

Model Report

Version •



Date/Time
Generated:
Author:

2017. 11. 22. 19:35:42

Oliver

EA Repository : C:\Users\Oliver\git\RiskGameJava\doc\UML.EAP

CREATED WITH



Table of Contents

Model	10
Use Case Model	10
Actors	10
Actors diagram	10
Player 1	10
Player 2	13
Player 3	13
Player 4	14
Application Start-up, Run and Shutdown	15
Application Start-up and Shutdown Use Cases diagram	15
Save Game	15
Players	16
Define Number Of Players	16
Game (Game Stage package)	16
Load Saved Gamestate	17
Preparation (Preparation Stage package)	17
Game Stage	19
Game Stage Use Cases diagram	19
Player 1	19
3 Card Exchange/Using/TradeIn	22
Exit Game	22
Game (contraction)	23
Main Battle Sequence	23
Reinforcement	24
Save Game	24
Transfer Units	24
Preparation Stage	26
Preparation Stage Use Cases diagram	26
Player	26
Define The Starting Player	26
Preparation Stage (contraction)	27

Primary Use Cases	28
Primary Use Cases diagram	28
Player 1	28
System Boundary	31
Note	31
Main Battle Sequence	31
Reinforcement	32
Dynamic View	33
Application Start-up, Run and Shutdown	33
Define number of players diagram	33
Player 1	34
RiskGameModel	36
RiskGameController	42
JFXAddPlayerView	47
JFXMainView	50
<anonymous>	56
Game diagram	57
Player 1	59
Load Game diagram	63
Player 1	64
RiskGameModel	67
RiskGameController	72
JFXMainView	77
<anonymous>	83
Check If Captured And Conquer	84
Define number of players	84
Define the starting player	84
Exchange Cards	85
Exit Game	85
Load Game	85
Main Battle Sequence	86
Main Game	86
Preparation stage	86
Reinforcement	87
Save Game	87
Transfer Units	87

want_load_game	88
want_save_game	88
Game Stage	89
Check If Captured And Conquer diagram	89
<anonymous>	89
Exchange Cards diagram	90
Player 1	90
Deck	93
RiskGameModel	95
RiskGameController	100
JFXMainView	105
<anonymous>	111
Exit Game diagram	113
Player 1	113
RiskGameModel	116
RiskGameController	122
JFXMainView	127
<anonymous>	133
Main Battle Sequence diagram	134
Player 1	135
Player 2	138
RiskGameModel	138
RiskGameController	144
JFXAttackView	149
JFXMainView	151
<anonymous>	157
Reinforcement diagram	160
Player 1	160
Player 2	163
RiskGameModel	163
RiskGameController	169
JFXMainView	174
<anonymous>	180
Save Game diagram	181
Player 1	182
RiskGameModel	185

RiskGameController	190
JFXMainView	195
<anonymous>	201
Transfer Units diagram	202
Player 1	202
RiskGameModel	205
RiskGameController	211
JFXMainView	216
JFXTransferView	222
<anonymous>	224
<anonymous>	225
<anonymous>	225
<anonymous>	225
Player1	226
attacker	226
attacker	226
defender	226
defender	226
Captured	226
Card loop	227
Close application	227
Close application	227
Close application	227
Save game?	227
canAttack	227
placingTheUnitsLoop	228
Preparation Stage	229
Define the starting player diagram	229
Player 1	229
RiskGameModel	232
RiskGameController	238
JFXMainView	243
<anonymous>	249
Preparation stage diagram	250
Player 1	251
RiskGameModel	253

RiskGameController	259
JFXMainView	264
<anonymous>	270
Alternative	271
Alternative1	271
Alternative2	271
Alternative3	271
Alternative4	271
Preparation loop	272
StartingPlayerLoop	272
Class Model	273
Class Model diagram	273
Note	273
Controller	274
View	274
Model	274
Application	274
Application diagram	274
App	275
Controller	276
Controller diagram	276
RiskGameModel	278
JFXAddPlayerView	283
JFXAttackView	286
JFXMainView	289
JFXRiskCardView	295
JFXTransferView	296
JFXViewBase	297
RiskGameController	299
View	305
View diagram	305
CountryPath	305
ImageViewPane	306
JFXAddPlayerView	307
JFXAttackView	310
JFXMainView	313

JFXRiskCardView	319
JFXTransferView	320
JFXViewBase	321
WorldBuilder	323
Country	323
Model	328
Model diagram	328
AttackResult	328
Card	329
Deck	330
Map	332
Country	333
Player	337
ResourceManager	339
RiskGameModel	340
SaveData	346
Territory	347
Color	349
Phase	350
Unit	350
User Interface Model	352
User Interface Model diagram	352
<anonymous>	352
<anonymous>	352
<anonymous>	353
<anonymous>	353
\$help://screen.htm	353
\$help://user_interface_group.htm	353
Primary Forms	353
Administrative Forms	354
Help System	354
Primary Forms	354
Primary Forms diagram	354
<anonymous>	355
Egységek képei	355
Egységek képei	356

Világtérkép	356
Világtérkép	356
<anonymous>	356
<anonymous>	356
<anonymous>	357
<anonymous>	358
<anonymous>	358
Főablak	358
1. játékos	358
2. játékos	358
3. játékos	358
4. játékos	359
5. játékos	359
Napló	359
Soron következő játékosok	359
Főablak	359
1. játékos	359
2. játékos	360
3. játékos	360
4. játékos	360
5. játékos	360
Soron következő játékosok	360
Adatlap	360
egységek száma	361
Következő erősítés	361
Le nem helyezett egységek száma	361
Főablak	361
Bónusz egységek számlálója	361
Change1	362
1. Játékos	362
2. játékos	362
3. játékos	362

4. játékos	362
5. játékos	363
Átcsoportosítás	363
Erősítés	363
Napló	363
Soron következő játékosok	363
Számláló	363
Támadás	364
UI Control	364
Támadás	364
Támadó dobókockái	364
Védekező dobókockái	364
Vizuális élmény	364
Átcsoportosítás	365
kijelölés	365
darab	365
darab	365
Elfogad	365
UI Control	365
Visszavonás	366
Administrative Forms	367
Administrative Forms diagram	367
<anonymous>	367
Help System	368
Help System diagram	368
<anonymous>	368
Content	368
Help	369
Toolbar	369
Index	369
Search	369

Model

Package in package "

Model
Version Phase 1.0 Proposed
Oliver created on 2017. 11. 22.. Last modified 2017. 11. 22.

Use Case Model

Package in package 'Model'

Use Case Model
Version Phase 1.0 Proposed
Szabi created on 2017. 02. 21.. Last modified 2017. 02. 21.

Actors

Package in package 'Use Case Model'

Actors
Version 1.0 Phase 1.0 Proposed
Szabi created on 2017. 02. 21.. Last modified 2017. 02. 21.

Actors diagram

Use Case diagram in package 'Actors'

Actors
Version 1.0
Oliver created on 2017. 02. 21.. Last modified 2017. 03. 23.

Figure 1: Actors

Player 1

Actor in package 'Actors'

Player 1
Version 1.0 Phase 1.0 Mandatory
Oliver created on 2017. 02. 21.. Last modified 2017. 03. 23.

OUTGOING BEHAVIORAL RELATIONSHIPS
Name: //Selects 2 Territories  Sequence from Player 1 to JFXMainView

OUTGOING BEHAVIORAL RELATIONSHIPS
Name: //select attacking units ↳ Sequence from Player 1 to JFXAttackView
↳ Sequence from Player 1 to Reinforcement
↳ Sequence from Player 1 to Load Game
↳ Sequence from Player 1 to Exit Game
↳ Sequence from Player 1 to Save Game
↳ Sequence from Player 1 to Exchange Cards
↳ Sequence from Player 1 to Define the starting player
↳ Sequence from Player 1 to Define number of players
Name: // Selects option ↳ Sequence from Player 1 to JFXMainView
Name: // Add number of players ↳ Sequence from Player 1 to JFXAddPlayerView
Name: //Set a Save Name then Apply ↳ Sequence from Player 1 to JFXMainView
Name: // click to begin action ↳ Sequence from Player 1 to JFXMainView
Name: // click to connect ↳ Sequence from Player 1 to JFXMainView
Name: //End Turn ↳ Sequence from Player 1 to JFXMainView
Name: // Selects exit game option ↳ Sequence from Player 1 to JFXMainView
Name: // Select saved game file then apply ↳ Sequence from Player 1 to JFXMainView

OUTGOING BEHAVIORAL RELATIONSHIPS	
 Sequence from Player 1 to Main Battle Sequence	
Name: //init attack	
 Sequence from Player 1 to JFXMainView	
Name: //end turn	
 Sequence from Player 1 to JFXMainView	
 Sequence from Player 1 to Transfer Units	
Name: // select Yes or No	
 Sequence from Player 1 to JFXMainView	
Name: //Selects Units then Apply	
 Sequence from Player 1 to JFXTransferView	
 Sequence from Player 1 to Preparation stage	
INCOMING BEHAVIORAL RELATIONSHIPS	
 Sequence from Transfer Units to Player 1	
 Sequence from Save Game to Player 1	
 Sequence from Main Battle Sequence to Player 1	
 Sequence from Preparation stage to Player 1	
 Sequence from Reinforcement to Player 1	
 Sequence from Exchange Cards to Player 1	
Name: // popup	
 Sequence from JFXMainView to Player 1	
Name: addTerritory()	
 Sequence from RiskGameModel to Player 1	
 Sequence from Define number of players to Player 1	
 Sequence from Exit Game to Player 1	

INCOMING BEHAVIORAL RELATIONSHIPS

- ➡ Sequence from Define the starting player to Player 1

- ➡ Sequence from Load Game to Player 1

CONNECTORS

- ## UseCaseLink Source -> Destination

From: Player 1 : Actor, Public

To: Reinforcement : UseCase, Public

-  **UseCaseLink** Source -> Destination

From: Player 1 : Actor, Public

To: Transfer Units : UseCase, Public

-  UseCaseLink Source -> Destination

From: Player 1 : Actor, Public

To: Main Battle Sequence : UseCase. Public

ASSOCIATIONS

- #### Association (direction: Unspecified)

Source: Public (Actor) Player 1

Target: Public (UseCase) Exit Game

Player 2

Actor in package 'Actors'

Player 2

Version 1.0 Phase 1.0 Mandatory

Oliver created on 2017. 03. 22.. Last modified 2017. 03. 23.

OUTGOING BEHAVIORAL RELATIONSHIPS

Name: //Selects a Territory

- ➡ Sequence from Player 2 to JFXMainView

Name: //select defending units

- ➡ Sequence from Player 2 to JFXAttackView

Player 3

Actor in package 'Actors'

Player 3
Version 1.0 Phase 1.0 Mandatory

Oliver created on 2017. 03. 22.. Last modified 2017. 03. 23.

Player 4

Actor in package 'Actors'

Player 4

Version 1.0 Phase 1.0 Mandatory

Oliver created on 2017. 03. 22.. Last modified 2017. 03. 23.

Application Start-up, Run and Shutdown

Package in package 'Use Case Model'

Application Start-up, Run and Shutdown
Version 1.0 Phase 1.0 Proposed
Szabó Dávid created on 2017. 02. 22.. Last modified 2017. 03. 28.

Application Start-up and Shutdown Use Cases diagram

Use Case diagram in package 'Application Start-up, Run and Shutdown'

Application Start-up and Shutdown Use Cases
Version 1.0
Szabó Dávid created on 2017. 02. 22.. Last modified 2017. 03. 28.

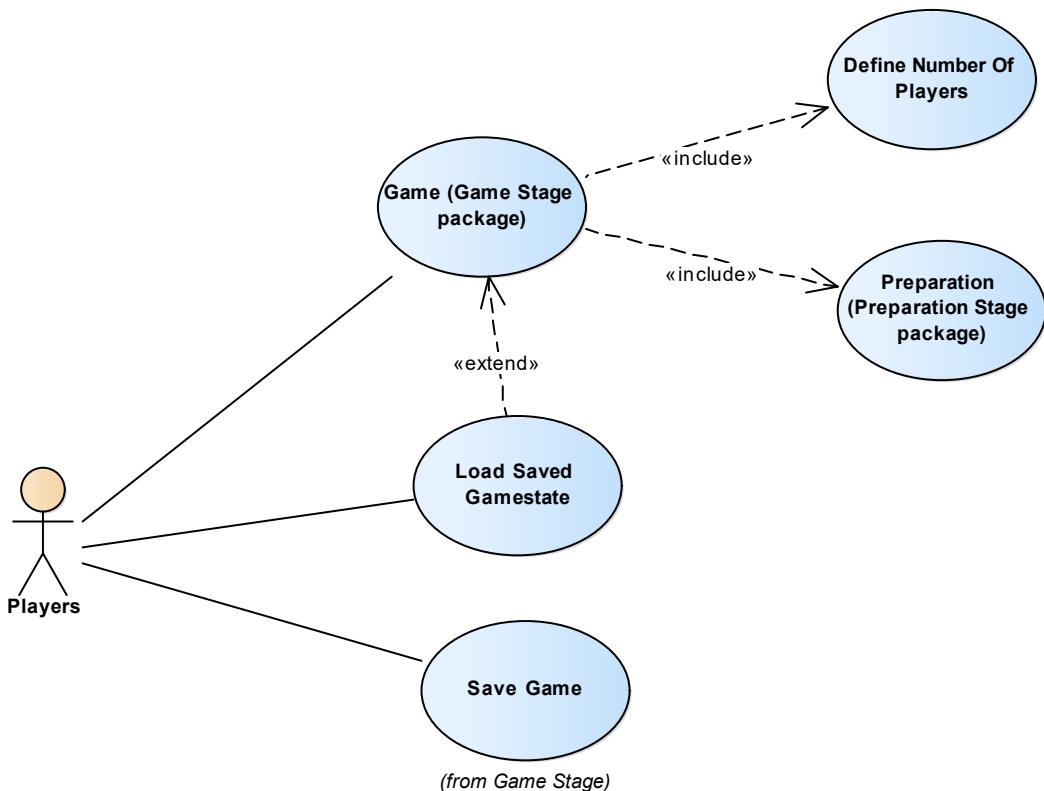


Figure 2: Application Start-up and Shutdown Use Cases

Save Game

UseCase in package 'Game Stage'

Aktualis jatekallas mentese

Save Game
Version 1.0 Phase 1.0 Proposed
Oliver created on 2017. 02. 23.. Last modified 2017. 03. 28.

ASSOCIATIONS

- Association (direction: Unspecified)
Source: Public (Actor) Players Target: Public (UseCase) Save Game

Players

Actor in package 'Application Start-up, Run and Shutdown'

Players
Version 1.0 Phase 1.0 Proposed
Szabó Dávid created on 2017. 02. 22.. Last modified 2017. 03. 28.

ASSOCIATIONS	
 Association (direction: Unspecified)	
Source: Public (Actor) Players	Target: Public (UseCase) Save Game
 Association (direction: Unspecified)	
Source: Public (Actor) Players	Target: Public (UseCase) Load Saved Gamestate
 Association (direction: Unspecified)	
Source: Public (Actor) Players	Target: Public (UseCase) Game (Game Stage package)

Define Number Of Players

UseCase in package 'Application Start-up, Run and Shutdown'

A jatek kezdete elott az also jatekosnak meg kell hataroznia, hogy osszesen hany jatekosnal kezdodjon el a jatek.

Define Number Of Players
Version 1.0 Phase 1.0 Proposed
Szabó Dávid created on 2017. 02. 22.. Last modified 2017. 03. 28.

CONNECTORS	
 Include «include» Source -> Destination From: Game (Game Stage package) : UseCase, Public To: Define Number Of Players : UseCase, Public	

Game (Game Stage package)

UseCase in package 'Application Start-up, Run and Shutdown'

A jatekosok egymas utan kovetkeznek. minden jatekos a sajat koreben eloszor elhelyezi az uj egysegeket, utana donthet, hogy akar-e tamadni. Korenek vegen atcsoportosithat egy teruletrol egysegeket egy masik teruletre.

Game (Game Stage package)
Version 1.0 Phase 1.0 Proposed

CONNECTORS

Include «include» Source -> Destination

From: Game (Game Stage package) : UseCase, Public
 To: Define Number Of Players : UseCase, Public

Include «include» Source -> Destination

From: Game (Game Stage package) : UseCase, Public
 To: Preparation (Preparation Stage package) : UseCase, Public

Extend «extend» Source -> Destination

From: Load Saved Gamestate : UseCase, Public
 To: Game (Game Stage package) : UseCase, Public

ASSOCIATIONS

Association (direction: Unspecified)

Source: Public (Actor) Players

Target: Public (UseCase) Game (Game Stage package)

Load Saved Gamestate

UseCase in package 'Application Start-up, Run and Shutdown'

Load Saved Gamestate

Version 1.0 Phase 1.0 Proposed

Szabó Dávid created on 2017. 02. 22.. Last modified 2017. 03. 28.

CONNECTORS

Extend «extend» Source -> Destination

From: Load Saved Gamestate : UseCase, Public
 To: Game (Game Stage package) : UseCase, Public

ASSOCIATIONS

Association (direction: Unspecified)

Source: Public (Actor) Players

Target: Public (UseCase) Load Saved Gamestate

Preparation (Preparation Stage package)

UseCase in package 'Application Start-up, Run and Shutdown'

A jatekosok egymás után elhelyezik a kezdeti egységeket a teruletekben.

Preparation (Preparation Stage package)

CONNECTORS **Include** «include» Source -> Destination

From: Game (Game Stage package) : UseCase, Public

To: Preparation (Preparation Stage package) : UseCase, Public

Game Stage

Package in package 'Use Case Model'

Game Stage
Version 1.0 Phase 1.0 Proposed
Szabó Dávid created on 2017. 02. 22.. Last modified 2017. 03. 28.

Game Stage Use Cases diagram

Use Case diagram in package 'Game Stage'

Game Stage Use Cases
Version 1.0
Szabi created on 2017. 02. 21.. Last modified 2017. 03. 28.

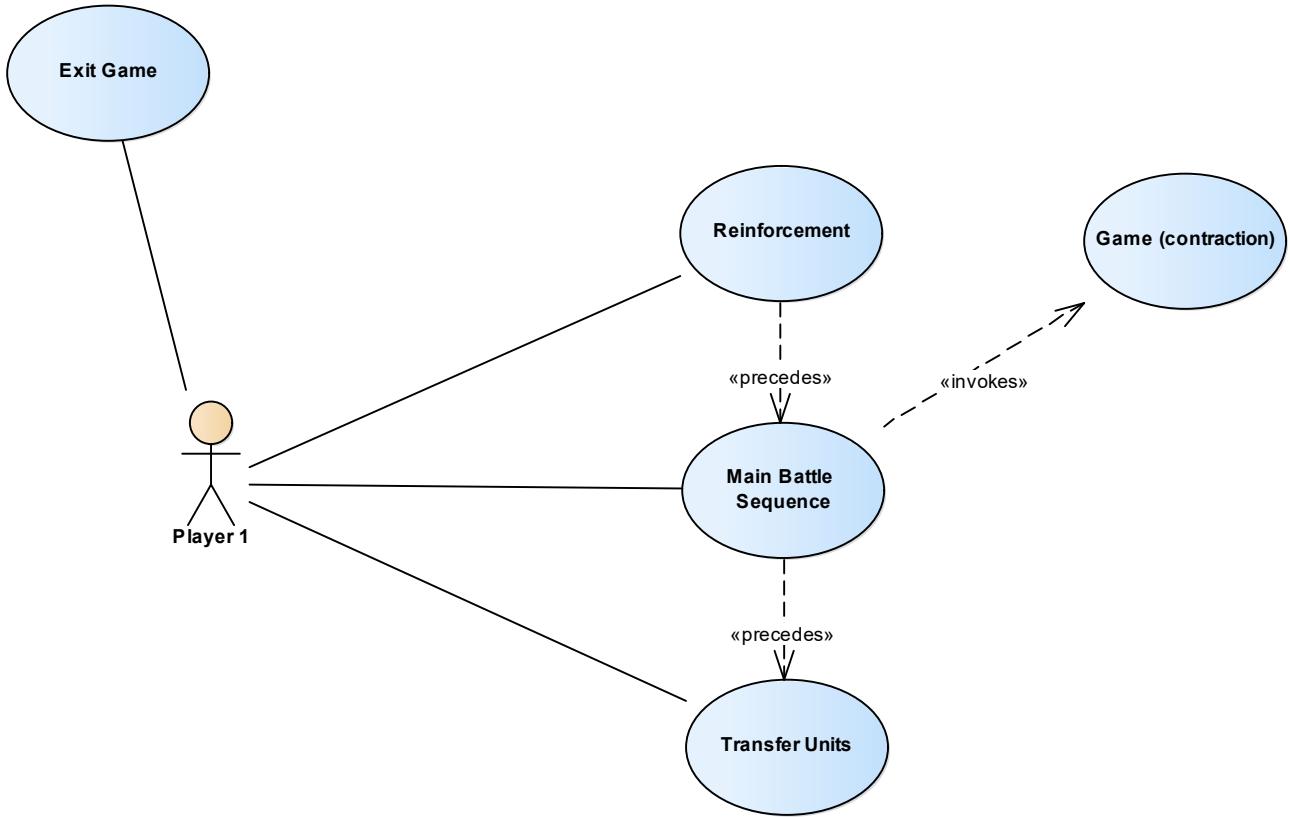


Figure 3: Game Stage Use Cases

Player 1

Actor in package 'Actors'

Player 1
Version 1.0 Phase 1.0 Mandatory
Oliver created on 2017. 02. 21.. Last modified 2017. 03. 23.

OUTGOING BEHAVIORAL RELATIONSHIPS
Name: //Selects 2 Territories Sequence from Player 1 to JFXMainView
Name: //select attacking units

OUTGOING BEHAVIORAL RELATIONSHIPS
↳ Sequence from Player 1 to JFXAttackView
↳ Sequence from Player 1 to Reinforcement
↳ Sequence from Player 1 to Load Game
↳ Sequence from Player 1 to Exit Game
↳ Sequence from Player 1 to Save Game
↳ Sequence from Player 1 to Exchange Cards
↳ Sequence from Player 1 to Define the starting player
↳ Sequence from Player 1 to Define number of players
Name: // Selects option ↳ Sequence from Player 1 to JFXMainView
Name: // Add number of players ↳ Sequence from Player 1 to JFXAddPlayerView
Name: //Set a Save Name then Apply ↳ Sequence from Player 1 to JFXMainView
Name: // click to begin action ↳ Sequence from Player 1 to JFXMainView
Name: // click to connect ↳ Sequence from Player 1 to JFXMainView
Name: //End Turn ↳ Sequence from Player 1 to JFXMainView
Name: // Selects exit game option ↳ Sequence from Player 1 to JFXMainView
Name: // Select saved game file then apply ↳ Sequence from Player 1 to JFXMainView
↳ Sequence from Player 1 to Main Battle Sequence

OUTGOING BEHAVIORAL RELATIONSHIPS	
Name: //init attack	Sequence from Player 1 to JFXMainView
Name: //end turn	Sequence from Player 1 to JFXMainView
	Sequence from Player 1 to Transfer Units
Name: // select Yes or No	Sequence from Player 1 to JFXMainView
Name: //Selects Units then Apply	Sequence from Player 1 to JFXTransferView
	Sequence from Player 1 to Preparation stage
INCOMING BEHAVIORAL RELATIONSHIPS	
Sequence from Transfer Units to Player 1	
Sequence from Save Game to Player 1	
Sequence from Main Battle Sequence to Player 1	
Sequence from Preparation stage to Player 1	
Sequence from Reinforcement to Player 1	
Sequence from Exchange Cards to Player 1	
Name: // popup	Sequence from JFXMainView to Player 1
Name: addTerritory()	Sequence from RiskGameModel to Player 1
Sequence from Define number of players to Player 1	
Sequence from Exit Game to Player 1	
Sequence from Define the starting player to Player 1	

INCOMING BEHAVIORAL RELATIONSHIPS	
 Sequence from Load Game to Player 1	
CONNECTORS	
 UseCaseLink Source -> Destination	
From: Player 1 : Actor, Public	
To: Reinforcement : UseCase, Public	
 UseCaseLink Source -> Destination	
From: Player 1 : Actor, Public	
To: Transfer Units : UseCase, Public	
 UseCaseLink Source -> Destination	
From: Player 1 : Actor, Public	
To: Main Battle Sequence : UseCase, Public	

ASSOCIATIONS	
 Association (direction: Unspecified)	
Source: Public (Actor) Player 1	Target: Public (UseCase) Exit Game

3 Card Exchange/Using/TradeIn

UseCase in package 'Game Stage'

3 Card Exchange/Using/TradeIn
 Version 1.0 Phase 1.0 Mandatory
 Szabi created on 2017. 04. 04.. Last modified 2017. 04. 04.

Exit Game

UseCase in package 'Game Stage'

Aktualis jatekból való kilepés. Egy felugro pop-up megkerdezi a jatekost, hogy ki szeretne-e lejni. Amennyiben ítt az igen gombra kattint, a program megkerdezi, hogy menteni szeretne-e az aktualis jatekallast. Amennyiben az igen gombra kattint, abban az esetben a program elmenti a jatekallast. Amennyiben a fenti ket eset kozul barmelyiknel is "nem"-re kattint, abban az esetben a jatekból kilep.

Exit Game
 Version 1.0 Phase 1.0 Proposed
 Szabó Dávid created on 2017. 02. 22.. Last modified 2017. 03. 28.

ASSOCIATIONS	
 Association (direction: Unspecified)	

ASSOCIATIONS

Source: Public (Actor) Player 1

Target: Public (UseCase) Exit Game

Game (contraction)*UseCase in package 'Game Stage'*

A tamado 1, 2 vagy 3 egyseggel tamad, a vedekezo 1 vagy 2 egyseggel vedekezik.

Annyi kockaval dobunk ahany egyseg harcol. A ket fel legnagyobb dobasait paronkent osszehasonlitjuk.

Minden osszehasonlitasnal, ha a vedekezo legalabb akkorat dobott mint a tamado akkor a tamado veszit egy egysemet.
Ellenekzo esetben a vedekezo veszit egy egysemet.

Game (contraction)

Version 1.0 Phase 1.0 Proposed

Szabi created on 2017. 02. 21.. Last modified 2017. 03. 28.

CONNECTORS **Dependency** «invokes» Source -> Destination

From: Main Battle Sequence : UseCase, Public
To: Game (contraction) : UseCase, Public

Main Battle Sequence*UseCase in package 'Game Stage'*

A jatekos a sajat teruleteirol szomszedos teruleteket tamadhat.

Megjeloli:

- Tamadast indito terulet
- Tamadott terulet
- Tamado egysegek szama (1, 2 vagy 3. Legalabb 1 egysegnek maradnia kell a tamado teruleten)

Main Battle Sequence

Version 1.0 Phase 1.0 Proposed

Szabi created on 2017. 02. 21.. Last modified 2017. 03. 28.

CONNECTORS **Dependency** «precedes» Source -> Destination

From: Main Battle Sequence : UseCase, Public
To: Transfer Units : UseCase, Public

 **Dependency** «invokes» Source -> Destination

From: Main Battle Sequence : UseCase, Public
To: Game (contraction) : UseCase, Public

 **Dependency** «precedes» Source -> Destination

From: Reinforcement : UseCase, Public
To: Main Battle Sequence : UseCase, Public

CONNECTORS

 **UseCaseLink** Source -> Destination

From: Player 1 : Actor, Public
 To: Main Battle Sequence : UseCase, Public

Reinforcement

UseCase in package 'Game Stage'

A jatekos a fordulo kezdeten erositest kerhet

- A birtokolt terulet harmada alapjan
- Minimum 3
- Extra szabalyok..

Reinforcement

Version 1.0 Phase 1.0 Mandatory

Szabi created on 2017. 02. 21.. Last modified 2017. 03. 28.

CONNECTORS

 **Dependency** «precedes» Source -> Destination

From: Reinforcement : UseCase, Public
 To: Main Battle Sequence : UseCase, Public

 **UseCaseLink** Source -> Destination

From: Player 1 : Actor, Public
 To: Reinforcement : UseCase, Public

Save Game

UseCase in package 'Game Stage'

Aktualis jatekallas mentese

Save Game

Version 1.0 Phase 1.0 Proposed

Oliver created on 2017. 02. 23.. Last modified 2017. 03. 28.

ASSOCIATIONS

 **Association** (direction: Unspecified)

Source: Public (Actor) Players

Target: Public (UseCase) Save Game

Transfer Units

UseCase in package 'Game Stage'

Egy teruletrol szomszedos teruletekre rakhatja az egysegeket.
 Legalabb egy egysegnek lennie kell minden teruleten.

Transfer Units
Version 1.0 Phase 1.0 Mandatory
Szabi created on 2017. 02. 21.. Last modified 2017. 03. 28.

CONNECTORS

Dependency «precedes» Source -> Destination

From: Main Battle Sequence : UseCase, Public
To: Transfer Units : UseCase, Public

UseCaseLink Source -> Destination

From: Player 1 : Actor, Public
To: Transfer Units : UseCase, Public

Preparation Stage

Package in package 'Use Case Model'

Preparation Stage
Version 1.0 Phase 1.0 Proposed
Szabó Dávid created on 2017. 02. 22.. Last modified 2017. 03. 28.

Preparation Stage Use Cases diagram

Use Case diagram in package 'Preparation Stage'

Preparation Stage Use Cases
Version 1.0
Szabó Dávid created on 2017. 02. 22.. Last modified 2017. 03. 28.

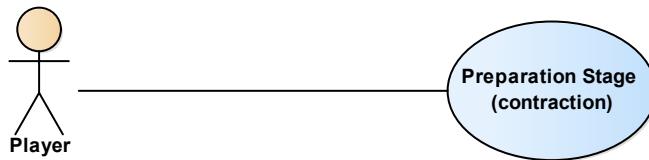


Figure 4: Preparation Stage Use Cases

Player

Actor in package 'Preparation Stage'

Player
Version 1.0 Phase 1.0 Proposed
Szabó Dávid created on 2017. 02. 22.. Last modified 2017. 03. 28.

ASSOCIATIONS	
Association (direction: Unspecified)	
Source: Public (Actor) Player	Target: Public (UseCase) Preparation Stage (contraction)

Define The Starting Player

UseCase in package 'Preparation Stage'

A rendszer kiválasztja, hogy melyik játékos kezdheti elhelyezni az egységeket.

Define The Starting Player
Version 1.0 Phase 1.0 Proposed
Szabó Dávid created on 2017. 02. 22.. Last modified 2017. 03. 28.

CONNECTORS

CONNECTORS

 **Dependency** «precedes» Source -> Destination
From: Define The Starting Player : UseCase, Public
To: Preparation Stage (contraction) : UseCase, Public

Preparation Stage (contraction)

UseCase in package 'Preparation Stage'

A jatekosok egyesevvel elhelyezik az osszes kezdeti egyseget a szabad teruletekre.

Eloszor egy teruletre csak egy egyseget lehet tenni. Ha mar nincs tobb szabad terulet akkor minden jatekos csak a sajat teruleteire tehet egyseget.

Preparation Stage (contraction)
Version 1.0 Phase 1.0 Proposed
Szabó Dávid created on 2017. 02. 22.. Last modified 2017. 03. 28.

CONNECTORS

 **Dependency** «precedes» Source -> Destination
From: Define The Starting Player : UseCase, Public
To: Preparation Stage (contraction) : UseCase, Public

ASSOCIATIONS

 Association (direction: Unspecified)

Source: Public (Actor) Player

Target: Public (UseCase) Preparation Stage
(contraction)

Primary Use Cases

Package in package 'Use Case Model'

Primary Use Cases
Version 1.0 Phase 1.0 Proposed
Szabi created on 2017. 02. 21.. Last modified 2017. 02. 21.

Primary Use Cases diagram

Use Case diagram in package 'Primary Use Cases'

Primary Use Cases
Version 1.0
Oliver created on 2017. 02. 21.. Last modified 2017. 02. 21.

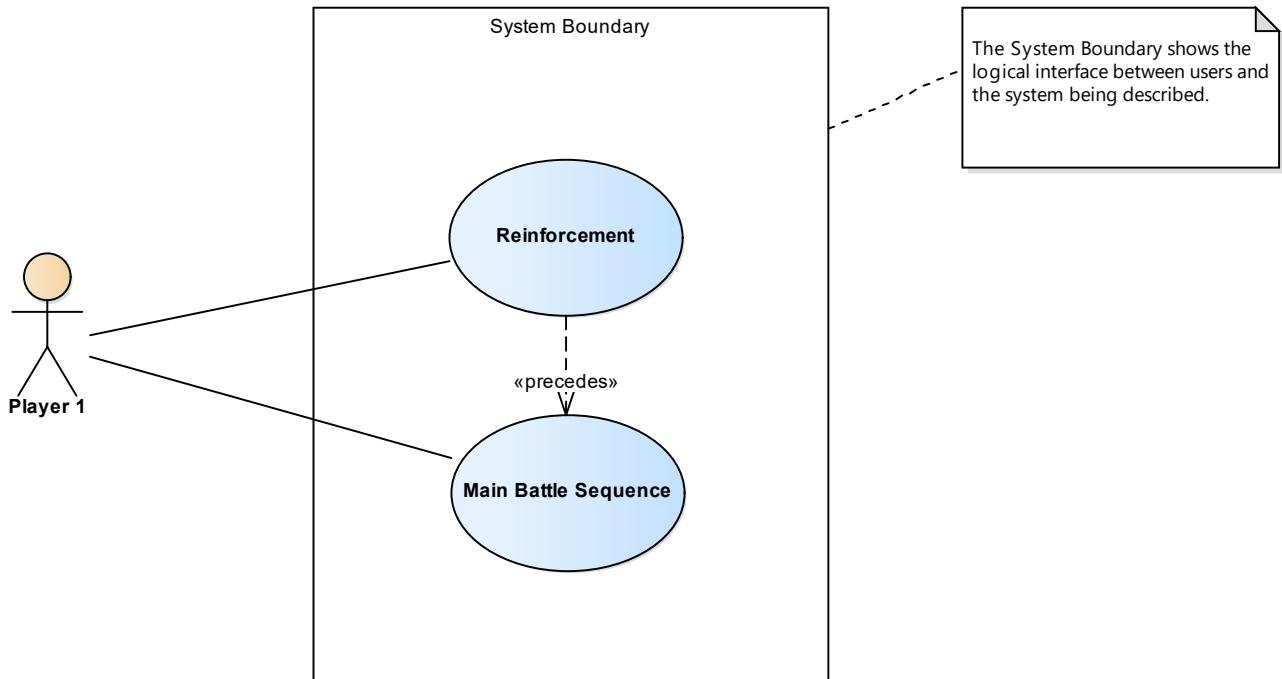


Figure 5: Primary Use Cases

Player 1

Actor in package 'Actors'

Player 1
Version 1.0 Phase 1.0 Mandatory
Oliver created on 2017. 02. 21.. Last modified 2017. 03. 23.

OUTGOING BEHAVIORAL RELATIONSHIPS
Name: //Selects 2 Territories Sequence from Player 1 to JFXMainView
Name: //select attacking units Sequence from Player 1 to JFXAttackView
Sequence from Player 1 to Reinforcement

OUTGOING BEHAVIORAL RELATIONSHIPS
↳ Sequence from Player 1 to Load Game
↳ Sequence from Player 1 to Exit Game
↳ Sequence from Player 1 to Save Game
↳ Sequence from Player 1 to Exchange Cards
↳ Sequence from Player 1 to Define the starting player
↳ Sequence from Player 1 to Define number of players
Name: // Selects option ↳ Sequence from Player 1 to JFXMainView
Name: // Add number of players ↳ Sequence from Player 1 to JFXAddPlayerView
Name: //Set a Save Name then Apply ↳ Sequence from Player 1 to JFXMainView
Name: // click to begin action ↳ Sequence from Player 1 to JFXMainView
Name: // click to connect ↳ Sequence from Player 1 to JFXMainView
Name: //End Turn ↳ Sequence from Player 1 to JFXMainView
Name: // Selects exit game option ↳ Sequence from Player 1 to JFXMainView
Name: // Select saved game file then apply ↳ Sequence from Player 1 to JFXMainView
↳ Sequence from Player 1 to Main Battle Sequence
Name: //init attack ↳ Sequence from Player 1 to JFXMainView

OUTGOING BEHAVIORAL RELATIONSHIPS	
Name: //end turn	Sequence from Player 1 to JFXMainView
Sequence from Player 1 to Transfer Units	
Name: // select Yes or No	Sequence from Player 1 to JFXMainView
Name: //Selects Units then Apply	Sequence from Player 1 to JFXTransferView
Sequence from Player 1 to Preparation stage	
INCOMING BEHAVIORAL RELATIONSHIPS	
Sequence from Transfer Units to Player 1	
Sequence from Save Game to Player 1	
Sequence from Main Battle Sequence to Player 1	
Sequence from Preparation stage to Player 1	
Sequence from Reinforcement to Player 1	
Sequence from Exchange Cards to Player 1	
Name: // popup	Sequence from JFXMainView to Player 1
Name: addTerritory()	Sequence from RiskGameModel to Player 1
Sequence from Define number of players to Player 1	
Sequence from Exit Game to Player 1	
Sequence from Define the starting player to Player 1	
Sequence from Load Game to Player 1	

CONNECTORS	
 UseCaseLink	Source -> Destination From: Player 1 : Actor, Public To: Reinforcement : UseCase, Public
 UseCaseLink	Source -> Destination From: Player 1 : Actor, Public To: Transfer Units : UseCase, Public
 UseCaseLink	Source -> Destination From: Player 1 : Actor, Public To: Main Battle Sequence : UseCase, Public
ASSOCIATIONS	
 Association (direction: Unspecified)	Source: Public (Actor) Player 1 Target: Public (UseCase) Exit Game

System Boundary

Boundary in package 'Primary Use Cases'

System Boundary
Version 1.0 Phase 1.0 Proposed
Szabi created on 2017. 02. 21.. Last modified 2017. 02. 21.
Extends

Note

Note in package 'Primary Use Cases'

The System Boundary shows the logical interface between users and the system being described.

Note
Version 1.0 Phase 1.0 Proposed
Szabi created on 2017. 02. 21.. Last modified 2017. 02. 21.
Extends

Main Battle Sequence

UseCase in package 'Game Stage'

A jatekos a sajat teruleteirol szomszedos teruleteket tamadhat.

Megjeloli:

- Tamadast indito terulet
- Tamadott terulet
- Tamado egysegek szama (1, 2 vagy 3. Legalabb 1 egysegnek maradnia kell a tamado teruleten)

Main Battle Sequence
Version 1.0 Phase 1.0 Proposed

CONNECTORS
<p> Dependency «precedes» Source -> Destination From: Main Battle Sequence : UseCase, Public To: Transfer Units : UseCase, Public</p>
<p> Dependency «invokes» Source -> Destination From: Main Battle Sequence : UseCase, Public To: Game (contraction) : UseCase, Public</p>
<p> Dependency «precedes» Source -> Destination From: Reinforcement : UseCase, Public To: Main Battle Sequence : UseCase, Public</p>
<p> UseCaseLink Source -> Destination From: Player 1 : Actor, Public To: Main Battle Sequence : UseCase, Public</p>

Reinforcement

UseCase in package 'Game Stage'

A jatekos a fordulo kezdeten erositest kerhet
 -A birtokolt terulet harmada alapjan
 -Minimum 3
 -Extra szabalyok..

Reinforcement
Version 1.0 Phase 1.0 Mandatory
Szabi created on 2017. 02. 21.. Last modified 2017. 03. 28.

CONNECTORS
<p> Dependency «precedes» Source -> Destination From: Reinforcement : UseCase, Public To: Main Battle Sequence : UseCase, Public</p>
<p> UseCaseLink Source -> Destination From: Player 1 : Actor, Public To: Reinforcement : UseCase, Public</p>

Dynamic View

Package in package 'Model'

Dynamic View
Version Phase 1.0 Proposed
Szabi created on 2017. 03. 07.. Last modified 2017. 03. 07.

Application Start-up, Run and Shutdown

Package in package 'Dynamic View'

Application Start-up, Run and Shutdown
Version 1.0 Phase 1.0 Proposed
Oliver created on 2017. 03. 28.. Last modified 2017. 03. 28.

Define number of players diagram

Interaction diagram in package 'Application Start-up, Run and Shutdown'

Define number of players
Version 1.0
fodorad created on 2017. 03. 22.. Last modified 2017. 09. 21.

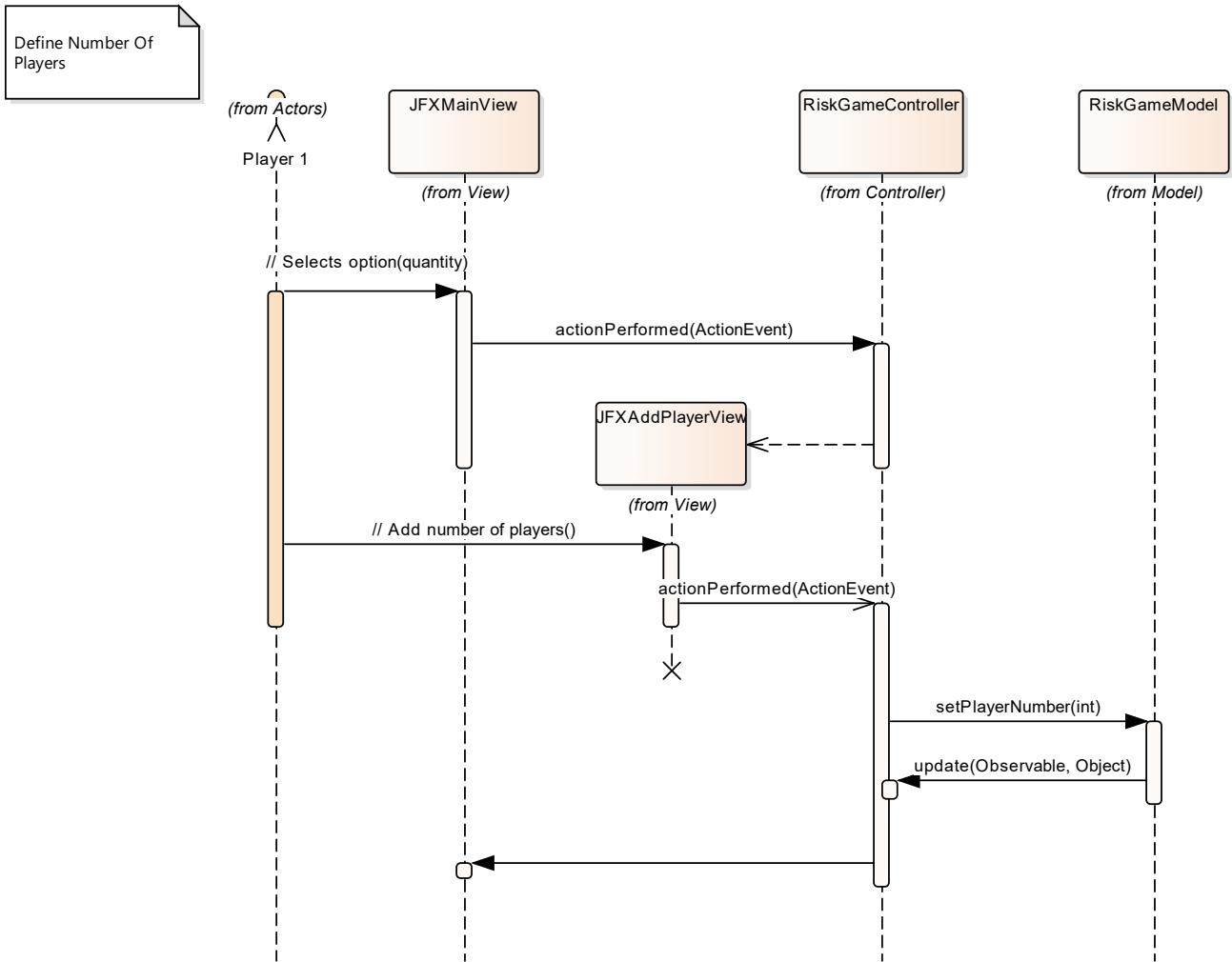


Figure 6: Define number of players

Player 1

Actor in package 'Actors'

Player 1
Version 1.0 Phase 1.0 Mandatory
Oliver created on 2017. 02. 21.. Last modified 2017. 03. 23.

OUTGOING BEHAVIORAL RELATIONSHIPS
Name: //Selects 2 Territories ↳ Sequence from Player 1 to JFXMainView
Name: //select attacking units ↳ Sequence from Player 1 to JFXAttackView
↳ Sequence from Player 1 to Reinforcement
↳ Sequence from Player 1 to Load Game
↳ Sequence from Player 1 to Exit Game
↳ Sequence from Player 1 to Save Game
↳ Sequence from Player 1 to Exchange Cards
↳ Sequence from Player 1 to Define the starting player
↳ Sequence from Player 1 to Define number of players
Name: // Selects option ↳ Sequence from Player 1 to JFXMainView
Name: // Add number of players ↳ Sequence from Player 1 to JFXAddPlayerView
Name: //Set a Save Name then Apply ↳ Sequence from Player 1 to JFXMainView
Name: // click to begin action ↳ Sequence from Player 1 to JFXMainView
Name: // click to connect ↳ Sequence from Player 1 to JFXMainView

OUTGOING BEHAVIORAL RELATIONSHIPS	
Name: //End Turn	Sequence from Player 1 to JFXMainView
Name: // Selects exit game option	Sequence from Player 1 to JFXMainView
Name: // Select saved game file then apply	Sequence from Player 1 to JFXMainView
	Sequence from Player 1 to Main Battle Sequence
Name: //init attack	Sequence from Player 1 to JFXMainView
Name: //end turn	Sequence from Player 1 to JFXMainView
	Sequence from Player 1 to Transfer Units
Name: // select Yes or No	Sequence from Player 1 to JFXMainView
Name: //Selects Units then Apply	Sequence from Player 1 to JFXTransferView
	Sequence from Player 1 to Preparation stage
INCOMING BEHAVIORAL RELATIONSHIPS	
	Sequence from Transfer Units to Player 1
	Sequence from Save Game to Player 1
	Sequence from Main Battle Sequence to Player 1
	Sequence from Preparation stage to Player 1
	Sequence from Reinforcement to Player 1
	Sequence from Exchange Cards to Player 1
Name: // popup	

INCOMING BEHAVIORAL RELATIONSHIPS
➡ Sequence from JFXMainView to Player 1
Name: addTerritory() ➡ Sequence from RiskGameModel to Player 1
➡ Sequence from Define number of players to Player 1
➡ Sequence from Exit Game to Player 1
➡ Sequence from Define the starting player to Player 1
➡ Sequence from Load Game to Player 1

CONNECTORS
↗ UseCaseLink Source -> Destination From: Player 1 : Actor, Public To: Reinforcement : UseCase, Public
↗ UseCaseLink Source -> Destination From: Player 1 : Actor, Public To: Transfer Units : UseCase, Public
↗ UseCaseLink Source -> Destination From: Player 1 : Actor, Public To: Main Battle Sequence : UseCase, Public

ASSOCIATIONS
↙ Association (direction: Unspecified) Source: Public (Actor) Player 1 Target: Public (UseCase) Exit Game

RiskGameModel

Class in package 'Model'

Az MVC architektura Model részét megvalósító osztálya.

RiskGameModel
Version 1.0 Phase 1.0 Proposed
Szabi created on 2017. 02. 21.. Last modified 2017. 11. 22.

OUTGOING STRUCTURAL RELATIONSHIPS

OUTGOING STRUCTURAL RELATIONSHIPS	
↳ Aggregation from RiskGameModel to RiskGameController	[Direction is 'Source -> Destination'.]
INCOMING STRUCTURAL RELATIONSHIPS	
⇒ Aggregation from Map to RiskGameModel	[Direction is 'Source -> Destination'.]
⇒ Aggregation from Deck to RiskGameModel	[Direction is 'Source -> Destination'.]
ATTRIBUTES	
◆ calculatedCircleNumber : int Private	[Is static False. Containment is Not Specified.]
◆ capturedThisTurn : Boolean Private	[Is static False. Containment is Not Specified.]
◆ circlenumber : int Private	[Is static False. Containment is Not Specified.]
◆ currentPlayer : int Private	[Is static False. Containment is Not Specified.]
◆ deck : Deck Private	[Is static False. Containment is Not Specified.]
◆ hasTransferred : boolean Private	[Is static False. Containment is Not Specified.]
◆ map : Map Private	[Is static False. Containment is Not Specified.]
◆ misnumber : int Private	[Is static False. Containment is Not Specified.]
◆ phase : Phase Private	[Is static False. Containment is Not Specified.]
◆ playersList : List<Player> Private	[Is static False. Containment is Not Specified.]
◆ r : Random Private = new Random()	

ATTRIBUTES	
	[Is static False. Containment is Not Specified.]
◆ waitForUnitsTemp : Territory Private = new Territory[2]	[Is static False. Containment is Not Specified.]
ASSOCIATIONS	
✓ . Association (direction: Source -> Destination)	
Source: Public (Class) RiskGameModel	Target: Private waitForUnitsTemp (Class) Territory
✓ . Association (direction: Source -> Destination)	
Source: Public (Class) RiskGameModel	Target: Private playersList (Class) Player Cardinality: [0..*]
✓ . Association (direction: Source -> Destination)	
Source: Public (Class) RiskGameModel	Target: Public deck (Class) Player
OPERATIONS	
◆ addPlayer (name : String , color : Color) : boolean Public	
Add player with name and color @return boolean	[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ addPlayerToPlayerList (player : Player) : void Public	
Adder for playerList	
Properties: throws = Exception ea_guid = {45684F69-D727-4875-BEF7-5232B4E9D1C9}	[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ addPlayerToPlayerList (map : java.util.Map<Color, String>) : void Public	
Pairs overload	
Properties: throws = Exception	[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ attackTerritoryWithResult (defenderID : int , attackerID : int , defendUnits : int , attackUnits : int) : AttackResult Public	
Return the result of the attack If the attacker have more than 3 units, the fighting attacker units will reduce to 3 If the defender have more than 2 units, the fighting defender units will reduce to 2	
Properties:	

OPERATIONS	
throws = Exception	[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ calculateCircle () : void Private	
Properties:	
throws = Exception	[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ checkAttackPossible (defender : Territory , attacker : Territory , defendUnits : int , attackUnits : int) : boolean Public	
Csak akkor tĂˇmadhat, ha a terĂĽleten elegendL' egysĂ©ge van. Igazzal tĂ©r vissza, ha a tamadas lehetsĂ©ges.	
@return boolean	[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ checkAttackPossible (defenderID : int , attackerID : int) : boolean Public	
Attack checking method	
@return	[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ checkIfCapturedAndConquer (defender : Territory) : boolean Public	
Capture and conquer checking method	
@return boolean	
Properties:	
ea_guid = {BFF4F3B6-2538-4778-9372-EAA3B468EDD8}	[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ checkIfTerritoryIsEmpty (territory : Territory) : boolean Public	
Empty territory checking method	
@return boolean	
Properties:	
ea_guid = {BE4F4E07-A4C4-4e91-9F00-EF1EEB65D26C}	[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ checkTransferPossible (from : int , to : int) : boolean Public	
Transfer checking method	
@return boolean	[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ endBattlePhase () : boolean Public	
Battle	
@return boolean	
Properties:	
throws = Exception	[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]

OPERATIONS
<p>❖ endTurn () : int Public</p> <p>Kor vege Returns next player's id @return int</p> <p>Properties: throws = Exception ea_guid = {F5F8D5D6-8997-43de-A5AD-DDDC647E775F} [Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]</p>
<p>❖ getCurrentPlayer () : int Public</p> <p>Getter for current player @return int</p> <p>[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]</p>
<p>❖ getPhase () : Phase Public</p> <p>Getter for Phase @return phase</p> <p>[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]</p>
<p>❖ getPlayerCards (id : int) : List<Integer> Public</p> <p>[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]</p>
<p>❖ getPlayerName (id : int) : String Public</p> <p>Getter for Player's name @return String</p> <p>[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]</p>
<p>❖ getPlayers () : List<String> Public</p> <p>Getter for Players list @return List<Player></p> <p>[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]</p>
<p>❖ getTerritory (id : int) : Territory Public</p> <p>Getter for Territory @return Territory</p> <p>[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]</p>
<p>❖ getUnitsLeftToReinforce (playerID : int) : int Public</p> <p>Getter for left units in reinforce phase @return int</p> <p>[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]</p>
<p>❖ loadGame (file : File) : boolean Public</p> <p>Load data method</p>

OPERATIONS
<p>@return boolean</p> <p>Properties:</p> <p>throws = Exception</p> <p>[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]</p>
<p>◆ moveUnits (units : int) : boolean Public</p> <p>Move units method</p> <p>@return boolean</p> <p>Properties:</p> <p>throws = Exception</p> <p>[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]</p>
<p>◆ nextPhase () : Phase Private</p> <p>Calculate the next phase</p> <p>@return Phase</p> <p>Properties:</p> <p>throws = Exception</p> <p>[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]</p>
<p>◆ nextPlayer () : void Protected</p> <p>Initializes the next Player's turn. Calculates the amount of Units to place.</p> <p>Properties:</p> <p>throws = Exception</p> <p>ea_guid = {806D8A15-D81F-4852-9BFA-E0988D68A2D8}</p> <p>[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]</p>
<p>◆ reinforce (territoryId : int) : boolean Public</p> <p>Reinforce method</p> <p>@return boolean</p> <p>Properties:</p> <p>throws = Exception</p> <p>[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]</p>
<p>◆ RiskGameModel () : Public</p> <p>The amount of Units left to Place in the Reinforcement Stage of the current Player's current turn.</p> <p>[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]</p>
<p>◆ saveGame (file : File) : boolean Public</p> <p>Save data method</p> <p>@return</p> <p>Properties:</p> <p>throws = IOException</p> <p>[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]</p>

OPERATIONS
<p>◆ selectStartingPlayer (list : List<Player>) : Player Public Random select the starting player <code>@return Player</code> <code>[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]</code></p>
<p>◆ transfer (fromID : int , toID : int , units : int) : boolean Public <code>@return boolean</code> Properties: <code>throws = Exception</code> <code>[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]</code></p>
<p>◆ transfer (from : Territory , to : Territory , units : int) : boolean Public Transfer method <code>@return boolean</code> Properties: <code>throws = Exception</code> <code>ea_guid = {23034260-4F89-40e8-BC43-DD7B2D6023D5}</code> <code>[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]</code></p>
<p>◆ winGame (winner : int) : void Private <code>[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]</code></p>

RiskGameController

Class in package 'Controller'

Az MVC architektura Controller részet megvalósító osztálya. A Controller a Modell és a View közötti kapcsolatért felelős. Rajta keresztül hajtják végre a View-on történt változások, melyek a Model-t érintik, illetve a Model a Controller-en keresztül módosítja a View-t.

RiskGameController
 Version 1.0 Phase 1.0 Proposed
 Szabó Dávid created on 2017. 03. 01.. Last modified 2017. 11. 22.
 Extends java.util.Observable

INCOMING STRUCTURAL RELATIONSHIPS
<p>⇒ Aggregation from JFXAttackView to RiskGameController <code>[Direction is 'Source -> Destination'.]</code></p>
<p>⇒ Aggregation from JFXAddPlayerView to RiskGameController <code>[Direction is 'Source -> Destination'.]</code></p>
<p>⇒ Aggregation from JFXTransferView to RiskGameController <code>[Direction is 'Source -> Destination'.]</code></p>

INCOMING STRUCTURAL RELATIONSHIPS	
⇒ Aggregation from RiskGameModel to RiskGameController	[Direction is 'Source -> Destination'.]
⇒ Aggregation from JFXMainView to RiskGameController	[Direction is 'Source -> Destination'.]
ATTRIBUTES	
◆ addPlayerView : JFXAddPlayerView Private	[Is static False. Containment is Not Specified.]
◆ attackResultView : JFXAttackResultView Private	[Is static False. Containment is Not Specified.]
◆ attackView : JFXAttackView Private	[Is static False. Containment is Not Specified.]
◆ cardGroup : Group Private	[Is static False. Containment is Not Specified.]
◆ cardScene : Scene Private	[Is static False. Containment is Not Specified.]
◆ cardView : JFXCardView Private	
CardView stuffs	[Is static False. Containment is Not Specified.]
◆ lastPhaseUpdate : Phase Private	[Is static False. Containment is Not Specified.]
◆ lastPlayerUpdate : int Private	[Is static False. Containment is Not Specified.]
◆ mainView : JFXMainView Private	
View objects	[Is static False. Containment is Not Specified.]
◆ model : RiskGameModel Private	[Is static False. Containment is Not Specified.]
◆ popupStage : Stage Private	[Is static False. Containment is Not Specified.]

ATTRIBUTES	
◆ preStage : Stage	Private Const Stages [Is static False. Containment is Not Specified.]
◆ previouslySelectedTerritory : int	Private = -1 [Is static False. Containment is Not Specified.]
◆ primaryStage : Stage	Private [Is static False. Containment is Not Specified.]
◆ titledPaneStage : Stage	Private [Is static False. Containment is Not Specified.]
◆ transferView : JFXTransferView	Private private JFXRiskCardView cardView; [Is static False. Containment is Not Specified.]
◆ victoryStage : Stage	Private [Is static False. Containment is Not Specified.]
◆ victoryView : JFXVictoryView	Private VictoryView stuffs [Is static False. Containment is Not Specified.]

ASSOCIATIONS	
◆ Association (direction: Source -> Destination)	Source: Public (Class) JFXViewBase Target: Protected controller (Class) RiskGameController

OPERATIONS	
◆ addLog (log : String) : void	Public A főablak naplójába küld egy új bejegyzést [Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ attackAccepted (defender : int , attacker : int , defenderUnits : int , attackerUnits : int) : void	Public [Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ attackRetreat () : void	Public [Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]

OPERATIONS	
◆ attackShowed (attacker : int , defender : int) : void	Public [Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ closePopupWindow () : void	Private [Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ convertToCountry (ID : int) : Country	Private [Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ convertToTerritoryID (country : Country) : int	Private [Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ countrySelected (country : Country) : void	Public [Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ getPhase () : Phase	Public [Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ getPreviouslySelectedTerritory () : int	Public [Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ getTerritoryData () : List<Territory>	Public [Is static True. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ isVictory () : boolean	Private [Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ loadGame (loadFile : File) : void	Public [Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ moveCardView () : void	Public [Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ nextPlayerSelected () : void	Public [Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ reinforce (territoryID : int , country : String) : void	Private
Properties:	
throws = Exception	[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ RiskGameController (stage : Stage) : Public	[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]

OPERATIONS	
◆ saveGame (saveFile : File) : void Public	[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ setModel (model : RiskGameModel) : void Public	Eltárolja a modellt és feliratkoztatja magát az eseményeire [Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ showAddPlayerView () : void Public	Add player view [Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ showAttackResultView () : void Private	[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ showAttackView (defenderID : int , attackerID : int) : void Private	Attack View [Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ showCardView () : void Public	[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ showMainView () : void Public	Main View [Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ showTransferView (fromID : int , toID : int) : void Private	Transfer View [Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ showVictoryView () : void Public	[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ switchToMain (map : Map<Color, String>) : void Public	Stage switch Pre -> Main [Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ transferAccepted (from : int , to : int , units : int) : void Public	[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ transferPhaseSelected () : void Public	[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]

OPERATIONS	
◆ UpdateAllTerritoriesOnMap () : void	Private [Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ UpdateCurrentDeck () : void	Private [Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ UpdateCurrentPhase () : void	Private [Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ UpdateCurrentPlayer () : void	Private [Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ UpdateTerritoryOnMap (territoryID : int) : void	Private [Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]

JFXAddPlayerView

Class in package 'View'

Ezen a View-en keresztul tortenik a leendo jatekosok hozzaadasa a jatekhoz.

JFXAddPlayerView
Version 1.0 Phase 1.0 Proposed
fodorad created on 2017. 03. 22.. Last modified 2017. 11. 22.
Extends JFXViewBase

OUTGOING STRUCTURAL RELATIONSHIPS	
↳ Aggregation from JFXAddPlayerView to RiskGameController	[Direction is 'Source -> Destination'.]
↳ Generalization from JFXAddPlayerView to JFXViewBase	[Direction is 'Source -> Destination'.]

ATTRIBUTES	
◆ btnAddPlayer : Button	Private Properties: annotations = @FXML [Is static False. Containment is Not Specified.]
◆ btnDeletePlayer4 : Button	Private Properties: annotations = @FXML [Is static False. Containment is Not Specified.]

ATTRIBUTES	
◆ btnDeletePlayer5 : Button	Private Properties: annotations = @FXML [Is static False. Containment is Not Specified.]
◆ btnDeletePlayer6 : Button	Private Properties: annotations = @FXML [Is static False. Containment is Not Specified.]
◆ btnReady : Button	Private Properties: annotations = @FXML [Is static False. Containment is Not Specified.]
◆ lblPlayer4 : Label	Private Properties: annotations = @FXML [Is static False. Containment is Not Specified.]
◆ lblPlayer5 : Label	Private Properties: annotations = @FXML [Is static False. Containment is Not Specified.]
◆ lblPlayer6 : Label	Private Properties: annotations = @FXML [Is static False. Containment is Not Specified.]
◆ stage : Stage	Package [Is static False. Containment is Not Specified.]
◆ tfList : ArrayList<TextField>	Private [Is static False. Containment is Not Specified.]
◆ tfPlayer1 : TextField	Private Properties: annotations = @FXML [Is static False. Containment is Not Specified.]
◆ tfPlayer2 : TextField	Private Properties: annotations = @FXML

ATTRIBUTES	
	[Is static False. Containment is Not Specified.]
◆ tfPlayer3 : TextField Private	
Properties: annotations = @FXML	[Is static False. Containment is Not Specified.]
◆ tfPlayer4 : TextField Private	
Properties: annotations = @FXML	[Is static False. Containment is Not Specified.]
◆ tfPlayer5 : TextField Private	
Properties: annotations = @FXML	[Is static False. Containment is Not Specified.]
◆ tfPlayer6 : TextField Private	
Properties: annotations = @FXML	[Is static False. Containment is Not Specified.]

OPERATIONS	
◆ addPlayerAction (event : ActionEvent) : void Protected	
Properties: annotations = @FXML	[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ finalize () : void Public	
Properties: throws = Throwable annotations = @Override	[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ initialize (location : URL , resources : ResourceBundle) : void Public	
Properties: annotations = @Override	[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ JFXAddPlayerView (stage : Stage) : Public	[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ readyOnAction (event : ActionEvent) : void Protected	

OPERATIONS	
Properties:	
annotations = @FXML	[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
 removePlayer4Action (event : ActionEvent) : void Protected	
Properties:	
annotations = @FXML	[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
 removePlayer5Action (event : ActionEvent) : void Protected	
Properties:	
annotations = @FXML	[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
 removePlayer6Action (event : ActionEvent) : void Protected	
Properties:	
annotations = @FXML	[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
 UpdateViewState (name : String) : void Public	[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]

JFXMainView

Class in package 'View'

Az MVC architektura View részet megvalósító osztálya. Ez a View a játék főkepernyője, ezen történik maga a játék (egysegek elhelyezése, csata stb.).

JFXMainView
Version 1.0 Phase 1.0 Proposed
Szabó Dávid created on 2017. 03. 01.. Last modified 2017. 11. 22.
Extends JFXViewBase

OUTGOING STRUCTURAL RELATIONSHIPS	
 Generalization from JFXMainView to JFXViewBase	[Direction is 'Source -> Destination'.]
 Aggregation from JFXMainView to RiskGameController	[Direction is 'Source -> Destination'.]

ATTRIBUTES	
 BASE_HEIGHT : double Private Const = 600	[Is static False. Containment is Not Specified.]

ATTRIBUTES
◆ BASE_WIDTH : double Private Const = 815 [Is static False. Containment is Not Specified.]
◆ borderPane : BorderPane Private Properties: annotations = @FXML [Is static False. Containment is Not Specified.]
◆ btnBattlePhase : Button Private Properties: annotations = @FXML [Is static False. Containment is Not Specified.]
◆ btnControlList : ArrayList<Button> Private [Is static False. Containment is Not Specified.]
◆ btnLoad : Button Private Properties: annotations = @FXML [Is static False. Containment is Not Specified.]
◆ btnLog : Button Private Properties: annotations = @FXML [Is static False. Containment is Not Specified.]
◆ btnNextPlayer : Button Private Properties: annotations = @FXML [Is static False. Containment is Not Specified.]
◆ btnReinforcePhase : Button Private Properties: annotations = @FXML [Is static False. Containment is Not Specified.]
◆ btnSave : Button Private Properties: annotations = @FXML [Is static False. Containment is Not Specified.]
◆ btnSettings : Button Private

ATTRIBUTES	
Properties: annotations = @FXML	[Is static False. Containment is Not Specified.]
◆ btnTransferPhase : Button Private	
Properties: annotations = @FXML	[Is static False. Containment is Not Specified.]
◆ centerStackPane : StackPane Private	
Properties: annotations = @FXML	[Is static False. Containment is Not Specified.]
◆ dimLog : Dimension Package	[Is static False. Containment is Not Specified.]
◆ imageView : ImageView Private	
Properties: annotations = @FXML	[Is static False. Containment is Not Specified.]
◆ isLogOpened : boolean Package	[Is static False. Containment is Not Specified.]
◆ isSLOpened : boolean Package	[Is static False. Containment is Not Specified.]
◆ lblPlayerName1 : Label Private	
Properties: annotations = @FXML	[Is static False. Containment is Not Specified.]
◆ lblPlayerName2 : Label Private	
Properties: annotations = @FXML	[Is static False. Containment is Not Specified.]
◆ lblPlayerName3 : Label Private	
Properties: annotations = @FXML	[Is static False. Containment is Not Specified.]
◆ lblPlayerName4 : Label Private	

ATTRIBUTES
<p>Properties: annotations = @FXML [Is static False. Containment is Not Specified.]</p>
<p>◆ lblPlayerName5 : Label Private Properties: annotations = @FXML [Is static False. Containment is Not Specified.]</p>
<p>◆ lblPlayerName6 : Label Private Properties: annotations = @FXML [Is static False. Containment is Not Specified.]</p>
<p>◆ lblPlayerNameList : ArrayList<Label> Private [Is static False. Containment is Not Specified.]</p>
<p>◆ pain : Pane Private Properties: annotations = @FXML [Is static False. Containment is Not Specified.]</p>
<p>◆ panePlayerColor1 : Pane Private Properties: annotations = @FXML [Is static False. Containment is Not Specified.]</p>
<p>◆ panePlayerColor2 : Pane Private Properties: annotations = @FXML [Is static False. Containment is Not Specified.]</p>
<p>◆ panePlayerColor3 : Pane Private Properties: annotations = @FXML [Is static False. Containment is Not Specified.]</p>
<p>◆ panePlayerColor4 : Pane Private Properties: annotations = @FXML [Is static False. Containment is Not Specified.]</p>
<p>◆ panePlayerColor5 : Pane Private</p>

ATTRIBUTES	
Properties:	
annotations = @FXML	[Is static False. Containment is Not Specified.]
panePlayerColor6 : Pane Private	
Properties:	
annotations = @FXML	[Is static False. Containment is Not Specified.]
panePlayerColorList : ArrayList<Pane> Private	[Is static False. Containment is Not Specified.]
stage : Stage Private	[Is static False. Containment is Not Specified.]
taLog : TextArea Private	
Properties:	
annotations = @FXML	[Is static False. Containment is Not Specified.]
world : World Private	[Is static False. Containment is Not Specified.]

OPERATIONS	
AddControllerListener (controller : RiskGameController) : void Public	
Properties:	
annotations = @Override	[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
appendLog (s : String) : void Public	[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
closeLog () : void Public	[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
finalize () : void Public	
Properties:	
throws = Throwable	
annotations = @Override	[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
fit (height : double , width : double) : void Public	[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]

OPERATIONS	
◆ <code>getRoot () : Parent</code>	Public Properties: <code>annotations = @Override</code> [Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ <code>getWorld () : World</code>	Public [Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ <code>handleLogButton (event : ActionEvent) : void</code>	Private Properties: <code>annotations = @FXML</code> [Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ <code>hangleLoadButton (event : ActionEvent) : void</code>	Private Properties: <code>annotations = @FXML</code> [Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ <code>hangleSaveButton (event : ActionEvent) : void</code>	Private Properties: <code>annotations = @FXML</code> [Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ <code>hangleSettingsButton (event : ActionEvent) : void</code>	Private Properties: <code>annotations = @FXML</code> [Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ <code>initialize (location : URL , resources : ResourceBundle) : void</code>	Public Properties: <code>annotations = @Override</code> [Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ <code>isLogOpened () : boolean</code>	Public [Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ <code>isSLOpened () : boolean</code>	Public [Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ <code>JFXMainView (stage : Stage) : Public</code>	[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ <code>openLog () : void</code>	Public [Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]

OPERATIONS
<ul style="list-style-type: none"> ◆ UpdateConnectedPlayers (playerList : List<String>) : void Public [Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
<ul style="list-style-type: none"> ◆ UpdateCurrentDeck (cards : List<Integer>) : void Public <p>TODO:</p> <p>[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]</p>
<ul style="list-style-type: none"> ◆ UpdateCurrentPhase (phase : Phase) : void Public [Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
<ul style="list-style-type: none"> ◆ UpdateCurrentPlayer (playerID : int) : void Public <p>@param player</p> <p>[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]</p>
<ul style="list-style-type: none"> ◆ updateIcons (attacker : int, defender : int) : void Public [Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
<ul style="list-style-type: none"> ◆ UpdateViewState (territory : Territory) : void Public <p>@param territories</p> <p>Properties: ea_guid = {0E811D72-EC4E-47ad-8B46-C39084EAD8F7} [Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]</p>
<ul style="list-style-type: none"> ◆ UpdateViewState (territories : List<Territory>) : void Public [Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]

<anonymous>

Note in package 'Application Start-up, Run and Shutdown'

Define Number Of Players

<anonymous>
Version 1.0 Phase 1.0 Proposed
Oliver created on 2017. 03. 23.. Last modified 2017. 03. 28.
Extends

INTERACTION MESSAGES

<p>✉ 1.0 '// Selects option' from 'Player 1' sent to 'JFXMainView'.</p> <p>Synchronous Call. Returns void.</p>	[Return is False. Iteration is False. New group is False.]
<p>✉ 1.1 'actionPerformed(ActionEvent)' from 'JFXMainView' sent to 'RiskGameController'.</p> <p>Synchronous Call.</p>	[Return is False. Iteration is False. New group is False. Life cycle is <none>.]
<p>✉ 1.2 '' from 'RiskGameController' sent to 'JFXAddPlayerView'.</p> <p>Synchronous Call. Returns void.</p>	[Return is False. Iteration is False. New group is False. Life cycle is New.]
<p>✉ 1.3 '// Add number of players' from 'Player 1' sent to 'JFXAddPlayerView'.</p> <p>Synchronous Call. Returns void.</p>	[Return is False. Iteration is False. New group is False.]
<p>✉ 1.4 'actionPerformed(ActionEvent)' from 'JFXAddPlayerView' sent to 'RiskGameController'.</p> <p>Asynchronous Call.</p>	[Return is False. Iteration is False. New group is False.]
<p>✉ 1.5 'setPlayerNumber(int)' from 'RiskGameController' sent to 'RiskGameModel'.</p> <p>Synchronous Call.</p>	[Return is False. Iteration is False. New group is False.]
<p>✉ 1.6 'update(Observable, Object)' from 'RiskGameModel' sent to 'RiskGameController'.</p> <p>Synchronous Call.</p>	[Return is False. Iteration is False. New group is False.]
<p>✉ 1.7 '' from 'RiskGameController' sent to 'JFXMainView'.</p> <p>Synchronous Call. Returns void.</p>	[Return is False. Iteration is False. New group is False.]

Game diagram

Interaction diagram in package 'Application Start-up, Run and Shutdown'

Game
Version 1.0
Oliver created on 2017. 04. 19.. Last modified 2017. 04. 19.

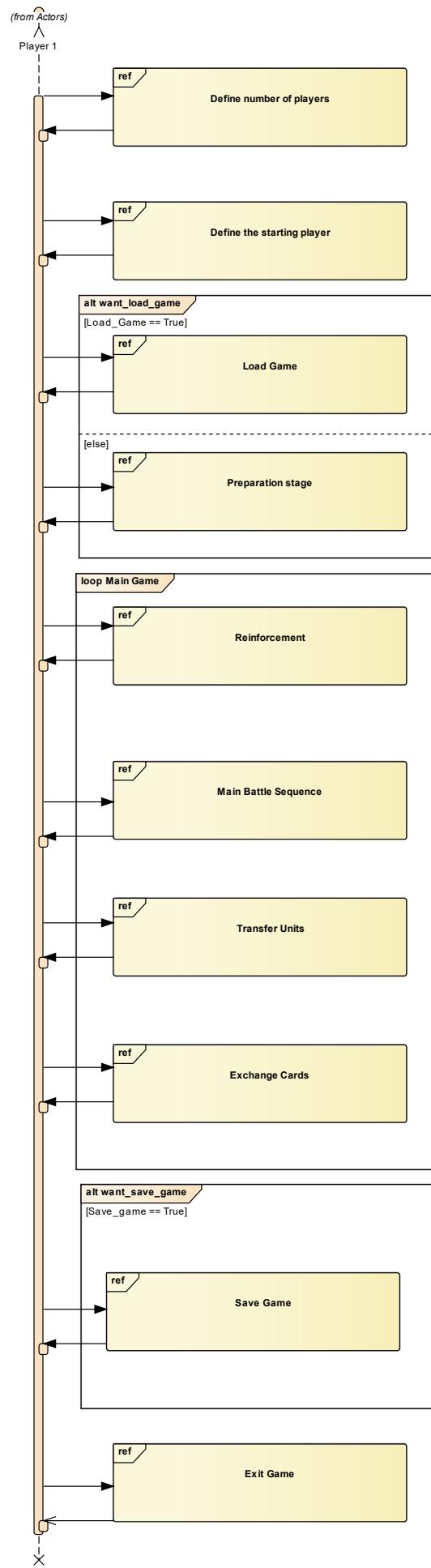


Figure 7: Game

Player 1*Actor in package 'Actors'*

Player 1
Version 1.0 Phase 1.0 Mandatory
Oliver created on 2017. 02. 21.. Last modified 2017. 03. 23.

OUTGOING BEHAVIORAL RELATIONSHIPS	
Name: //Selects 2 Territories	
↳ Sequence from Player 1 to JFXMainView	
Name: //select attacking units	
↳ Sequence from Player 1 to JFXAttackView	
↳ Sequence from Player 1 to Reinforcement	
↳ Sequence from Player 1 to Load Game	
↳ Sequence from Player 1 to Exit Game	
↳ Sequence from Player 1 to Save Game	
↳ Sequence from Player 1 to Exchange Cards	
↳ Sequence from Player 1 to Define the starting player	
↳ Sequence from Player 1 to Define number of players	
Name: // Selects option	
↳ Sequence from Player 1 to JFXMainView	
Name: // Add number of players	
↳ Sequence from Player 1 to JFXAddPlayerView	
Name: //Set a Save Name then Apply	
↳ Sequence from Player 1 to JFXMainView	
Name: // click to begin action	
↳ Sequence from Player 1 to JFXMainView	
Name: // click to connect	
↳ Sequence from Player 1 to JFXMainView	

OUTGOING BEHAVIORAL RELATIONSHIPS
Name: //End Turn ↳ Sequence from Player 1 to JFXMainView
Name: // Selects exit game option ↳ Sequence from Player 1 to JFXMainView
Name: // Select saved game file then apply ↳ Sequence from Player 1 to JFXMainView
↳ Sequence from Player 1 to Main Battle Sequence
Name: //init attack ↳ Sequence from Player 1 to JFXMainView
Name: //end turn ↳ Sequence from Player 1 to JFXMainView
↳ Sequence from Player 1 to Transfer Units
Name: // select Yes or No ↳ Sequence from Player 1 to JFXMainView
Name: //Selects Units then Apply ↳ Sequence from Player 1 to JFXTransferView
↳ Sequence from Player 1 to Preparation stage

INCOMING BEHAVIORAL RELATIONSHIPS
↗ Sequence from Transfer Units to Player 1
↗ Sequence from Save Game to Player 1
↗ Sequence from Main Battle Sequence to Player 1
↗ Sequence from Preparation stage to Player 1
↗ Sequence from Reinforcement to Player 1
↗ Sequence from Exchange Cards to Player 1

INCOMING BEHAVIORAL RELATIONSHIPS	
Name: // popup	
➡ Sequence from JFXMainView to Player 1	
Name: addTerritory()	
➡ Sequence from RiskGameModel to Player 1	
➡ Sequence from Define number of players to Player 1	
➡ Sequence from Exit Game to Player 1	
➡ Sequence from Define the starting player to Player 1	
➡ Sequence from Load Game to Player 1	

CONNECTORS	
↗ UseCaseLink	Source -> Destination
From:	Player 1 : Actor, Public
To:	Reinforcement : UseCase, Public
↗ UseCaseLink	Source -> Destination
From:	Player 1 : Actor, Public
To:	Transfer Units : UseCase, Public
↗ UseCaseLink	Source -> Destination
From:	Player 1 : Actor, Public
To:	Main Battle Sequence : UseCase, Public

ASSOCIATIONS	
↙ Association (direction: Unspecified)	
Source: Public (Actor) Player 1	Target: Public (UseCase) Exit Game

INTERACTION MESSAGES	
✉ 1.0 "	from 'Player 1' sent to 'Define number of players'.
Synchronous Call. Returns void.	[Return is False. Iteration is False. New group is False.]

✉ 1.1 " from 'Define number of players' sent to 'Player 1'.

Synchronous Call. Returns void.

[Return is False. Iteration is False. New group is False.]

✉ 1.2 " from 'Player 1' sent to 'Define the starting player'.

Synchronous Call. Returns void.

[Return is False. Iteration is False. New group is False.]

✉ 1.3 " from 'Define the starting player' sent to 'Player 1'.

Synchronous Call. Returns void.

[Return is False. Iteration is False. New group is False.]

✉ 1.4 " from 'Player 1' sent to 'Load Game'.

Synchronous Call. Returns void.

[Return is False. Iteration is False. New group is False.]

✉ 1.5 " from 'Load Game' sent to 'Player 1'.

Synchronous Call. Returns void.

[Return is False. Iteration is False. New group is False.]

✉ 1.6 " from 'Player 1' sent to 'Preparation stage'.

Synchronous Call. Returns void.

[Return is False. Iteration is False. New group is False.]

✉ 1.7 " from 'Preparation stage' sent to 'Player 1'.

Synchronous Call. Returns void.

[Return is False. Iteration is False. New group is False.]

✉ 1.8 " from 'Player 1' sent to 'Reinforcement'.

Synchronous Call. Returns void.

[Return is False. Iteration is False. New group is False.]

✉ 1.9 " from 'Reinforcement' sent to 'Player 1'.

Synchronous Call. Returns void.

[Return is False. Iteration is False. New group is False.]

✉ 1.10 " from 'Player 1' sent to 'Main Battle Sequence'.

Synchronous Call. Returns void.

[Return is False. Iteration is False. New group is False.]

✉ 1.11 " from 'Main Battle Sequence' sent to 'Player 1'.

Synchronous Call. Returns void.	[Return is False. Iteration is False. New group is False.]
✉ 1.12 " from 'Player 1' sent to 'Transfer Units'.	
Synchronous Call. Returns void.	[Return is False. Iteration is False. New group is False.]
✉ 1.13 " from 'Transfer Units' sent to 'Player 1'.	
Synchronous Call. Returns void.	[Return is False. Iteration is False. New group is False.]
✉ 1.14 " from 'Player 1' sent to 'Exchange Cards'.	
Synchronous Call. Returns void.	[Return is False. Iteration is False. New group is False.]
✉ 1.15 " from 'Exchange Cards' sent to 'Player 1'.	
Synchronous Call. Returns void.	[Return is False. Iteration is False. New group is False.]
✉ 1.16 " from 'Player 1' sent to 'Save Game'.	
Synchronous Call. Returns void.	[Return is False. Iteration is False. New group is False.]
✉ 1.17 " from 'Save Game' sent to 'Player 1'.	
Synchronous Call. Returns void.	[Return is False. Iteration is False. New group is False.]
✉ 1.18 " from 'Player 1' sent to 'Exit Game'.	
Synchronous Call. Returns void.	[Return is False. Iteration is False. New group is False.]
✉ 1.19 " from 'Exit Game' sent to 'Player 1'.	
Asynchronous Call. Returns void.	[Return is False. Iteration is False. New group is False. Life cycle is Delete.]

Load Game diagram

Interaction diagram in package 'Application Start-up, Run and Shutdown'

Load Game
Version 1.0
fodorad created on 2017. 03. 22.. Last modified 2017. 09. 21.

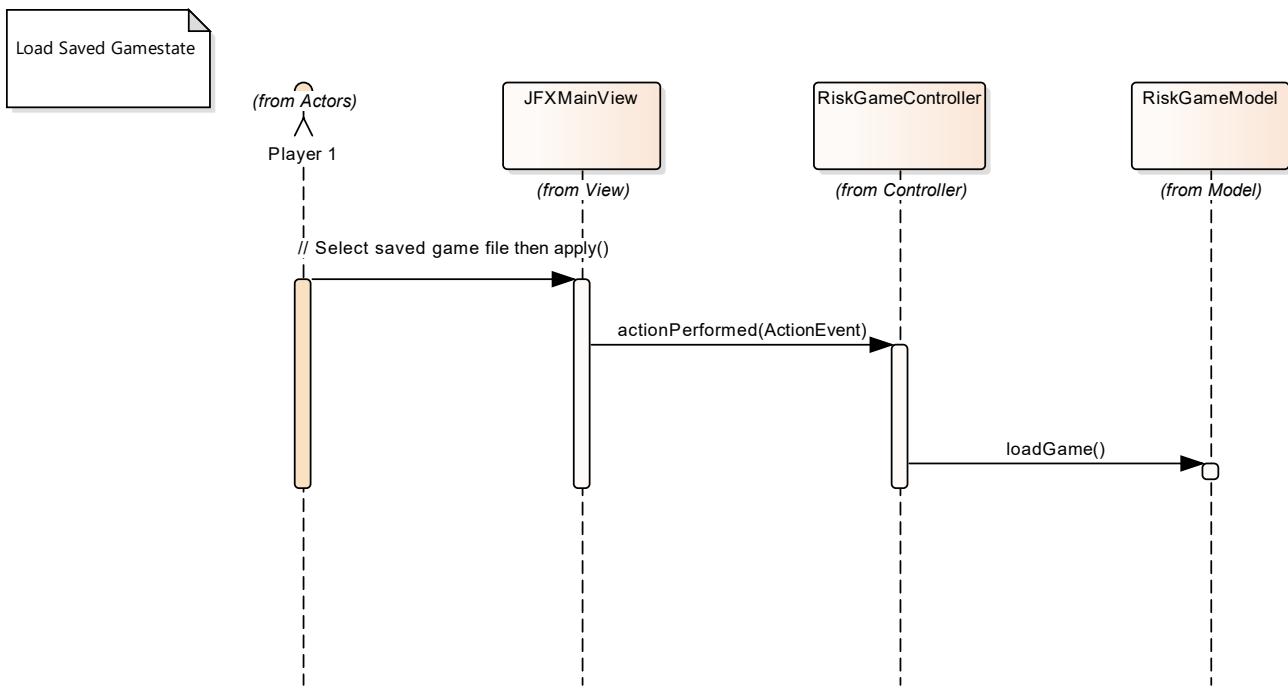


Figure 8: Load Game

Player 1

Actor in package 'Actors'

Player 1
 Version 1.0 Phase 1.0 Mandatory
 Oliver created on 2017. 02. 21.. Last modified 2017. 03. 23.

OUTGOING BEHAVIORAL RELATIONSHIPS
Name: //Selects 2 Territories Sequence from Player 1 to JFXMainView
Name: //select attacking units Sequence from Player 1 to JFXAttackView
Sequence from Player 1 to Reinforcement
Sequence from Player 1 to Load Game
Sequence from Player 1 to Exit Game
Sequence from Player 1 to Save Game
Sequence from Player 1 to Exchange Cards
Sequence from Player 1 to Define the starting player

OUTGOING BEHAVIORAL RELATIONSHIPS
<p>↳ Sequence from Player 1 to Define number of players</p>
<p>Name: // Selects option ↳ Sequence from Player 1 to JFXMainView</p>
<p>Name: // Add number of players ↳ Sequence from Player 1 to JFXAddPlayerView</p>
<p>Name: //Set a Save Name then Apply ↳ Sequence from Player 1 to JFXMainView</p>
<p>Name: // click to begin action ↳ Sequence from Player 1 to JFXMainView</p>
<p>Name: // click to connect ↳ Sequence from Player 1 to JFXMainView</p>
<p>Name: //End Turn ↳ Sequence from Player 1 to JFXMainView</p>
<p>Name: // Selects exit game option ↳ Sequence from Player 1 to JFXMainView</p>
<p>Name: // Select saved game file then apply ↳ Sequence from Player 1 to JFXMainView</p>
<p>↳ Sequence from Player 1 to Main Battle Sequence</p>
<p>Name: //init attack ↳ Sequence from Player 1 to JFXMainView</p>
<p>Name: //end turn ↳ Sequence from Player 1 to JFXMainView</p>
<p>↳ Sequence from Player 1 to Transfer Units</p>
<p>Name: // select Yes or No ↳ Sequence from Player 1 to JFXMainView</p>
<p>Name: //Selects Units then Apply ↳ Sequence from Player 1 to JFXTransferView</p>
<p>↳ Sequence from Player 1 to Preparation stage</p>

OUTGOING BEHAVIORAL RELATIONSHIPS
INCOMING BEHAVIORAL RELATIONSHIPS
<ul style="list-style-type: none"> ➡ Sequence from Transfer Units to Player 1
<ul style="list-style-type: none"> ➡ Sequence from Save Game to Player 1
<ul style="list-style-type: none"> ➡ Sequence from Main Battle Sequence to Player 1
<ul style="list-style-type: none"> ➡ Sequence from Preparation stage to Player 1
<ul style="list-style-type: none"> ➡ Sequence from Reinforcement to Player 1
<ul style="list-style-type: none"> ➡ Sequence from Exchange Cards to Player 1
<p>Name: //popup</p> <ul style="list-style-type: none"> ➡ Sequence from JFXMainView to Player 1
<p>Name: addTerritory()</p> <ul style="list-style-type: none"> ➡ Sequence from RiskGameModel to Player 1
<ul style="list-style-type: none"> ➡ Sequence from Define number of players to Player 1
<ul style="list-style-type: none"> ➡ Sequence from Exit Game to Player 1
<ul style="list-style-type: none"> ➡ Sequence from Define the starting player to Player 1
<ul style="list-style-type: none"> ➡ Sequence from Load Game to Player 1
CONNECTORS
<p>↗ UseCaseLink Source -> Destination</p> <p>From: Player 1 : Actor, Public</p> <p>To: Reinforcement : UseCase, Public</p>
<p>↗ UseCaseLink Source -> Destination</p> <p>From: Player 1 : Actor, Public</p> <p>To: Transfer Units : UseCase, Public</p>
<p>↗ UseCaseLink Source -> Destination</p> <p>From: Player 1 : Actor, Public</p> <p>To: Main Battle Sequence : UseCase, Public</p>

ASSOCIATIONS

↙ Association (direction: Unspecified)

Source: Public (Actor) Player 1

Target: Public (UseCase) Exit Game

RiskGameModel

Class in package 'Model'

Az MVC architektura Model részét megvalósító osztálya.

RiskGameModel
Version 1.0 Phase 1.0 Proposed
Szabi created on 2017. 02. 21.. Last modified 2017. 11. 22.

OUTGOING STRUCTURAL RELATIONSHIPS

⬅ Aggregation from RiskGameModel to RiskGameController

[Direction is 'Source -> Destination'.]

INCOMING STRUCTURAL RELATIONSHIPS

➡ Aggregation from Map to RiskGameModel

[Direction is 'Source -> Destination'.]

➡ Aggregation from Deck to RiskGameModel

[Direction is 'Source -> Destination'.]

ATTRIBUTES

◆ calculatedCircleNumber : int Private

[Is static False. Containment is Not Specified.]

◆ capturedThisTurn : Boolean Private

[Is static False. Containment is Not Specified.]

◆ circlenumber : int Private

[Is static False. Containment is Not Specified.]

◆ currentPlayer : int Private

[Is static False. Containment is Not Specified.]

◆ deck : Deck Private

[Is static False. Containment is Not Specified.]

◆ hasTransferred : boolean Private

[Is static False. Containment is Not Specified.]

ATTRIBUTES	
◆ map : Map	Private [Is static False. Containment is Not Specified.]
◆ miscnumber : int	Private [Is static False. Containment is Not Specified.]
◆ phase : Phase	Private [Is static False. Containment is Not Specified.]
◆ playersList : List<Player>	Private [Is static False. Containment is Not Specified.]
◆ r : Random	Private = new Random() [Is static False. Containment is Not Specified.]
◆ waitForUnitsTemp : Territory	Private = new Territory[2] [Is static False. Containment is Not Specified.]

ASSOCIATIONS	
↙ Association (direction: Source -> Destination)	Source: Public (Class) RiskGameModel Target: Private waitForUnitsTemp (Class) Territory
↙ Association (direction: Source -> Destination)	Source: Public (Class) RiskGameModel Target: Private playersList (Class) Player Cardinality: [0..*]
↙ Association (direction: Source -> Destination)	Source: Public (Class) RiskGameModel Target: Public deck (Class) Player

OPERATIONS	
◆ addPlayer (name : String , color : Color) : boolean	Public
Add player with name and color @return boolean	[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ addPlayerToPlayerList (player : Player) : void	Public
Adder for playerList Properties: throws = Exception	

OPERATIONS	
<code>ea_guid = {45684F69-D727-4875-BEF7-5232B4E9D1C9}</code>	[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
 <code>addPlayerToPlayerList (map : java.util.Map<Color, String>) : void</code> Public	Pairs overload
Properties: <code>throws = Exception</code>	[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
 <code>attackTerritoryWithResult (defenderID : int, attackerID : int, defendUnits : int, attackUnits : int) : AttackResult</code> Public	Return the result of the attack If the attacker have more than 3 units, the fighting attacker units will reduce to 3 If the defender have more than 2 units, the fighting defender units will reduce to 2
Properties: <code>throws = Exception</code>	[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
 <code>calculateCircle () : void</code> Private	
Properties: <code>throws = Exception</code>	[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
 <code>checkAttackPossible (defender : Territory, attacker : Territory, defendUnits : int, attackUnits : int) : boolean</code> Public	Csak akkor tĂˇmadhat, ha a terĂĽleten elegendL' egysĂ©ge van. Igazzal tĂ©r vissza, ha a tamadas lehetsĂ©ges. @return boolean
	[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
 <code>checkAttackPossible (defenderID : int, attackerID : int) : boolean</code> Public	Attack checking method @return
	[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
 <code>checkIfCapturedAndConquer (defender : Territory) : boolean</code> Public	
Capture and conquer checking method @return boolean	
Properties: <code>ea_guid = {BFF4F3B6-2538-4778-9372-EAA3B468EDD8}</code>	[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
 <code>checkIfTerrotiryIsEmpty (territory : Territory) : boolean</code> Public	
Empty territory checking method @return boolean	
Properties: <code>ea_guid = {BE4F4E07-A4C4-4e91-9F00-EF1EEB65D26C}</code>	

OPERATIONS	
[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]	
<p>◆ checkTransferPossible (from : int , to : int) : boolean Public</p> <p>Transfer checking method @return boolean</p> <p>[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]</p>	
<p>◆ endBattlePhase () : boolean Public</p> <p>Battle @return boolean</p> <p>Properties: throws = Exception</p> <p>[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]</p>	
<p>◆ endTurn () : int Public</p> <p>Kor vege Returns next player's id @return int</p> <p>Properties: throws = Exception ea_guid = {F5F8D5D6-8997-43de-A5AD-DDDC647E775F}</p> <p>[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]</p>	
<p>◆ getCurrentPlayer () : int Public</p> <p>Getter for current player @return int</p> <p>[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]</p>	
<p>◆ getPhase () : Phase Public</p> <p>Getter for Phase @return phase</p> <p>[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]</p>	
<p>◆ getPlayerCards (id : int) : List<Integer> Public</p> <p>[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]</p>	
<p>◆ getPlayerName (id : int) : String Public</p> <p>Getter for Player's name @return String</p> <p>[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]</p>	
<p>◆ getPlayers () : List<String> Public</p> <p>Getter for Players list @return List<Player></p> <p>[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]</p>	

OPERATIONS
<p>◆ getTerritory (id : int) : Territory Public Getter for Territory @return Territory [Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]</p>
<p>◆ getUnitsLeftToReinforce (playerID : int) : int Public Getter for left units in reinforce phase @return int [Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]</p>
<p>◆ loadGame (file : File) : boolean Public Load data method @return boolean Properties: throws = Exception [Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]</p>
<p>◆ moveUnits (units : int) : boolean Public Move units method @return boolean Properties: throws = Exception [Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]</p>
<p>◆ nextPhase () : Phase Private Calculate the next phase @return Phase Properties: throws = Exception [Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]</p>
<p>◆ nextPlayer () : void Protected Initializes the next Player's turn. Calculates the amount of Units to place. Properties: throws = Exception ea_guid = {806D8A15-D81F-4852-9BFA-E0988D68A2D8} [Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]</p>
<p>◆ reinforce (territoryId : int) : boolean Public Reinforce method @return boolean</p>

OPERATIONS	
Properties:	
throws = Exception	[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ RiskGameModel () : Public	
The amount of Units left to Place in the Reinforcement Stage of the current Player's current turn.	
[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]	
◆ saveGame (file : File) : boolean Public	
Save data method	
@return	
Properties:	
throws = IOException	[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ selectStartingPlayer (list : List<Player>) : Player Public	
Random select the starting player	
@return Player	[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ transfer (fromID : int, toID : int, units : int) : boolean Public	
@return boolean	
Properties:	
throws = Exception	[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ transfer (from : Territory, to : Territory, units : int) : boolean Public	
Transfer method	
@return boolean	
Properties:	
throws = Exception	[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
ea_guid = {23034260-4F89-40e8-BC43-DD7B2D6023D5}	
[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]	
◆ winGame (winner : int) : void Private	
[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]	

RiskGameController

Class in package 'Controller'

Az MVC architektura Controller részet megvalósító osztálya. A Controller a Modell és a View kozotti kapcsolatért felelős. Rajta keresztül hajtódnak vegre a View-on történo változások, melyek a Model-t érintik, illetve a Model a Controller-en keresztül módosítja a View-t.

RiskGameController
Version 1.0 Phase 1.0 Proposed
Szabó Dávid created on 2017. 03. 01.. Last modified 2017. 11. 22.
Extends java.util.Observable

INCOMING STRUCTURAL RELATIONSHIPS

- ⇒ Aggregation from JFXAttackView to RiskGameController [Direction is 'Source -> Destination'.]
- ⇒ Aggregation from JFXAddPlayerView to RiskGameController [Direction is 'Source -> Destination'.]
- ⇒ Aggregation from JFXTransferView to RiskGameController [Direction is 'Source -> Destination'.]
- ⇒ Aggregation from RiskGameModel to RiskGameController [Direction is 'Source -> Destination'.]
- ⇒ Aggregation from JFXMainView to RiskGameController [Direction is 'Source -> Destination'.]

ATTRIBUTES

- ◆ addPlayerView : JFXAddPlayerView Private [Is static False. Containment is Not Specified.]
- ◆ attackResultView : JFXAttackResultView Private [Is static False. Containment is Not Specified.]
- ◆ attackView : JFXAttackView Private [Is static False. Containment is Not Specified.]
- ◆ cardGroup : Group Private [Is static False. Containment is Not Specified.]
- ◆ cardScene : Scene Private [Is static False. Containment is Not Specified.]
- ◆ cardView : JFXCardView Private CardView stuffs [Is static False. Containment is Not Specified.]
- ◆ lastPhaseUpdate : Phase Private [Is static False. Containment is Not Specified.]

ATTRIBUTES	
◆ lastPlayerUpdate : int	Private [Is static False. Containment is Not Specified.]
◆ mainView : JFXMainView	Private View objects [Is static False. Containment is Not Specified.]
◆ model : RiskGameModel	Private [Is static False. Containment is Not Specified.]
◆ popupStage : Stage	Private [Is static False. Containment is Not Specified.]
◆ preStage : Stage	Private Const Stages [Is static False. Containment is Not Specified.]
◆ previouslySelectedTerritory : int	Private = -1 [Is static False. Containment is Not Specified.]
◆ primaryStage : Stage	Private [Is static False. Containment is Not Specified.]
◆ titledPaneStage : Stage	Private [Is static False. Containment is Not Specified.]
◆ transferView : JFXTransferView	Private private JFXRiskCardView cardView; [Is static False. Containment is Not Specified.]
◆ victoryStage : Stage	Private [Is static False. Containment is Not Specified.]
◆ victoryView : JFXVictoryView	Private VictoryView stuffs [Is static False. Containment is Not Specified.]

ASSOCIATIONS	
◆ Association (direction: Source -> Destination)	
Source: Public (Class) JFXViewBase	Target: Protected controller (Class) RiskGameController

OPERATIONS	
◆ addLog (log : String) : void	Public A főablak naplójába küld egy új bejegyzést [Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ attackAccepted (defender : int, attacker : int, defenderUnits : int, attackerUnits : int) : void	Public [Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ attackRetreat () : void	Public [Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ attackShowed (attacker : int, defender : int) : void	Public [Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ closePopupWindow () : void	Private [Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ convertToCountry (ID : int) : Country	Private [Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ convertToTerritoryID (country : Country) : int	Private [Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ countrySelected (country : Country) : void	Public [Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ getPhase () : Phase	Public [Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ getPreviouslySelectedTerritory () : int	Public [Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ getTerritoryData () : List<Territory>	Public [Is static True. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ isVictory () : boolean	Private [Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ loadGame (loadFile : File) : void	Public [Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ moveCardView () : void	Public [Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]

OPERATIONS	
◆ nextPlayerSelected () : void	Public [Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ reinforce (territoryID : int , country : String) : void	Private Properties: throws = Exception [Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ RiskGameController (stage : Stage) : Public	[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ saveGame (saveFile : File) : void	Public [Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ setModel (model : RiskGameModel) : void	Public Eltárolja a modellt és feliratkoztatja magát az eseményeire [Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ showAddPlayerView () : void	Public Add player view [Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ showAttackResultView () : void	Private [Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ showAttackView (defenderID : int , attackerID : int) : void	Private Attack View [Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ showCardView () : void	Public [Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ showMainView () : void	Public Main View [Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ showTransferView (fromID : int , toID : int) : void	Private Transfer View [Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ showVictoryView () : void	Public [Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]

OPERATIONS	
◆ switchToMain (map : Map<Color, String>) : void Public	Stage switch Pre -> Main [Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ transferAccepted (from : int , to : int , units : int) : void Public	[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ transferPhaseSelected () : void Public	[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ UpdateAllTerritoriesOnMap () : void Private	[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ UpdateCurrentDeck () : void Private	[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ UpdateCurrentPhase () : void Private	[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ UpdateCurrentPlayer () : void Private	[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ UpdateTerritoryOnMap (territoryID : int) : void Private	[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]

JFXMainView

Class in package 'View'

Az MVC architektura View részet megvalósító osztalya. Ez a View a játék főkepernyője, ezen történik maga a játék (egysegek elhelyezése, csata stb.).

JFXMainView
Version 1.0 Phase 1.0 Proposed
Szabó Dávid created on 2017. 03. 01.. Last modified 2017. 11. 22.
Extends JFXViewBase

OUTGOING STRUCTURAL RELATIONSHIPS	
↳ Generalization from JFXMainView to JFXViewBase	[Direction is 'Source -> Destination'.]
↳ Aggregation from JFXMainView to RiskGameController	[Direction is 'Source -> Destination'.]

ATTRIBUTES
◆ BASE_HEIGHT : double Private Const = 600 [Is static False. Containment is Not Specified.]
◆ BASE_WIDTH : double Private Const = 815 [Is static False. Containment is Not Specified.]
◆ borderPane : BorderPane Private Properties: annotations = @FXML [Is static False. Containment is Not Specified.]
◆ btnBattlePhase : Button Private Properties: annotations = @FXML [Is static False. Containment is Not Specified.]
◆ btnControlList : ArrayList<Button> Private [Is static False. Containment is Not Specified.]
◆ btnLoad : Button Private Properties: annotations = @FXML [Is static False. Containment is Not Specified.]
◆ btnLog : Button Private Properties: annotations = @FXML [Is static False. Containment is Not Specified.]
◆ btnNextPlayer : Button Private Properties: annotations = @FXML [Is static False. Containment is Not Specified.]
◆ btnReinforcePhase : Button Private Properties: annotations = @FXML [Is static False. Containment is Not Specified.]
◆ btnSave : Button Private Properties: annotations = @FXML [Is static False. Containment is Not Specified.]

ATTRIBUTES
<p>◆ btnSettings : Button Private</p> <p>Properties:</p> <p> annotations = @FXML</p> <p>[Is static False. Containment is Not Specified.]</p>
<p>◆ btnTransferPhase : Button Private</p> <p>Properties:</p> <p> annotations = @FXML</p> <p>[Is static False. Containment is Not Specified.]</p>
<p>◆ centerStackPane : StackPane Private</p> <p>Properties:</p> <p> annotations = @FXML</p> <p>[Is static False. Containment is Not Specified.]</p>
<p>◆ dimLog : Dimension Package</p> <p>[Is static False. Containment is Not Specified.]</p>
<p>◆ imageView : ImageView Private</p> <p>Properties:</p> <p> annotations = @FXML</p> <p>[Is static False. Containment is Not Specified.]</p>
<p>◆ isLogOpened : boolean Package</p> <p>[Is static False. Containment is Not Specified.]</p>
<p>◆ isSLOpened : boolean Package</p> <p>[Is static False. Containment is Not Specified.]</p>
<p>◆ lblPlayerName1 : Label Private</p> <p>Properties:</p> <p> annotations = @FXML</p> <p>[Is static False. Containment is Not Specified.]</p>
<p>◆ lblPlayerName2 : Label Private</p> <p>Properties:</p> <p> annotations = @FXML</p> <p>[Is static False. Containment is Not Specified.]</p>
<p>◆ lblPlayerName3 : Label Private</p> <p>Properties:</p> <p> annotations = @FXML</p>

ATTRIBUTES	
	[Is static False. Containment is Not Specified.]
◆ lblPlayerName4 : Label Private	
Properties: annotations = @FXML	[Is static False. Containment is Not Specified.]
◆ lblPlayerName5 : Label Private	
Properties: annotations = @FXML	[Is static False. Containment is Not Specified.]
◆ lblPlayerName6 : Label Private	
Properties: annotations = @FXML	[Is static False. Containment is Not Specified.]
◆ lblPlayerNameList : ArrayList<Label> Private	[Is static False. Containment is Not Specified.]
◆ pain : Pane Private	
Properties: annotations = @FXML	[Is static False. Containment is Not Specified.]
◆ panePlayerColor1 : Pane Private	
Properties: annotations = @FXML	[Is static False. Containment is Not Specified.]
◆ panePlayerColor2 : Pane Private	
Properties: annotations = @FXML	[Is static False. Containment is Not Specified.]
◆ panePlayerColor3 : Pane Private	
Properties: annotations = @FXML	[Is static False. Containment is Not Specified.]
◆ panePlayerColor4 : Pane Private	
Properties: annotations = @FXML	[Is static False. Containment is Not Specified.]

ATTRIBUTES	
◆ panePlayerColor5 : Pane Private	
Properties:	
annotations = @FXML	[Is static False. Containment is Not Specified.]
◆ panePlayerColor6 : Pane Private	
Properties:	
annotations = @FXML	[Is static False. Containment is Not Specified.]
◆ panePlayerColorList : ArrayList<Pane> Private	[Is static False. Containment is Not Specified.]
◆ stage : Stage Private	[Is static False. Containment is Not Specified.]
◆ taLog : TextArea Private	
Properties:	
annotations = @FXML	[Is static False. Containment is Not Specified.]
◆ world : World Private	[Is static False. Containment is Not Specified.]

OPERATIONS	
◆ AddControllerListener (controller : RiskGameController) : void Public	
Properties:	
annotations = @Override	[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ appendLog (s : String) : void Public	[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ closeLog () : void Public	[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ finalize () : void Public	
Properties:	
throws = Throwable	
annotations = @Override	[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]

OPERATIONS	
◆ fit (height : double , width : double) : void	Public [Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ getRoot () : Parent	Public Properties: annotations = @Override [Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ getWorld () : World	Public [Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ handleLogButton (event : ActionEvent) : void	Private Properties: annotations = @FXML [Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ handleLoadButton (event : ActionEvent) : void	Private Properties: annotations = @FXML [Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ handleSaveButton (event : ActionEvent) : void	Private Properties: annotations = @FXML [Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ handleSettingsButton (event : ActionEvent) : void	Private Properties: annotations = @FXML [Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ initialize (location : URL , resources : ResourceBundle) : void	Public Properties: annotations = @Override [Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ isLogOpened () : boolean	Public [Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ isSLOpened () : boolean	Public [Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ JFXMainView (stage : Stage) : Public	[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]

OPERATIONS	
◆ openLog () : void Public	[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ UpdateConnectedPlayers (playerList : List<String>) : void Public	[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ UpdateCurrentDeck (cards : List<Integer>) : void Public	
TODO:	[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ UpdateCurrentPhase (phase : Phase) : void Public	[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ UpdateCurrentPlayer (playerID : int) : void Public	
@param player	[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ updateIcons (attacker : int, defender : int) : void Public	[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ UpdateViewState (territory : Territory) : void Public	
@param territories	
Properties:	
ea_guid = {0E811D72-EC4E-47ad-8B46-C39084EAD8F7}	[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ UpdateViewState (territories : List<Territory>) : void Public	[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]

<anonymous>

Note in package 'Application Start-up, Run and Shutdown'

Load Saved Gamestate

<anonymous>
Version 1.0 Phase 1.0 Proposed
Oliver created on 2017. 03. 28.. Last modified 2017. 03. 28.
Extends

INTERACTION MESSAGES

1.0 '/* Select saved game file then apply' from 'Player 1' sent to 'JFXMainView'.

Synchronous Call. Returns void.

[Return is False. Iteration is False. New group is False.]

1.1 'actionPerformed(ActionEvent)' from 'JFXMainView' sent to 'RiskGameController'.

Synchronous Call.

[Return is False. Iteration is False. New group is False.]

1.2 'loadGame()' from 'RiskGameController' sent to 'RiskGameModel'.

Synchronous Call.

[Return is False. Iteration is False. New group is False.]

Check If Captured And Conquer

InteractionOccurrence in package 'Application Start-up, Run and Shutdown'

Check If Captured And Conquer
Version 1.0 Phase 1.0 Proposed

Oliver created on 2017. 04. 19.. Last modified 2017. 04. 19.

Define number of players

InteractionOccurrence in package 'Application Start-up, Run and Shutdown'

Define number of players
Version 1.0 Phase 1.0 Proposed

Oliver created on 2017. 04. 19.. Last modified 2017. 04. 19.

OUTGOING BEHAVIORAL RELATIONSHIPS

Sequence from Define number of players to Player 1

INCOMING BEHAVIORAL RELATIONSHIPS

Sequence from Player 1 to Define number of players

Define the starting player

InteractionOccurrence in package 'Application Start-up, Run and Shutdown'

Define the starting player
Version 1.0 Phase 1.0 Proposed

Oliver created on 2017. 04. 19.. Last modified 2017. 04. 19.

OUTGOING BEHAVIORAL RELATIONSHIPS

OUTGOING BEHAVIORAL RELATIONSHIPS

 Sequence from Define the starting player to Player 1

INCOMING BEHAVIORAL RELATIONSHIPS

 Sequence from Player 1 to Define the starting player

Exchange Cards

Interaction Occurrence in package 'Application Start-up, Run and Shutdown'

Exchange Cards

Version 1.0 Phase 1.0 Proposed

Oliver created on 2017. 04. 19.. Last modified 2017. 04. 19.

OUTGOING BEHAVIORAL RELATIONSHIPS

 Sequence from Exchange Cards to Player 1

INCOMING BEHAVIORAL RELATIONSHIPS

 Sequence from Player 1 to Exchange Cards

Exit Game

Interaction Occurrence in package 'Application Start-up, Run and Shutdown'

Exit Game

Version 1.0 Phase 1.0 Proposed

Oliver created on 2017. 04. 19.. Last modified 2017. 04. 19.

OUTGOING BEHAVIORAL RELATIONSHIPS

 Sequence from Exit Game to Player 1

INCOMING BEHAVIORAL RELATIONSHIPS

 Sequence from Player 1 to Exit Game

Load Game

Interaction Occurrence in package 'Application Start-up, Run and Shutdown'

Load Game

Version 1.0 Phase 1.0 Proposed

Oliver created on 2017. 04. 19.. Last modified 2017. 04. 19.

OUTGOING BEHAVIORAL RELATIONSHIPS

 Sequence from Load Game to Player 1

INCOMING BEHAVIORAL RELATIONSHIPS

 Sequence from Player 1 to Load Game

Main Battle Sequence

InteractionOccurrence in package 'Application Start-up, Run and Shutdown'

Main Battle Sequence

Version 1.0 Phase 1.0 Proposed

Oliver created on 2017. 04. 19.. Last modified 2017. 04. 19.

OUTGOING BEHAVIORAL RELATIONSHIPS

 Sequence from Main Battle Sequence to Player 1

INCOMING BEHAVIORAL RELATIONSHIPS

 Sequence from Player 1 to Main Battle Sequence

Main Game

InteractionFragment in package 'Application Start-up, Run and Shutdown'

Main Game

Version 1.0 Phase 1.0 Proposed

Oliver created on 2017. 04. 19.. Last modified 2017. 04. 19.

Preparation stage

InteractionOccurrence in package 'Application Start-up, Run and Shutdown'

Preparation stage

Version 1.0 Phase 1.0 Proposed

Oliver created on 2017. 04. 19.. Last modified 2017. 04. 19.

OUTGOING BEHAVIORAL RELATIONSHIPS

 Sequence from Preparation stage to Player 1

INCOMING BEHAVIORAL RELATIONSHIPS

 Sequence from Player 1 to Preparation stage

Reinforcement

InteractionOccurrence in package 'Application Start-up, Run and Shutdown'

Reinforcement
Version 1.0 Phase 1.0 Proposed
Oliver created on 2017. 04. 19.. Last modified 2017. 04. 19.

OUTGOING BEHAVIORAL RELATIONSHIPS

↳ Sequence from Reinforcement to Player 1

INCOMING BEHAVIORAL RELATIONSHIPS

↗ Sequence from Player 1 to Reinforcement

Save Game

InteractionOccurrence in package 'Application Start-up, Run and Shutdown'

Save Game
Version 1.0 Phase 1.0 Proposed
Oliver created on 2017. 04. 19.. Last modified 2017. 04. 19.

OUTGOING BEHAVIORAL RELATIONSHIPS

↳ Sequence from Save Game to Player 1

INCOMING BEHAVIORAL RELATIONSHIPS

↗ Sequence from Player 1 to Save Game

Transfer Units

InteractionOccurrence in package 'Application Start-up, Run and Shutdown'

Transfer Units
Version 1.0 Phase 1.0 Proposed
Oliver created on 2017. 04. 19.. Last modified 2017. 04. 19.

OUTGOING BEHAVIORAL RELATIONSHIPS

↳ Sequence from Transfer Units to Player 1

INCOMING BEHAVIORAL RELATIONSHIPS

↗ Sequence from Player 1 to Transfer Units

INCOMING BEHAVIORAL RELATIONSHIPS**want_load_game***InteractionFragment in package 'Application Start-up, Run and Shutdown'*

want_load_game

Version 1.0 Phase 1.0 Proposed

Oliver created on 2017. 04. 19.. Last modified 2017. 04. 19.

want_save_game*InteractionFragment in package 'Application Start-up, Run and Shutdown'*

want_save_game

Version 1.0 Phase 1.0 Proposed

Oliver created on 2017. 04. 19.. Last modified 2017. 04. 19.

Game Stage

Package in package 'Dynamic View'

Game Stage
Version 1.0 Phase 1.0 Proposed
Szabó Dávid created on 2017. 03. 22.. Last modified 2017. 03. 28.

Check If Captured And Conquer diagram

Interaction diagram in package 'Game Stage'

Check If Captured And Conquer
Version 1.0
Szabi created on 2017. 03. 07.. Last modified 2017. 03. 28.

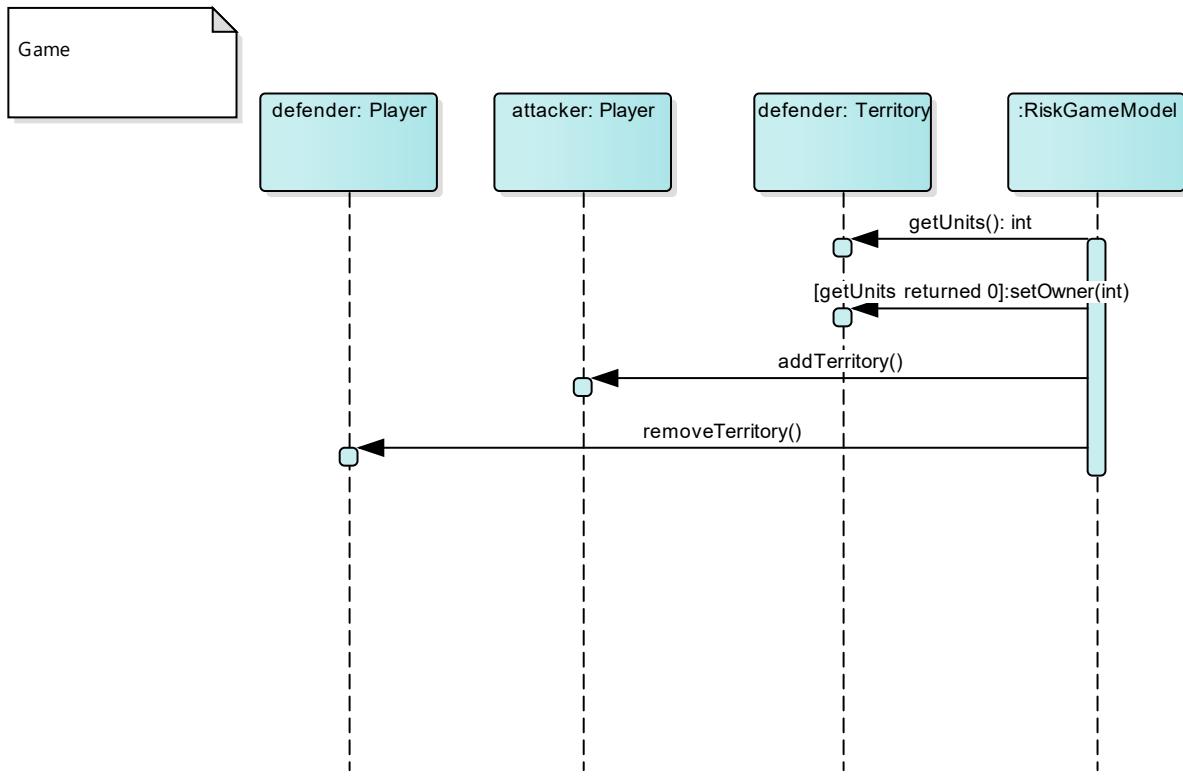


Figure 9: Check If Captured And Conquer

<anonymous>

Note in package 'Game Stage'

Game

<anonymous>
Version 1.0 Phase 1.0 Proposed
Oliver created on 2017. 03. 28.. Last modified 2017. 03. 28.
Extends

INTERACTION MESSAGES

<p>✉ 1.0 'getUnits()' from ':RiskGameModel' sent to 'defender: Territory'.</p> <p>Synchronous Call. Returns int.</p> <p>[Return is False. Iteration is False. New group is False.]</p>	
<p>✉ 1.1 'setOwner(int)' from ':RiskGameModel' sent to 'defender: Territory'.</p> <p>When "getUnits returned 0". Synchronous Call. Returns void.</p> <p>[Return is False. Iteration is False. New group is False.]</p>	
<p>✉ 1.2 'addTerritory() from ':RiskGameModel' sent to 'attacker: Player'.</p> <p>Synchronous Call.</p> <p>territoryCount++</p> <p>[Return is False. Iteration is False. New group is False.]</p>	
<p>✉ 1.3 'removeTerritory()' from ':RiskGameModel' sent to 'defender: Player'.</p> <p>Synchronous Call.</p> <p>[Return is False. Iteration is False. New group is False.]</p>	

Exchange Cards diagram

Interaction diagram in package 'Game Stage'

Exchange Cards
Version 1.0

Szabi created on 2017. 03. 29.. Last modified 2017. 09. 21.

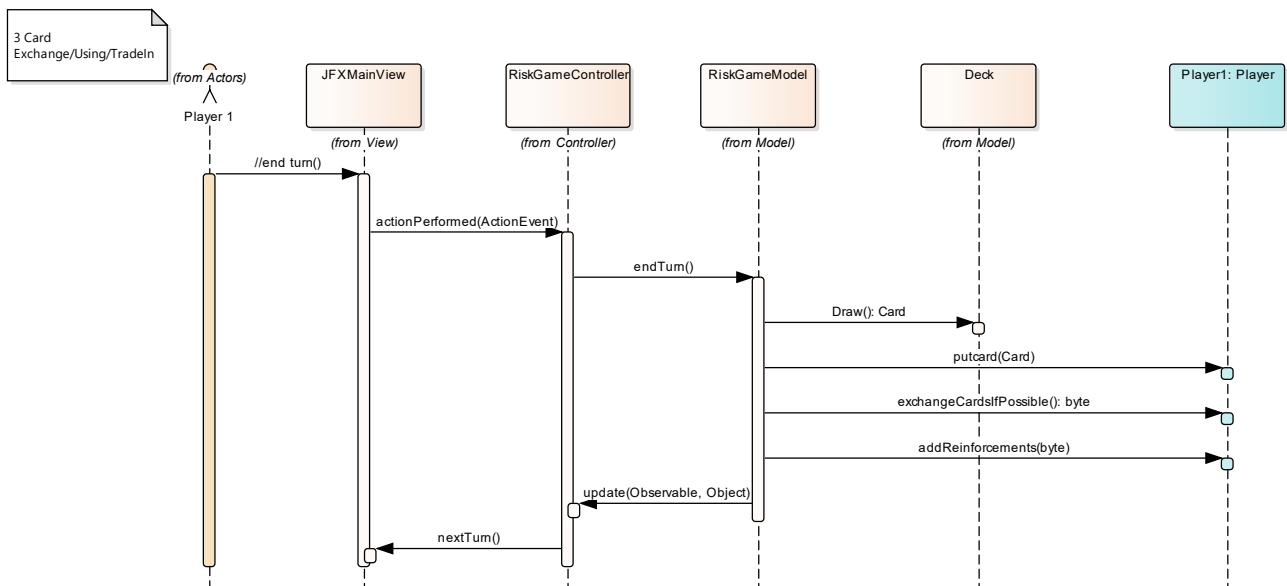


Figure 10: Exchange Cards

Player 1

Actor in package 'Actors'

Player 1
Version 1.0 Phase 1.0 Mandatory
Oliver created on 2017. 02. 21.. Last modified 2017. 03. 23.

OUTGOING BEHAVIORAL RELATIONSHIPS

Name: //Selects 2 Territories

↳ Sequence from Player 1 to JFXMainView

Name: //select attacking units

↳ Sequence from Player 1 to JFXAttackView

↳ Sequence from Player 1 to Reinforcement

↳ Sequence from Player 1 to Load Game

↳ Sequence from Player 1 to Exit Game

↳ Sequence from Player 1 to Save Game

↳ Sequence from Player 1 to Exchange Cards

↳ Sequence from Player 1 to Define the starting player

↳ Sequence from Player 1 to Define number of players

Name: // Selects option

↳ Sequence from Player 1 to JFXMainView

Name: // Add number of players

↳ Sequence from Player 1 to JFXAddPlayerView

Name: //Set a Save Name then Apply

↳ Sequence from Player 1 to JFXMainView

Name: // click to begin action

↳ Sequence from Player 1 to JFXMainView

Name: // click to connect

↳ Sequence from Player 1 to JFXMainView

Name: //End Turn

↳ Sequence from Player 1 to JFXMainView

OUTGOING BEHAVIORAL RELATIONSHIPS
Name: // Selects exit game option ➡ Sequence from Player 1 to JFXMainView
Name: // Select saved game file then apply ➡ Sequence from Player 1 to JFXMainView
➡ Sequence from Player 1 to Main Battle Sequence
Name: //init attack ➡ Sequence from Player 1 to JFXMainView
➡ Sequence from Player 1 to Transfer Units
Name: // select Yes or No ➡ Sequence from Player 1 to JFXMainView
Name: //Selects Units then Apply ➡ Sequence from Player 1 to JFXTransferView
➡ Sequence from Player 1 to Preparation stage

INCOMING BEHAVIORAL RELATIONSHIPS
➡ Sequence from Transfer Units to Player 1
➡ Sequence from Save Game to Player 1
➡ Sequence from Main Battle Sequence to Player 1
➡ Sequence from Preparation stage to Player 1
➡ Sequence from Reinforcement to Player 1
➡ Sequence from Exchange Cards to Player 1
Name: // popup ➡ Sequence from JFXMainView to Player 1
Name: addTerritory()

INCOMING BEHAVIORAL RELATIONSHIPS	
 Sequence from RiskGameModel to Player 1	
 Sequence from Define number of players to Player 1	
 Sequence from Exit Game to Player 1	
 Sequence from Define the starting player to Player 1	
 Sequence from Load Game to Player 1	

CONNECTORS	
 UseCaseLink Source -> Destination	
From: Player 1 : Actor, Public	
To: Reinforcement : UseCase, Public	
 UseCaseLink Source -> Destination	
From: Player 1 : Actor, Public	
To: Transfer Units : UseCase, Public	
 UseCaseLink Source -> Destination	
From: Player 1 : Actor, Public	
To: Main Battle Sequence : UseCase, Public	

ASSOCIATIONS	
 Association (direction: Unspecified)	
Source: Public (Actor) Player 1	Target: Public (UseCase) Exit Game

Deck

Class in package 'Model'

Deck
Version 1.0 Phase 1.0 Proposed
Szabi created on 2017. 02. 28.. Last modified 2017. 11. 22.
Extends Serializable

OUTGOING STRUCTURAL RELATIONSHIPS	
 Aggregation from Deck to RiskGameModel	[Direction is 'Source -> Destination'.]

INCOMING STRUCTURAL RELATIONSHIPS	
-----------------------------------	--

INCOMING STRUCTURAL RELATIONSHIPS	
⇒ Aggregation from Card to Deck	[Direction is 'Source -> Destination'.]
ATTRIBUTES	
◆ cardsSet : Set<Card> Private	[Is static False. Containment is Not Specified.]
◆ r : Random Package = new Random()	[Is static False. Containment is Not Specified.]
◆ serialVersionUID : long Private Const = 1L	[Is static True. Containment is Not Specified.]
ASSOCIATIONS	
↙ Association (direction: Source -> Destination)	
Source: Public (Class) SaveData	Target: Public deck (Class) Deck
↙ Association (direction: Source -> Destination)	
Source: Public (Class) Player	Target: Private deck (Class) Deck
OPERATIONS	
◆ Deck () : Public	[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ Draw () : Card Public	
A kihuzott kartya eltunik a paklibol	
Properties:	
ea_guid = {A9BC1E05-34BF-47e7-B78F-61F3E6E58456}	
[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]	
◆ finalize () : void Public	
Properties:	
throws = Throwable	
annotations = @Override	
[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]	
◆ Put (card : Card) : Deck Public	
@param kartya kartya	
[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]	

OPERATIONS
<p>◆ <code>readAllCard () : void</code> Private Read cards from <code>territorycards.properties</code> [Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]</p>

RiskGameModel

Class in package 'Model'

Az MVC architektura Model részét megvalósító osztalya.

RiskGameModel
Version 1.0 Phase 1.0 Proposed
Szabi created on 2017. 02. 21.. Last modified 2017. 11. 22.

OUTGOING STRUCTURAL RELATIONSHIPS
<p>↳ Aggregation from RiskGameModel to RiskGameController [Direction is 'Source -> Destination'.]</p>

INCOMING STRUCTURAL RELATIONSHIPS
<p>⇒ Aggregation from Map to RiskGameModel [Direction is 'Source -> Destination'.]</p> <p>⇒ Aggregation from Deck to RiskGameModel [Direction is 'Source -> Destination'.]</p>

ATTRIBUTES
<p>◆ <code>calculatedCircleNumber : int</code> Private [Is static False. Containment is Not Specified.]</p>
<p>◆ <code>capturedThisTurn : Boolean</code> Private [Is static False. Containment is Not Specified.]</p>
<p>◆ <code>circlenumber : int</code> Private [Is static False. Containment is Not Specified.]</p>
<p>◆ <code>currentPlayer : int</code> Private [Is static False. Containment is Not Specified.]</p>
<p>◆ <code>deck : Deck</code> Private [Is static False. Containment is Not Specified.]</p>
<p>◆ <code>hasTransferred : boolean</code> Private</p>

ATTRIBUTES	
	[Is static False. Containment is Not Specified.]
◆ map : Map Private	[Is static False. Containment is Not Specified.]
◆ misnumber : int Private	[Is static False. Containment is Not Specified.]
◆ phase : Phase Private	[Is static False. Containment is Not Specified.]
◆ playersList : List<Player> Private	[Is static False. Containment is Not Specified.]
◆ r : Random Private = new Random()	[Is static False. Containment is Not Specified.]
◆ waitForUnitsTemp : Territory Private = new Territory[2]	[Is static False. Containment is Not Specified.]

ASSOCIATIONS	
✓ Association (direction: Source -> Destination)	
Source: Public (Class) RiskGameModel	Target: Private waitForUnitsTemp (Class) Territory
✓ Association (direction: Source -> Destination)	
Source: Public (Class) RiskGameModel	Target: Private playersList (Class) Player Cardinality: [0..*]
✓ Association (direction: Source -> Destination)	
Source: Public (Class) RiskGameModel	Target: Public deck (Class) Player

OPERATIONS	
◆ addPlayer (name : String , color : Color) : boolean Public	
Add player with name and color @return boolean	[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ addPlayerToPlayerList (player : Player) : void Public	
Adder for playerList	
Properties:	

OPERATIONS	
throws = Exception ea_guid = {45684F69-D727-4875-BEF7-5232B4E9D1C9} [Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]	
◆ addPlayerToPlayerList (map : java.util.Map<Color, String>) : void Public	Pairs overload
Properties: throws = Exception [Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]	
◆ attackTerritoryWithResult (defenderID : int, attackerID : int, defendUnits : int, attackUnits : int) : AttackResult Public Return the result of the attack If the attacker have more than 3 units, the fighting attacker units will reduce to 3 If the defender have more than 2 units, the fighting defender units will reduce to 2	Properties: throws = Exception [Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ calculateCircle () : void Private Properties: throws = Exception [Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]	
◆ checkAttackPossible (defender : Territory, attacker : Territory, defendUnits : int, attackUnits : int) : boolean Public Csak akkor tĂˇmadhat, ha a terĂĽleten elegendL‘ egysĂ©ge van. Igazzal tĂˇr vissza, ha a tamadas lehetsĂ©ges. @return boolean [Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]	Attack checking method @return [Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ checkAttackPossible (defenderID : int, attackerID : int) : boolean Public Capture and conquer checking method @return boolean Properties: ea_guid = {BFF4F3B6-2538-4778-9372-EAA3B468EDD8} [Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]	
◆ checkIfCapturedAndConquer (defender : Territory) : boolean Public Capture and conquer checking method @return boolean Properties:	Empty territory checking method @return boolean

OPERATIONS	
<code>ea_guid = {BE4F4E07-A4C4-4e91-9F00-EF1EEB65D26C}</code>	[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
 <code>checkTransferPossible (from : int , to : int) : boolean</code> Public	Transfer checking method @return boolean [Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
 <code>endBattlePhase () : boolean</code> Public	Battle @return boolean
Properties: <code>throws = Exception</code>	[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
 <code>endTurn () : int</code> Public	Kor vege Returns next player's id @return int
Properties: <code>throws = Exception</code> <code>ea_guid = {F5F8D5D6-8997-43de-A5AD-DDDC647E775F}</code>	[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
 <code>getCurrentPlayer () : int</code> Public	Getter for current player @return int [Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
 <code>getPhase () : Phase</code> Public	Getter for Phase @return phase [Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
 <code>getPlayerCards (id : int) : List<Integer></code> Public	[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
 <code>getPlayerName (id : int) : String</code> Public	Getter for Player's name @return String [Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
 <code>getPlayers () : List<Player></code> Public	Getter for Players list @return List<Player>

OPERATIONS	
[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]	
<p>◆ getTerritory (id : int) : Territory Public</p> <p>Getter for Territory @return Territory</p> <p>[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]</p>	
<p>◆ getUnitsLeftToReinforce (playerID : int) : int Public</p> <p>Getter for left units in reinforce phase @return int</p> <p>[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]</p>	
<p>◆ loadGame (file : File) : boolean Public</p> <p>Load data method @return boolean</p> <p>Properties: throws = Exception</p> <p>[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]</p>	
<p>◆ moveUnits (units : int) : boolean Public</p> <p>Move units method @return boolean</p> <p>Properties: throws = Exception</p> <p>[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]</p>	
<p>◆ nextPhase () : Phase Private</p> <p>Calculate the next phase @return Phase</p> <p>Properties: throws = Exception</p> <p>[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]</p>	
<p>◆ nextPlayer () : void Protected</p> <p>Initializes the next Player's turn. Calculates the amount of Units to place.</p> <p>Properties: throws = Exception ea_guid = {806D8A15-D81F-4852-9BFA-E0988D68A2D8}</p> <p>[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]</p>	
<p>◆ reinforce (territoryId : int) : boolean Public</p> <p>Reinforce method @return boolean</p>	

OPERATIONS	
Properties:	
throws = Exception	[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ RiskGameModel () : Public	The amount of Units left to Place in the Reinforcement Stage of the current Player's current turn. [Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ saveGame (file : File) : boolean Public	Save data method @return
Properties:	
throws = IOException	[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ selectStartingPlayer (list : List<Player>) : Player Public	Random select the starting player @return Player [Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ transfer (fromID : int, toID : int, units : int) : boolean Public	@return boolean
Properties:	
throws = Exception	[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ transfer (from : Territory, to : Territory, units : int) : boolean Public	Transfer method @return boolean
Properties:	
throws = Exception	
ea_guid = {23034260-4F89-40e8-BC43-DD7B2D6023D5}	[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ winGame (winner : int) : void Private	[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]

RiskGameController

Class in package 'Controller'

Az MVC architektura Controller részet megvalósító osztálya. A Controller a Modell és a View kozotti kapcsolatért felelős. Rajta keresztül hajtoddnak végre a View-on történo változások, melyek a Model-t érintik, illetve a Model a

Controller-en keresztul módosítja a View-t.

RiskGameController
Version 1.0 Phase 1.0 Proposed
Szabó Dávid created on 2017. 03. 01.. Last modified 2017. 11. 22.
Extends java.util.Observable

INCOMING STRUCTURAL RELATIONSHIPS

- ⇒ Aggregation from JFXAttackView to RiskGameController
[Direction is 'Source -> Destination'.]
- ⇒ Aggregation from JFXAddPlayerView to RiskGameController
[Direction is 'Source -> Destination'.]
- ⇒ Aggregation from JFXTransferView to RiskGameController
[Direction is 'Source -> Destination'.]
- ⇒ Aggregation from RiskGameModel to RiskGameController
[Direction is 'Source -> Destination'.]
- ⇒ Aggregation from JFXMainView to RiskGameController
[Direction is 'Source -> Destination'.]

ATTRIBUTES

- ◆ addPlayerView : JFXAddPlayerView Private
[Is static False. Containment is Not Specified.]
- ◆ attackResultView : JFXAttackResultView Private
[Is static False. Containment is Not Specified.]
- ◆ attackView : JFXAttackView Private
[Is static False. Containment is Not Specified.]
- ◆ cardGroup : Group Private
[Is static False. Containment is Not Specified.]
- ◆ cardScene : Scene Private
[Is static False. Containment is Not Specified.]
- ◆ cardView : JFXCardView Private
CardView stuffs
[Is static False. Containment is Not Specified.]
- ◆ lastPhaseUpdate : Phase Private
[Is static False. Containment is Not Specified.]

ATTRIBUTES	
◆ lastPlayerUpdate : int	Private [Is static False. Containment is Not Specified.]
◆ mainView : JFXMainView	Private
View objects	[Is static False. Containment is Not Specified.]
◆ model : RiskGameModel	Private [Is static False. Containment is Not Specified.]
◆ popupStage : Stage	Private [Is static False. Containment is Not Specified.]
◆ preStage : Stage	Private Const
Stages	[Is static False. Containment is Not Specified.]
◆ previouslySelectedTerritory : int	Private = -1 [Is static False. Containment is Not Specified.]
◆ primaryStage : Stage	Private [Is static False. Containment is Not Specified.]
◆ titledPaneStage : Stage	Private [Is static False. Containment is Not Specified.]
◆ transferView : JFXTransferView	Private
private JFXRiskCardView cardView;	[Is static False. Containment is Not Specified.]
◆ victoryStage : Stage	Private [Is static False. Containment is Not Specified.]
◆ victoryView : JFXVictoryView	Private
VictoryView stuffs	[Is static False. Containment is Not Specified.]

ASSOCIATIONS	
◆ Association (direction: Source -> Destination)	
Source: Public (Class) JFXViewBase	Target: Protected controller (Class) RiskGameController

ASSOCIATIONS
OPERATIONS
<p>◆ addLog (log : String) : void Public A főablak naplójába küld egy új bejegyzést [Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]</p>
<p>◆ attackAccepted (defender : int, attacker : int, defenderUnits : int, attackerUnits : int) : void Public [Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]</p>
<p>◆ attackRetreat () : void Public [Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]</p>
<p>◆ attackShowed (attacker : int, defender : int) : void Public [Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]</p>
<p>◆ closePopupWindow () : void Private [Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]</p>
<p>◆ convertToCountry (ID : int) : Country Private [Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]</p>
<p>◆ convertToTerritoryID (country : Country) : int Private [Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]</p>
<p>◆ countrySelected (country : Country) : void Public [Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]</p>
<p>◆ getPhase () : Phase Public [Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]</p>
<p>◆ getPreviouslySelectedTerritory () : int Public [Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]</p>
<p>◆ getTerritoryData () : List<Territory> Public [Is static True. Is abstract False. Is return array False. Is query False. Is synchronized False.]</p>
<p>◆ isVictory () : boolean Private [Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]</p>
<p>◆ loadGame (loadFile : File) : void Public [Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]</p>
<p>◆ moveCardView () : void Public</p>

OPERATIONS	
[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]	
<p>◆ nextPlayerSelected () : void Public [Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]</p>	
<p>◆ reinforce (territoryID : int , country : String) : void Private</p> <p>Properties: throws = Exception [Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]</p>	
<p>◆ RiskGameController (stage : Stage) : Public [Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]</p>	
<p>◆ saveGame (saveFile : File) : void Public [Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]</p>	
<p>◆ setModel (model : RiskGameModel) : void Public Eltárolja a modellt és feliratkoztatja magát az eseményeire [Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]</p>	
<p>◆ showAddPlayerView () : void Public Add player view [Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]</p>	
<p>◆ showAttackResultView () : void Private [Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]</p>	
<p>◆ showAttackView (defenderID : int , attackerID : int) : void Private Attack View [Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]</p>	
<p>◆ showCardView () : void Public [Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]</p>	
<p>◆ showMainView () : void Public Main View [Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]</p>	
<p>◆ showTransferView (fromID : int , toID : int) : void Private Transfer View [Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]</p>	

OPERATIONS	
◆ showVictoryView () : void Public	[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ switchToMain (map : Map<Color, String>) : void Public	
Stage switch Pre -> Main	[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ transferAccepted (from : int, to : int, units : int) : void Public	[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ transferPhaseSelected () : void Public	[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ UpdateAllTerritoriesOnMap () : void Private	[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ UpdateCurrentDeck () : void Private	[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ UpdateCurrentPhase () : void Private	[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ UpdateCurrentPlayer () : void Private	[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ UpdateTerritoryOnMap (territoryID : int) : void Private	[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]

JFXMainView

Class in package 'View'

Az MVC architektura View részet megvalósító osztálya. Ez a View a játék főképernyője, ezen történik maga a játék (egysegek elhelyezése, csata stb.).

JFXMainView
Version 1.0 Phase 1.0 Proposed
Szabó Dávid created on 2017. 03. 01.. Last modified 2017. 11. 22.
Extends JFXViewBase

OUTGOING STRUCTURAL RELATIONSHIPS	
↳ Generalization from JFXMainView to JFXViewBase	[Direction is 'Source -> Destination'.]

OUTGOING STRUCTURAL RELATIONSHIPS	
↳ Aggregation from JFXMainView to RiskGameController	[Direction is 'Source -> Destination'.]
ATTRIBUTES	
<p>◆ BASE_HEIGHT : double Private Const = 600 [Is static False. Containment is Not Specified.]</p>	
<p>◆ BASE_WIDTH : double Private Const = 815 [Is static False. Containment is Not Specified.]</p>	
<p>◆ borderPane : BorderPane Private Properties: annotations = @FXML [Is static False. Containment is Not Specified.]</p>	
<p>◆ btnBattlePhase : Button Private Properties: annotations = @FXML [Is static False. Containment is Not Specified.]</p>	
<p>◆ btnControlList : ArrayList<Button> Private [Is static False. Containment is Not Specified.]</p>	
<p>◆ btnLoad : Button Private Properties: annotations = @FXML [Is static False. Containment is Not Specified.]</p>	
<p>◆ btnLog : Button Private Properties: annotations = @FXML [Is static False. Containment is Not Specified.]</p>	
<p>◆ btnNextPlayer : Button Private Properties: annotations = @FXML [Is static False. Containment is Not Specified.]</p>	
<p>◆ btnReinforcePhase : Button Private Properties: annotations = @FXML [Is static False. Containment is Not Specified.]</p>	

ATTRIBUTES
◆ btnSave : Button Private Properties: annotations = @FXML [Is static False. Containment is Not Specified.]
◆ btnSettings : Button Private Properties: annotations = @FXML [Is static False. Containment is Not Specified.]
◆ btnTransferPhase : Button Private Properties: annotations = @FXML [Is static False. Containment is Not Specified.]
◆ centerStackPane : StackPane Private Properties: annotations = @FXML [Is static False. Containment is Not Specified.]
◆ dimLog : Dimension Package [Is static False. Containment is Not Specified.]
◆ imageView : ImageView Private Properties: annotations = @FXML [Is static False. Containment is Not Specified.]
◆ isLogOpened : boolean Package [Is static False. Containment is Not Specified.]
◆ isSLOpened : boolean Package [Is static False. Containment is Not Specified.]
◆ lblPlayerName1 : Label Private Properties: annotations = @FXML [Is static False. Containment is Not Specified.]
◆ lblPlayerName2 : Label Private Properties: annotations = @FXML [Is static False. Containment is Not Specified.]

ATTRIBUTES
<p>◆ lblPlayerName3 : Label Private</p> <p>Properties:</p> <p> annotations = @FXML</p> <p style="text-align: right;">[Is static False. Containment is Not Specified.]</p>
<p>◆ lblPlayerName4 : Label Private</p> <p>Properties:</p> <p> annotations = @FXML</p> <p style="text-align: right;">[Is static False. Containment is Not Specified.]</p>
<p>◆ lblPlayerName5 : Label Private</p> <p>Properties:</p> <p> annotations = @FXML</p> <p style="text-align: right;">[Is static False. Containment is Not Specified.]</p>
<p>◆ lblPlayerName6 : Label Private</p> <p>Properties:</p> <p> annotations = @FXML</p> <p style="text-align: right;">[Is static False. Containment is Not Specified.]</p>
<p>◆ lblPlayerNameList : ArrayList<Label> Private</p> <p style="text-align: right;">[Is static False. Containment is Not Specified.]</p>
<p>◆ pain : Pane Private</p> <p>Properties:</p> <p> annotations = @FXML</p> <p style="text-align: right;">[Is static False. Containment is Not Specified.]</p>
<p>◆ panePlayerColor1 : Pane Private</p> <p>Properties:</p> <p> annotations = @FXML</p> <p style="text-align: right;">[Is static False. Containment is Not Specified.]</p>
<p>◆ panePlayerColor2 : Pane Private</p> <p>Properties:</p> <p> annotations = @FXML</p> <p style="text-align: right;">[Is static False. Containment is Not Specified.]</p>
<p>◆ panePlayerColor3 : Pane Private</p> <p>Properties:</p> <p> annotations = @FXML</p> <p style="text-align: right;">[Is static False. Containment is Not Specified.]</p>

ATTRIBUTES	
◆ panePlayerColor4 : Pane Private	Properties: annotations = @FXML [Is static False. Containment is Not Specified.]
◆ panePlayerColor5 : Pane Private	Properties: annotations = @FXML [Is static False. Containment is Not Specified.]
◆ panePlayerColor6 : Pane Private	Properties: annotations = @FXML [Is static False. Containment is Not Specified.]
◆ panePlayerColorList : ArrayList<Pane> Private	[Is static False. Containment is Not Specified.]
◆ stage : Stage Private	[Is static False. Containment is Not Specified.]
◆ taLog : TextArea Private	Properties: annotations = @FXML [Is static False. Containment is Not Specified.]
◆ world : World Private	[Is static False. Containment is Not Specified.]

OPERATIONS	
◆ AddControllerListener (controller : RiskGameController) : void Public	Properties: annotations = @Override [Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ appendLog (s : String) : void Public	[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ closeLog () : void Public	[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ finalize () : void Public	

OPERATIONS	
Properties:	
throws = Throwable	
annotations = @Override	[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ fit (height : double , width : double) : void Public	[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ getRoot () : Parent Public	
Properties:	
annotations = @Override	[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ getWorld () : World Public	[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ handleLogButton (event : ActionEvent) : void Private	
Properties:	
annotations = @FXML	[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ hangleLoadButton (event : ActionEvent) : void Private	
Properties:	
annotations = @FXML	[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ hangleSaveButton (event : ActionEvent) : void Private	
Properties:	
annotations = @FXML	[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ hangleSettingsButton (event : ActionEvent) : void Private	
Properties:	
annotations = @FXML	[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ initialize (location : URL , resources : ResourceBundle) : void Public	
Properties:	
annotations = @Override	[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ isLogOpened () : boolean Public	[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]

OPERATIONS	
◆ isSLOpened () : boolean Public	[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ JFXMainView (stage : Stage) : Public	[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ openLog () : void Public	[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ UpdateConnectedPlayers (playerList : List<String>) : void Public	[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ UpdateCurrentDeck (cards : List<Integer>) : void Public	
TODO:	[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ UpdateCurrentPhase (phase : Phase) : void Public	[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ UpdateCurrentPlayer (playerID : int) : void Public	
@param player	[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ updateIcons (attacker : int, defender : int) : void Public	[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ UpdateViewState (territory : Territory) : void Public	
@param territories	
Properties:	
ea_guid = {0E811D72-EC4E-47ad-8B46-C39084EAD8F7}	[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ UpdateViewState (territories : List<Territory>) : void Public	[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]

<anonymous>

Note in package 'Game Stage'

3 Card Exchange/Using/TradeIn

<anonymous>
Version 1.0 Phase 1.0 Proposed
Oliver created on 2017. 04. 04.. Last modified 2017. 04. 04.

INTERACTION MESSAGES

✉ 1.0 '//end turn' from 'Player 1' sent to 'JFXMainView'.

Synchronous Call. Returns void.

[Return is False. Iteration is False. New group is False.]

✉ 1.1 'actionPerformed(ActionEvent)' from 'JFXMainView' sent to 'RiskGameController'.

Synchronous Call.

[Return is False. Iteration is False. New group is False.]

✉ 1.2 'endTurn()' from 'RiskGameController' sent to 'RiskGameModel'.

Synchronous Call.

[Return is False. Iteration is False. New group is False.]

✉ 1.3 'Draw()' from 'RiskGameModel' sent to 'Deck'.

Synchronous Call. Returns Card.

[Return is False. Iteration is False. New group is False.]

✉ 1.4 'putcard(Card)' from 'RiskGameModel' sent to 'Player1: Player'.

Synchronous Call.

[Return is False. Iteration is False. New group is False.]

✉ 1.5 'exchangeCardsIfPossible()' from 'RiskGameModel' sent to 'Player1: Player'.

Synchronous Call. Returns byte.

Ha van 3 cserélhető kártya, vegyük el

[Return is False. Iteration is False. New group is False.]

✉ 1.6 'addReinforcements(byte)' from 'RiskGameModel' sent to 'Player1: Player'.

Synchronous Call.

[Return is False. Iteration is False. New group is False.]

✉ 1.7 'update(Observable, Object)' from 'RiskGameModel' sent to 'RiskGameController'.

Synchronous Call.

[Return is False. Iteration is False. New group is False.]

✉ 1.8 'nextTurn()' from 'RiskGameController' sent to 'JFXMainView'.

Synchronous Call.

[Return is False. Iteration is False. New group is False.]

Exit Game diagram

Interaction diagram in package 'Game Stage'

Exit Game
Version 1.0

fodorad created on 2017. 03. 22.. Last modified 2017. 09. 21.

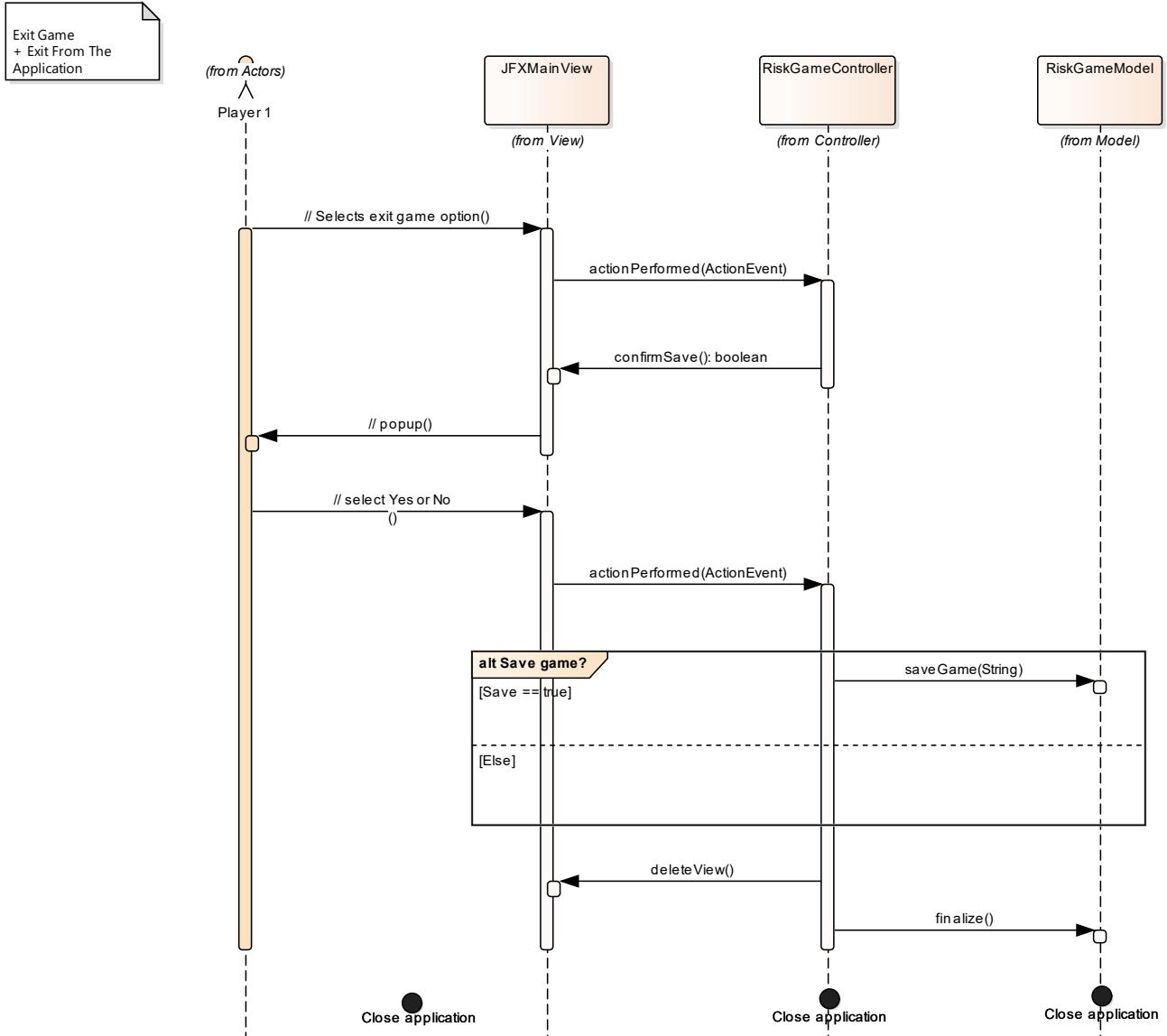


Figure 11: Exit Game

Player 1

Actor in package 'Actors'

Player 1

Version 1.0 Phase 1.0 Mandatory

Oliver created on 2017. 02. 21.. Last modified 2017. 03. 23.

OUTGOING BEHAVIORAL RELATIONSHIPS

OUTGOING BEHAVIORAL RELATIONSHIPS
Name: //Selects 2 Territories ↳ Sequence from Player 1 to JFXMainView
Name: //select attacking units ↳ Sequence from Player 1 to JFXAttackView
↳ Sequence from Player 1 to Reinforcement
↳ Sequence from Player 1 to Load Game
↳ Sequence from Player 1 to Exit Game
↳ Sequence from Player 1 to Save Game
↳ Sequence from Player 1 to Exchange Cards
↳ Sequence from Player 1 to Define the starting player
↳ Sequence from Player 1 to Define number of players
Name: // Selects option ↳ Sequence from Player 1 to JFXMainView
Name: // Add number of players ↳ Sequence from Player 1 to JFXAddPlayerView
Name: //Set a Save Name then Apply ↳ Sequence from Player 1 to JFXMainView
Name: // click to begin action ↳ Sequence from Player 1 to JFXMainView
Name: // click to connect ↳ Sequence from Player 1 to JFXMainView
Name: //End Turn ↳ Sequence from Player 1 to JFXMainView
Name: // Selects exit game option ↳ Sequence from Player 1 to JFXMainView
Name: // Select saved game file then apply ↳ Sequence from Player 1 to JFXMainView

OUTGOING BEHAVIORAL RELATIONSHIPS
<p>↳ Sequence from Player 1 to Main Battle Sequence</p>
<p>Name: //init attack ↳ Sequence from Player 1 to JFXMainView</p>
<p>Name: //end turn ↳ Sequence from Player 1 to JFXMainView</p>
<p>↳ Sequence from Player 1 to Transfer Units</p>
<p>Name: // select Yes or No ↳ Sequence from Player 1 to JFXMainView</p>
<p>Name: //Selects Units then Apply ↳ Sequence from Player 1 to JFXTransferView</p>
<p>↳ Sequence from Player 1 to Preparation stage</p>

INCOMING BEHAVIORAL RELATIONSHIPS
<p>↗ Sequence from Transfer Units to Player 1</p>
<p>↗ Sequence from Save Game to Player 1</p>
<p>↗ Sequence from Main Battle Sequence to Player 1</p>
<p>↗ Sequence from Preparation stage to Player 1</p>
<p>↗ Sequence from Reinforcement to Player 1</p>
<p>↗ Sequence from Exchange Cards to Player 1</p>
<p>Name: // popup ↗ Sequence from JFXMainView to Player 1</p>
<p>Name: addTerritory() ↗ Sequence from RiskGameModel to Player 1</p>
<p>↗ Sequence from Define number of players to Player 1</p>

INCOMING BEHAVIORAL RELATIONSHIPS

- ➡ Sequence from Exit Game to Player 1

- ➡ Sequence from Define the starting player to Player 1

- ➡ Sequence from Load Game to Player 1

CONNECTORS**UseCaseLink** Source -> Destination

From: Player 1 : Actor, Public
 To: Reinforcement : UseCase, Public

UseCaseLink Source -> Destination

From: Player 1 : Actor, Public
 To: Transfer Units : UseCase, Public

UseCaseLink Source -> Destination

From: Player 1 : Actor, Public
 To: Main Battle Sequence : UseCase, Public

ASSOCIATIONS**Association** (direction: Unspecified)

Source: Public (Actor) Player 1

Target: Public (UseCase) Exit Game

RiskGameModel*Class in package 'Model'*

Az MVC architektura Model részét megvalósító osztalya.

RiskGameModel
 Version 1.0 Phase 1.0 Proposed
 Szabi created on 2017. 02. 21.. Last modified 2017. 11. 22.

OUTGOING STRUCTURAL RELATIONSHIPS**Aggregation** from RiskGameModel to RiskGameController

[Direction is 'Source -> Destination'.]

INCOMING STRUCTURAL RELATIONSHIPS**Aggregation** from Map to RiskGameModel

[Direction is 'Source -> Destination'.]

INCOMING STRUCTURAL RELATIONSHIPS	
⇒ Aggregation from Deck to RiskGameModel	[Direction is 'Source -> Destination'.]
ATTRIBUTES	
◆ calculatedCircleNumber : int Private	[Is static False. Containment is Not Specified.]
◆ capturedThisTurn : Boolean Private	[Is static False. Containment is Not Specified.]
◆ circlenumber : int Private	[Is static False. Containment is Not Specified.]
◆ currentPlayer : int Private	[Is static False. Containment is Not Specified.]
◆ deck : Deck Private	[Is static False. Containment is Not Specified.]
◆ hasTransferred : boolean Private	[Is static False. Containment is Not Specified.]
◆ map : Map Private	[Is static False. Containment is Not Specified.]
◆ miscnumber : int Private	[Is static False. Containment is Not Specified.]
◆ phase : Phase Private	[Is static False. Containment is Not Specified.]
◆ playersList : List<Player> Private	[Is static False. Containment is Not Specified.]
◆ r : Random Private = new Random()	[Is static False. Containment is Not Specified.]
◆ waitForUnitsTemp : Territory Private = new Territory[2]	[Is static False. Containment is Not Specified.]
ASSOCIATIONS	
↙ Association (direction: Source -> Destination)	

ASSOCIATIONS	
Source: Public (Class) RiskGameModel	Target: Private waitForUnitsTemp (Class) Territory
Association (direction: Source -> Destination)	
Source: Public (Class) RiskGameModel	Target: Private playersList (Class) Player Cardinality: [0..*]
Association (direction: Source -> Destination)	
Source: Public (Class) RiskGameModel	Target: Public deck (Class) Player

OPERATIONS	
addPlayer (name : String , color : Color) : boolean Public	
Add player with name and color @return boolean	[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
addPlayerToPlayerList (player : Player) : void Public	
Adder for playerList	
Properties: throws = Exception ea_guid = {45684F69-D727-4875-BEF7-5232B4E9D1C9}	[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
addPlayerToPlayerList (map : java.util.Map<Color, String>) : void Public	
Pairs overload	
Properties: throws = Exception	[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
attackTerritoryWithResult (defenderID : int , attackerID : int , defendUnits : int , attackUnits : int) : AttackResult Public	
Return the result of the attack If the attacker have more than 3 units, the fighting attacker units will reduce to 3 If the defender have more than 2 units, the fighting defender units will reduce to 2	
Properties: throws = Exception	[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
calculateCircle () : void Private	
Properties: throws = Exception	[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]

OPERATIONS

◆ checkAttackPossible (defender : Territory , attacker : Territory , defendUnits : int , attackUnits : int) : boolean Public

Csak akkor tĂˇmadhat, ha a terĂĽleten elegendL‘ egysĂ©ge van. Igazzal tĂ©r vissza, ha a tamadas lehetsĂ©ges.

@return boolean

[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]

◆ checkAttackPossible (defenderID : int , attackerID : int) : boolean Public

Attack checking method

@return

[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]

◆ checkIfCapturedAndConquer (defender : Territory) : boolean Public

Capture and conquer checking method

@return boolean

Properties:

ea_guid = {BFF4F3B6-2538-4778-9372-EAA3B468EDD8}

[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]

◆ checkIfTerritoryIsEmpty (territory : Territory) : boolean Public

Empty territory checking method

@return boolean

Properties:

ea_guid = {BE4F4E07-A4C4-4e91-9F00-EF1EEB65D26C}

[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]

◆ checkTransferPossible (from : int , to : int) : boolean Public

Transfer checking method

@return boolean

[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]

◆ endBattlePhase () : boolean Public

Battle

@return boolean

Properties:

throws = Exception

[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]

◆ endTurn () : int Public

Kor vege Returns next player's id

@return int

Properties:

throws = Exception

ea_guid = {F5F8D5D6-8997-43de-A5AD-DDDC647E775F}

[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]

OPERATIONS	
◆ <code>getCurrentPlayer () : int</code> Public	
Getter for current player @return int	[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ <code>getPhase () : Phase</code> Public	
Getter for Phase @return phase	[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ <code>getPlayerCards (id : int) : List<Integer></code> Public	[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ <code>getPlayerName (id : int) : String</code> Public	
Getter for Player's name @return String	[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ <code>getPlayers () : List<String></code> Public	
Getter for Players list @return List<Player>	[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ <code>getTerritory (id : int) : Territory</code> Public	
Getter for Territory @return Territory	[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ <code>getUnitsLeftToReinforce (playerID : int) : int</code> Public	
Getter for left units in reinforce phase @return int	[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ <code>loadGame (file : File) : boolean</code> Public	
Load data method @return boolean	
Properties: throws = Exception	[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ <code>moveUnits (units : int) : boolean</code> Public	
Move units method @return boolean	

OPERATIONS
<p>Properties: throws = Exception [Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]</p>
<p>◆ nextPhase () : Phase Private Calculate the next phase @return Phase</p> <p>Properties: throws = Exception [Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]</p>
<p>◆ nextPlayer () : void Protected Initializes the next Player's turn. Calculates the amount of Units to place.</p> <p>Properties: throws = Exception ea_guid = {806D8A15-D81F-4852-9BFA-E0988D68A2D8} [Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]</p>
<p>◆ reinforce (territoryId : int) : boolean Public Reinforce method @return boolean</p> <p>Properties: throws = Exception [Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]</p>
<p>◆ RiskGameModel () : Public The amount of Units left to Place in the Reinforcement Stage of the current Player's current turn. [Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]</p>
<p>◆ saveGame (file : File) : boolean Public Save data method @return</p> <p>Properties: throws = IOException [Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]</p>
<p>◆ selectStartingPlayer (list : List<Player>) : Player Public Random select the starting player @return Player [Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]</p>
<p>◆ transfer (fromID : int, toID : int, units : int) : boolean Public</p>

OPERATIONS
@return boolean
Properties: throws = Exception [Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ transfer (from : Territory , to : Territory , units : int) : boolean Public Transfer method @return boolean
Properties: throws = Exception ea_guid = {23034260-4F89-40e8-BC43-DD7B2D6023D5} [Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ winGame (winner : int) : void Private [Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]

RiskGameController

Class in package 'Controller'

Az MVC architektura Controller részet megvalósító osztálya. A Controller a Modell és a View kozotti kapcsolatért felelős. Rajta keresztül hajtoddnak végre a View-on történo változások, melyek a Model-t érintik, illetve a Model a Controller-en keresztül módosítja a View-t.

RiskGameController
Version 1.0 Phase 1.0 Proposed
Szabó Dávid created on 2017. 03. 01.. Last modified 2017. 11. 22.
Extends java.util.Observable

INCOMING STRUCTURAL RELATIONSHIPS
⇒ Aggregation from JFXAttackView to RiskGameController [Direction is 'Source -> Destination'.]
⇒ Aggregation from JFXAddPlayerView to RiskGameController [Direction is 'Source -> Destination'.]
⇒ Aggregation from JFXTransferView to RiskGameController [Direction is 'Source -> Destination'.]
⇒ Aggregation from RiskGameModel to RiskGameController [Direction is 'Source -> Destination'.]
⇒ Aggregation from JFXMainView to RiskGameController [Direction is 'Source -> Destination'.]

ATTRIBUTES

ATTRIBUTES	
◆ addPlayerView : JFXAddPlayerView	Private [Is static False. Containment is Not Specified.]
◆ attackResultView : JFXAttackResultView	Private [Is static False. Containment is Not Specified.]
◆ attackView : JFXAttackView	Private [Is static False. Containment is Not Specified.]
◆ cardGroup : Group	Private [Is static False. Containment is Not Specified.]
◆ cardScene : Scene	Private [Is static False. Containment is Not Specified.]
◆ cardView : JFXCardView	Private CardView stuffs [Is static False. Containment is Not Specified.]
◆ lastPhaseUpdate : Phase	Private [Is static False. Containment is Not Specified.]
◆ lastPlayerUpdate : int	Private [Is static False. Containment is Not Specified.]
◆ mainView : JFXMainView	Private View objects [Is static False. Containment is Not Specified.]
◆ model : RiskGameModel	Private [Is static False. Containment is Not Specified.]
◆ popupStage : Stage	Private [Is static False. Containment is Not Specified.]
◆ preStage : Stage	Private Const Stages [Is static False. Containment is Not Specified.]
◆ previouslySelectedTerritory : int	Private = -1 [Is static False. Containment is Not Specified.]
◆ primaryStage : Stage	Private

ATTRIBUTES	
	[Is static False. Containment is Not Specified.]
◆ titledPaneStage : Stage Private	[Is static False. Containment is Not Specified.]
◆ transferView : JFXTransferView Private private JFXRiskCardView cardView;	[Is static False. Containment is Not Specified.]
◆ victoryStage : Stage Private	[Is static False. Containment is Not Specified.]
◆ victoryView : JFXVictoryView Private VictoryView stuffs	[Is static False. Containment is Not Specified.]

ASSOCIATIONS	
↙ Association (direction: Source -> Destination)	Source: Public (Class) JFXViewBase Target: Protected controller (Class) RiskGameController

OPERATIONS	
◆ addLog (log : String) : void Public	
A főablak naplójába küld egy új bejegyzést	[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ attackAccepted (defender : int, attacker : int, defenderUnits : int, attackerUnits : int) : void Public	[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ attackRetreat () : void Public	[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ attackShowed (attacker : int, defender : int) : void Public	[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ closePopupWindow () : void Private	[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ convertToCountry (ID : int) : Country Private	[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]

OPERATIONS	
◆ convertToTerritoryID (country : Country) : int	Private [Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ countrySelected (country : Country) : void	Public [Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ getPhase () : Phase	Public [Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ getPreviouslySelectedTerritory () : int	Public [Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ getTerritoryData () : List<Territory>	Public [Is static True. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ isVictory () : boolean	Private [Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ loadGame (loadFile : File) : void	Public [Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ moveCardView () : void	Public [Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ nextPlayerSelected () : void	Public [Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ reinforce (territoryID : int, country : String) : void	Private Properties: throws = Exception [Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ RiskGameController (stage : Stage) : Public	[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ saveGame (saveFile : File) : void	Public [Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ setModel (model : RiskGameModel) : void	Public Eltárolja a modelt és feliratkoztatja magát az eseményeire [Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ showAddPlayerView () : void	Public Add player view

OPERATIONS	
[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]	
<p>◊ showAttackResultView () : void Private [Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]</p>	
<p>◊ showAttackView (defenderID : int , attackerID : int) : void Private Attack View [Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]</p>	
<p>◊ showCardView () : void Public [Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]</p>	
<p>◊ showMainView () : void Public Main View [Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]</p>	
<p>◊ showTransferView (fromID : int , toID : int) : void Private Transfer View [Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]</p>	
<p>◊ showVictoryView () : void Public [Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]</p>	
<p>◊ switchToMain (map : Map<Color, String>) : void Public Stage switch Pre -> Main [Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]</p>	
<p>◊ transferAccepted (from : int , to : int , units : int) : void Public [Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]</p>	
<p>◊ transferPhaseSelected () : void Public [Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]</p>	
<p>◊ UpdateAllTerritoriesOnMap () : void Private [Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]</p>	
<p>◊ UpdateCurrentDeck () : void Private [Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]</p>	
<p>◊ UpdateCurrentPhase () : void Private [Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]</p>	
<p>◊ UpdateCurrentPlayer () : void Private</p>	

OPERATIONS	
[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]	
◆ UpdateTerritoryOnMap (territoryID : int) : void Private	[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]

JFXMainView

Class in package 'View'

Az MVC architektura View részét megvalósító osztálya. Ez a View a játék főkepernyője, ezen történik maga a játék (egysegek elhelyezése, csata stb.).

JFXMainView
Version 1.0 Phase 1.0 Proposed
Szabó Dávid created on 2017. 03. 01.. Last modified 2017. 11. 22.
Extends JFXViewBase

OUTGOING STRUCTURAL RELATIONSHIPS	
↳ Generalization from JFXMainView to JFXViewBase	[Direction is 'Source -> Destination'.]
↳ Aggregation from JFXMainView to RiskGameController	[Direction is 'Source -> Destination'.]

ATTRIBUTES	
◆ BASE_HEIGHT : double Private Const = 600	[Is static False. Containment is Not Specified.]
◆ BASE_WIDTH : double Private Const = 815	[Is static False. Containment is Not Specified.]
◆ borderPane : BorderPane Private	
Properties: annotations = @FXML	[Is static False. Containment is Not Specified.]
◆ btnBattlePhase : Button Private	
Properties: annotations = @FXML	[Is static False. Containment is Not Specified.]
◆ btnControlList : ArrayList<Button> Private	[Is static False. Containment is Not Specified.]

ATTRIBUTES	
◆ btnLoad : Button	Private
Properties:	
annotations = @FXML	[Is static False. Containment is Not Specified.]
◆ btnLog : Button	Private
Properties:	
annotations = @FXML	[Is static False. Containment is Not Specified.]
◆ btnNextPlayer : Button	Private
Properties:	
annotations = @FXML	[Is static False. Containment is Not Specified.]
◆ btnReinforcePhase : Button	Private
Properties:	
annotations = @FXML	[Is static False. Containment is Not Specified.]
◆ btnSave : Button	Private
Properties:	
annotations = @FXML	[Is static False. Containment is Not Specified.]
◆ btnSettings : Button	Private
Properties:	
annotations = @FXML	[Is static False. Containment is Not Specified.]
◆ btnTransferPhase : Button	Private
Properties:	
annotations = @FXML	[Is static False. Containment is Not Specified.]
◆ centerStackPane : StackPane	Private
Properties:	
annotations = @FXML	[Is static False. Containment is Not Specified.]
◆ dimLog : Dimension	Package
	[Is static False. Containment is Not Specified.]
◆ imageView : ImageView	Private

ATTRIBUTES
Properties: annotations = @FXML [Is static False. Containment is Not Specified.]
◆ isLogOpened : boolean Package [Is static False. Containment is Not Specified.]
◆ isSLOpened : boolean Package [Is static False. Containment is Not Specified.]
◆ lblPlayerName1 : Label Private Properties: annotations = @FXML [Is static False. Containment is Not Specified.]
◆ lblPlayerName2 : Label Private Properties: annotations = @FXML [Is static False. Containment is Not Specified.]
◆ lblPlayerName3 : Label Private Properties: annotations = @FXML [Is static False. Containment is Not Specified.]
◆ lblPlayerName4 : Label Private Properties: annotations = @FXML [Is static False. Containment is Not Specified.]
◆ lblPlayerName5 : Label Private Properties: annotations = @FXML [Is static False. Containment is Not Specified.]
◆ lblPlayerName6 : Label Private Properties: annotations = @FXML [Is static False. Containment is Not Specified.]
◆ lblPlayerNameList : ArrayList<Label> Private [Is static False. Containment is Not Specified.]

ATTRIBUTES	
◆ pain : Pane	Private Properties: annotations = @FXML [Is static False. Containment is Not Specified.]
◆ panePlayerColor1 : Pane	Private Properties: annotations = @FXML [Is static False. Containment is Not Specified.]
◆ panePlayerColor2 : Pane	Private Properties: annotations = @FXML [Is static False. Containment is Not Specified.]
◆ panePlayerColor3 : Pane	Private Properties: annotations = @FXML [Is static False. Containment is Not Specified.]
◆ panePlayerColor4 : Pane	Private Properties: annotations = @FXML [Is static False. Containment is Not Specified.]
◆ panePlayerColor5 : Pane	Private Properties: annotations = @FXML [Is static False. Containment is Not Specified.]
◆ panePlayerColor6 : Pane	Private Properties: annotations = @FXML [Is static False. Containment is Not Specified.]
◆ panePlayerColorList : ArrayList<Pane>	Private [Is static False. Containment is Not Specified.]
◆ stage : Stage	Private [Is static False. Containment is Not Specified.]
◆ taLog : TextArea	Private Properties: annotations = @FXML

ATTRIBUTES	
	[Is static False. Containment is Not Specified.]
◆ world : World Private	[Is static False. Containment is Not Specified.]
OPERATIONS	
<p>◆ AddControllerListener (controller : RiskGameController) : void Public</p> <p>Properties: annotations = @Override [Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]</p>	
<p>◆ appendLog (s : String) : void Public [Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]</p>	
<p>◆ closeLog () : void Public [Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]</p>	
<p>◆ finalize () : void Public</p> <p>Properties: throws = Throwable annotations = @Override [Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]</p>	
<p>◆ fit (height : double, width : double) : void Public [Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]</p>	
<p>◆ getRoot () : Parent Public</p> <p>Properties: annotations = @Override [Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]</p>	
<p>◆ getWorld () : World Public [Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]</p>	
<p>◆ handleLogButton (event : ActionEvent) : void Private</p> <p>Properties: annotations = @FXML [Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]</p>	
<p>◆ handleLoadButton (event : ActionEvent) : void Private</p> <p>Properties: annotations = @FXML [Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]</p>	

OPERATIONS	
◆ handleSaveButton (event : ActionEvent) : void Private	
Properties:	
annotations = @FXML	[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ handleSettingsButton (event : ActionEvent) : void Private	
Properties:	
annotations = @FXML	[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ initialize (location : URL, resources : ResourceBundle) : void Public	
Properties:	
annotations = @Override	[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ isLogOpened () : boolean Public	[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ isSLOpened () : boolean Public	[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ JFXMainView (stage : Stage) : Public	[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ openLog () : void Public	[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ UpdateConnectedPlayers (playerList : List<String>) : void Public	[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ UpdateCurrentDeck (cards : List<Integer>) : void Public	
TODO:	[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ UpdateCurrentPhase (phase : Phase) : void Public	[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ UpdateCurrentPlayer (playerID : int) : void Public	
@param player	[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ updateIcons (attacker : int, defender : int) : void Public	[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]

OPERATIONS
<p>◆ UpdateViewState (territory : Territory) : void Public <code>@param territories</code></p> <p>Properties: <code>ea_guid = {0E811D72-EC4E-47ad-8B46-C39084EAD8F7}</code> [Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]</p>
<p>◆ UpdateViewState (territories : List<Territory>) : void Public [Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]</p>

<anonymous>

Note in package 'Game Stage'

Exit Game

+ Exit From The Application

<anonymous>

Version 1.0 Phase 1.0 Proposed

Oliver created on 2017. 03. 28.. Last modified 2017. 03. 28.

Extends

INTERACTION MESSAGES
<p>✉ 1.0 '// Selects exit game option' from 'Player 1' sent to 'JFXMainView'. Synchronous Call. Returns void. [Return is False. Iteration is False. New group is False.]</p>
<p>✉ 1.1 'actionPerformed(ActionEvent)' from 'JFXMainView' sent to 'RiskGameController'. Synchronous Call. [Return is False. Iteration is False. New group is False.]</p>
<p>✉ 1.2 'confirmSave()' from 'RiskGameController' sent to 'JFXMainView'. Synchronous Call. Returns boolean. [Return is False. Iteration is False. New group is False.]</p>
<p>✉ 1.3 '// popup' from 'JFXMainView' sent to 'Player 1'. Synchronous Call. Returns void. [Return is False. Iteration is False. New group is False.]</p>
<p>✉ 1.4 '// select Yes or No' from 'Player 1' sent to 'JFXMainView'. Synchronous Call. Returns void.</p>

	[Return is False. Iteration is False. New group is False.]
✉ 1.5 'actionPerformed(ActionEvent)' from 'JFXMainView' sent to 'RiskGameController'.	Synchronous Call. [Return is False. Iteration is False. New group is False.]
✉ 1.6 'saveGame(String)' from 'RiskGameController' sent to 'RiskGameModel'.	Synchronous Call. [Return is False. Iteration is False. New group is False.]
✉ 1.7 'deleteView()' from 'RiskGameController' sent to 'JFXMainView'.	Synchronous Call. [Return is False. Iteration is False. New group is False.]
✉ 1.8 'finalize()' from 'RiskGameController' sent to 'RiskGameModel'.	Synchronous Call. Returns void. [Return is False. Iteration is False. New group is False.]

Main Battle Sequence diagram

Interaction diagram in package 'Game Stage'

Main Battle Sequence
Version 1.0
Szabi created on 2017. 03. 07.. Last modified 2017. 09. 21.

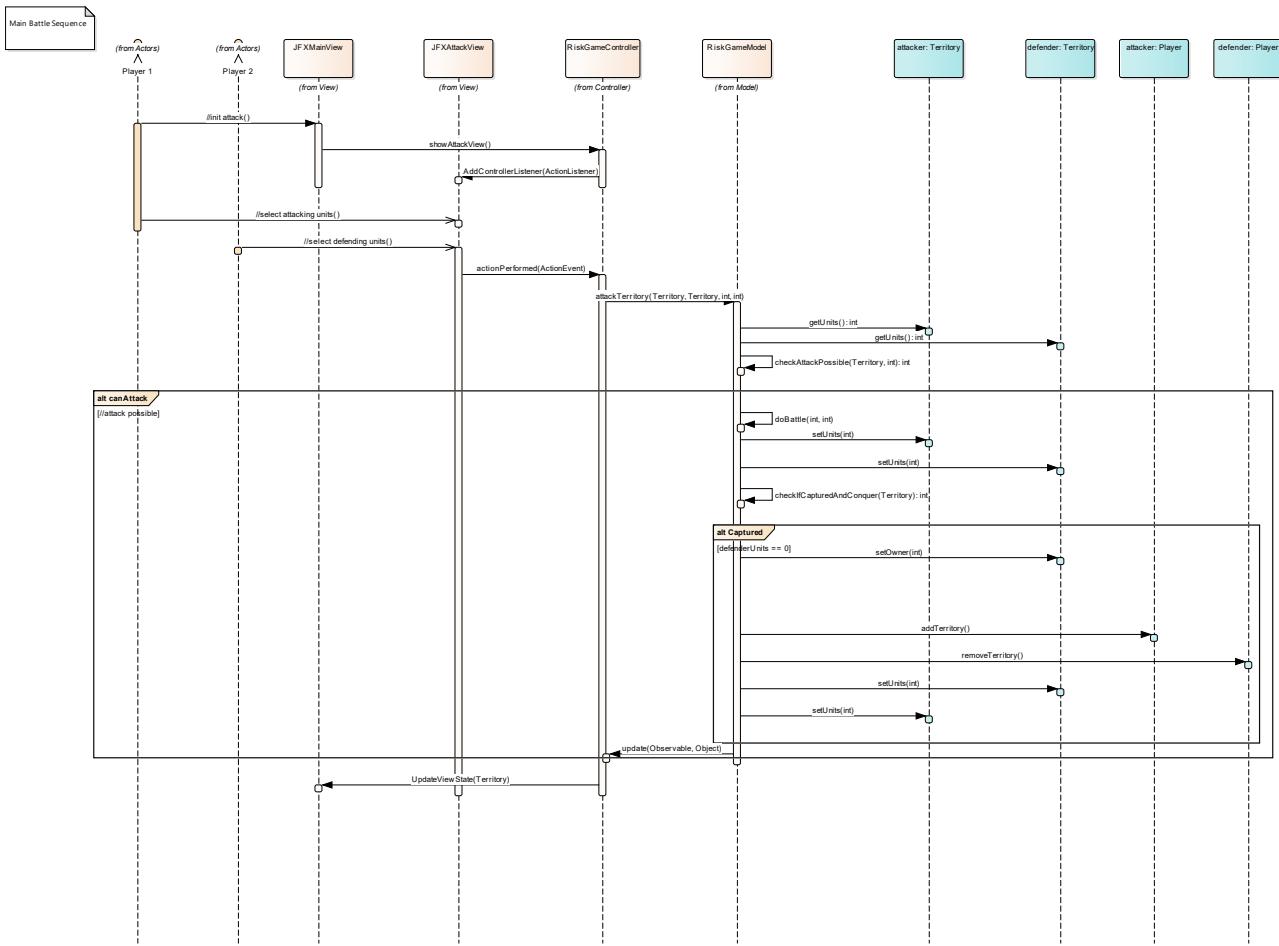


Figure 12: Main Battle Sequence

Player 1

Actor in package 'Actors'

Player 1

Version 1.0 Phase 1.0 Mandatory

Oliver created on 2017. 02. 21.. Last modified 2017. 03. 23.

OUTGOING BEHAVIORAL RELATIONSHIPS
Name: //Selects 2 Territories Sequence from Player 1 to JF XMainView
Name: //select attacking units Sequence from Player 1 to JFXAttackView
Sequence from Player 1 to Reinforcement
Sequence from Player 1 to Load Game
Sequence from Player 1 to Exit Game
Sequence from Player 1 to Save Game

OUTGOING BEHAVIORAL RELATIONSHIPS
↳ Sequence from Player 1 to Exchange Cards
↳ Sequence from Player 1 to Define the starting player
↳ Sequence from Player 1 to Define number of players
Name: // Selects option ↳ Sequence from Player 1 to JFXMainView
Name: // Add number of players ↳ Sequence from Player 1 to JFXAddPlayerView
Name: //Set a Save Name then Apply ↳ Sequence from Player 1 to JFXMainView
Name: // click to begin action ↳ Sequence from Player 1 to JFXMainView
Name: // click to connect ↳ Sequence from Player 1 to JFXMainView
Name: //End Turn ↳ Sequence from Player 1 to JFXMainView
Name: // Selects exit game option ↳ Sequence from Player 1 to JFXMainView
Name: // Select saved game file then apply ↳ Sequence from Player 1 to JFXMainView
↳ Sequence from Player 1 to Main Battle Sequence
Name: //init attack ↳ Sequence from Player 1 to JFXMainView
Name: //end turn ↳ Sequence from Player 1 to JFXMainView
↳ Sequence from Player 1 to Transfer Units
Name: // select Yes or No ↳ Sequence from Player 1 to JFXMainView

OUTGOING BEHAVIORAL RELATIONSHIPS

Name: //Selects Units then Apply
➡ Sequence from Player 1 to JFXTransferView

➡ Sequence from Player 1 to Preparation stage

INCOMING BEHAVIORAL RELATIONSHIPS

➡ Sequence from Transfer Units to Player 1

➡ Sequence from Save Game to Player 1

➡ Sequence from Main Battle Sequence to Player 1

➡ Sequence from Preparation stage to Player 1

➡ Sequence from Reinforcement to Player 1

➡ Sequence from Exchange Cards to Player 1

Name: // popup

➡ Sequence from JFXMainView to Player 1

Name: addTerritory()

➡ Sequence from RiskGameModel to Player 1

➡ Sequence from Define number of players to Player 1

➡ Sequence from Exit Game to Player 1

➡ Sequence from Define the starting player to Player 1

➡ Sequence from Load Game to Player 1

CONNECTORS

↗ **UseCaseLink** Source -> Destination

From: Player 1 : Actor, Public

To: Reinforcement : UseCase, Public

CONNECTORS	
 UseCaseLink	Source -> Destination
From:	Player 1 : Actor, Public
To:	Transfer Units : UseCase, Public
 UseCaseLink	Source -> Destination
From:	Player 1 : Actor, Public
To:	Main Battle Sequence : UseCase, Public
ASSOCIATIONS	
 Association (direction: Unspecified)	
Source: Public (Actor) Player 1	Target: Public (UseCase) Exit Game

Player 2

Actor in package 'Actors'

Player 2
Version 1.0 Phase 1.0 Mandatory
Oliver created on 2017. 03. 22.. Last modified 2017. 03. 23.

OUTGOING BEHAVIORAL RELATIONSHIPS	
Name: //Selects a Territory	
 Sequence from Player 2 to JFXMainView	
Name: //select defending units	
 Sequence from Player 2 to JFXAttackView	

RiskGameModel

Class in package 'Model'

Az MVC architektura Model részét megvalósító osztálya.

RiskGameModel
Version 1.0 Phase 1.0 Proposed
Szabi created on 2017. 02. 21.. Last modified 2017. 11. 22.

OUTGOING STRUCTURAL RELATIONSHIPS	
 Aggregation from RiskGameModel to RiskGameController	[Direction is 'Source -> Destination'.]

INCOMING STRUCTURAL RELATIONSHIPS	
-----------------------------------	--

INCOMING STRUCTURAL RELATIONSHIPS	
⇒ Aggregation from Map to RiskGameModel	[Direction is 'Source -> Destination'.]
⇒ Aggregation from Deck to RiskGameModel	[Direction is 'Source -> Destination'.]
ATTRIBUTES	
◆ calculatedCircleNumber : int Private	[Is static False. Containment is Not Specified.]
◆ capturedThisTurn : Boolean Private	[Is static False. Containment is Not Specified.]
◆ circlenumber : int Private	[Is static False. Containment is Not Specified.]
◆ currentPlayer : int Private	[Is static False. Containment is Not Specified.]
◆ deck : Deck Private	[Is static False. Containment is Not Specified.]
◆ hasTransferred : boolean Private	[Is static False. Containment is Not Specified.]
◆ map : Map Private	[Is static False. Containment is Not Specified.]
◆ miscnumber : int Private	[Is static False. Containment is Not Specified.]
◆ phase : Phase Private	[Is static False. Containment is Not Specified.]
◆ playersList : List<Player> Private	[Is static False. Containment is Not Specified.]
◆ r : Random Private = new Random()	[Is static False. Containment is Not Specified.]
◆ waitForUnitsTemp : Territory Private = new Territory[2]	[Is static False. Containment is Not Specified.]

ASSOCIATIONS	
 Association (direction: Source -> Destination)	
Source: Public (Class) RiskGameModel	Target: Private waitForUnitsTemp (Class) Territory
 Association (direction: Source -> Destination)	
Source: Public (Class) RiskGameModel	Target: Private playersList (Class) Player Cardinality: [0..*]
 Association (direction: Source -> Destination)	
Source: Public (Class) RiskGameModel	Target: Public deck (Class) Player
OPERATIONS	
 addPlayer (name : String , color : Color) : boolean Public	
Add player with name and color @return boolean	[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
 addPlayerToPlayerList (player : Player) : void Public	
Adder for playerList	
Properties: throws = Exception ea_guid = {45684F69-D727-4875-BEF7-5232B4E9D1C9} [Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]	
 addPlayerToPlayerList (map : java.util.Map<Color, String>) : void Public	
Pairs overload	
Properties: throws = Exception [Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]	
 attackTerritoryWithResult (defenderID : int , attackerID : int , defendUnits : int , attackUnits : int) : AttackResult Public	
Return the result of the attack If the attacker have more than 3 units, the fighting attacker units will reduce to 3 If the defender have more than 2 units, the fighting defender units will reduce to 2	
Properties: throws = Exception [Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]	
 calculateCircle () : void Private	
Properties: throws = Exception [Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]	

OPERATIONS
<p>◆ checkAttackPossible (defender : Territory , attacker : Territory , defendUnits : int , attackUnits : int) : boolean Public Csak akkor tĂˇmadhat, ha a terĂĽleten elegendL‘ egysĂ©ge van. Igazzal tĂ©r vissza, ha a tamadas lehetsĂ©ges. <code>@return boolean</code> [Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]</p>
<p>◆ checkAttackPossible (defenderID : int , attackerID : int) : boolean Public Attack checking method <code>@return</code> [Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]</p>
<p>◆ checkIfCapturedAndConquer (defender : Territory) : boolean Public Capture and conquer checking method <code>@return boolean</code> Properties: <code>ea_guid = {BFF4F3B6-2538-4778-9372-EAA3B468EDD8}</code> [Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]</p>
<p>◆ checkIfTerrotiryIsEmpty (territory : Territory) : boolean Public Empty territory checking method <code>@return boolean</code> Properties: <code>ea_guid = {BE4F4E07-A4C4-4e91-9F00-EF1EEB65D26C}</code> [Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]</p>
<p>◆ checkTransferPossible (from : int , to : int) : boolean Public Transfer checking method <code>@return boolean</code> [Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]</p>
<p>◆ endBattlePhase () : boolean Public Battle <code>@return boolean</code> Properties: <code>throws = Exception</code> [Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]</p>
<p>◆ endTurn () : int Public Kor vege Returns next player's id <code>@return int</code> Properties: <code>throws = Exception</code> <code>ea_guid = {F5F8D5D6-8997-43de-A5AD-DDDC647E775F}</code></p>

OPERATIONS	
[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]	
<p>◆ getCurrentPlayer () : int Public</p> <p>Getter for current player @return int</p> <p>[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]</p>	
<p>◆ getPhase () : Phase Public</p> <p>Getter for Phase @return phase</p> <p>[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]</p>	
<p>◆ getPlayerCards (id : int) : List<Integer> Public</p> <p>[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]</p>	
<p>◆ getPlayerName (id : int) : String Public</p> <p>Getter for Player's name @return String</p> <p>[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]</p>	
<p>◆ getPlayers () : List<String> Public</p> <p>Getter for Players list @return List<Player></p> <p>[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]</p>	
<p>◆ getTerritory (id : int) : Territory Public</p> <p>Getter for Territory @return Territory</p> <p>[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]</p>	
<p>◆ getUnitsLeftToReinforce (playerID : int) : int Public</p> <p>Getter for left units in reinforce phase @return int</p> <p>[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]</p>	
<p>◆ loadGame (file : File) : boolean Public</p> <p>Load data method @return boolean</p> <p>Properties: throws = Exception</p> <p>[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]</p>	
<p>◆ moveUnits (units : int) : boolean Public</p>	

OPERATIONS	
Move units method @return boolean	Properties: throws = Exception [Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ nextPhase () : Phase Private	Calculate the next phase @return Phase
Properties: throws = Exception	[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ nextPlayer () : void Protected	Initializes the next Player's turn. Calculates the amount of Units to place.
Properties: throws = Exception ea_guid = {806D8A15-D81F-4852-9BFA-E0988D68A2D8}	[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ reinforce (territoryId : int) : boolean Public	Reinforce method @return boolean
Properties: throws = Exception	[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ RiskGameModel () : Public	The amount of Units left to Place in the Reinforcement Stage of the current Player's current turn. [Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ saveGame (file : File) : boolean Public	Save data method @return
Properties: throws = IOException	[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ selectStartingPlayer (list : List<Player>) : Player Public	Random select the starting player @return Player
	[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]

OPERATIONS

◆ transfer (fromID : int , toID : int , units : int) : boolean Public

@return boolean

Properties:

throws = Exception

[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]

◆ transfer (from : Territory , to : Territory , units : int) : boolean Public

Transfer method

@return boolean

Properties:

throws = Exception

ea_guid = {23034260-4F89-40e8-BC43-DD7B2D6023D5}

[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]

◆ winGame (winner : int) : void Private

[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]

RiskGameController

Class in package 'Controller'

Az MVC architektura Controller részét megvalósító osztálya. A Controller a Modell és a View közötti kapcsolatért felelős. Rajta keresztül hajtják végre a View-on történt változások, melyek a Model-t érintik, illetve a Model a Controller-en keresztül módosítja a View-t.

RiskGameController

Version 1.0 Phase 1.0 Proposed

Szabó Dávid created on 2017. 03. 01.. Last modified 2017. 11. 22.

Extends java.util.Observable

INCOMING STRUCTURAL RELATIONSHIPS

⇒ Aggregation from JFXAttackView to RiskGameController

[Direction is 'Source -> Destination'.]

⇒ Aggregation from JFXAddPlayerView to RiskGameController

[Direction is 'Source -> Destination'.]

⇒ Aggregation from JFXTransferView to RiskGameController

[Direction is 'Source -> Destination'.]

⇒ Aggregation from RiskGameModel to RiskGameController

[Direction is 'Source -> Destination'.]

⇒ Aggregation from JFXMainView to RiskGameController

[Direction is 'Source -> Destination'.]

ATTRIBUTES	
◆ addPlayerView : JFXAddPlayerView	Private [Is static False. Containment is Not Specified.]
◆ attackResultView : JFXAttackResultView	Private [Is static False. Containment is Not Specified.]
◆ attackView : JFXAttackView	Private [Is static False. Containment is Not Specified.]
◆ cardGroup : Group	Private [Is static False. Containment is Not Specified.]
◆ cardScene : Scene	Private [Is static False. Containment is Not Specified.]
◆ cardView : JFXCardView	Private CardView stuffs [Is static False. Containment is Not Specified.]
◆ lastPhaseUpdate : Phase	Private [Is static False. Containment is Not Specified.]
◆ lastPlayerUpdate : int	Private [Is static False. Containment is Not Specified.]
◆ mainView : JFXMainView	Private View objects [Is static False. Containment is Not Specified.]
◆ model : RiskGameModel	Private [Is static False. Containment is Not Specified.]
◆ popupStage : Stage	Private [Is static False. Containment is Not Specified.]
◆ preStage : Stage	Private Const Stages [Is static False. Containment is Not Specified.]
◆ previouslySelectedTerritory : int	Private = -1 [Is static False. Containment is Not Specified.]

ATTRIBUTES	
◆ primaryStage : Stage	Private [Is static False. Containment is Not Specified.]
◆ titledPaneStage : Stage	Private [Is static False. Containment is Not Specified.]
◆ transferView : JFXTransferView	Private private JFXRiskCardView cardView; [Is static False. Containment is Not Specified.]
◆ victoryStage : Stage	Private [Is static False. Containment is Not Specified.]
◆ victoryView : JFXVictoryView	Private VictoryView stuffs [Is static False. Containment is Not Specified.]

ASSOCIATIONS	
✍ Association (direction: Source -> Destination)	Source: Public (Class) JFXViewBase Target: Protected controller (Class) RiskGameController

OPERATIONS	
◆ addLog (log : String) : void	Public A főablak naplójába küld egy új bejegyzést [Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ attackAccepted (defender : int, attacker : int, defenderUnits : int, attackerUnits : int) : void	Public [Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ attackRetreat () : void	Public [Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ attackShowed (attacker : int, defender : int) : void	Public [Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ closePopupWindow () : void	Private [Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ convertToCountry (ID : int) : Country	Private [Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]

OPERATIONS	
◆ convertToTerritoryID (country : Country) : int	Private [Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ countrySelected (country : Country) : void	Public [Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ getPhase () : Phase	Public [Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ getPreviouslySelectedTerritory () : int	Public [Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ getTerritoryData () : List<Territory>	Public [Is static True. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ isVictory () : boolean	Private [Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ loadGame (loadFile : File) : void	Public [Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ moveCardView () : void	Public [Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ nextPlayerSelected () : void	Public [Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ reinforce (territoryID : int, country : String) : void	Private Properties: throws = Exception [Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ RiskGameController (stage : Stage) : void	Public [Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ saveGame (saveFile : File) : void	Public [Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ setModel (model : RiskGameModel) : void	Public Eltárolja a modellt és feliratkoztatja magát az eseményre [Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ showAddPlayerView () : void	Public

OPERATIONS	
Add player view	[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◊ showAttackResultView () : void Private	[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◊ showAttackView (defenderID : int , attackerID : int) : void Private	
Attack View	[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◊ showCardView () : void Public	[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◊ showMainView () : void Public	
Main View	[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◊ showTransferView (fromID : int , toID : int) : void Private	
Transfer View	[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◊ showVictoryView () : void Public	[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◊ switchToMain (map : Map<Color, String>) : void Public	
Stage switch Pre -> Main	[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◊ transferAccepted (from : int , to : int , units : int) : void Public	[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◊ transferPhaseSelected () : void Public	[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◊ UpdateAllTerritoriesOnMap () : void Private	[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◊ UpdateCurrentDeck () : void Private	[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◊ UpdateCurrentPhase () : void Private	[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]

OPERATIONS

- ◆ UpdateCurrentPlayer () : void Private
[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
- ◆ UpdateTerritoryOnMap (territoryID : int) : void Private
[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]

JFXAttackView*Class in package 'View'*

Ezen a View-en keresztul jelölheti meg az egyik játékos a másik játékos (ellenfel) azon területet, amelyet tamadni szeretne.

JFXAttackView
Version 1.0 Phase 1.0 Proposed
Szabó Dávid created on 2017. 03. 08.. Last modified 2017. 11. 22.
Extends JFXViewBase

OUTGOING STRUCTURAL RELATIONSHIPS

- ↳ Generalization from JFXAttackView to JFXViewBase
[Direction is 'Source -> Destination'.]
- ↳ Aggregation from JFXAttackView to RiskGameController
[Direction is 'Source -> Destination'.]

ATTRIBUTES

- ◆ attackerID : int Private
[Is static False. Containment is Not Specified.]
- ◆ attackerUnits : int Private
[Is static False. Containment is Not Specified.]
- ◆ barUnits : ProgressBar Private

Properties:
annotations = @FXML
[Is static False. Containment is Not Specified.]
- ◆ btnAttack : Button Private

Properties:
annotations = @FXML
[Is static False. Containment is Not Specified.]
- ◆ btnRetreat : Button Private

Properties:

ATTRIBUTES	
annotations = @FXML	[Is static False. Containment is Not Specified.]
◆ defenderID : int Private	[Is static False. Containment is Not Specified.]
◆ defenderUnits : int Private	[Is static False. Containment is Not Specified.]
◆ lblAttackerAlive : Label Private	
Properties:	
annotations = @FXML	[Is static False. Containment is Not Specified.]
◆ lblAttackerName : Label Private	
Properties:	
annotations = @FXML	[Is static False. Containment is Not Specified.]
◆ lblCountryAttacker : Label Private	
Properties:	
annotations = @FXML	[Is static False. Containment is Not Specified.]
◆ lblCountryDefender : Label Private	
Properties:	
annotations = @FXML	[Is static False. Containment is Not Specified.]
◆ lblDefenderAlive : Label Private	
Properties:	
annotations = @FXML	[Is static False. Containment is Not Specified.]
◆ lblDefenderName : Label Private	
Properties:	
annotations = @FXML	[Is static False. Containment is Not Specified.]
◆ spinneratt : Spinner Private	
Properties:	
annotations = @FXML	[Is static False. Containment is Not Specified.]

ATTRIBUTES	
◆ spinnerdef : Spinner	Private
Properties:	
annotations = @FXML	
[Is static False. Containment is Not Specified.]	
OPERATIONS	
◆ finalize () : void	
Properties:	
throws = Throwable	
annotations = @Override	
[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]	
◆ initialize (location : URL , resources : ResourceBundle) : void	
Properties:	
annotations = @Override	
[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]	
◆ JFXAttackView () : Public	
[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]	
◆ UpdateViewState (attackResult : AttackResult) : void	
[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]	
◆ UpdateViewState (defenderName : String , attackerName : String , defender : Territory , attacker : Territory) : void	
[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]	

JFXMainView

Class in package 'View'

Az MVC architektura View részet megvalósító osztálya. Ez a View a játék főkepernyője, ezen történik maga a játék (egysegek elhelyezése, csata stb.).

JFXMainView
Version 1.0 Phase 1.0 Proposed
Szabó Dávid created on 2017. 03. 01.. Last modified 2017. 11. 22.
Extends JFXViewBase

OUTGOING STRUCTURAL RELATIONSHIPS	
↳ Generalization from JFXMainView to JFXViewBase	[Direction is 'Source -> Destination'.]
↳ Aggregation from JFXMainView to RiskGameController	[Direction is 'Source -> Destination'.]

ATTRIBUTES
◆ BASE_HEIGHT : double Private Const = 600 [Is static False. Containment is Not Specified.]
◆ BASE_WIDTH : double Private Const = 815 [Is static False. Containment is Not Specified.]
◆ borderPane : BorderPane Private Properties: annotations = @FXML [Is static False. Containment is Not Specified.]
◆ btnBattlePhase : Button Private Properties: annotations = @FXML [Is static False. Containment is Not Specified.]
◆ btnControlList : ArrayList<Button> Private [Is static False. Containment is Not Specified.]
◆ btnLoad : Button Private Properties: annotations = @FXML [Is static False. Containment is Not Specified.]
◆ btnLog : Button Private Properties: annotations = @FXML [Is static False. Containment is Not Specified.]
◆ btnNextPlayer : Button Private Properties: annotations = @FXML [Is static False. Containment is Not Specified.]
◆ btnReinforcePhase : Button Private Properties: annotations = @FXML [Is static False. Containment is Not Specified.]
◆ btnSave : Button Private Properties: annotations = @FXML [Is static False. Containment is Not Specified.]

ATTRIBUTES
<p>◆ btnSettings : Button Private</p> <p>Properties:</p> <p> annotations = @FXML</p> <p>[Is static False. Containment is Not Specified.]</p>
<p>◆ btnTransferPhase : Button Private</p> <p>Properties:</p> <p> annotations = @FXML</p> <p>[Is static False. Containment is Not Specified.]</p>
<p>◆ centerStackPane : StackPane Private</p> <p>Properties:</p> <p> annotations = @FXML</p> <p>[Is static False. Containment is Not Specified.]</p>
<p>◆ dimLog : Dimension Package</p> <p>[Is static False. Containment is Not Specified.]</p>
<p>◆ imageView : ImageView Private</p> <p>Properties:</p> <p> annotations = @FXML</p> <p>[Is static False. Containment is Not Specified.]</p>
<p>◆ isLogOpened : boolean Package</p> <p>[Is static False. Containment is Not Specified.]</p>
<p>◆ isSLOpened : boolean Package</p> <p>[Is static False. Containment is Not Specified.]</p>
<p>◆ lblPlayerName1 : Label Private</p> <p>Properties:</p> <p> annotations = @FXML</p> <p>[Is static False. Containment is Not Specified.]</p>
<p>◆ lblPlayerName2 : Label Private</p> <p>Properties:</p> <p> annotations = @FXML</p> <p>[Is static False. Containment is Not Specified.]</p>
<p>◆ lblPlayerName3 : Label Private</p> <p>Properties:</p> <p> annotations = @FXML</p>

ATTRIBUTES	
	[Is static False. Containment is Not Specified.]
◆ lblPlayerName4 : Label Private	
Properties: annotations = @FXML	[Is static False. Containment is Not Specified.]
◆ lblPlayerName5 : Label Private	
Properties: annotations = @FXML	[Is static False. Containment is Not Specified.]
◆ lblPlayerName6 : Label Private	
Properties: annotations = @FXML	[Is static False. Containment is Not Specified.]
◆ lblPlayerNameList : ArrayList<Label> Private	[Is static False. Containment is Not Specified.]
◆ pain : Pane Private	
Properties: annotations = @FXML	[Is static False. Containment is Not Specified.]
◆ panePlayerColor1 : Pane Private	
Properties: annotations = @FXML	[Is static False. Containment is Not Specified.]
◆ panePlayerColor2 : Pane Private	
Properties: annotations = @FXML	[Is static False. Containment is Not Specified.]
◆ panePlayerColor3 : Pane Private	
Properties: annotations = @FXML	[Is static False. Containment is Not Specified.]
◆ panePlayerColor4 : Pane Private	
Properties: annotations = @FXML	[Is static False. Containment is Not Specified.]

ATTRIBUTES	
◆ panePlayerColor5 : Pane Private	
Properties:	
annotations = @FXML	[Is static False. Containment is Not Specified.]
◆ panePlayerColor6 : Pane Private	
Properties:	
annotations = @FXML	[Is static False. Containment is Not Specified.]
◆ panePlayerColorList : ArrayList<Pane> Private	[Is static False. Containment is Not Specified.]
◆ stage : Stage Private	[Is static False. Containment is Not Specified.]
◆ taLog : TextArea Private	
Properties:	
annotations = @FXML	[Is static False. Containment is Not Specified.]
◆ world : World Private	[Is static False. Containment is Not Specified.]

OPERATIONS	
◆ AddControllerListener (controller : RiskGameController) : void Public	
Properties:	
annotations = @Override	[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ appendLog (s : String) : void Public	[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ closeLog () : void Public	[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ finalize () : void Public	
Properties:	
throws = Throwable	
annotations = @Override	[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]

OPERATIONS	
◆ fit (height : double , width : double) : void	Public [Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ getRoot () : Parent	Public Properties: annotations = @Override [Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ getWorld () : World	Public [Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ handleLogButton (event : ActionEvent) : void	Private Properties: annotations = @FXML [Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ hangleLoadButton (event : ActionEvent) : void	Private Properties: annotations = @FXML [Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ hangleSaveButton (event : ActionEvent) : void	Private Properties: annotations = @FXML [Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ hangleSettingsButton (event : ActionEvent) : void	Private Properties: annotations = @FXML [Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ initialize (location : URL , resources : ResourceBundle) : void	Public Properties: annotations = @Override [Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ isLogOpened () : boolean	Public [Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ isSLOpened () : boolean	Public [Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ JFXMainView (stage : Stage) : Public	[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]

OPERATIONS	
◆ openLog () : void Public	[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ UpdateConnectedPlayers (playerList : List<String>) : void Public	[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ UpdateCurrentDeck (cards : List<Integer>) : void Public	
TODO:	[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ UpdateCurrentPhase (phase : Phase) : void Public	[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ UpdateCurrentPlayer (playerID : int) : void Public	
@param player	[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ updateIcons (attacker : int, defender : int) : void Public	[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ UpdateViewState (territory : Territory) : void Public	
@param territories	
Properties:	
ea_guid = {0E811D72-EC4E-47ad-8B46-C39084EAD8F7}	[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ UpdateViewState (territories : List<Territory>) : void Public	[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]

<anonymous>

Note in package 'Game Stage'

Main Battle Sequence

<anonymous>
Version 1.0 Phase 1.0 Proposed
Oliver created on 2017. 03. 28.. Last modified 2017. 03. 28.
Extends

INTERACTION MESSAGES

✉ 1.0 '//init attack' from 'Player 1' sent to 'JFXMainView'.

Synchronous Call. Returns void.

[Return is False. Iteration is False. New group is False.]

✉ 1.1 'showAttackView()' from 'JFXMainView' sent to 'RiskGameController'.

Synchronous Call.

[Return is False. Iteration is False. New group is False.]

✉ 1.2 'AddControllerListener(ActionListener)' from 'RiskGameController' sent to 'JFXAttackView'.

Synchronous Call.

[Return is False. Iteration is False. New group is False.]

✉ 1.3 '//select attacking units' from 'Player 1' sent to 'JFXAttackView'.

Asynchronous Signal.

[Return is False. Iteration is False. New group is False.]

✉ 1.4 '//select defending units' from 'Player 2' sent to 'JFXAttackView'.

Asynchronous Signal.

[Return is False. Iteration is False. New group is False.]

✉ 1.5 'actionPerformed(ActionEvent)' from 'JFXAttackView' sent to 'RiskGameController'.

Synchronous Call.

[Return is False. Iteration is False. New group is False.]

✉ 1.6 'attackTerritory(Territory, Territory, int, int)' from 'RiskGameController' sent to 'RiskGameModel'.

Synchronous Call.

[Return is False. Iteration is False. New group is False.]

✉ 1.7 'getUnits()' from 'RiskGameModel' sent to 'attacker: Territory'.

Synchronous Call. Returns int.

[Return is False. Iteration is False. New group is False.]

✉ 1.8 'getUnits()' from 'RiskGameModel' sent to 'defender: Territory'.

Synchronous Call. Returns int.

[Return is False. Iteration is False. New group is False.]

✉ 1.9 'checkAttackPossible(Territory, int)' from 'RiskGameModel' sent to 'RiskGameModel'.

Synchronous Call. Returns int.

[Return is False. Iteration is False. New group is False.]

✉ 1.10 'doBattle(int, int)' from 'RiskGameModel' sent to 'RiskGameModel'.

Synchronous Call.	[Return is False. Iteration is False. New group is False.]
✉ 1.11 'setUnits(int)' from 'RiskGameModel' sent to 'attacker: Territory'.	
Synchronous Call. Returns void.	[Return is False. Iteration is False. New group is False.]
✉ 1.12 'setUnits(int)' from 'RiskGameModel' sent to 'defender: Territory'.	
Synchronous Call. Returns void.	[Return is False. Iteration is False. New group is False.]
✉ 1.13 'checkIfCapturedAndConquer(Territory)' from 'RiskGameModel' sent to 'RiskGameModel'.	
Synchronous Call. Returns int.	[Return is False. Iteration is False. New group is False.]
✉ 1.14 'setOwner(int)' from 'RiskGameModel' sent to 'defender: Territory'.	
Synchronous Call. Returns void.	[Return is False. Iteration is False. New group is False.]
✉ 1.15 'addTerritory()' from 'RiskGameModel' sent to 'attacker: Player'.	
Synchronous Call.	[Return is False. Iteration is False. New group is False.]
✉ 1.16 'removeTerritory()' from 'RiskGameModel' sent to 'defender: Player'.	
Synchronous Call.	[Return is False. Iteration is False. New group is False.]
✉ 1.17 'setUnits(int)' from 'RiskGameModel' sent to 'defender: Territory'.	
Synchronous Call. Returns void.	[Return is False. Iteration is False. New group is False.]
✉ 1.18 'setUnits(int)' from 'RiskGameModel' sent to 'attacker: Territory'.	
Synchronous Call. Returns void.	[Return is False. Iteration is False. New group is False.]
✉ 1.19 'update(Observable, Object)' from 'RiskGameModel' sent to 'RiskGameController'.	
Synchronous Call.	[Return is False. Iteration is False. New group is False.]
✉ 1.20 'UpdateViewState(Territory)' from 'RiskGameController' sent to 'JFXMainView'.	
Synchronous Call.	[Return is False. Iteration is False. New group is False.]

Reinforcement diagram

Interaction diagram in package 'Game Stage'

Reinforcement
Version 1.0

Szabó Dávid created on 2017. 03. 22.. Last modified 2017. 09. 21.

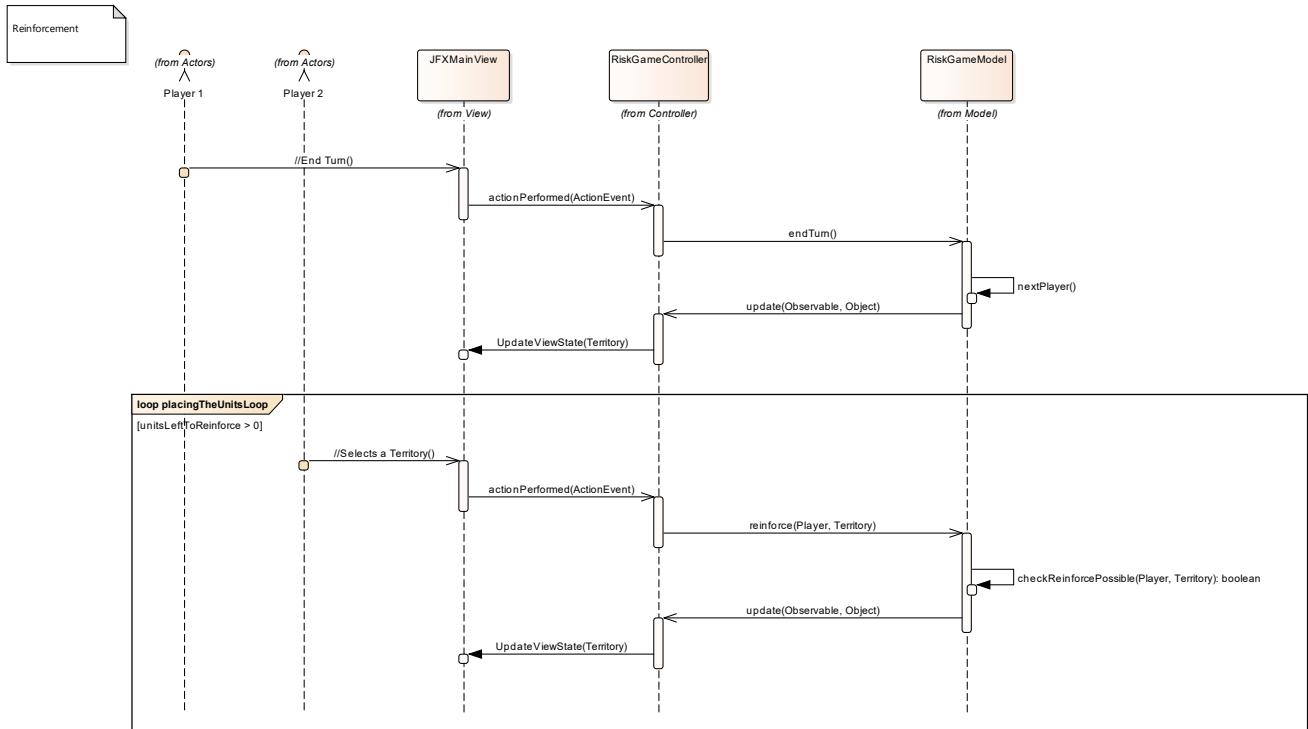


Figure 13: Reinforcement

Player 1

Actor in package 'Actors'

Player 1

Version 1.0 Phase 1.0 Mandatory

Oliver created on 2017. 02. 21.. Last modified 2017. 03. 23.

OUTGOING BEHAVIORAL RELATIONSHIPS
Name: //Selects 2 Territories ↳ Sequence from Player 1 to JFXMainView
Name: //select attacking units ↳ Sequence from Player 1 to JFXAttackView
↳ Sequence from Player 1 to Reinforcement
↳ Sequence from Player 1 to Load Game

OUTGOING BEHAVIORAL RELATIONSHIPS
↳ Sequence from Player 1 to Exit Game
↳ Sequence from Player 1 to Save Game
↳ Sequence from Player 1 to Exchange Cards
↳ Sequence from Player 1 to Define the starting player
↳ Sequence from Player 1 to Define number of players
Name: // Selects option ↳ Sequence from Player 1 to JFXMainView
Name: // Add number of players ↳ Sequence from Player 1 to JFXAddPlayerView
Name: //Set a Save Name then Apply ↳ Sequence from Player 1 to JFXMainView
Name: // click to begin action ↳ Sequence from Player 1 to JFXMainView
Name: // click to connect ↳ Sequence from Player 1 to JFXMainView
Name: //End Turn ↳ Sequence from Player 1 to JFXMainView
Name: // Selects exit game option ↳ Sequence from Player 1 to JFXMainView
Name: // Select saved game file then apply ↳ Sequence from Player 1 to JFXMainView
↳ Sequence from Player 1 to Main Battle Sequence
Name: //init attack ↳ Sequence from Player 1 to JFXMainView
Name: //end turn ↳ Sequence from Player 1 to JFXMainView
↳ Sequence from Player 1 to Transfer Units

OUTGOING BEHAVIORAL RELATIONSHIPS
Name: // select Yes or No ➡ Sequence from Player 1 to JFXMainView
Name: //Selects Units then Apply ➡ Sequence from Player 1 to JFXTransferView
➡ Sequence from Player 1 to Preparation stage
INCOMING BEHAVIORAL RELATIONSHIPS
➡ Sequence from Transfer Units to Player 1
➡ Sequence from Save Game to Player 1
➡ Sequence from Main Battle Sequence to Player 1
➡ Sequence from Preparation stage to Player 1
➡ Sequence from Reinforcement to Player 1
➡ Sequence from Exchange Cards to Player 1
Name: // popup ➡ Sequence from JFXMainView to Player 1
Name: addTerritory() ➡ Sequence from RiskGameModel to Player 1
➡ Sequence from Define number of players to Player 1
➡ Sequence from Exit Game to Player 1
➡ Sequence from Define the starting player to Player 1
➡ Sequence from Load Game to Player 1
CONNECTORS

CONNECTORS	
 UseCaseLink	Source -> Destination From: Player 1 : Actor, Public To: Reinforcement : UseCase, Public
 UseCaseLink	Source -> Destination From: Player 1 : Actor, Public To: Transfer Units : UseCase, Public
 UseCaseLink	Source -> Destination From: Player 1 : Actor, Public To: Main Battle Sequence : UseCase, Public

ASSOCIATIONS	
 Association (direction: Unspecified)	Source: Public (Actor) Player 1 Target: Public (UseCase) Exit Game

Player 2

Actor in package 'Actors'

Player 2
Version 1.0 Phase 1.0 Mandatory
Oliver created on 2017. 03. 22.. Last modified 2017. 03. 23.

OUTGOING BEHAVIORAL RELATIONSHIPS	
Name: //Selects a Territory	 Sequence from Player 2 to JFXMainView
Name: //select defending units	 Sequence from Player 2 to JFXAttackView

RiskGameModel

Class in package 'Model'

Az MVC architektura Model részét megvalósító osztálya.

RiskGameModel
Version 1.0 Phase 1.0 Proposed
Szabi created on 2017. 02. 21.. Last modified 2017. 11. 22.

OUTGOING STRUCTURAL RELATIONSHIPS	
-----------------------------------	--

OUTGOING STRUCTURAL RELATIONSHIPS	
↳ Aggregation from RiskGameModel to RiskGameController	[Direction is 'Source -> Destination'.]
INCOMING STRUCTURAL RELATIONSHIPS	
⇒ Aggregation from Map to RiskGameModel	[Direction is 'Source -> Destination'.]
⇒ Aggregation from Deck to RiskGameModel	[Direction is 'Source -> Destination'.]
ATTRIBUTES	
◆ calculatedCircleNumber : int Private	[Is static False. Containment is Not Specified.]
◆ capturedThisTurn : Boolean Private	[Is static False. Containment is Not Specified.]
◆ circlenumber : int Private	[Is static False. Containment is Not Specified.]
◆ currentPlayer : int Private	[Is static False. Containment is Not Specified.]
◆ deck : Deck Private	[Is static False. Containment is Not Specified.]
◆ hasTransferred : boolean Private	[Is static False. Containment is Not Specified.]
◆ map : Map Private	[Is static False. Containment is Not Specified.]
◆ misnumber : int Private	[Is static False. Containment is Not Specified.]
◆ phase : Phase Private	[Is static False. Containment is Not Specified.]
◆ playersList : List<Player> Private	[Is static False. Containment is Not Specified.]
◆ r : Random Private = new Random()	

ATTRIBUTES	
	[Is static False. Containment is Not Specified.]
◆ waitForUnitsTemp : Territory Private = new Territory[2]	[Is static False. Containment is Not Specified.]
ASSOCIATIONS	
✓ . Association (direction: Source -> Destination)	
Source: Public (Class) RiskGameModel	Target: Private waitForUnitsTemp (Class) Territory
✓ . Association (direction: Source -> Destination)	
Source: Public (Class) RiskGameModel	Target: Private playersList (Class) Player Cardinality: [0..*]
✓ . Association (direction: Source -> Destination)	
Source: Public (Class) RiskGameModel	Target: Public deck (Class) Player
OPERATIONS	
◆ addPlayer (name : String , color : Color) : boolean Public	
Add player with name and color @return boolean	[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ addPlayerToPlayerList (player : Player) : void Public	
Adder for playerList	
Properties: throws = Exception ea_guid = {45684F69-D727-4875-BEF7-5232B4E9D1C9}	[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ addPlayerToPlayerList (map : java.util.Map<Color, String>) : void Public	
Pairs overload	
Properties: throws = Exception	[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ attackTerritoryWithResult (defenderID : int , attackerID : int , defendUnits : int , attackUnits : int) : AttackResult Public	
Return the result of the attack If the attacker have more than 3 units, the fighting attacker units will reduce to 3 If the defender have more than 2 units, the fighting defender units will reduce to 2	
Properties:	

OPERATIONS	
throws = Exception	[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ calculateCircle () : void Private	
Properties:	
throws = Exception	[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ checkAttackPossible (defender : Territory , attacker : Territory , defendUnits : int , attackUnits : int) : boolean Public	
Csak akkor tĂˇmadhat, ha a terĂĽleten elegendL' egysĂ©ge van. Igazzal tĂ©r vissza, ha a tamadas lehetsĂ©ges.	
@return boolean	[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ checkAttackPossible (defenderID : int , attackerID : int) : boolean Public	
Attack checking method	
@return	[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ checkIfCapturedAndConquer (defender : Territory) : boolean Public	
Capture and conquer checking method	
@return boolean	
Properties:	
ea_guid = {BFF4F3B6-2538-4778-9372-EAA3B468EDD8}	[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ checkIfTerritoryIsEmpty (territory : Territory) : boolean Public	
Empty territory checking method	
@return boolean	
Properties:	
ea_guid = {BE4F4E07-A4C4-4e91-9F00-EF1EEB65D26C}	[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ checkTransferPossible (from : int , to : int) : boolean Public	
Transfer checking method	
@return boolean	[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ endBattlePhase () : boolean Public	
Battle	
@return boolean	
Properties:	
throws = Exception	[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]

OPERATIONS
<p>❖ endTurn () : int Public</p> <p>Kor vege Returns next player's id @return int</p> <p>Properties: throws = Exception ea_guid = {F5F8D5D6-8997-43de-A5AD-DDDC647E775F} [Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]</p>
<p>❖ getCurrentPlayer () : int Public</p> <p>Getter for current player @return int</p> <p>[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]</p>
<p>❖ getPhase () : Phase Public</p> <p>Getter for Phase @return phase</p> <p>[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]</p>
<p>❖ getPlayerCards (id : int) : List<Integer> Public</p> <p>[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]</p>
<p>❖ getPlayerName (id : int) : String Public</p> <p>Getter for Player's name @return String</p> <p>[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]</p>
<p>❖ getPlayers () : List<String> Public</p> <p>Getter for Players list @return List<Player></p> <p>[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]</p>
<p>❖ getTerritory (id : int) : Territory Public</p> <p>Getter for Territory @return Territory</p> <p>[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]</p>
<p>❖ getUnitsLeftToReinforce (playerID : int) : int Public</p> <p>Getter for left units in reinforce phase @return int</p> <p>[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]</p>
<p>❖ loadGame (file : File) : boolean Public</p> <p>Load data method</p>

OPERATIONS	
@return boolean	
Properties:	
throws = Exception	[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ moveUnits (units : int) : boolean Public	
Move units method	
@return boolean	
Properties:	
throws = Exception	[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ nextPhase () : Phase Private	
Calculate the next phase	
@return Phase	
Properties:	
throws = Exception	[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ nextPlayer () : void Protected	
Initializes the next Player's turn. Calculates the amount of Units to place.	
Properties:	
throws = Exception	
ea_guid = {806D8A15-D81F-4852-9BFA-E0988D68A2D8}	[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ reinforce (territoryId : int) : boolean Public	
Reinforce method	
@return boolean	
Properties:	
throws = Exception	[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ RiskGameModel () : Public	
The amount of Units left to Place in the Reinforcement Stage of the current Player's current turn.	
[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]	
◆ saveGame (file : File) : boolean Public	
Save data method	
@return	
Properties:	
throws = IOException	[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]

OPERATIONS
<p>◆ selectStartingPlayer (list : List<Player>) : Player Public Random select the starting player <code>@return Player</code> <code>[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]</code></p>
<p>◆ transfer (fromID : int , toID : int , units : int) : boolean Public <code>@return boolean</code> Properties: <code>throws = Exception</code> <code>[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]</code></p>
<p>◆ transfer (from : Territory , to : Territory , units : int) : boolean Public Transfer method <code>@return boolean</code> Properties: <code>throws = Exception</code> <code>ea_guid = {23034260-4F89-40e8-BC43-DD7B2D6023D5}</code> <code>[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]</code></p>
<p>◆ winGame (winner : int) : void Private <code>[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]</code></p>

RiskGameController

Class in package 'Controller'

Az MVC architektura Controller részet megvalósító osztálya. A Controller a Modell és a View közötti kapcsolatért felelős. Rajta keresztül hajtják végre a View-on történt változások, melyek a Model-t érintik, illetve a Model a Controller-en keresztül módosítja a View-t.

RiskGameController
 Version 1.0 Phase 1.0 Proposed
 Szabó Dávid created on 2017. 03. 01.. Last modified 2017. 11. 22.
 Extends java.util.Observable

INCOMING STRUCTURAL RELATIONSHIPS
<p>⇒ Aggregation from JFXAttackView to RiskGameController <code>[Direction is 'Source -> Destination'.]</code></p>
<p>⇒ Aggregation from JFXAddPlayerView to RiskGameController <code>[Direction is 'Source -> Destination'.]</code></p>
<p>⇒ Aggregation from JFXTransferView to RiskGameController <code>[Direction is 'Source -> Destination'.]</code></p>

INCOMING STRUCTURAL RELATIONSHIPS	
⇒ Aggregation from RiskGameModel to RiskGameController	[Direction is 'Source -> Destination'.]
⇒ Aggregation from JFXMainView to RiskGameController	[Direction is 'Source -> Destination'.]
ATTRIBUTES	
◆ addPlayerView : JFXAddPlayerView Private	[Is static False. Containment is Not Specified.]
◆ attackResultView : JFXAttackResultView Private	[Is static False. Containment is Not Specified.]
◆ attackView : JFXAttackView Private	[Is static False. Containment is Not Specified.]
◆ cardGroup : Group Private	[Is static False. Containment is Not Specified.]
◆ cardScene : Scene Private	[Is static False. Containment is Not Specified.]
◆ cardView : JFXCardView Private	
CardView stuffs	[Is static False. Containment is Not Specified.]
◆ lastPhaseUpdate : Phase Private	[Is static False. Containment is Not Specified.]
◆ lastPlayerUpdate : int Private	[Is static False. Containment is Not Specified.]
◆ mainView : JFXMainView Private	
View objects	[Is static False. Containment is Not Specified.]
◆ model : RiskGameModel Private	[Is static False. Containment is Not Specified.]
◆ popupStage : Stage Private	[Is static False. Containment is Not Specified.]

ATTRIBUTES	
◆ preStage : Stage Private Const	
Stages	[Is static False. Containment is Not Specified.]
◆ previouslySelectedTerritory : int Private = -1	[Is static False. Containment is Not Specified.]
◆ primaryStage : Stage Private	[Is static False. Containment is Not Specified.]
◆ titledPaneStage : Stage Private	[Is static False. Containment is Not Specified.]
◆ transferView : JFXTransferView Private	
private JFXRiskCardView cardView;	[Is static False. Containment is Not Specified.]
◆ victoryStage : Stage Private	[Is static False. Containment is Not Specified.]
◆ victoryView : JFXVictoryView Private	
VictoryView stuffs	[Is static False. Containment is Not Specified.]

ASSOCIATIONS	
◆ Association (direction: Source -> Destination)	

Source: Public (Class) JFXViewBase

Target: Protected controller (Class)
RiskGameController

OPERATIONS	
◆ addLog (log : String) : void Public	
A főablak naplójába küld egy új bejegyzést	[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ attackAccepted (defender : int , attacker : int , defenderUnits : int , attackerUnits : int) : void Public	[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ attackRetreat () : void Public	[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]

OPERATIONS	
◆ attackShowed (attacker : int , defender : int) : void	Public [Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ closePopupWindow () : void	Private [Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ convertToCountry (ID : int) : Country	Private [Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ convertToTerritoryID (country : Country) : int	Private [Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ countrySelected (country : Country) : void	Public [Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ getPhase () : Phase	Public [Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ getPreviouslySelectedTerritory () : int	Public [Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ getTerritoryData () : List<Territory>	Public [Is static True. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ isVictory () : boolean	Private [Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ loadGame (loadFile : File) : void	Public [Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ moveCardView () : void	Public [Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ nextPlayerSelected () : void	Public [Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ reinforce (territoryID : int , country : String) : void	Private
Properties:	
throws = Exception	[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ RiskGameController (stage : Stage) : Public	[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]

OPERATIONS	
◆ saveGame (saveFile : File) : void Public	[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ setModel (model : RiskGameModel) : void Public	Eltárolja a modellt és feliratkoztatja magát az eseményeire [Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ showAddPlayerView () : void Public	Add player view [Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ showAttackResultView () : void Private	[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ showAttackView (defenderID : int, attackerID : int) : void Private	Attack View [Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ showCardView () : void Public	[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ showMainView () : void Public	Main View [Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ showTransferView (fromID : int, toID : int) : void Private	Transfer View [Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ showVictoryView () : void Public	[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ switchToMain (map : Map<Color, String>) : void Public	Stage switch Pre -> Main [Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ transferAccepted (from : int, to : int, units : int) : void Public	[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ transferPhaseSelected () : void Public	[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]

OPERATIONS	
◆ UpdateAllTerritoriesOnMap () : void	Private [Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ UpdateCurrentDeck () : void	Private [Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ UpdateCurrentPhase () : void	Private [Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ UpdateCurrentPlayer () : void	Private [Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ UpdateTerritoryOnMap (territoryID : int) : void	Private [Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]

JFXMainView

Class in package 'View'

Az MVC architektura View részét megvalósító osztalya. Ez a View a játék főkepernyője, ezen történik maga a játék (egysegek elhelyezése, csata stb.).

JFXMainView
Version 1.0 Phase 1.0 Proposed
Szabó Dávid created on 2017. 03. 01.. Last modified 2017. 11. 22.
Extends JFXViewBase

OUTGOING STRUCTURAL RELATIONSHIPS	
↳ Generalization from JFXMainView to JFXViewBase	[Direction is 'Source -> Destination'.]
↳ Aggregation from JFXMainView to RiskGameController	[Direction is 'Source -> Destination'.]

ATTRIBUTES	
◆ BASE_HEIGHT : double	Private Const = 600 [Is static False. Containment is Not Specified.]
◆ BASE_WIDTH : double	Private Const = 815 [Is static False. Containment is Not Specified.]
◆ borderPane : BorderPane	Private Properties: annotations = @FXML [Is static False. Containment is Not Specified.]

ATTRIBUTES
<p>◆ btnBattlePhase : Button Private</p> <p>Properties: annotations = @FXML</p> <p>[Is static False. Containment is Not Specified.]</p>
<p>◆ btnControlList : ArrayList<Button> Private</p> <p>[Is static False. Containment is Not Specified.]</p>
<p>◆ btnLoad : Button Private</p> <p>Properties: annotations = @FXML</p> <p>[Is static False. Containment is Not Specified.]</p>
<p>◆ btnLog : Button Private</p> <p>Properties: annotations = @FXML</p> <p>[Is static False. Containment is Not Specified.]</p>
<p>◆ btnNextPlayer : Button Private</p> <p>Properties: annotations = @FXML</p> <p>[Is static False. Containment is Not Specified.]</p>
<p>◆ btnReinforcePhase : Button Private</p> <p>Properties: annotations = @FXML</p> <p>[Is static False. Containment is Not Specified.]</p>
<p>◆ btnSave : Button Private</p> <p>Properties: annotations = @FXML</p> <p>[Is static False. Containment is Not Specified.]</p>
<p>◆ btnSettings : Button Private</p> <p>Properties: annotations = @FXML</p> <p>[Is static False. Containment is Not Specified.]</p>
<p>◆ btnTransferPhase : Button Private</p> <p>Properties: annotations = @FXML</p> <p>[Is static False. Containment is Not Specified.]</p>

ATTRIBUTES
◆ centerStackPane : StackPane Private Properties: annotations = @FXML [Is static False. Containment is Not Specified.]
◆ dimLog : Dimension Package [Is static False. Containment is Not Specified.]
◆ imageView : ImageView Private Properties: annotations = @FXML [Is static False. Containment is Not Specified.]
◆ isLogOpened : boolean Package [Is static False. Containment is Not Specified.]
◆ isSLOpened : boolean Package [Is static False. Containment is Not Specified.]
◆ lblPlayerName1 : Label Private Properties: annotations = @FXML [Is static False. Containment is Not Specified.]
◆ lblPlayerName2 : Label Private Properties: annotations = @FXML [Is static False. Containment is Not Specified.]
◆ lblPlayerName3 : Label Private Properties: annotations = @FXML [Is static False. Containment is Not Specified.]
◆ lblPlayerName4 : Label Private Properties: annotations = @FXML [Is static False. Containment is Not Specified.]
◆ lblPlayerName5 : Label Private Properties: annotations = @FXML [Is static False. Containment is Not Specified.]

ATTRIBUTES
<p>◆ lblPlayerName6 : Label Private</p> <p>Properties:</p> <p>annotations = @FXML</p> <p>[Is static False. Containment is Not Specified.]</p>
<p>◆ lblPlayerNameList : ArrayList<Label> Private</p> <p>[Is static False. Containment is Not Specified.]</p>
<p>◆ pain : Pane Private</p> <p>Properties:</p> <p>annotations = @FXML</p> <p>[Is static False. Containment is Not Specified.]</p>
<p>◆ panePlayerColor1 : Pane Private</p> <p>Properties:</p> <p>annotations = @FXML</p> <p>[Is static False. Containment is Not Specified.]</p>
<p>◆ panePlayerColor2 : Pane Private</p> <p>Properties:</p> <p>annotations = @FXML</p> <p>[Is static False. Containment is Not Specified.]</p>
<p>◆ panePlayerColor3 : Pane Private</p> <p>Properties:</p> <p>annotations = @FXML</p> <p>[Is static False. Containment is Not Specified.]</p>
<p>◆ panePlayerColor4 : Pane Private</p> <p>Properties:</p> <p>annotations = @FXML</p> <p>[Is static False. Containment is Not Specified.]</p>
<p>◆ panePlayerColor5 : Pane Private</p> <p>Properties:</p> <p>annotations = @FXML</p> <p>[Is static False. Containment is Not Specified.]</p>
<p>◆ panePlayerColor6 : Pane Private</p> <p>Properties:</p> <p>annotations = @FXML</p> <p>[Is static False. Containment is Not Specified.]</p>

ATTRIBUTES	
◆ panePlayerColorList : ArrayList<Pane>	Private [Is static False. Containment is Not Specified.]
◆ stage : Stage	Private [Is static False. Containment is Not Specified.]
◆ taLog : TextArea	Private Properties: annotations = @FXML [Is static False. Containment is Not Specified.]
◆ world : World	Private [Is static False. Containment is Not Specified.]

OPERATIONS	
◆ AddControllerListener (controller : RiskGameController) : void	Public
Properties:	
annotations = @Override	[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ appendLog (s : String) : void	Public [Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ closeLog () : void	Public [Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ finalize () : void	Public Properties: throws = Throwable annotations = @Override [Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ fit (height : double, width : double) : void	Public [Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ getRoot () : Parent	Public Properties: annotations = @Override [Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ getWorld () : World	Public [Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]

OPERATIONS	
◆ handleLogButton (event : ActionEvent) : void Private	
Properties:	
annotations = @FXML	[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ handleLoadButton (event : ActionEvent) : void Private	
Properties:	
annotations = @FXML	[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ handleSaveButton (event : ActionEvent) : void Private	
Properties:	
annotations = @FXML	[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ handleSettingsButton (event : ActionEvent) : void Private	
Properties:	
annotations = @FXML	[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ initialize (location : URL , resources : ResourceBundle) : void Public	
Properties:	
annotations = @Override	[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ isLogOpened () : boolean Public	
	[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ isSLOpened () : boolean Public	
	[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ JFXMainView (stage : Stage) : Public	
	[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ openLog () : void Public	
	[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ UpdateConnectedPlayers (playerList : List<String>) : void Public	
	[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ UpdateCurrentDeck (cards : List<Integer>) : void Public	
TODO:	[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]

OPERATIONS	
◆ UpdateCurrentPhase (phase : Phase) : void Public	[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ UpdateCurrentPlayer (playerID : int) : void Public @param player	[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ updateIcons (attacker : int, defender : int) : void Public	[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ UpdateViewState (territory : Territory) : void Public @param territories	Properties: ea_guid = {0E811D72-EC4E-47ad-8B46-C39084EAD8F7} [Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ UpdateViewState (territories : List<Territory>) : void Public	[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]

<anonymous>

Note in package 'Game Stage'

Reinforcement

<anonymous>
Version 1.0 Phase 1.0 Proposed
Oliver created on 2017. 03. 28.. Last modified 2017. 03. 29.
Extends

INTERACTION MESSAGES	
✉ 1.0 '/*End Turn' from 'Player 1' sent to 'JFXMainView'.	
Asynchronous Call. Returns void.	[Return is False. Iteration is False. New group is False.]
✉ 1.1 'actionPerformed(ActionEvent)' from 'JFXMainView' sent to 'RiskGameController'.	
Asynchronous Call.	[Return is False. Iteration is False. New group is False.]
✉ 1.2 'endTurn()' from 'RiskGameController' sent to 'RiskGameModel'.	
Asynchronous Call.	

	[Return is False. Iteration is False. New group is False.]
✉ 1.3 'nextPlayer()' from 'RiskGameModel' sent to 'RiskGameModel'.	
Synchronous Call.	[Return is False. Iteration is False. New group is False.]
✉ 1.4 'update(Observable, Object)' from 'RiskGameModel' sent to 'RiskGameController'.	
Asynchronous Call.	[Return is False. Iteration is False. New group is False.]
✉ 1.5 'UpdateViewState(Territory)' from 'RiskGameController' sent to 'JFXMainView'.	
Synchronous Call.	[Return is False. Iteration is False. New group is False.]
✉ 1.6 '//Selects a Territory' from 'Player 2' sent to 'JFXMainView'.	
Asynchronous Call. Returns void.	[Return is False. Iteration is False. New group is False.]
✉ 1.7 'actionPerformed(ActionEvent)' from 'JFXMainView' sent to 'RiskGameController'.	
Asynchronous Call.	[Return is False. Iteration is False. New group is False.]
✉ 1.8 'reinforce(Player, Territory)' from 'RiskGameController' sent to 'RiskGameModel'.	
Asynchronous Call.	[Return is False. Iteration is False. New group is False.]
✉ 1.9 'checkReinforcePossible(Player, Territory)' from 'RiskGameModel' sent to 'RiskGameModel'.	
Synchronous Call. Returns boolean.	[Return is False. Iteration is False. New group is False.]
✉ 1.10 'update(Observable, Object)' from 'RiskGameModel' sent to 'RiskGameController'.	
Asynchronous Call.	[Return is False. Iteration is False. New group is False.]
✉ 1.11 'UpdateViewState(Territory)' from 'RiskGameController' sent to 'JFXMainView'.	
Synchronous Call.	[Return is False. Iteration is False. New group is False.]

Save Game diagram

Interaction diagram in package 'Game Stage'

Save Game
Version 1.0

Szabó Dávid created on 2017. 03. 22.. Last modified 2017. 09. 21.

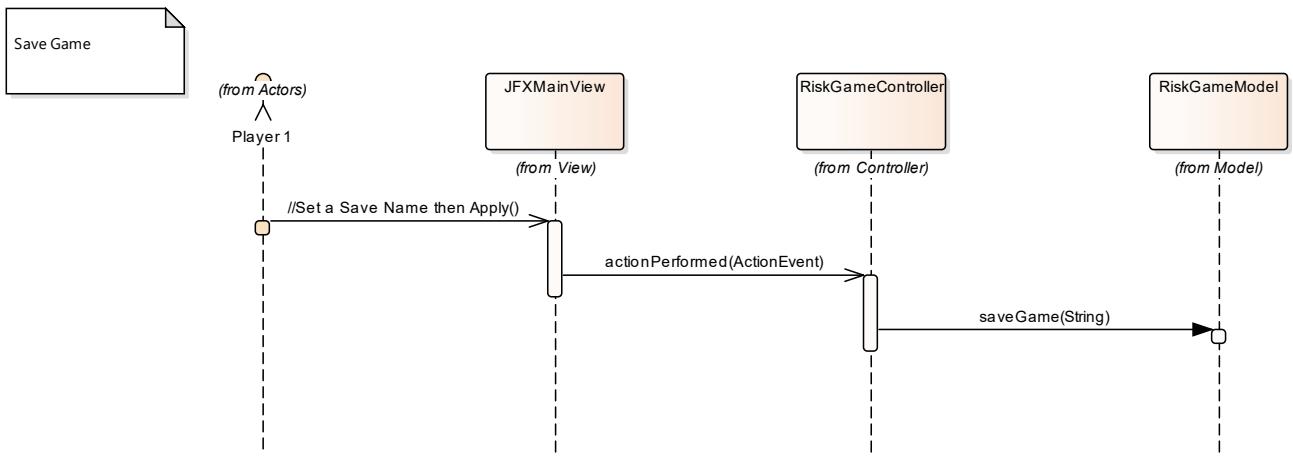


Figure 14: Save Game

Player 1

Actor in package 'Actors'

Player 1

Version 1.0 Phase 1.0 Mandatory

Oliver created on 2017. 02. 21.. Last modified 2017. 03. 23.

OUTGOING BEHAVIORAL RELATIONSHIPS

Name: //Selects 2 Territories

↳ Sequence from Player 1 to JFXMainView

Name: //select attacking units

↳ Sequence from Player 1 to JFXAttackView

↳ Sequence from Player 1 to Reinforcement

↳ Sequence from Player 1 to Load Game

↳ Sequence from Player 1 to Exit Game

↳ Sequence from Player 1 to Save Game

↳ Sequence from Player 1 to Exchange Cards

↳ Sequence from Player 1 to Define the starting player

↳ Sequence from Player 1 to Define number of players

OUTGOING BEHAVIORAL RELATIONSHIPS
Name: // Selects option ↳ Sequence from Player 1 to JFXMainView
Name: // Add number of players ↳ Sequence from Player 1 to JFXAddPlayerView
Name: //Set a Save Name then Apply ↳ Sequence from Player 1 to JFXMainView
Name: // click to begin action ↳ Sequence from Player 1 to JFXMainView
Name: // click to connect ↳ Sequence from Player 1 to JFXMainView
Name: //End Turn ↳ Sequence from Player 1 to JFXMainView
Name: // Selects exit game option ↳ Sequence from Player 1 to JFXMainView
Name: // Select saved game file then apply ↳ Sequence from Player 1 to JFXMainView
↳ Sequence from Player 1 to Main Battle Sequence
Name: //init attack ↳ Sequence from Player 1 to JFXMainView
Name: //end turn ↳ Sequence from Player 1 to JFXMainView
↳ Sequence from Player 1 to Transfer Units
Name: // select Yes or No ↳ Sequence from Player 1 to JFXMainView
Name: //Selects Units then Apply ↳ Sequence from Player 1 to JFXTransferView
↳ Sequence from Player 1 to Preparation stage
INCOMING BEHAVIORAL RELATIONSHIPS

INCOMING BEHAVIORAL RELATIONSHIPS
⇒ Sequence from Transfer Units to Player 1
⇒ Sequence from Save Game to Player 1
⇒ Sequence from Main Battle Sequence to Player 1
⇒ Sequence from Preparation stage to Player 1
⇒ Sequence from Reinforcement to Player 1
⇒ Sequence from Exchange Cards to Player 1
Name: //popup ⇒ Sequence from JFXMainView to Player 1
Name: addTerritory() ⇒ Sequence from RiskGameModel to Player 1
⇒ Sequence from Define number of players to Player 1
⇒ Sequence from Exit Game to Player 1
⇒ Sequence from Define the starting player to Player 1
⇒ Sequence from Load Game to Player 1

CONNECTORS
 UseCaseLink Source -> Destination From: Player 1 : Actor, Public To: Reinforcement : UseCase, Public
 UseCaseLink Source -> Destination From: Player 1 : Actor, Public To: Transfer Units : UseCase, Public
 UseCaseLink Source -> Destination From: Player 1 : Actor, Public To: Main Battle Sequence : UseCase, Public

ASSOCIATIONS

ASSOCIATIONS

 Association (direction: Unspecified)

Source: Public (Actor) Player 1

Target: Public (UseCase) Exit Game

RiskGameModel

Class in package 'Model'

Az MVC architektura Model részét megvalósító osztálya.

RiskGameModel
Version 1.0 Phase 1.0 Proposed
Szabi created on 2017. 02. 21.. Last modified 2017. 11. 22.

OUTGOING STRUCTURAL RELATIONSHIPS

 Aggregation from RiskGameModel to RiskGameController

[Direction is 'Source -> Destination'.]

INCOMING STRUCTURAL RELATIONSHIPS

 Aggregation from Map to RiskGameModel

[Direction is 'Source -> Destination'.]

 Aggregation from Deck to RiskGameModel

[Direction is 'Source -> Destination'.]

ATTRIBUTES

 calculatedCircleNumber : int Private

[Is static False. Containment is Not Specified.]

 capturedThisTurn : Boolean Private

[Is static False. Containment is Not Specified.]

 circlenumber : int Private

[Is static False. Containment is Not Specified.]

 currentPlayer : int Private

[Is static False. Containment is Not Specified.]

 deck : Deck Private

[Is static False. Containment is Not Specified.]

 hasTransferred : boolean Private

[Is static False. Containment is Not Specified.]

ATTRIBUTES	
◆ map : Map	Private [Is static False. Containment is Not Specified.]
◆ miscnumber : int	Private [Is static False. Containment is Not Specified.]
◆ phase : Phase	Private [Is static False. Containment is Not Specified.]
◆ playersList : List<Player>	Private [Is static False. Containment is Not Specified.]
◆ r : Random	Private = new Random() [Is static False. Containment is Not Specified.]
◆ waitForUnitsTemp : Territory	Private = new Territory[2] [Is static False. Containment is Not Specified.]

ASSOCIATIONS	
✓ Association (direction: Source -> Destination)	Source: Public (Class) RiskGameModel Target: Private waitForUnitsTemp (Class) Territory
✓ Association (direction: Source -> Destination)	Source: Public (Class) RiskGameModel Target: Private playersList (Class) Player Cardinality: [0..*]
✓ Association (direction: Source -> Destination)	Source: Public (Class) RiskGameModel Target: Public deck (Class) Player

OPERATIONS	
◆ addPlayer (name : String , color : Color) : boolean	Public Add player with name and color @return boolean [Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ addPlayerToPlayerList (player : Player) : void	Public Adder for playerList Properties: throws = Exception ea_guid = {45684F69-D727-4875-BEF7-5232B4E9D1C9} [Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]

OPERATIONS
<p>◆ addPlayerToPlayerList (map : java.util.Map<Color, String>) : void Public</p> <p>Pairs overload</p> <p>Properties: throws = Exception [Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]</p>
<p>◆ attackTerritoryWithResult (defenderID : int, attackerID : int, defendUnits : int, attackUnits : int) : AttackResult Public</p> <p>Return the result of the attack If the attacker have more than 3 units, the fighting attacker units will reduce to 3 If the defender have more than 2 units, the fighting defender units will reduce to 2</p> <p>Properties: throws = Exception [Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]</p>
<p>◆ calculateCircle () : void Private</p> <p>Properties: throws = Exception [Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]</p>
<p>◆ checkAttackPossible (defender : Territory, attacker : Territory, defendUnits : int, attackUnits : int) : boolean Public</p> <p>Csak akkor tĂĽmadhat, ha a terĂĽleten elegendL' egysĂ©ge van. Igazzal tĂ©r vissza, ha a tamadas lehetsĂ©ges. @return boolean [Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]</p>
<p>◆ checkAttackPossible (defenderID : int, attackerID : int) : boolean Public</p> <p>Attack checking method @return [Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]</p>
<p>◆ checkIfCapturedAndConquer (defender : Territory) : boolean Public</p> <p>Capture and conquer checking method @return boolean</p> <p>Properties: ea_guid = {BFF4F3B6-2538-4778-9372-EAA3B468EDD8} [Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]</p>
<p>◆ checkIfTerritoryIsEmpty (territory : Territory) : boolean Public</p> <p>Empty territory checking method @return boolean</p> <p>Properties: ea_guid = {BE4F4E07-A4C4-4e91-9F00-EF1EEB65D26C} [Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]</p>

OPERATIONS
<p>◆ checkTransferPossible (from : int , to : int) : boolean Public</p> <p>Transfer checking method @return boolean</p> <p>[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]</p>
<p>◆ endBattlePhase () : boolean Public</p> <p>Battle @return boolean</p> <p>Properties: throws = Exception</p> <p>[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]</p>
<p>◆ endTurn () : int Public</p> <p>Kor vege Returns next player's id @return int</p> <p>Properties: throws = Exception ea_guid = {F5F8D5D6-8997-43de-A5AD-DDDC647E775F}</p> <p>[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]</p>
<p>◆ getCurrentPlayer () : int Public</p> <p>Getter for current player @return int</p> <p>[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]</p>
<p>◆ getPhase () : Phase Public</p> <p>Getter for Phase @return phase</p> <p>[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]</p>
<p>◆ getPlayerCards (id : int) : List<Integer> Public</p> <p>[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]</p>
<p>◆ getPlayerName (id : int) : String Public</p> <p>Getter for Player's name @return String</p> <p>[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]</p>
<p>◆ getPlayers () : List<String> Public</p> <p>Getter for Players list @return List<Player></p> <p>[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]</p>

OPERATIONS	
◆ getTerritory (id : int) : Territory Public	
Getter for Territory @return Territory	[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ getUnitsLeftToReinforce (playerID : int) : int Public	
Getter for left units in reinforce phase @return int	[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ loadGame (file : File) : boolean Public	
Load data method @return boolean	
Properties: throws = Exception	[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ moveUnits (units : int) : boolean Public	
Move units method @return boolean	
Properties: throws = Exception	[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ nextPhase () : Phase Private	
Calculate the next phase @return Phase	
Properties: throws = Exception	[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ nextPlayer () : void Protected	
Initializes the next Player's turn. Calculates the amount of Units to place.	
Properties: throws = Exception ea_guid = {806D8A15-D81F-4852-9BFA-E0988D68A2D8}	[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ reinforce (territoryId : int) : boolean Public	
Reinforce method @return boolean	
Properties: throws = Exception	

OPERATIONS	
[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]	
<p>◆ RiskGameModel () : Public</p> <p>The amount of Units left to Place in the Reinforcement Stage of the current Player's current turn.</p>	
[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]	
<p>◆ saveGame (file : File) : boolean Public</p> <p>Save data method</p> <p>@return</p> <p>Properties:</p> <p>throws = IOException</p>	
[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]	
<p>◆ selectStartingPlayer (list : List<Player>) : Player Public</p> <p>Random select the starting player</p> <p>@return Player</p>	
[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]	
<p>◆ transfer (fromID : int , toID : int , units : int) : boolean Public</p> <p>@return boolean</p> <p>Properties:</p> <p>throws = Exception</p>	
[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]	
<p>◆ transfer (from : Territory , to : Territory , units : int) : boolean Public</p> <p>Transfer method</p> <p>@return boolean</p> <p>Properties:</p> <p>throws = Exception</p> <p>ea_guid = {23034260-4F89-40e8-BC43-DD7B2D6023D5}</p>	
[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]	
<p>◆ winGame (winner : int) : void Private</p>	
[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]	

RiskGameController

Class in package 'Controller'

Az MVC architektura Controller részét megvalósító osztálya. A Controller a Modell és a View közötti kapcsolatért felelős. Rajta keresztül hajtják végre a View-on történt változások, melyek a Model-t érintik, illetve a Model a Controller-en keresztül módosítja a View-t.

RiskGameController

INCOMING STRUCTURAL RELATIONSHIPS

- ⇒ Aggregation from JFXAttackView to RiskGameController
[Direction is 'Source -> Destination'.]
- ⇒ Aggregation from JFXAddPlayerView to RiskGameController
[Direction is 'Source -> Destination'.]
- ⇒ Aggregation from JFXTransferView to RiskGameController
[Direction is 'Source -> Destination'.]
- ⇒ Aggregation from RiskGameModel to RiskGameController
[Direction is 'Source -> Destination'.]
- ⇒ Aggregation from JFXMainView to RiskGameController
[Direction is 'Source -> Destination'.]

ATTRIBUTES

- ◆ addPlayerView : JFXAddPlayerView Private
[Is static False. Containment is Not Specified.]
- ◆ attackResultView : JFXAttackResultView Private
[Is static False. Containment is Not Specified.]
- ◆ attackView : JFXAttackView Private
[Is static False. Containment is Not Specified.]
- ◆ cardGroup : Group Private
[Is static False. Containment is Not Specified.]
- ◆ cardScene : Scene Private
[Is static False. Containment is Not Specified.]
- ◆ cardView : JFXCardView Private
CardView stuffs
[Is static False. Containment is Not Specified.]
- ◆ lastPhaseUpdate : Phase Private
[Is static False. Containment is Not Specified.]
- ◆ lastPlayerUpdate : int Private
[Is static False. Containment is Not Specified.]

ATTRIBUTES	
◆ mainView : JFXMainView Private	
View objects	
	[Is static False. Containment is Not Specified.]
◆ model : RiskGameModel Private	
	[Is static False. Containment is Not Specified.]
◆ popupStage : Stage Private	
	[Is static False. Containment is Not Specified.]
◆ preStage : Stage Private Const	
Stages	
	[Is static False. Containment is Not Specified.]
◆ previouslySelectedTerritory : int Private = -1	
	[Is static False. Containment is Not Specified.]
◆ primaryStage : Stage Private	
	[Is static False. Containment is Not Specified.]
◆ titledPaneStage : Stage Private	
	[Is static False. Containment is Not Specified.]
◆ transferView : JFXTransferView Private	
private JFXRiskCardView cardView;	
	[Is static False. Containment is Not Specified.]
◆ victoryStage : Stage Private	
	[Is static False. Containment is Not Specified.]
◆ victoryView : JFXVictoryView Private	
VictoryView stuffs	
	[Is static False. Containment is Not Specified.]

ASSOCIATIONS	
◆ Association (direction: Source -> Destination)	
Source: Public (Class) JFXViewBase	Target: Protected controller (Class) RiskGameController

OPERATIONS	
------------	--

OPERATIONS	
◆ addLog (log : String) : void Public	A főablak naplójába küld egy új bejegyzést [Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ attackAccepted (defender : int, attacker : int, defenderUnits : int, attackerUnits : int) : void Public	[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ attackRetreat () : void Public	[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ attackShowed (attacker : int, defender : int) : void Public	[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ closePopupWindow () : void Private	[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ convertToCountry (ID : int) : Country Private	[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ convertToTerritoryID (country : Country) : int Private	[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ countrySelected (country : Country) : void Public	[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ getPhase () : Phase Public	[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ getPreviouslySelectedTerritory () : int Public	[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ getTerritoryData () : List<Territory> Public	[Is static True. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ isVictory () : boolean Private	[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ loadGame (loadFile : File) : void Public	[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ moveCardView () : void Public	[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]

OPERATIONS	
◆ nextPlayerSelected () : void Public	[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ reinforce (territoryID : int , country : String) : void Private	
Properties:	
throws = Exception	[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ RiskGameController (stage : Stage) : Public	[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ saveGame (saveFile : File) : void Public	[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ setModel (model : RiskGameModel) : void Public	
Eltárolja a modellt és feliratkoztatja magát az eseményeire	[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ showAddPlayerView () : void Public	
Add player view	[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ showAttackResultView () : void Private	[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ showAttackView (defenderID : int , attackerID : int) : void Private	
Attack View	[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ showCardView () : void Public	[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ showMainView () : void Public	
Main View	[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ showTransferView (fromID : int , toID : int) : void Private	
Transfer View	[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ showVictoryView () : void Public	[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]

OPERATIONS	
◆	switchToMain (map : Map<Color, String>) : void Public Stage switch Pre -> Main [Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆	transferAccepted (from : int, to : int, units : int) : void Public [Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆	transferPhaseSelected () : void Public [Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆	UpdateAllTerritoriesOnMap () : void Private [Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆	UpdateCurrentDeck () : void Private [Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆	UpdateCurrentPhase () : void Private [Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆	UpdateCurrentPlayer () : void Private [Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆	UpdateTerritoryOnMap (territoryID : int) : void Private [Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]

JFXMainView

Class in package 'View'

Az MVC architektura View részét megvalósító osztálya. Ez a View a játék főkepernyője, ezen történik maga a játék (egysegek elhelyezése, csata stb.).

JFXMainView
Version 1.0 Phase 1.0 Proposed
Szabó Dávid created on 2017. 03. 01.. Last modified 2017. 11. 22.
Extends JFXViewBase

OUTGOING STRUCTURAL RELATIONSHIPS	
◀	Generalization from JFXMainView to JFXViewBase [Direction is 'Source -> Destination'.]
◀	Aggregation from JFXMainView to RiskGameController [Direction is 'Source -> Destination'.]

ATTRIBUTES
◆ BASE_HEIGHT : double Private Const = 600 [Is static False. Containment is Not Specified.]
◆ BASE_WIDTH : double Private Const = 815 [Is static False. Containment is Not Specified.]
◆ borderPane : BorderPane Private Properties: annotations = @FXML [Is static False. Containment is Not Specified.]
◆ btnBattlePhase : Button Private Properties: annotations = @FXML [Is static False. Containment is Not Specified.]
◆ btnControlList : ArrayList<Button> Private [Is static False. Containment is Not Specified.]
◆ btnLoad : Button Private Properties: annotations = @FXML [Is static False. Containment is Not Specified.]
◆ btnLog : Button Private Properties: annotations = @FXML [Is static False. Containment is Not Specified.]
◆ btnNextPlayer : Button Private Properties: annotations = @FXML [Is static False. Containment is Not Specified.]
◆ btnReinforcePhase : Button Private Properties: annotations = @FXML [Is static False. Containment is Not Specified.]
◆ btnSave : Button Private Properties: annotations = @FXML [Is static False. Containment is Not Specified.]

ATTRIBUTES
<p>◆ btnSettings : Button Private</p> <p>Properties:</p> <p> annotations = @FXML</p> <p>[Is static False. Containment is Not Specified.]</p>
<p>◆ btnTransferPhase : Button Private</p> <p>Properties:</p> <p> annotations = @FXML</p> <p>[Is static False. Containment is Not Specified.]</p>
<p>◆ centerStackPane : StackPane Private</p> <p>Properties:</p> <p> annotations = @FXML</p> <p>[Is static False. Containment is Not Specified.]</p>
<p>◆ dimLog : Dimension Package</p> <p>[Is static False. Containment is Not Specified.]</p>
<p>◆ imageView : ImageView Private</p> <p>Properties:</p> <p> annotations = @FXML</p> <p>[Is static False. Containment is Not Specified.]</p>
<p>◆ isLogOpened : boolean Package</p> <p>[Is static False. Containment is Not Specified.]</p>
<p>◆ isSLOpened : boolean Package</p> <p>[Is static False. Containment is Not Specified.]</p>
<p>◆ lblPlayerName1 : Label Private</p> <p>Properties:</p> <p> annotations = @FXML</p> <p>[Is static False. Containment is Not Specified.]</p>
<p>◆ lblPlayerName2 : Label Private</p> <p>Properties:</p> <p> annotations = @FXML</p> <p>[Is static False. Containment is Not Specified.]</p>
<p>◆ lblPlayerName3 : Label Private</p> <p>Properties:</p> <p> annotations = @FXML</p>

ATTRIBUTES	
	[Is static False. Containment is Not Specified.]
◆ lblPlayerName4 : Label Private	
Properties: annotations = @FXML	[Is static False. Containment is Not Specified.]
◆ lblPlayerName5 : Label Private	
Properties: annotations = @FXML	[Is static False. Containment is Not Specified.]
◆ lblPlayerName6 : Label Private	
Properties: annotations = @FXML	[Is static False. Containment is Not Specified.]
◆ lblPlayerNameList : ArrayList<Label> Private	[Is static False. Containment is Not Specified.]
◆ pain : Pane Private	
Properties: annotations = @FXML	[Is static False. Containment is Not Specified.]
◆ panePlayerColor1 : Pane Private	
Properties: annotations = @FXML	[Is static False. Containment is Not Specified.]
◆ panePlayerColor2 : Pane Private	
Properties: annotations = @FXML	[Is static False. Containment is Not Specified.]
◆ panePlayerColor3 : Pane Private	
Properties: annotations = @FXML	[Is static False. Containment is Not Specified.]
◆ panePlayerColor4 : Pane Private	
Properties: annotations = @FXML	[Is static False. Containment is Not Specified.]

ATTRIBUTES	
◆ panePlayerColor5 : Pane Private	
Properties:	
annotations = @FXML	[Is static False. Containment is Not Specified.]
◆ panePlayerColor6 : Pane Private	
Properties:	
annotations = @FXML	[Is static False. Containment is Not Specified.]
◆ panePlayerColorList : ArrayList<Pane> Private	[Is static False. Containment is Not Specified.]
◆ stage : Stage Private	[Is static False. Containment is Not Specified.]
◆ taLog : TextArea Private	
Properties:	
annotations = @FXML	[Is static False. Containment is Not Specified.]
◆ world : World Private	[Is static False. Containment is Not Specified.]

OPERATIONS	
◆ AddControllerListener (controller : RiskGameController) : void Public	
Properties:	
annotations = @Override	[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ appendLog (s : String) : void Public	[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ closeLog () : void Public	[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ finalize () : void Public	
Properties:	
throws = Throwable	
annotations = @Override	[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]

OPERATIONS	
◆ fit (height : double , width : double) : void	Public [Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ getRoot () : Parent	Public Properties: annotations = @Override [Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ getWorld () : World	Public [Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ handleLogButton (event : ActionEvent) : void	Private Properties: annotations = @FXML [Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ hangleLoadButton (event : ActionEvent) : void	Private Properties: annotations = @FXML [Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ hangleSaveButton (event : ActionEvent) : void	Private Properties: annotations = @FXML [Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ hangleSettingsButton (event : ActionEvent) : void	Private Properties: annotations = @FXML [Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ initialize (location : URL , resources : ResourceBundle) : void	Public Properties: annotations = @Override [Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ isLogOpened () : boolean	Public [Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ isSLOpened () : boolean	Public [Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ JFXMainView (stage : Stage) : Public	[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]

OPERATIONS	
◆ openLog () : void Public	[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ UpdateConnectedPlayers (playerList : List<String>) : void Public	[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ UpdateCurrentDeck (cards : List<Integer>) : void Public	
TODO:	[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ UpdateCurrentPhase (phase : Phase) : void Public	[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ UpdateCurrentPlayer (playerID : int) : void Public	
@param player	[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ updateIcons (attacker : int, defender : int) : void Public	[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ UpdateViewState (territory : Territory) : void Public	
@param territories	
Properties:	
ea_guid = {0E811D72-EC4E-47ad-8B46-C39084EAD8F7}	[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ UpdateViewState (territories : List<Territory>) : void Public	[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]

<anonymous>

Note in package 'Game Stage'

Save Game

<anonymous>
Version 1.0 Phase 1.0 Proposed
Oliver created on 2017. 03. 28.. Last modified 2017. 03. 28.
Extends

INTERACTION MESSAGES

✉ 1.0 //Set a Save Name then Apply' from 'Player 1' sent to 'JFXMainView'.

Asynchronous Call. Returns void.

[Return is False. Iteration is False. New group is False.]

✉ 1.1 'actionPerformed(ActionEvent)' from 'JFXMainView' sent to 'RiskGameController'.

Asynchronous Call.

[Return is False. Iteration is False. New group is False.]

✉ 1.2 'saveGame(String)' from 'RiskGameController' sent to 'RiskGameModel'.

Synchronous Call.

[Return is False. Iteration is False. New group is False.]

Transfer Units diagram

Interaction diagram in package 'Game Stage'

Transfer Units
Version 1.0

Szabó Dávid created on 2017. 03. 22.. Last modified 2017. 09. 21.

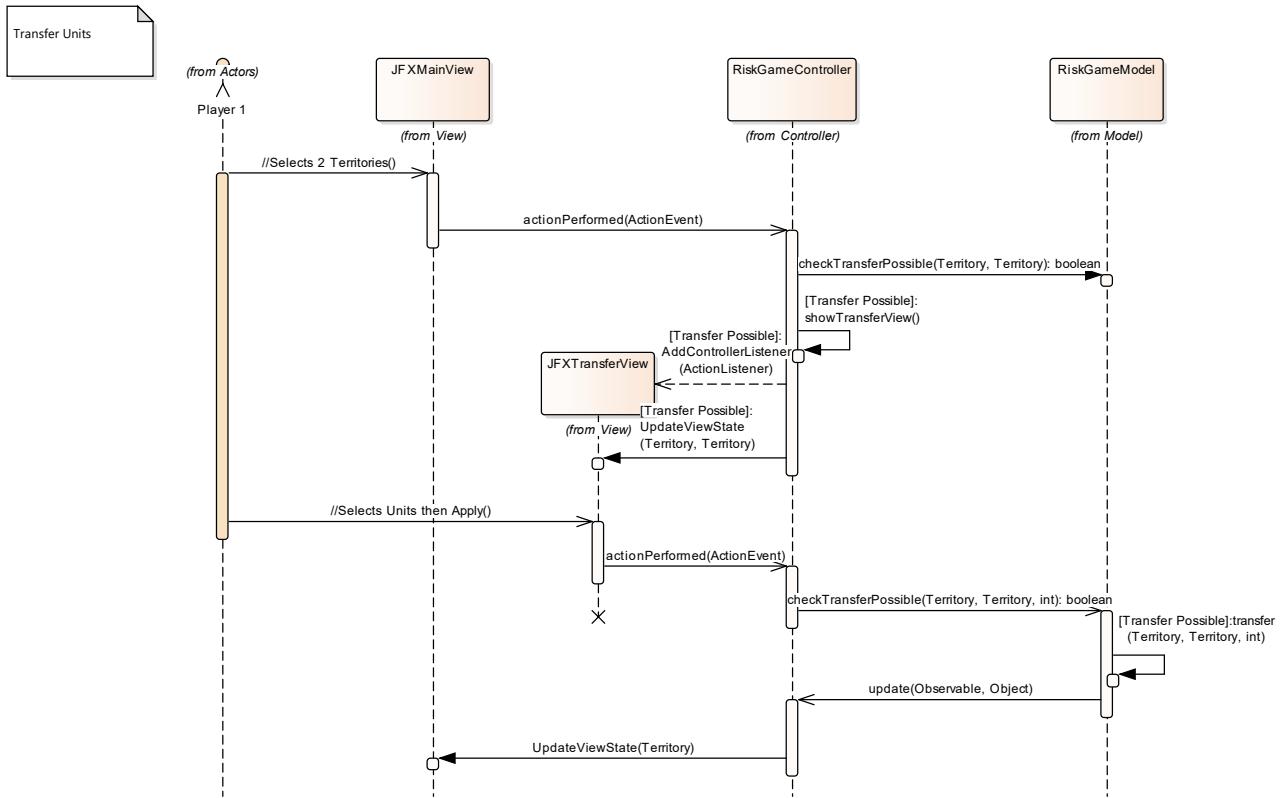


Figure 15: Transfer Units

Player 1

Actor in package 'Actors'

Player 1

Version 1.0 Phase 1.0 Mandatory

Oliver created on 2017. 02. 21.. Last modified 2017. 03. 23.

OUTGOING BEHAVIORAL RELATIONSHIPS

Name: //Selects 2 Territories

↳ Sequence from Player 1 to JFXMainView

Name: //select attacking units

↳ Sequence from Player 1 to JFXAttackView

↳ Sequence from Player 1 to Reinforcement

↳ Sequence from Player 1 to Load Game

↳ Sequence from Player 1 to Exit Game

↳ Sequence from Player 1 to Save Game

↳ Sequence from Player 1 to Exchange Cards

↳ Sequence from Player 1 to Define the starting player

↳ Sequence from Player 1 to Define number of players

Name: // Selects option

↳ Sequence from Player 1 to JFXMainView

Name: // Add number of players

↳ Sequence from Player 1 to JFXAddPlayerView

Name: //Set a Save Name then Apply

↳ Sequence from Player 1 to JFXMainView

Name: // click to begin action

↳ Sequence from Player 1 to JFXMainView

Name: // click to connect

↳ Sequence from Player 1 to JFXMainView

Name: //End Turn

↳ Sequence from Player 1 to JFXMainView

Name: // Selects exit game option

OUTGOING BEHAVIORAL RELATIONSHIPS	
↳ Sequence from Player 1 to JFXMainView	
Name: // Select saved game file then apply	
↳ Sequence from Player 1 to JFXMainView	
↳ Sequence from Player 1 to Main Battle Sequence	
Name: //init attack	
↳ Sequence from Player 1 to JFXMainView	
Name: //end turn	
↳ Sequence from Player 1 to JFXMainView	
↳ Sequence from Player 1 to Transfer Units	
Name: // select Yes or No	
↳ Sequence from Player 1 to JFXMainView	
Name: //Selects Units then Apply	
↳ Sequence from Player 1 to JFXTransferView	
↳ Sequence from Player 1 to Preparation stage	
INCOMING BEHAVIORAL RELATIONSHIPS	
↗ Sequence from Transfer Units to Player 1	
↗ Sequence from Save Game to Player 1	
↗ Sequence from Main Battle Sequence to Player 1	
↗ Sequence from Preparation stage to Player 1	
↗ Sequence from Reinforcement to Player 1	
↗ Sequence from Exchange Cards to Player 1	
Name: // popup	
↗ Sequence from JFXMainView to Player 1	
Name: addTerritory()	
↗ Sequence from RiskGameModel to Player 1	

INCOMING BEHAVIORAL RELATIONSHIPS

Sequence from Define number of players to Player 1

Sequence from Exit Game to Player 1

Sequence from Define the starting player to Player 1

Sequence from Load Game to Player 1

CONNECTORS

UseCaseLink Source -> Destination

From: Player 1 : Actor, Public
To: Reinforcement : UseCase, Public

UseCaseLink Source -> Destination

From: Player 1 : Actor, Public
To: Transfer Units : UseCase, Public

UseCaseLink Source -> Destination

From: Player 1 : Actor, Public
To: Main Battle Sequence : UseCase, Public

ASSOCIATIONS

Association (direction: Unspecified)

Source: Public (Actor) Player 1

Target: Public (UseCase) Exit Game

RiskGameModel

Class in package 'Model'

Az MVC architektura Model részét megvalósító osztálya.

RiskGameModel
Version 1.0 Phase 1.0 Proposed
Szabi created on 2017. 02. 21.. Last modified 2017. 11. 22.

OUTGOING STRUCTURAL RELATIONSHIPS

Aggregation from RiskGameModel to RiskGameController

[Direction is 'Source -> Destination'.]

INCOMING STRUCTURAL RELATIONSHIPS

INCOMING STRUCTURAL RELATIONSHIPS	
⇒ Aggregation from Map to RiskGameModel	[Direction is 'Source -> Destination'.]
⇒ Aggregation from Deck to RiskGameModel	[Direction is 'Source -> Destination'.]
ATTRIBUTES	
◆ calculatedCircleNumber : int Private	[Is static False. Containment is Not Specified.]
◆ capturedThisTurn : Boolean Private	[Is static False. Containment is Not Specified.]
◆ circlenumber : int Private	[Is static False. Containment is Not Specified.]
◆ currentPlayer : int Private	[Is static False. Containment is Not Specified.]
◆ deck : Deck Private	[Is static False. Containment is Not Specified.]
◆ hasTransferred : boolean Private	[Is static False. Containment is Not Specified.]
◆ map : Map Private	[Is static False. Containment is Not Specified.]
◆ miscnumber : int Private	[Is static False. Containment is Not Specified.]
◆ phase : Phase Private	[Is static False. Containment is Not Specified.]
◆ playersList : List<Player> Private	[Is static False. Containment is Not Specified.]
◆ r : Random Private = new Random()	[Is static False. Containment is Not Specified.]
◆ waitForUnitsTemp : Territory Private = new Territory[2]	[Is static False. Containment is Not Specified.]

ASSOCIATIONS	
 Association (direction: Source -> Destination)	
Source: Public (Class) RiskGameModel	Target: Private waitForUnitsTemp (Class) Territory
 Association (direction: Source -> Destination)	
Source: Public (Class) RiskGameModel	Target: Private playersList (Class) Player Cardinality: [0..*]
 Association (direction: Source -> Destination)	
Source: Public (Class) RiskGameModel	Target: Public deck (Class) Player

OPERATIONS	
 addPlayer (name : String , color : Color) : boolean Public	
Add player with name and color @return boolean	[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
 addPlayerToPlayerList (player : Player) : void Public	
Adder for playerList	
Properties: throws = Exception ea_guid = {45684F69-D727-4875-BEF7-5232B4E9D1C9}	[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
 addPlayerToPlayerList (map : java.util.Map<Color, String>) : void Public	
Pairs overload	
Properties: throws = Exception	[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
 attackTerritoryWithResult (defenderID : int , attackerID : int , defendUnits : int , attackUnits : int) : AttackResult Public	
Return the result of the attack If the attacker have more than 3 units, the fighting attacker units will reduce to 3 If the defender have more than 2 units, the fighting defender units will reduce to 2	
Properties: throws = Exception	[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
 calculateCircle () : void Private	
Properties: throws = Exception	[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]

OPERATIONS
<p>◆ checkAttackPossible (defender : Territory , attacker : Territory , defendUnits : int , attackUnits : int) : boolean Public Csak akkor tĂˇmadhat, ha a terĂĽleten elegendL‘ egysĂ©ge van. Igazzal tĂ©r vissza, ha a tamadas lehetsĂ©ges. <code>@return boolean</code> [Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]</p>
<p>◆ checkAttackPossible (defenderID : int , attackerID : int) : boolean Public Attack checking method <code>@return</code> [Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]</p>
<p>◆ checkIfCapturedAndConquer (defender : Territory) : boolean Public Capture and conquer checking method <code>@return boolean</code> Properties: <code>ea_guid = {BFF4F3B6-2538-4778-9372-EAA3B468EDD8}</code> [Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]</p>
<p>◆ checkIfTerrotiryIsEmpty (territory : Territory) : boolean Public Empty territory checking method <code>@return boolean</code> Properties: <code>ea_guid = {BE4F4E07-A4C4-4e91-9F00-EF1EEB65D26C}</code> [Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]</p>
<p>◆ checkTransferPossible (from : int , to : int) : boolean Public Transfer checking method <code>@return boolean</code> [Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]</p>
<p>◆ endBattlePhase () : boolean Public Battle <code>@return boolean</code> Properties: <code>throws = Exception</code> [Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]</p>
<p>◆ endTurn () : int Public Kor vege Returns next player's id <code>@return int</code> Properties: <code>throws = Exception</code> <code>ea_guid = {F5F8D5D6-8997-43de-A5AD-DDDC647E775F}</code></p>

OPERATIONS	
[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]	
<p>◆ getCurrentPlayer () : int Public</p> <p>Getter for current player @return int</p> <p>[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]</p>	
<p>◆ getPhase () : Phase Public</p> <p>Getter for Phase @return phase</p> <p>[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]</p>	
<p>◆ getPlayerCards (id : int) : List<Integer> Public</p> <p>[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]</p>	
<p>◆ getPlayerName (id : int) : String Public</p> <p>Getter for Player's name @return String</p> <p>[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]</p>	
<p>◆ getPlayers () : List<String> Public</p> <p>Getter for Players list @return List<Player></p> <p>[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]</p>	
<p>◆ getTerritory (id : int) : Territory Public</p> <p>Getter for Territory @return Territory</p> <p>[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]</p>	
<p>◆ getUnitsLeftToReinforce (playerID : int) : int Public</p> <p>Getter for left units in reinforce phase @return int</p> <p>[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]</p>	
<p>◆ loadGame (file : File) : boolean Public</p> <p>Load data method @return boolean</p> <p>Properties: throws = Exception</p> <p>[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]</p>	
<p>◆ moveUnits (units : int) : boolean Public</p>	

OPERATIONS	
Move units method @return boolean	Properties: throws = Exception [Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ nextPhase () : Phase Private	Calculate the next phase @return Phase
Properties: throws = Exception	[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ nextPlayer () : void Protected	Initializes the next Player's turn. Calculates the amount of Units to place.
Properties: throws = Exception ea_guid = {806D8A15-D81F-4852-9BFA-E0988D68A2D8}	[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ reinforce (territoryId : int) : boolean Public	Reinforce method @return boolean
Properties: throws = Exception	[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ RiskGameModel () : Public	The amount of Units left to Place in the Reinforcement Stage of the current Player's current turn. [Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ saveGame (file : File) : boolean Public	Save data method @return
Properties: throws = IOException	[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ selectStartingPlayer (list : List<Player>) : Player Public	Random select the starting player @return Player
	[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]

OPERATIONS

◆ transfer (fromID : int , toID : int , units : int) : boolean Public

@return boolean

Properties:

throws = Exception

[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]

◆ transfer (from : Territory , to : Territory , units : int) : boolean Public

Transfer method

@return boolean

Properties:

throws = Exception

ea_guid = {23034260-4F89-40e8-BC43-DD7B2D6023D5}

[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]

◆ winGame (winner : int) : void Private

[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]

RiskGameController

Class in package 'Controller'

Az MVC architektura Controller részét megvalósító osztálya. A Controller a Modell és a View közötti kapcsolatért felelős. Rajta keresztül hajtják végre a View-on történt változások, melyek a Model-t érintik, illetve a Model a Controller-en keresztül módosítja a View-t.

RiskGameController

Version 1.0 Phase 1.0 Proposed

Szabó Dávid created on 2017. 03. 01.. Last modified 2017. 11. 22.

Extends java.util.Observable

INCOMING STRUCTURAL RELATIONSHIPS

⇒ Aggregation from JFXAttackView to RiskGameController

[Direction is 'Source -> Destination'.]

⇒ Aggregation from JFXAddPlayerView to RiskGameController

[Direction is 'Source -> Destination'.]

⇒ Aggregation from JFXTransferView to RiskGameController

[Direction is 'Source -> Destination'.]

⇒ Aggregation from RiskGameModel to RiskGameController

[Direction is 'Source -> Destination'.]

⇒ Aggregation from JFXMainView to RiskGameController

[Direction is 'Source -> Destination'.]

ATTRIBUTES	
◆ addPlayerView : JFXAddPlayerView	Private [Is static False. Containment is Not Specified.]
◆ attackResultView : JFXAttackResultView	Private [Is static False. Containment is Not Specified.]
◆ attackView : JFXAttackView	Private [Is static False. Containment is Not Specified.]
◆ cardGroup : Group	Private [Is static False. Containment is Not Specified.]
◆ cardScene : Scene	Private [Is static False. Containment is Not Specified.]
◆ cardView : JFXCardView	Private CardView stuffs [Is static False. Containment is Not Specified.]
◆ lastPhaseUpdate : Phase	Private [Is static False. Containment is Not Specified.]
◆ lastPlayerUpdate : int	Private [Is static False. Containment is Not Specified.]
◆ mainView : JFXMainView	Private View objects [Is static False. Containment is Not Specified.]
◆ model : RiskGameModel	Private [Is static False. Containment is Not Specified.]
◆ popupStage : Stage	Private [Is static False. Containment is Not Specified.]
◆ preStage : Stage	Private Const Stages [Is static False. Containment is Not Specified.]
◆ previouslySelectedTerritory : int	Private = -1 [Is static False. Containment is Not Specified.]

ATTRIBUTES	
◆ primaryStage : Stage	Private [Is static False. Containment is Not Specified.]
◆ titledPaneStage : Stage	Private [Is static False. Containment is Not Specified.]
◆ transferView : JFXTransferView	Private private JFXRiskCardView cardView; [Is static False. Containment is Not Specified.]
◆ victoryStage : Stage	Private [Is static False. Containment is Not Specified.]
◆ victoryView : JFXVictoryView	Private VictoryView stuffs [Is static False. Containment is Not Specified.]

ASSOCIATIONS	
✍ Association (direction: Source -> Destination)	Source: Public (Class) JFXViewBase Target: Protected controller (Class) RiskGameController

OPERATIONS	
◆ addLog (log : String) : void	Public A főablak naplójába küld egy új bejegyzést [Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ attackAccepted (defender : int, attacker : int, defenderUnits : int, attackerUnits : int) : void	Public [Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ attackRetreat () : void	Public [Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ attackShowed (attacker : int, defender : int) : void	Public [Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ closePopupWindow () : void	Private [Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ convertToCountry (ID : int) : Country	Private [Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]

OPERATIONS	
◆ convertToTerritoryID (country : Country) : int	Private [Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ countrySelected (country : Country) : void	Public [Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ getPhase () : Phase	Public [Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ getPreviouslySelectedTerritory () : int	Public [Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ getTerritoryData () : List<Territory>	Public [Is static True. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ isVictory () : boolean	Private [Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ loadGame (loadFile : File) : void	Public [Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ moveCardView () : void	Public [Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ nextPlayerSelected () : void	Public [Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ reinforce (territoryID : int, country : String) : void	Private Properties: throws = Exception [Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ RiskGameController (stage : Stage) : void	Public [Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ saveGame (saveFile : File) : void	Public [Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ setModel (model : RiskGameModel) : void	Public Eltárolja a modellt és feliratkoztatja magát az eseményre [Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ showAddPlayerView () : void	Public

OPERATIONS	
Add player view	[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◊ showAttackResultView () : void Private	[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◊ showAttackView (defenderID : int , attackerID : int) : void Private	
Attack View	[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◊ showCardView () : void Public	[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◊ showMainView () : void Public	
Main View	[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◊ showTransferView (fromID : int , toID : int) : void Private	
Transfer View	[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◊ showVictoryView () : void Public	[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◊ switchToMain (map : Map<Color, String>) : void Public	
Stage switch Pre -> Main	[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◊ transferAccepted (from : int , to : int , units : int) : void Public	[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◊ transferPhaseSelected () : void Public	[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◊ UpdateAllTerritoriesOnMap () : void Private	[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◊ UpdateCurrentDeck () : void Private	[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◊ UpdateCurrentPhase () : void Private	[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]

OPERATIONS

- ◆ UpdateCurrentPlayer () : void Private
[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
- ◆ UpdateTerritoryOnMap (territoryID : int) : void Private
[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]

JFXMainView*Class in package 'View'*

Az MVC architektura View részet megvalósító osztálya. Ez a View a játék főkepernyője, ezen történik maga a játék (egyések elhelyezése, csata stb.).

JFXMainView
Version 1.0 Phase 1.0 Proposed
Szabó Dávid created on 2017. 03. 01.. Last modified 2017. 11. 22.
Extends JFXViewBase

OUTGOING STRUCTURAL RELATIONSHIPS

- ↳ Generalization from JFXMainView to JFXViewBase
[Direction is 'Source -> Destination'.]
- ↳ Aggregation from JFXMainView to RiskGameController
[Direction is 'Source -> Destination'.]

ATTRIBUTES

- ◆ BASE_HEIGHT : double Private Const = 600
[Is static False. Containment is Not Specified.]
- ◆ BASE_WIDTH : double Private Const = 815
[Is static False. Containment is Not Specified.]
- ◆ borderPane : BorderPane Private

Properties:
annotations = @FXML
[Is static False. Containment is Not Specified.]
- ◆ btnBattlePhase : Button Private

Properties:
annotations = @FXML
[Is static False. Containment is Not Specified.]
- ◆ btnControlList : ArrayList<Button> Private

[Is static False. Containment is Not Specified.]

ATTRIBUTES
◆ btnLoad : Button Private Properties: annotations = @FXML [Is static False. Containment is Not Specified.]
◆ btnLog : Button Private Properties: annotations = @FXML [Is static False. Containment is Not Specified.]
◆ btnNextPlayer : Button Private Properties: annotations = @FXML [Is static False. Containment is Not Specified.]
◆ btnReinforcePhase : Button Private Properties: annotations = @FXML [Is static False. Containment is Not Specified.]
◆ btnSave : Button Private Properties: annotations = @FXML [Is static False. Containment is Not Specified.]
◆ btnSettings : Button Private Properties: annotations = @FXML [Is static False. Containment is Not Specified.]
◆ btnTransferPhase : Button Private Properties: annotations = @FXML [Is static False. Containment is Not Specified.]
◆ centerStackPane : StackPane Private Properties: annotations = @FXML [Is static False. Containment is Not Specified.]
◆ dimLog : Dimension Package [Is static False. Containment is Not Specified.]

ATTRIBUTES	
◆ imageView : ImageView	Private Properties: annotations = @FXML [Is static False. Containment is Not Specified.]
◆ isLogOpened : boolean	Package [Is static False. Containment is Not Specified.]
◆ isSLOpened : boolean	Package [Is static False. Containment is Not Specified.]
◆ lblPlayerName1 : Label	Private Properties: annotations = @FXML [Is static False. Containment is Not Specified.]
◆ lblPlayerName2 : Label	Private Properties: annotations = @FXML [Is static False. Containment is Not Specified.]
◆ lblPlayerName3 : Label	Private Properties: annotations = @FXML [Is static False. Containment is Not Specified.]
◆ lblPlayerName4 : Label	Private Properties: annotations = @FXML [Is static False. Containment is Not Specified.]
◆ lblPlayerName5 : Label	Private Properties: annotations = @FXML [Is static False. Containment is Not Specified.]
◆ lblPlayerName6 : Label	Private Properties: annotations = @FXML [Is static False. Containment is Not Specified.]
◆ lblPlayerNameList : ArrayList<Label>	Private [Is static False. Containment is Not Specified.]

ATTRIBUTES
◆ pain : Pane Private Properties: annotations = @FXML [Is static False. Containment is Not Specified.]
◆ panePlayerColor1 : Pane Private Properties: annotations = @FXML [Is static False. Containment is Not Specified.]
◆ panePlayerColor2 : Pane Private Properties: annotations = @FXML [Is static False. Containment is Not Specified.]
◆ panePlayerColor3 : Pane Private Properties: annotations = @FXML [Is static False. Containment is Not Specified.]
◆ panePlayerColor4 : Pane Private Properties: annotations = @FXML [Is static False. Containment is Not Specified.]
◆ panePlayerColor5 : Pane Private Properties: annotations = @FXML [Is static False. Containment is Not Specified.]
◆ panePlayerColor6 : Pane Private Properties: annotations = @FXML [Is static False. Containment is Not Specified.]
◆ panePlayerColorList : ArrayList<Pane> Private [Is static False. Containment is Not Specified.]
◆ stage : Stage Private [Is static False. Containment is Not Specified.]
◆ taLog : TextArea Private Properties:

ATTRIBUTES	
annotations = @FXML	[Is static False. Containment is Not Specified.]
世界 : World Private	[Is static False. Containment is Not Specified.]
OPERATIONS	
◆ AddControllerListener (controller : RiskGameController) : void Public	
Properties:	
annotations = @Override	[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ appendLog (s : String) : void Public	
	[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ closeLog () : void Public	
	[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ finalize () : void Public	
Properties:	
throws = Throwable	
annotations = @Override	[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ fit (height : double, width : double) : void Public	
	[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ getRoot () : Parent Public	
Properties:	
annotations = @Override	[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ getWorld () : World Public	
	[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ handleLogButton (event : ActionEvent) : void Private	
Properties:	
annotations = @FXML	[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ handleLoadButton (event : ActionEvent) : void Private	
Properties:	
annotations = @FXML	[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]

OPERATIONS
<p>◆ handleSaveButton (event : ActionEvent) : void Private</p> <p>Properties:</p> <pre>annotations = @FXML [Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]</pre>
<p>◆ handleSettingsButton (event : ActionEvent) : void Private</p> <p>Properties:</p> <pre>annotations = @FXML [Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]</pre>
<p>◆ initialize (location : URL, resources : ResourceBundle) : void Public</p> <p>Properties:</p> <pre>annotations = @Override [Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]</pre>
<p>◆ isLogOpened () : boolean Public</p> <pre>[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]</pre>
<p>◆ isSLOpened () : boolean Public</p> <pre>[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]</pre>
<p>◆ JFXMainView (stage : Stage) : Public</p> <pre>[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]</pre>
<p>◆ openLog () : void Public</p> <pre>[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]</pre>
<p>◆ UpdateConnectedPlayers (playerList : List<String>) : void Public</p> <pre>[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]</pre>
<p>◆ UpdateCurrentDeck (cards : List<Integer>) : void Public</p> <p>TODO:</p> <pre>[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]</pre>
<p>◆ UpdateCurrentPhase (phase : Phase) : void Public</p> <pre>[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]</pre>
<p>◆ UpdateCurrentPlayer (playerID : int) : void Public</p> <p>@param player</p> <pre>[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]</pre>
<p>◆ updateIcons (attacker : int, defender : int) : void Public</p>

OPERATIONS	
[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]	
◆ UpdateViewState (territory : Territory) : void	Public
@param territories	
Properties:	
ea_guid = {0E811D72-EC4E-47ad-8B46-C39084EAD8F7}	[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ UpdateViewState (territories : List<Territory>) : void	Public
	[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]

JFXTransferView

Class in package 'View'

Ezen a View-en valaszthatja ki a jatekos, hogy a sajat teruleten belül hova helyezi az egységeit.

JFXTransferView
Version 1.0 Phase 1.0 Proposed
Szabó Dávid created on 2017. 03. 08.. Last modified 2017. 11. 22.
Extends JFXViewBase

OUTGOING STRUCTURAL RELATIONSHIPS	
↳ Generalization from JFXTransferView to JFXViewBase	[Direction is 'Source -> Destination'.]
↳ Aggregation from JFXTransferView to RiskGameController	[Direction is 'Source -> Destination'.]

ATTRIBUTES	
◆ btnAccept : Button	Private
Properties:	
annotations = @FXML	[Is static False. Containment is Not Specified.]
◆ fromID : int	Private
	[Is static False. Containment is Not Specified.]
◆ lblFromCountryName : Label	Private
Properties:	
annotations = @FXML	[Is static False. Containment is Not Specified.]
◆ lblFromQuantity : Label	Private

ATTRIBUTES	
Properties: annotations = @FXML	[Is static False. Containment is Not Specified.]
◆ lblPlayerName : Label Private	
Properties: annotations = @FXML	[Is static False. Containment is Not Specified.]
◆ lblSelectedQuantity : Label Private	
Properties: annotations = @FXML	[Is static False. Containment is Not Specified.]
◆ lblToCountryName : Label Private	
Properties: annotations = @FXML	[Is static False. Containment is Not Specified.]
◆ lblToQuantity : Label Private	
Properties: annotations = @FXML	[Is static False. Containment is Not Specified.]
◆ sldSoilderQuantity : Slider Private	
Properties: annotations = @FXML	[Is static False. Containment is Not Specified.]
◆ toID : int Private	[Is static False. Containment is Not Specified.]
OPERATIONS	
◆ finalize () : void Public	
Properties: annotations = @Override throws = Throwable	[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ initialize (location : URL , resources : ResourceBundle) : void Public	
Properties: annotations = @Override	[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]

OPERATIONS
<p>◆ JFXTransferView () : Public [Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]</p>
<p>◆ UpdateViewState (player : String , from : Territory , to : Territory) : void Public [Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]</p>

<anonymous>

Note in package 'Game Stage'

Transfer Units

<anonymous>
Version 1.0 Phase 1.0 Proposed
Oliver created on 2017. 03. 28.. Last modified 2017. 03. 28.
Extends

INTERACTION MESSAGES
<p>✉ 1.0 '/*Selects 2 Territories' from 'Player 1' sent to 'JFXMainView'. Asynchronous Call. [Return is False. Iteration is False. New group is False.]</p>
<p>✉ 1.1 'actionPerformed(ActionEvent)' from 'JFXMainView' sent to 'RiskGameController'. Asynchronous Call. [Return is False. Iteration is False. New group is False.]</p>
<p>✉ 1.2 'checkTransferPossible(Territory, Territory)' from 'RiskGameController' sent to 'RiskGameModel'. Synchronous Call. Returns boolean. [Return is False. Iteration is False. New group is False.]</p>
<p>✉ 1.3 'showTransferView()' from 'RiskGameController' sent to 'RiskGameController'. When "Transfer Possible". Synchronous Call. [Return is False. Iteration is False. New group is False.]</p>
<p>✉ 1.4 'AddControllerListener(ActionListener)' from 'RiskGameController' sent to 'JFXTransferView'. When "Transfer Possible". Synchronous Call. [Return is False. Iteration is False. New group is False. Life cycle is New.]</p>
<p>✉ 1.5 'UpdateViewState(Territory, Territory)' from 'RiskGameController' sent to 'JFXTransferView'. When "Transfer Possible". Synchronous Call.</p>

	[Return is False. Iteration is False. New group is False. Life cycle is <none>.]
✉ 1.6 '//Selects Units then Apply' from 'Player 1' sent to 'JFXTransferView'.	
Asynchronous Call. Returns void.	[Return is False. Iteration is False. New group is False.]
✉ 1.7 'actionPerformed(ActionEvent)' from 'JFXTransferView' sent to 'RiskGameController'.	
Asynchronous Call.	[Return is False. Iteration is False. New group is False.]
✉ 1.8 'checkTransferPossible(Territory, Territory, int)' from 'RiskGameController' sent to 'RiskGameModel'.	
Asynchronous Call. Returns boolean.	[Return is False. Iteration is False. New group is False. Life cycle is <none>.]
✉ 1.9 'transfer(Territory, Territory, int)' from 'RiskGameModel' sent to 'RiskGameModel'.	
When "Transfer Possible". Synchronous Call.	[Return is False. Iteration is False. New group is False.]
✉ 1.10 'update(Observable, Object)' from 'RiskGameModel' sent to 'RiskGameController'.	
Asynchronous Call.	[Return is False. Iteration is False. New group is False.]
✉ 1.11 'UpdateViewState(Territory)' from 'RiskGameController' sent to 'JFXMainView'.	
Synchronous Call.	[Return is False. Iteration is False. New group is False.]

<anonymous>

Object in package 'Game Stage'

<anonymous>
Version 1.0 Phase 1.0 Proposed
Szabi created on 2017. 03. 07.. Last modified 2017. 03. 28.

<anonymous>

Object in package 'Game Stage'

<anonymous>
Version 1.0 Phase 1.0 Proposed
Szabi created on 2017. 03. 07.. Last modified 2017. 03. 28.

<anonymous>

Object in package 'Game Stage'

<anonymous>

Version 1.0 Phase 1.0 Proposed
Szabi created on 2017. 03. 07.. Last modified 2017. 03. 28.

Player1

Object in package 'Game Stage'

Player1

Version 1.0 Phase 1.0 Proposed
Szabi created on 2017. 03. 29.. Last modified 2017. 03. 29.

attacker

Object in package 'Game Stage'

attacker

Version 1.0 Phase 1.0 Proposed
Szabi created on 2017. 03. 07.. Last modified 2017. 03. 28.

attacker

Object in package 'Game Stage'

attacker

Version 1.0 Phase 1.0 Proposed
Szabi created on 2017. 03. 07.. Last modified 2017. 03. 28.

defender

Object in package 'Game Stage'

defender

Version 1.0 Phase 1.0 Proposed
Szabi created on 2017. 03. 07.. Last modified 2017. 03. 28.

defender

Object in package 'Game Stage'

defender

Version 1.0 Phase 1.0 Proposed
Szabi created on 2017. 03. 07.. Last modified 2017. 03. 28.

Captured

InteractionFragment in package 'Game Stage'

Captured

Version 1.0 Phase 1.0 Proposed

Card loop

InteractionFragment in package 'Game Stage'

Card loop

Version 1.0 Phase 1.0 Proposed

Oliver created on 2017. 03. 28.. Last modified 2017. 03. 28.

Close application

MessageEnd in package 'Game Stage'

Close application

Version 1.0 Phase 1.0 Proposed

fodorad created on 2017. 03. 23.. Last modified 2017. 03. 28.

Close application

MessageEnd in package 'Game Stage'

Close application

Version 1.0 Phase 1.0 Proposed

fodorad created on 2017. 03. 23.. Last modified 2017. 03. 28.

Close application

MessageEnd in package 'Game Stage'

Close application

Version 1.0 Phase 1.0 Proposed

fodorad created on 2017. 03. 23.. Last modified 2017. 03. 28.

Save game?

InteractionFragment in package 'Game Stage'

Save game?

Version 1.0 Phase 1.0 Proposed

fodorad created on 2017. 03. 23.. Last modified 2017. 03. 28.

canAttack

InteractionFragment in package 'Game Stage'

canAttack

Version 1.0 Phase 1.0 Proposed

Szabi created on 2017. 04. 04.. Last modified 2017. 04. 04.

placingTheUnitsLoop

InteractionFragment in package 'Game Stage'

placingTheUnitsLoop

Version 1.0 Phase 1.0 Proposed

Szabó Dávid created on 2017. 03. 29.. Last modified 2017. 03. 30.

Preparation Stage

Package in package 'Dynamic View'

Preparation Stage
Version 1.0 Phase 1.0 Proposed
Oliver created on 2017. 03. 22.. Last modified 2017. 03. 28.

Define the starting player diagram

Interaction diagram in package 'Preparation Stage'

Define the starting player
Version 1.0
Oliver created on 2017. 03. 22.. Last modified 2017. 04. 19.

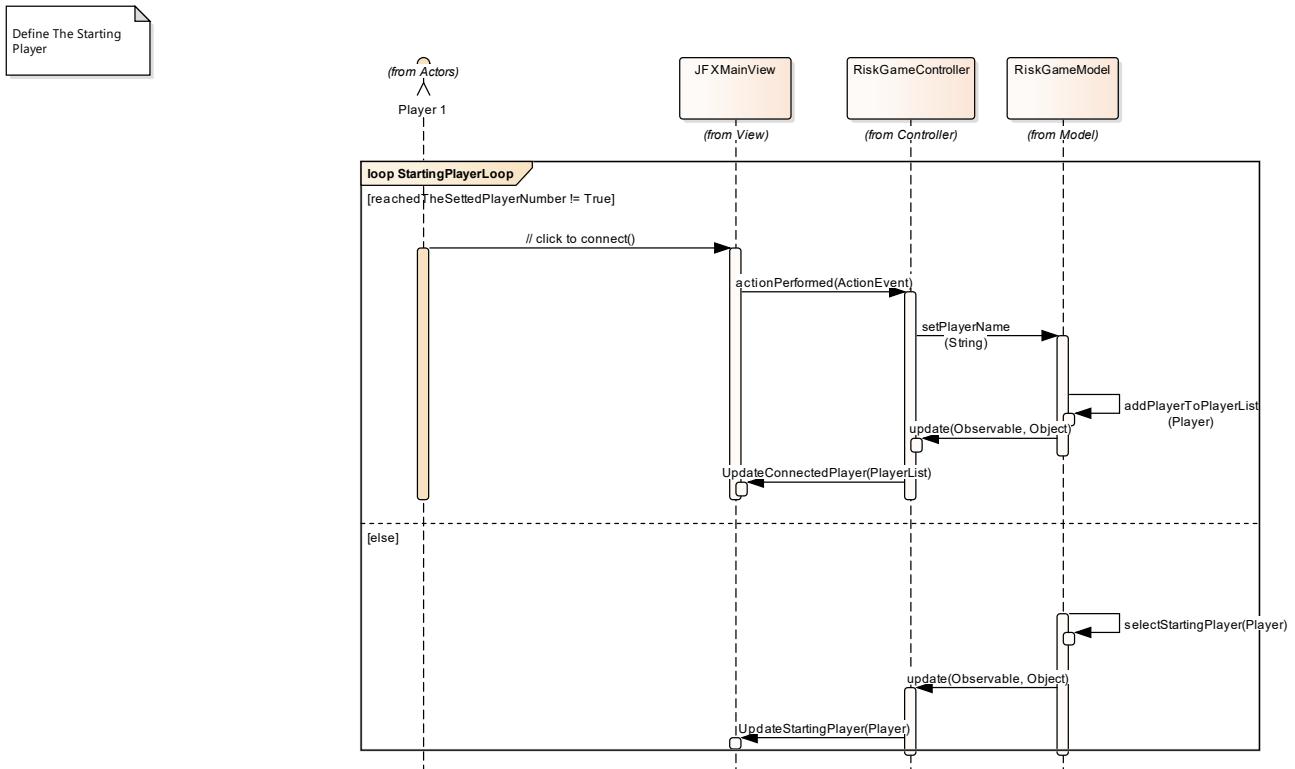


Figure 16: Define the starting player

Player 1

Actor in package 'Actors'

Player 1
Version 1.0 Phase 1.0 Mandatory
Oliver created on 2017. 02. 21.. Last modified 2017. 03. 23.

OUTGOING BEHAVIORAL RELATIONSHIPS
Name: //Selects 2 Territories Sequence from Player 1 to JFXMainView
Name: //select attacking units Sequence from Player 1 to JFXAttackView

OUTGOING BEHAVIORAL RELATIONSHIPS
↳ Sequence from Player 1 to Reinforcement
↳ Sequence from Player 1 to Load Game
↳ Sequence from Player 1 to Exit Game
↳ Sequence from Player 1 to Save Game
↳ Sequence from Player 1 to Exchange Cards
↳ Sequence from Player 1 to Define the starting player
↳ Sequence from Player 1 to Define number of players
Name: // Selects option ↳ Sequence from Player 1 to JFXMainView
Name: // Add number of players ↳ Sequence from Player 1 to JFXAddPlayerView
Name: //Set a Save Name then Apply ↳ Sequence from Player 1 to JFXMainView
Name: // click to begin action ↳ Sequence from Player 1 to JFXMainView
Name: // click to connect ↳ Sequence from Player 1 to JFXMainView
Name: //End Turn ↳ Sequence from Player 1 to JFXMainView
Name: // Selects exit game option ↳ Sequence from Player 1 to JFXMainView
Name: // Select saved game file then apply ↳ Sequence from Player 1 to JFXMainView
↳ Sequence from Player 1 to Main Battle Sequence
Name: //init attack

OUTGOING BEHAVIORAL RELATIONSHIPS	
 Sequence from Player 1 to JFXMainView	
Name: //end turn	
 Sequence from Player 1 to JFXMainView	
 Sequence from Player 1 to Transfer Units	
Name: // select Yes or No	
 Sequence from Player 1 to JFXMainView	
Name: //Selects Units then Apply	
 Sequence from Player 1 to JFXTransferView	
 Sequence from Player 1 to Preparation stage	
INCOMING BEHAVIORAL RELATIONSHIPS	
 Sequence from Transfer Units to Player 1	
 Sequence from Save Game to Player 1	
 Sequence from Main Battle Sequence to Player 1	
 Sequence from Preparation stage to Player 1	
 Sequence from Reinforcement to Player 1	
 Sequence from Exchange Cards to Player 1	
Name: // popup	
 Sequence from JFXMainView to Player 1	
Name: addTerritory()	
 Sequence from RiskGameModel to Player 1	
 Sequence from Define number of players to Player 1	
 Sequence from Exit Game to Player 1	
 Sequence from Define the starting player to Player 1	

INCOMING BEHAVIORAL RELATIONSHIPS

➡ Sequence from Load Game to Player 1

CONNECTORS

↗ **UseCaseLink** Source -> Destination

From: Player 1 : Actor, Public
To: Reinforcement : UseCase, Public

↗ **UseCaseLink** Source -> Destination

From: Player 1 : Actor, Public
To: Transfer Units : UseCase, Public

↗ **UseCaseLink** Source -> Destination

From: Player 1 : Actor, Public
To: Main Battle Sequence : UseCase, Public

ASSOCIATIONS

↙ Association (direction: Unspecified)

Source: Public (Actor) Player 1

Target: Public (UseCase) Exit Game

RiskGameModel

Class in package 'Model'

Az MVC architektura Model részét megvalósító osztalya.

RiskGameModel
Version 1.0 Phase 1.0 Proposed
Szabi created on 2017. 02. 21.. Last modified 2017. 11. 22.

OUTGOING STRUCTURAL RELATIONSHIPS

⬅ Aggregation from RiskGameModel to RiskGameController

[Direction is 'Source -> Destination'.]

INCOMING STRUCTURAL RELATIONSHIPS

➡ Aggregation from Map to RiskGameModel

[Direction is 'Source -> Destination'.]

➡ Aggregation from Deck to RiskGameModel

[Direction is 'Source -> Destination'.]

ATTRIBUTES

◆ calculatedCircleNumber : int Private

ATTRIBUTES	
	[Is static False. Containment is Not Specified.]
◆ capturedThisTurn : Boolean Private	[Is static False. Containment is Not Specified.]
◆ circlenumber : int Private	[Is static False. Containment is Not Specified.]
◆ currentPlayer : int Private	[Is static False. Containment is Not Specified.]
◆ deck : Deck Private	[Is static False. Containment is Not Specified.]
◆ hasTransferred : boolean Private	[Is static False. Containment is Not Specified.]
◆ map : Map Private	[Is static False. Containment is Not Specified.]
◆ misnumber : int Private	[Is static False. Containment is Not Specified.]
◆ phase : Phase Private	[Is static False. Containment is Not Specified.]
◆ playersList : List<Player> Private	[Is static False. Containment is Not Specified.]
◆ r : Random Private = new Random()	[Is static False. Containment is Not Specified.]
◆ waitForUnitsTemp : Territory Private = new Territory[2]	[Is static False. Containment is Not Specified.]

ASSOCIATIONS	
✓ Association (direction: Source -> Destination)	
Source: Public (Class) RiskGameModel	Target: Private waitForUnitsTemp (Class) Territory
✓ Association (direction: Source -> Destination)	
Source: Public (Class) RiskGameModel	Target: Private playersList (Class) Player Cardinality: [0..*]

ASSOCIATIONS

 Association (direction: Source -> Destination)

Source: Public (Class) RiskGameModel

Target: Public deck (Class) Player

OPERATIONS

 addPlayer (name : String , color : Color) : boolean Public

Add player with name and color

@return boolean

[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]

 addPlayerToPlayerList (player : Player) : void Public

Adder for playerList

Properties:

throws = Exception

ea_guid = {45684F69-D727-4875-BEF7-5232B4E9D1C9}

[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]

 addPlayerToPlayerList (map : java.util.Map<Color, String>) : void Public

Pairs overload

Properties:

throws = Exception

[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]

 attackTerritoryWithResult (defenderID : int , attackerID : int , defendUnits : int , attackUnits : int) : AttackResult Public

Return the result of the attack If the attacker have more than 3 units, the fighting attacker units will reduce to 3 If the defender have more than 2 units, the fighting defender units will reduce to 2

Properties:

throws = Exception

[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]

 calculateCircle () : void Private

Properties:

throws = Exception

[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]

 checkAttackPossible (defender : Territory , attacker : Territory , defendUnits : int , attackUnits : int) : boolean Public

Csak akkor tĂˇmadhat, ha a terĂĽleten elegendL' egysĂ©ge van. Igazzal tĂ©r vissza, ha a tamadas lehetsĂ©ges.
@return boolean

[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]

 checkAttackPossible (defenderID : int , attackerID : int) : boolean Public

OPERATIONS	
Attack checking method	
@return	[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ checkIfCapturedAndConquer (defender : Territory) : boolean Public	
Capture and conquer checking method	
@return boolean	
Properties:	
ea_guid = {BFF4F3B6-2538-4778-9372-EAA3B468EDD8}	
[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]	
◆ checkIfTerritoryIsEmpty (territory : Territory) : boolean Public	
Empty territory checking method	
@return boolean	
Properties:	
ea_guid = {BE4F4E07-A4C4-4e91-9F00-EF1EEB65D26C}	
[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]	
◆ checkTransferPossible (from : int, to : int) : boolean Public	
Transfer checking method	
@return boolean	[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ endBattlePhase () : boolean Public	
Battle	
@return boolean	
Properties:	
throws = Exception	
[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]	
◆ endTurn () : int Public	
Kor vege Returns next player's id	
@return int	
Properties:	
throws = Exception	
ea_guid = {F5F8D5D6-8997-43de-A5AD-DDDC647E775F}	
[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]	
◆ getCurrentPlayer () : int Public	
Getter for current player	
@return int	[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ getPhase () : Phase Public	

OPERATIONS	
Getter for Phase @return phase	[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◊ getPlayerCards (id : int) : List<Integer> Public [Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]	
◊ getPlayerName (id : int) : String Public Getter for Player's name @return String	[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◊ getPlayers () : List<String> Public Getter for Players list @return List<Player>	[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◊ getTerritory (id : int) : Territory Public Getter for Territory @return Territory	[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◊ getUnitsLeftToReinforce (playerID : int) : int Public Getter for left units in reinforce phase @return int	[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◊ loadGame (file : File) : boolean Public Load data method @return boolean Properties: throws = Exception	[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◊ moveUnits (units : int) : boolean Public Move units method @return boolean Properties: throws = Exception	[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◊ nextPhase () : Phase Private Calculate the next phase	

OPERATIONS	
@return Phase	
Properties: throws = Exception	[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ nextPlayer () : void Protected	Initializes the next Player's turn. Calculates the amount of Units to place.
Properties: throws = Exception ea_guid = {806D8A15-D81F-4852-9BFA-E0988D68A2D8}	[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ reinforce (territoryId : int) : boolean Public	Reinforce method
@return boolean	
Properties: throws = Exception	[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ RiskGameModel () : Public	The amount of Units left to Place in the Reinforcement Stage of the current Player's current turn.
	[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ saveGame (file : File) : boolean Public	Save data method
@return	
Properties: throws = IOException	[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ selectStartingPlayer (list : List<Player>) : Player Public	Random select the starting player
@return Player	[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ transfer (fromID : int , toID : int , units : int) : boolean Public	
@return boolean	
Properties: throws = Exception	[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ transfer (from : Territory , to : Territory , units : int) : boolean Public	

OPERATIONS	
Transfer method	
@return boolean	
Properties:	
throws = Exception	
ea_guid = {23034260-4F89-40e8-BC43-DD7B2D6023D5}	[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]	
◆ winGame (winner : int) : void Private	
[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]	

RiskGameController

Class in package 'Controller'

Az MVC architektura Controller részet megvalósító osztálya. A Controller a Modell és a View kozotti kapcsolatért felelős. Rajta keresztül hajtoddnak végre a View-on történt változások, melyek a Model-t érintik, illetve a Model a Controller-en keresztül módosítja a View-t.

RiskGameController
Version 1.0 Phase 1.0 Proposed
Szabó Dávid created on 2017. 03. 01.. Last modified 2017. 11. 22.
Extends java.util.Observable

INCOMING STRUCTURAL RELATIONSHIPS	
⇒ Aggregation from JFXAttackView to RiskGameController	[Direction is 'Source -> Destination'.]
⇒ Aggregation from JFXAddPlayerView to RiskGameController	[Direction is 'Source -> Destination'.]
⇒ Aggregation from JFXTransferView to RiskGameController	[Direction is 'Source -> Destination'.]
⇒ Aggregation from RiskGameModel to RiskGameController	[Direction is 'Source -> Destination'.]
⇒ Aggregation from JFXMainView to RiskGameController	[Direction is 'Source -> Destination'.]

ATTRIBUTES	
◆ addPlayerView : JFXAddPlayerView Private	[Is static False. Containment is Not Specified.]
◆ attackResultView : JFXAttackResultView Private	[Is static False. Containment is Not Specified.]

ATTRIBUTES	
◆ attackView : JFXAttackView	Private [Is static False. Containment is Not Specified.]
◆ cardGroup : Group	Private [Is static False. Containment is Not Specified.]
◆ cardScene : Scene	Private [Is static False. Containment is Not Specified.]
◆ cardView : JFXCardView	Private
CardView stuffs	[Is static False. Containment is Not Specified.]
◆ lastPhaseUpdate : Phase	Private [Is static False. Containment is Not Specified.]
◆ lastPlayerUpdate : int	Private [Is static False. Containment is Not Specified.]
◆ mainView : JFXMainView	Private
View objects	[Is static False. Containment is Not Specified.]
◆ model : RiskGameModel	Private [Is static False. Containment is Not Specified.]
◆ popupStage : Stage	Private [Is static False. Containment is Not Specified.]
◆ preStage : Stage	Private Const [Is static False. Containment is Not Specified.]
◆ previouslySelectedTerritory : int	Private = -1 [Is static False. Containment is Not Specified.]
◆ primaryStage : Stage	Private [Is static False. Containment is Not Specified.]
◆ titledPaneStage : Stage	Private [Is static False. Containment is Not Specified.]
◆ transferView : JFXTransferView	Private

ATTRIBUTES	
private JFXRiskCardView cardView;	[Is static False. Containment is Not Specified.]
◆ victoryStage : Stage Private	[Is static False. Containment is Not Specified.]
◆ victoryView : JFXVictoryView Private	
VictoryView stuffs	[Is static False. Containment is Not Specified.]
ASSOCIATIONS	
◆ Association (direction: Source -> Destination)	
Source: Public (Class) JFXViewBase	Target: Protected controller (Class) RiskGameController
OPERATIONS	
◆ addLog (log : String) : void Public	
A főablak naplójába küld egy új bejegyzést	[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ attackAccepted (defender : int, attacker : int, defenderUnits : int, attackerUnits : int) : void Public	[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ attackRetreat () : void Public	[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ attackShowed (attacker : int, defender : int) : void Public	[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ closePopupWindow () : void Private	[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ convertToCountry (ID : int) : Country Private	[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ convertToTerritoryID (country : Country) : int Private	[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ countrySelected (country : Country) : void Public	[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]

OPERATIONS	
◆ getPhase () : Phase	Public [Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ getPreviouslySelectedTerritory () : int	Public [Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ getTerritoryData () : List<Territory>	Public [Is static True. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ isVictory () : boolean	Private [Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ loadGame (loadFile : File) : void	Public [Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ moveCardView () : void	Public [Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ nextPlayerSelected () : void	Public [Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ reinforce (territoryID : int , country : String) : void	Private Properties: throws = Exception [Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ RiskGameController (stage : Stage) : void	Public [Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ saveGame (saveFile : File) : void	Public [Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ setModel (model : RiskGameModel) : void	Public Eltárolja a modellt és feliratkoztatja magát az eseményeire [Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ showAddPlayerView () : void	Public Add player view [Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ showAttackResultView () : void	Private [Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ showAttackView (defenderID : int , attackerID : int) : void	Private [Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]

OPERATIONS	
Attack View	[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◊ showCardView () : void Public	[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
Main View	[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◊ showMainView () : void Public	
Transfer View	[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◊ showTransferView (fromID : int , toID : int) : void Private	
Stage switch Pre -> Main	[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◊ switchToMain (map : Map<Color, String>) : void Public	
Transfer	[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◊ transferAccepted (from : int , to : int , units : int) : void Public	[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◊ transferPhaseSelected () : void Public	[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◊ UpdateAllTerritoriesOnMap () : void Private	[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◊ UpdateCurrentDeck () : void Private	[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◊ UpdateCurrentPhase () : void Private	[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◊ UpdateCurrentPlayer () : void Private	[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◊ UpdateTerritoryOnMap (territoryID : int) : void Private	[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]

JFXMainView

Class in package 'View'

Az MVC architektura View részet megvalósító osztálya. Ez a View a játék főkepernyője, ezen történik maga a játék (egyégek elhelyezése, csata stb.).

JFXMainView
Version 1.0 Phase 1.0 Proposed
Szabó Dávid created on 2017. 03. 01.. Last modified 2017. 11. 22.
Extends JFXViewBase

OUTGOING STRUCTURAL RELATIONSHIPS

↳ Generalization from JFXMainView to JFXViewBase

[Direction is 'Source -> Destination'.]

↳ Aggregation from JFXMainView to RiskGameController

[Direction is 'Source -> Destination'.]

ATTRIBUTES

◆ BASE_HEIGHT : double Private Const = 600

[Is static False. Containment is Not Specified.]

◆ BASE_WIDTH : double Private Const = 815

[Is static False. Containment is Not Specified.]

◆ borderPane : BorderPane Private

Properties:

annotations = @FXML

[Is static False. Containment is Not Specified.]

◆ btnBattlePhase : Button Private

Properties:

annotations = @FXML

[Is static False. Containment is Not Specified.]

◆ btnControlList : ArrayList<Button> Private

[Is static False. Containment is Not Specified.]

◆ btnLoad : Button Private

Properties:

annotations = @FXML

[Is static False. Containment is Not Specified.]

◆ btnLog : Button Private

Properties:

ATTRIBUTES	
annotations = @FXML	[Is static False. Containment is Not Specified.]
◆ btnNextPlayer : Button Private	
Properties:	
annotations = @FXML	[Is static False. Containment is Not Specified.]
◆ btnReinforcePhase : Button Private	
Properties:	
annotations = @FXML	[Is static False. Containment is Not Specified.]
◆ btnSave : Button Private	
Properties:	
annotations = @FXML	[Is static False. Containment is Not Specified.]
◆ btnSettings : Button Private	
Properties:	
annotations = @FXML	[Is static False. Containment is Not Specified.]
◆ btnTransferPhase : Button Private	
Properties:	
annotations = @FXML	[Is static False. Containment is Not Specified.]
◆ centerStackPane : StackPane Private	
Properties:	
annotations = @FXML	[Is static False. Containment is Not Specified.]
◆ dimLog : Dimension Package	[Is static False. Containment is Not Specified.]
◆ imageView : ImageView Private	
Properties:	
annotations = @FXML	[Is static False. Containment is Not Specified.]
◆ isLogOpened : boolean Package	[Is static False. Containment is Not Specified.]

ATTRIBUTES	
◆ isSLOpened : boolean	Package [Is static False. Containment is Not Specified.]
◆ lblPlayerName1 : Label	Private Properties: annotations = @FXML [Is static False. Containment is Not Specified.]
◆ lblPlayerName2 : Label	Private Properties: annotations = @FXML [Is static False. Containment is Not Specified.]
◆ lblPlayerName3 : Label	Private Properties: annotations = @FXML [Is static False. Containment is Not Specified.]
◆ lblPlayerName4 : Label	Private Properties: annotations = @FXML [Is static False. Containment is Not Specified.]
◆ lblPlayerName5 : Label	Private Properties: annotations = @FXML [Is static False. Containment is Not Specified.]
◆ lblPlayerName6 : Label	Private Properties: annotations = @FXML [Is static False. Containment is Not Specified.]
◆ lblPlayerNameList : ArrayList<Label>	Private [Is static False. Containment is Not Specified.]
◆ pain : Pane	Private Properties: annotations = @FXML [Is static False. Containment is Not Specified.]
◆ panePlayerColor1 : Pane	Private Properties: annotations = @FXML

ATTRIBUTES	
	[Is static False. Containment is Not Specified.]
◆ panePlayerColor2 : Pane Private	
Properties: annotations = @FXML	[Is static False. Containment is Not Specified.]
◆ panePlayerColor3 : Pane Private	
Properties: annotations = @FXML	[Is static False. Containment is Not Specified.]
◆ panePlayerColor4 : Pane Private	
Properties: annotations = @FXML	[Is static False. Containment is Not Specified.]
◆ panePlayerColor5 : Pane Private	
Properties: annotations = @FXML	[Is static False. Containment is Not Specified.]
◆ panePlayerColor6 : Pane Private	
Properties: annotations = @FXML	[Is static False. Containment is Not Specified.]
◆ panePlayerColorList : ArrayList<Pane> Private	[Is static False. Containment is Not Specified.]
◆ stage : Stage Private	[Is static False. Containment is Not Specified.]
◆ taLog : TextArea Private	
Properties: annotations = @FXML	[Is static False. Containment is Not Specified.]
◆ world : World Private	[Is static False. Containment is Not Specified.]

OPERATIONS

◆ AddControllerListener (controller : RiskGameController) : void Public

OPERATIONS
<p>Properties:</p> <p>annotations = @Override [Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]</p>
<p>◆ appendLog (s : String) : void Public [Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]</p>
<p>◆ closeLog () : void Public [Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]</p>
<p>◆ finalize () : void Public</p>
<p>Properties:</p> <p>throws = Throwable annotations = @Override [Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]</p>
<p>◆ fit (height : double, width : double) : void Public [Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]</p>
<p>◆ getRoot () : Parent Public</p>
<p>Properties:</p> <p>annotations = @Override [Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]</p>
<p>◆ getWorld () : World Public [Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]</p>
<p>◆ handleLogButton (event : ActionEvent) : void Private</p>
<p>Properties:</p> <p>annotations = @FXML [Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]</p>
<p>◆ hangleLoadButton (event : ActionEvent) : void Private</p>
<p>Properties:</p> <p>annotations = @FXML [Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]</p>
<p>◆ hangleSaveButton (event : ActionEvent) : void Private</p>
<p>Properties:</p> <p>annotations = @FXML [Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]</p>
<p>◆ hangleSettingsButton (event : ActionEvent) : void Private</p>

OPERATIONS	
Properties:	
annotations = @FXML	[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
initialize (location : URL , resources : ResourceBundle) : void Public	
Properties:	
annotations = @Override	[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
isLogOpened () : boolean Public	[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
isSLOpened () : boolean Public	[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
JFXMainView (stage : Stage) : Public	[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
openLog () : void Public	[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
UpdateConnectedPlayers (playerList : List<String>) : void Public	[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
UpdateCurrentDeck (cards : List<Integer>) : void Public	
TODO:	[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
UpdateCurrentPhase (phase : Phase) : void Public	[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
UpdateCurrentPlayer (playerID : int) : void Public	
@param player	[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
updateIcons (attacker : int , defender : int) : void Public	[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
UpdateViewState (territory : Territory) : void Public	
@param territories	
Properties:	
ea_guid = {0E811D72-EC4E-47ad-8B46-C39084EAD8F7}	[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]

OPERATIONS

- ❖ UpdateViewState (territories : List<Territory>) : void Public
[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]

<anonymous>

Note in package 'Preparation Stage'

Define The Starting Player

<anonymous>

Version 1.0 Phase 1.0 Proposed

Oliver created on 2017. 03. 28.. Last modified 2017. 03. 28.

Extends

INTERACTION MESSAGES

- ✉ 1.0 '// click to connect' from 'Player 1' sent to 'JFXMainView'.

Synchronous Call. Returns void.

[Return is False. Iteration is False. New group is False.]

- ✉ 1.1 'actionPerformed(ActionEvent)' from 'JFXMainView' sent to 'RiskGameController'.

Synchronous Call.

[Return is False. Iteration is False. New group is False.]

- ✉ 1.2 'setPlayerName(String)' from 'RiskGameController' sent to 'RiskGameModel'.

Synchronous Call.

[Return is False. Iteration is False. New group is False.]

- ✉ 1.3 'addPlayerToPlayerList(Player)' from 'RiskGameModel' sent to 'RiskGameModel'.

Synchronous Call. Returns void.

[Return is False. Iteration is False. New group is False.]

- ✉ 1.4 'update(Observable, Object)' from 'RiskGameModel' sent to 'RiskGameController'.

Synchronous Call.

[Return is False. Iteration is False. New group is False.]

- ✉ 1.5 'UpdateConnectedPlayer(PlayerList)' from 'RiskGameController' sent to 'JFXMainView'.

Synchronous Call.

[Return is False. Iteration is False. New group is False.]

- ✉ 1.6 'selectStartingPlayer(Player)' from 'RiskGameModel' sent to 'RiskGameModel'.

Synchronous Call.	[Return is False. Iteration is False. New group is False.]
✉ 1.7 'update(Observable, Object)' from 'RiskGameModel' sent to 'RiskGameController'.	
Synchronous Call.	[Return is False. Iteration is False. New group is False.]

✉ 1.8 'UpdateStartingPlayer(Player)' from 'RiskGameController' sent to 'JFXMainView'.	
Synchronous Call.	[Return is False. Iteration is False. New group is False.]

Preparation stage diagram

Interaction diagram in package 'Preparation Stage'

Preparation stage
Version 1.0

fodorad created on 2017. 03. 22.. Last modified 2017. 04. 19.

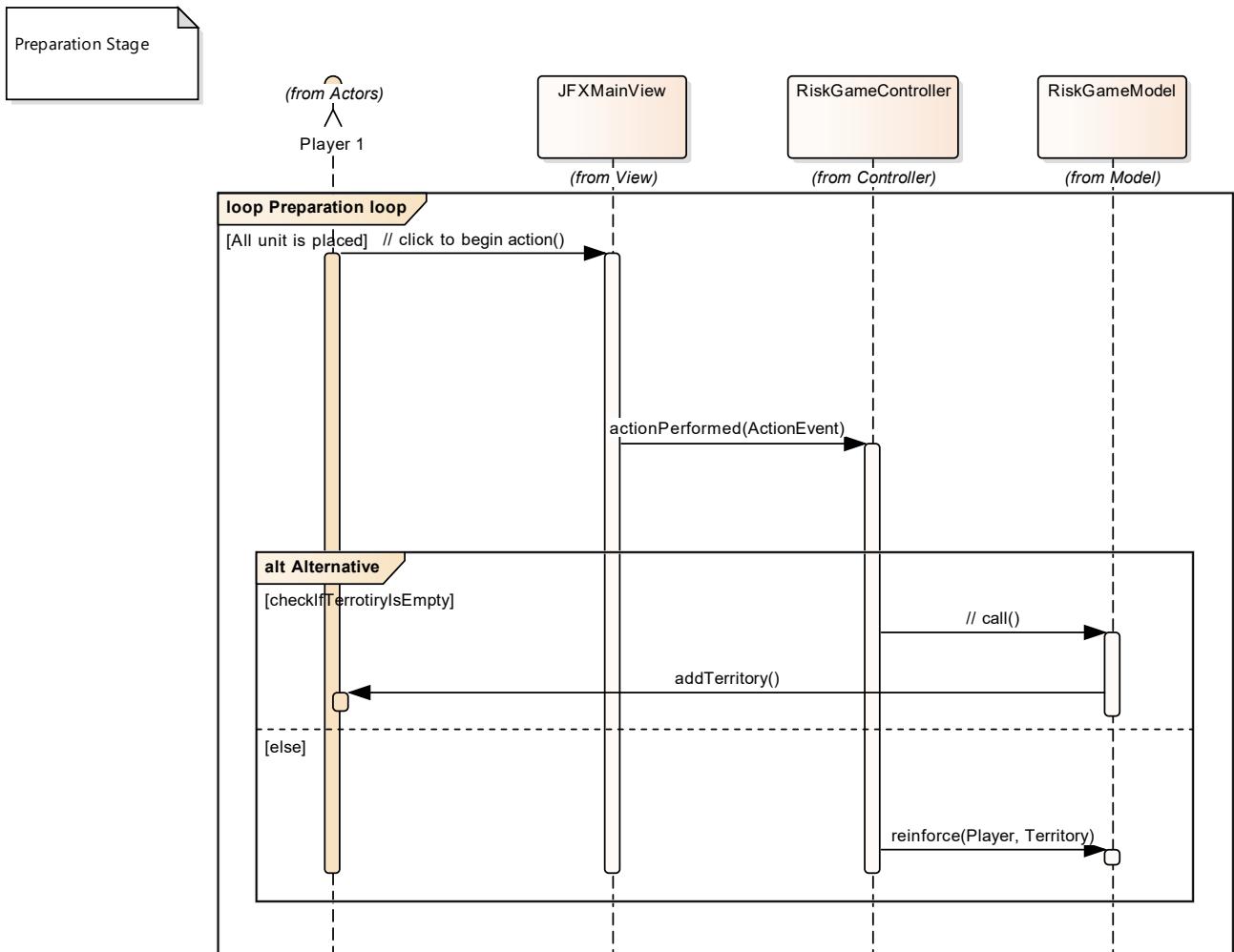


Figure 17: Preparation stage

Player 1

Actor in package 'Actors'

Player 1
Version 1.0 Phase 1.0 Mandatory
Oliver created on 2017. 02. 21.. Last modified 2017. 03. 23.

OUTGOING BEHAVIORAL RELATIONSHIPS
Name: //Selects 2 Territories ⌚ Sequence from Player 1 to JFXMainView
Name: //select attacking units ⌚ Sequence from Player 1 to JFXAttackView
⌚ Sequence from Player 1 to Reinforcement
⌚ Sequence from Player 1 to Load Game
⌚ Sequence from Player 1 to Exit Game
⌚ Sequence from Player 1 to Save Game
⌚ Sequence from Player 1 to Exchange Cards
⌚ Sequence from Player 1 to Define the starting player
⌚ Sequence from Player 1 to Define number of players
Name: // Selects option ⌚ Sequence from Player 1 to JFXMainView
Name: // Add number of players ⌚ Sequence from Player 1 to JFXAddPlayerView
Name: //Set a Save Name then Apply ⌚ Sequence from Player 1 to JFXMainView
Name: // click to begin action ⌚ Sequence from Player 1 to JFXMainView
Name: // click to connect ⌚ Sequence from Player 1 to JFXMainView
Name: //End Turn

OUTGOING BEHAVIORAL RELATIONSHIPS	
↳ Sequence from Player 1 to JFXMainView	
Name: // Selects exit game option	
↳ Sequence from Player 1 to JFXMainView	
Name: // Select saved game file then apply	
↳ Sequence from Player 1 to JFXMainView	
↳ Sequence from Player 1 to Main Battle Sequence	
Name: //init attack	
↳ Sequence from Player 1 to JFXMainView	
Name: //end turn	
↳ Sequence from Player 1 to JFXMainView	
↳ Sequence from Player 1 to Transfer Units	
Name: // select Yes or No	
↳ Sequence from Player 1 to JFXMainView	
Name: //Selects Units then Apply	
↳ Sequence from Player 1 to JFXTransferView	
↳ Sequence from Player 1 to Preparation stage	
INCOMING BEHAVIORAL RELATIONSHIPS	
↗ Sequence from Transfer Units to Player 1	
↗ Sequence from Save Game to Player 1	
↗ Sequence from Main Battle Sequence to Player 1	
↗ Sequence from Preparation stage to Player 1	
↗ Sequence from Reinforcement to Player 1	
↗ Sequence from Exchange Cards to Player 1	
Name: // popup	
↗ Sequence from JFXMainView to Player 1	

INCOMING BEHAVIORAL RELATIONSHIPS

Name: addTerritory()
→ Sequence from RiskGameModel to Player 1

➡ Sequence from Define number of players to Player 1

➡ Sequence from Exit Game to Player 1

➡ Sequence from Define the starting player to Player 1

➡ Sequence from Load Game to Player 1

CONNECTORS

 **UseCaseLink** Source -> Destination
From: Player 1 : Actor, Public
To: Reinforcement : UseCase, Public

 **UseCaseLink** Source -> Destination
From: Player 1 : Actor, Public
To: Transfer Units : UseCase, Public

 **UseCaseLink** Source -> Destination
From: Player 1 : Actor, Public
To: Main Battle Sequence : UseCase, Public

ASSOCIATIONS

Association (direction: Unspecified)

Source: Public (Actor) Player 1

Target: Public (UseCase) Exit Game

RiskGameModel

Class in package 'Model'

Az MVC architektura Model részét megvalósító osztalya.

RiskGameModel
Version 1.0 Phase 1.0 Proposed
Szabi created on 2017. 02. 21.. Last modified 2017. 11. 22.

OUTGOING STRUCTURAL RELATIONSHIPS

OUTGOING STRUCTURAL RELATIONSHIPS	
↳ Aggregation from RiskGameModel to RiskGameController	[Direction is 'Source -> Destination'.]
INCOMING STRUCTURAL RELATIONSHIPS	
⇒ Aggregation from Map to RiskGameModel	[Direction is 'Source -> Destination'.]
⇒ Aggregation from Deck to RiskGameModel	[Direction is 'Source -> Destination'.]
ATTRIBUTES	
◆ calculatedCircleNumber : int Private	[Is static False. Containment is Not Specified.]
◆ capturedThisTurn : Boolean Private	[Is static False. Containment is Not Specified.]
◆ circlenumber : int Private	[Is static False. Containment is Not Specified.]
◆ currentPlayer : int Private	[Is static False. Containment is Not Specified.]
◆ deck : Deck Private	[Is static False. Containment is Not Specified.]
◆ hasTransferred : boolean Private	[Is static False. Containment is Not Specified.]
◆ map : Map Private	[Is static False. Containment is Not Specified.]
◆ misnumber : int Private	[Is static False. Containment is Not Specified.]
◆ phase : Phase Private	[Is static False. Containment is Not Specified.]
◆ playersList : List<Player> Private	[Is static False. Containment is Not Specified.]
◆ r : Random Private = new Random()	

ATTRIBUTES	
	[Is static False. Containment is Not Specified.]
◆ waitForUnitsTemp : Territory Private = new Territory[2]	[Is static False. Containment is Not Specified.]
ASSOCIATIONS	
✓ . Association (direction: Source -> Destination)	
Source: Public (Class) RiskGameModel	Target: Private waitForUnitsTemp (Class) Territory
✓ . Association (direction: Source -> Destination)	
Source: Public (Class) RiskGameModel	Target: Private playersList (Class) Player Cardinality: [0..*]
✓ . Association (direction: Source -> Destination)	
Source: Public (Class) RiskGameModel	Target: Public deck (Class) Player
OPERATIONS	
◆ addPlayer (name : String , color : Color) : boolean Public	
Add player with name and color @return boolean	[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ addPlayerToPlayerList (player : Player) : void Public	
Adder for playerList	
Properties: throws = Exception ea_guid = {45684F69-D727-4875-BEF7-5232B4E9D1C9}	[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ addPlayerToPlayerList (map : java.util.Map<Color, String>) : void Public	
Pairs overload	
Properties: throws = Exception	[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ attackTerritoryWithResult (defenderID : int , attackerID : int , defendUnits : int , attackUnits : int) : AttackResult Public	
Return the result of the attack If the attacker have more than 3 units, the fighting attacker units will reduce to 3 If the defender have more than 2 units, the fighting defender units will reduce to 2	
Properties:	

OPERATIONS	
throws = Exception	[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ calculateCircle () : void Private	
Properties:	
throws = Exception	[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ checkAttackPossible (defender : Territory , attacker : Territory , defendUnits : int , attackUnits : int) : boolean Public	
Csak akkor tĂˇmadhat, ha a terĂĽleten elegendL' egysĂ©ge van. Igazzal tĂ©r vissza, ha a tamadas lehetsĂ©ges.	
@return boolean	[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ checkAttackPossible (defenderID : int , attackerID : int) : boolean Public	
Attack checking method	
@return	[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ checkIfCapturedAndConquer (defender : Territory) : boolean Public	
Capture and conquer checking method	
@return boolean	
Properties:	
ea_guid = {BFF4F3B6-2538-4778-9372-EAA3B468EDD8}	[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ checkIfTerritoryIsEmpty (territory : Territory) : boolean Public	
Empty territory checking method	
@return boolean	
Properties:	
ea_guid = {BE4F4E07-A4C4-4e91-9F00-EF1EEB65D26C}	[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ checkTransferPossible (from : int , to : int) : boolean Public	
Transfer checking method	
@return boolean	[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ endBattlePhase () : boolean Public	
Battle	
@return boolean	
Properties:	
throws = Exception	[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]

OPERATIONS
<p>❖ endTurn () : int Public</p> <p>Kor vege Returns next player's id @return int</p> <p>Properties: throws = Exception ea_guid = {F5F8D5D6-8997-43de-A5AD-DDDC647E775F} [Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]</p>
<p>❖ getCurrentPlayer () : int Public</p> <p>Getter for current player @return int</p> <p>[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]</p>
<p>❖ getPhase () : Phase Public</p> <p>Getter for Phase @return phase</p> <p>[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]</p>
<p>❖ getPlayerCards (id : int) : List<Integer> Public</p> <p>[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]</p>
<p>❖ getPlayerName (id : int) : String Public</p> <p>Getter for Player's name @return String</p> <p>[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]</p>
<p>❖ getPlayers () : List<String> Public</p> <p>Getter for Players list @return List<Player></p> <p>[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]</p>
<p>❖ getTerritory (id : int) : Territory Public</p> <p>Getter for Territory @return Territory</p> <p>[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]</p>
<p>❖ getUnitsLeftToReinforce (playerID : int) : int Public</p> <p>Getter for left units in reinforce phase @return int</p> <p>[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]</p>
<p>❖ loadGame (file : File) : boolean Public</p> <p>Load data method</p>

OPERATIONS
<p>@return boolean</p> <p>Properties:</p> <p>throws = Exception</p> <p>[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]</p>
<p>◆ moveUnits (units : int) : boolean Public</p> <p>Move units method</p> <p>@return boolean</p> <p>Properties:</p> <p>throws = Exception</p> <p>[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]</p>
<p>◆ nextPhase () : Phase Private</p> <p>Calculate the next phase</p> <p>@return Phase</p> <p>Properties:</p> <p>throws = Exception</p> <p>[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]</p>
<p>◆ nextPlayer () : void Protected</p> <p>Initializes the next Player's turn. Calculates the amount of Units to place.</p> <p>Properties:</p> <p>throws = Exception</p> <p>ea_guid = {806D8A15-D81F-4852-9BFA-E0988D68A2D8}</p> <p>[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]</p>
<p>◆ reinforce (territoryId : int) : boolean Public</p> <p>Reinforce method</p> <p>@return boolean</p> <p>Properties:</p> <p>throws = Exception</p> <p>[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]</p>
<p>◆ RiskGameModel () : Public</p> <p>The amount of Units left to Place in the Reinforcement Stage of the current Player's current turn.</p> <p>[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]</p>
<p>◆ saveGame (file : File) : boolean Public</p> <p>Save data method</p> <p>@return</p> <p>Properties:</p> <p>throws = IOException</p> <p>[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]</p>

OPERATIONS
<p>◆ selectStartingPlayer (list : List<Player>) : Player Public Random select the starting player <code>@return Player</code> <code>[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]</code></p>
<p>◆ transfer (fromID : int , toID : int , units : int) : boolean Public <code>@return boolean</code> Properties: <code>throws = Exception</code> <code>[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]</code></p>
<p>◆ transfer (from : Territory , to : Territory , units : int) : boolean Public Transfer method <code>@return boolean</code> Properties: <code>throws = Exception</code> <code>ea_guid = {23034260-4F89-40e8-BC43-DD7B2D6023D5}</code> <code>[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]</code></p>
<p>◆ winGame (winner : int) : void Private <code>[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]</code></p>

RiskGameController

Class in package 'Controller'

Az MVC architektura Controller részet megvalósító osztálya. A Controller a Modell és a View közötti kapcsolatért felelős. Rajta keresztül hajtják végre a View-on történt változások, melyek a Model-t érintik, illetve a Model a Controller-en keresztül módosítja a View-t.

RiskGameController
 Version 1.0 Phase 1.0 Proposed
 Szabó Dávid created on 2017. 03. 01.. Last modified 2017. 11. 22.
 Extends java.util.Observable

INCOMING STRUCTURAL RELATIONSHIPS
<p>⇒ Aggregation from JFXAttackView to RiskGameController <code>[Direction is 'Source -> Destination'.]</code></p>
<p>⇒ Aggregation from JFXAddPlayerView to RiskGameController <code>[Direction is 'Source -> Destination'.]</code></p>
<p>⇒ Aggregation from JFXTransferView to RiskGameController <code>[Direction is 'Source -> Destination'.]</code></p>

INCOMING STRUCTURAL RELATIONSHIPS	
⇒ Aggregation from RiskGameModel to RiskGameController	[Direction is 'Source -> Destination'.]
⇒ Aggregation from JFXMainView to RiskGameController	[Direction is 'Source -> Destination'.]
ATTRIBUTES	
◆ addPlayerView : JFXAddPlayerView Private	[Is static False. Containment is Not Specified.]
◆ attackResultView : JFXAttackResultView Private	[Is static False. Containment is Not Specified.]
◆ attackView : JFXAttackView Private	[Is static False. Containment is Not Specified.]
◆ cardGroup : Group Private	[Is static False. Containment is Not Specified.]
◆ cardScene : Scene Private	[Is static False. Containment is Not Specified.]
◆ cardView : JFXCardView Private	
CardView stuffs	[Is static False. Containment is Not Specified.]
◆ lastPhaseUpdate : Phase Private	[Is static False. Containment is Not Specified.]
◆ lastPlayerUpdate : int Private	[Is static False. Containment is Not Specified.]
◆ mainView : JFXMainView Private	
View objects	[Is static False. Containment is Not Specified.]
◆ model : RiskGameModel Private	[Is static False. Containment is Not Specified.]
◆ popupStage : Stage Private	[Is static False. Containment is Not Specified.]

ATTRIBUTES	
◆ preStage : Stage Private Const	
Stages	[Is static False. Containment is Not Specified.]
◆ previouslySelectedTerritory : int Private = -1	[Is static False. Containment is Not Specified.]
◆ primaryStage : Stage Private	[Is static False. Containment is Not Specified.]
◆ titledPaneStage : Stage Private	[Is static False. Containment is Not Specified.]
◆ transferView : JFXTransferView Private	
private JFXRiskCardView cardView;	[Is static False. Containment is Not Specified.]
◆ victoryStage : Stage Private	[Is static False. Containment is Not Specified.]
◆ victoryView : JFXVictoryView Private	
VictoryView stuffs	[Is static False. Containment is Not Specified.]

ASSOCIATIONS	
◆ Association (direction: Source -> Destination)	

Source: Public (Class) JFXViewBase

Target: Protected controller (Class)
RiskGameController

OPERATIONS	
◆ addLog (log : String) : void Public	
A főablak naplójába küld egy új bejegyzést	[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ attackAccepted (defender : int , attacker : int , defenderUnits : int , attackerUnits : int) : void Public	[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ attackRetreat () : void Public	[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]

OPERATIONS	
◆ attackShowed (attacker : int , defender : int) : void	Public [Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ closePopupWindow () : void	Private [Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ convertToCountry (ID : int) : Country	Private [Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ convertToTerritoryID (country : Country) : int	Private [Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ countrySelected (country : Country) : void	Public [Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ getPhase () : Phase	Public [Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ getPreviouslySelectedTerritory () : int	Public [Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ getTerritoryData () : List<Territory>	Public [Is static True. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ isVictory () : boolean	Private [Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ loadGame (loadFile : File) : void	Public [Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ moveCardView () : void	Public [Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ nextPlayerSelected () : void	Public [Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ reinforce (territoryID : int , country : String) : void	Private
Properties:	
throws = Exception	[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ RiskGameController (stage : Stage) : Public	[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]

OPERATIONS	
◆ saveGame (saveFile : File) : void Public	[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ setModel (model : RiskGameModel) : void Public	Eltárolja a modellt és feliratkoztatja magát az eseményeire [Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ showAddPlayerView () : void Public	Add player view [Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ showAttackResultView () : void Private	[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ showAttackView (defenderID : int , attackerID : int) : void Private	Attack View [Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ showCardView () : void Public	[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ showMainView () : void Public	Main View [Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ showTransferView (fromID : int , toID : int) : void Private	Transfer View [Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ showVictoryView () : void Public	[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ switchToMain (map : Map<Color, String>) : void Public	Stage switch Pre -> Main [Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ transferAccepted (from : int , to : int , units : int) : void Public	[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ transferPhaseSelected () : void Public	[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]

OPERATIONS	
◆ UpdateAllTerritoriesOnMap () : void	Private [Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ UpdateCurrentDeck () : void	Private [Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ UpdateCurrentPhase () : void	Private [Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ UpdateCurrentPlayer () : void	Private [Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ UpdateTerritoryOnMap (territoryID : int) : void	Private [Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]

JFXMainView

Class in package 'View'

Az MVC architektura View részét megvalósító osztalya. Ez a View a játék főkepernyője, ezen történik maga a játék (egysegek elhelyezése, csata stb.).

JFXMainView
Version 1.0 Phase 1.0 Proposed
Szabó Dávid created on 2017. 03. 01.. Last modified 2017. 11. 22.
Extends JFXViewBase

OUTGOING STRUCTURAL RELATIONSHIPS	
↳ Generalization from JFXMainView to JFXViewBase	[Direction is 'Source -> Destination'.]
↳ Aggregation from JFXMainView to RiskGameController	[Direction is 'Source -> Destination'.]

ATTRIBUTES	
◆ BASE_HEIGHT : double	Private Const = 600 [Is static False. Containment is Not Specified.]
◆ BASE_WIDTH : double	Private Const = 815 [Is static False. Containment is Not Specified.]
◆ borderPane : BorderPane	Private Properties: annotations = @FXML [Is static False. Containment is Not Specified.]

ATTRIBUTES
<p>◆ btnBattlePhase : Button Private</p> <p>Properties: annotations = @FXML</p> <p>[Is static False. Containment is Not Specified.]</p>
<p>◆ btnControlList : ArrayList<Button> Private</p> <p>[Is static False. Containment is Not Specified.]</p>
<p>◆ btnLoad : Button Private</p> <p>Properties: annotations = @FXML</p> <p>[Is static False. Containment is Not Specified.]</p>
<p>◆ btnLog : Button Private</p> <p>Properties: annotations = @FXML</p> <p>[Is static False. Containment is Not Specified.]</p>
<p>◆ btnNextPlayer : Button Private</p> <p>Properties: annotations = @FXML</p> <p>[Is static False. Containment is Not Specified.]</p>
<p>◆ btnReinforcePhase : Button Private</p> <p>Properties: annotations = @FXML</p> <p>[Is static False. Containment is Not Specified.]</p>
<p>◆ btnSave : Button Private</p> <p>Properties: annotations = @FXML</p> <p>[Is static False. Containment is Not Specified.]</p>
<p>◆ btnSettings : Button Private</p> <p>Properties: annotations = @FXML</p> <p>[Is static False. Containment is Not Specified.]</p>
<p>◆ btnTransferPhase : Button Private</p> <p>Properties: annotations = @FXML</p> <p>[Is static False. Containment is Not Specified.]</p>

ATTRIBUTES
◆ centerStackPane : StackPane Private Properties: annotations = @FXML [Is static False. Containment is Not Specified.]
◆ dimLog : Dimension Package [Is static False. Containment is Not Specified.]
◆ imageView : ImageView Private Properties: annotations = @FXML [Is static False. Containment is Not Specified.]
◆ isLogOpened : boolean Package [Is static False. Containment is Not Specified.]
◆ isSLOpened : boolean Package [Is static False. Containment is Not Specified.]
◆ lblPlayerName1 : Label Private Properties: annotations = @FXML [Is static False. Containment is Not Specified.]
◆ lblPlayerName2 : Label Private Properties: annotations = @FXML [Is static False. Containment is Not Specified.]
◆ lblPlayerName3 : Label Private Properties: annotations = @FXML [Is static False. Containment is Not Specified.]
◆ lblPlayerName4 : Label Private Properties: annotations = @FXML [Is static False. Containment is Not Specified.]
◆ lblPlayerName5 : Label Private Properties: annotations = @FXML [Is static False. Containment is Not Specified.]

ATTRIBUTES
<p>◆ lblPlayerName6 : Label Private</p> <p>Properties:</p> <p>annotations = @FXML</p> <p>[Is static False. Containment is Not Specified.]</p>
<p>◆ lblPlayerNameList : ArrayList<Label> Private</p> <p>[Is static False. Containment is Not Specified.]</p>
<p>◆ pain : Pane Private</p> <p>Properties:</p> <p>annotations = @FXML</p> <p>[Is static False. Containment is Not Specified.]</p>
<p>◆ panePlayerColor1 : Pane Private</p> <p>Properties:</p> <p>annotations = @FXML</p> <p>[Is static False. Containment is Not Specified.]</p>
<p>◆ panePlayerColor2 : Pane Private</p> <p>Properties:</p> <p>annotations = @FXML</p> <p>[Is static False. Containment is Not Specified.]</p>
<p>◆ panePlayerColor3 : Pane Private</p> <p>Properties:</p> <p>annotations = @FXML</p> <p>[Is static False. Containment is Not Specified.]</p>
<p>◆ panePlayerColor4 : Pane Private</p> <p>Properties:</p> <p>annotations = @FXML</p> <p>[Is static False. Containment is Not Specified.]</p>
<p>◆ panePlayerColor5 : Pane Private</p> <p>Properties:</p> <p>annotations = @FXML</p> <p>[Is static False. Containment is Not Specified.]</p>
<p>◆ panePlayerColor6 : Pane Private</p> <p>Properties:</p> <p>annotations = @FXML</p> <p>[Is static False. Containment is Not Specified.]</p>

ATTRIBUTES	
◆ panePlayerColorList : ArrayList<Pane>	Private [Is static False. Containment is Not Specified.]
◆ stage : Stage	Private [Is static False. Containment is Not Specified.]
◆ taLog : TextArea	Private Properties: annotations = @FXML [Is static False. Containment is Not Specified.]
◆ world : World	Private [Is static False. Containment is Not Specified.]
OPERATIONS	
◆ AddControllerListener (controller : RiskGameController) : void	Public
Properties:	
annotations = @Override	[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ appendLog (s : String) : void	Public [Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ closeLog () : void	Public [Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ finalize () : void	Public Properties: throws = Throwable annotations = @Override [Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ fit (height : double, width : double) : void	Public [Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ getRoot () : Parent	Public Properties: annotations = @Override [Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ getWorld () : World	Public [Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]

OPERATIONS	
◆ handleLogButton (event : ActionEvent) : void Private	
Properties:	
annotations = @FXML	[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ handleLoadButton (event : ActionEvent) : void Private	
Properties:	
annotations = @FXML	[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ handleSaveButton (event : ActionEvent) : void Private	
Properties:	
annotations = @FXML	[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ handleSettingsButton (event : ActionEvent) : void Private	
Properties:	
annotations = @FXML	[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ initialize (location : URL , resources : ResourceBundle) : void Public	
Properties:	
annotations = @Override	[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ isLogOpened () : boolean Public	
	[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ isSLOpened () : boolean Public	
	[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ JFXMainView (stage : Stage) : Public	
	[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ openLog () : void Public	
	[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ UpdateConnectedPlayers (playerList : List<String>) : void Public	
	[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ UpdateCurrentDeck (cards : List<Integer>) : void Public	
TODO:	[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]

OPERATIONS	
◆ UpdateCurrentPhase (phase : Phase) : void Public	[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ UpdateCurrentPlayer (playerID : int) : void Public @param player	[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ updateIcons (attacker : int, defender : int) : void Public	[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ UpdateViewState (territory : Territory) : void Public @param territories	Properties: ea_guid = {0E811D72-EC4E-47ad-8B46-C39084EAD8F7} [Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ UpdateViewState (territories : List<Territory>) : void Public	[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]

<anonymous>

Note in package 'Preparation Stage'

Preparation Stage

<anonymous>
Version 1.0 Phase 1.0 Proposed
Oliver created on 2017. 03. 28.. Last modified 2017. 03. 28.
Extends

INTERACTION MESSAGES	
✉ 1.0 '/* click to begin action' from 'Player 1' sent to 'JFXMainView'.	Synchronous Call. Returns void. [Return is False. Iteration is False. New group is False.]
✉ 1.1 'actionPerformed(ActionEvent)' from 'JFXMainView' sent to 'RiskGameController'.	Synchronous Call. [Return is False. Iteration is False. New group is False.]
✉ 1.2 '/* call' from 'RiskGameController' sent to 'RiskGameModel'.	Synchronous Call. Returns void.

	[Return is False. Iteration is False. New group is False.]
✉ 1.3 'addTerritory()' from 'RiskGameModel' sent to 'Player 1'.	Synchronous Call. Returns void. [Return is False. Iteration is False. New group is False.]
✉ 1.4 'reinforce(Player, Territory)' from 'RiskGameController' sent to 'RiskGameModel'.	Synchronous Call. [Return is False. Iteration is False. New group is False.]

Alternative

InteractionFragment in package 'Preparation Stage'

Alternative
Version 1.0 Phase 1.0 Proposed
fodorad created on 2017. 04. 18.. Last modified 2017. 04. 18.

Alternative1

InteractionFragment in package 'Preparation Stage'

Alternative1
Version 1.0 Phase 1.0 Proposed
fodorad created on 2017. 03. 22.. Last modified 2017. 04. 18.

Alternative2

InteractionFragment in package 'Preparation Stage'

Alternative2
Version 1.0 Phase 1.0 Proposed
fodorad created on 2017. 03. 22.. Last modified 2017. 03. 28.

Alternative3

InteractionFragment in package 'Preparation Stage'

Alternative3
Version 1.0 Phase 1.0 Proposed
fodorad created on 2017. 03. 22.. Last modified 2017. 03. 28.

Alternative4

InteractionFragment in package 'Preparation Stage'

Alternative4
Version 1.0 Phase 1.0 Proposed
fodorad created on 2017. 03. 22.. Last modified 2017. 03. 28.

Preparation loop

InteractionFragment in package 'Preparation Stage'

Preparation loop
Version 1.0 Phase 1.0 Proposed
fodorad created on 2017. 03. 22.. Last modified 2017. 03. 28.

StartingPlayerLoop

InteractionFragment in package 'Preparation Stage'

StartingPlayerLoop
Version 1.0 Phase 1.0 Proposed
Oliver created on 2017. 04. 04.. Last modified 2017. 04. 04.

Class Model

Package in package 'Model'

Class Model

Version Phase 1.0 Proposed

Szabi created on 2017. 02. 21.. Last modified 2017. 02. 21.

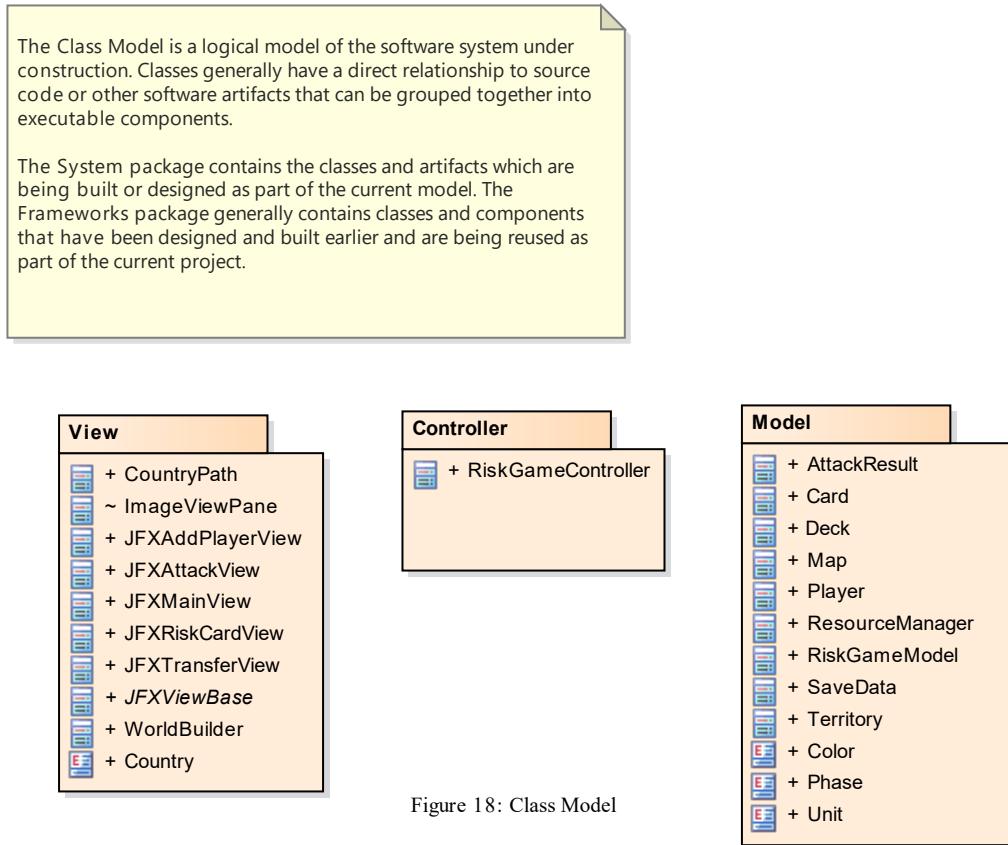
Class Model diagram

Class diagram in package 'Class Model'

Class Model

Version 1.0

Oliver created on 2017. 02. 21.. Last modified 2017. 03. 28.



Note

Note in package 'Class Model'

The Class Model is a logical model of the software system under construction. Classes generally have a direct relationship to source code or other software artifacts that can be grouped together into executable components.

The System package contains the classes and artifacts which are being built or designed as part of the current model. The Frameworks package generally contains classes and components that have been designed and built earlier and are being reused as part of the current project.

Note

Version 1.0 Phase 1.0 Proposed

Szabi created on 2017. 02. 21.. Last modified 2017. 02. 21.

Extends

Controller

Package in package 'Class Model'

Controller
Version 1.0 Phase 1.0 Proposed
Szabó Dávid created on 2017. 03. 01.. Last modified 2017. 03. 01.

View

Package in package 'Class Model'

View
Version 1.0 Phase 1.0 Proposed
Szabó Dávid created on 2017. 03. 01.. Last modified 2017. 03. 01.

Model

Package in package 'Class Model'

Model
Version 1.0 Phase 1.0 Proposed
Szabi created on 2017. 02. 21.. Last modified 2017. 03. 01.

Application

Package in package 'Class Model'

Application
Version 1.0 Phase 1.0 Proposed
Oliver created on 2017. 05. 04.. Last modified 2017. 05. 04.

Application diagram

Class diagram in package 'Application'

Application
Version 1.0
Oliver created on 2017. 05. 04.. Last modified 2017. 09. 21.

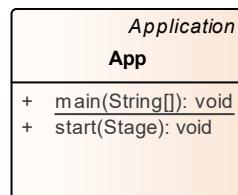


Figure 19: Application

App

Class in package 'Application'

App
Version 1.0 Phase 1.0 Proposed
Oliver created on 2017. 05. 04.. Last modified 2017. 11. 22.
Extends Application

OPERATIONS

◆ main (args : String[]) : void Public
[Is static True. Is abstract False. Is return array False. Is query False. Is synchronized False.]

◆ start (primaryStage : Stage) : void Public

Properties:

annotations = @Override
throws = Exception
[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]

Controller

Package in package 'Class Model'

Controller
Version 1.0 Phase 1.0 Proposed
Szabó Dávid created on 2017. 03. 01.. Last modified 2017. 03. 01.

Controller diagram

Class diagram in package 'Controller'

Controller
Version 1.0
Szabó Dávid created on 2017. 03. 01.. Last modified 2017. 11. 22.

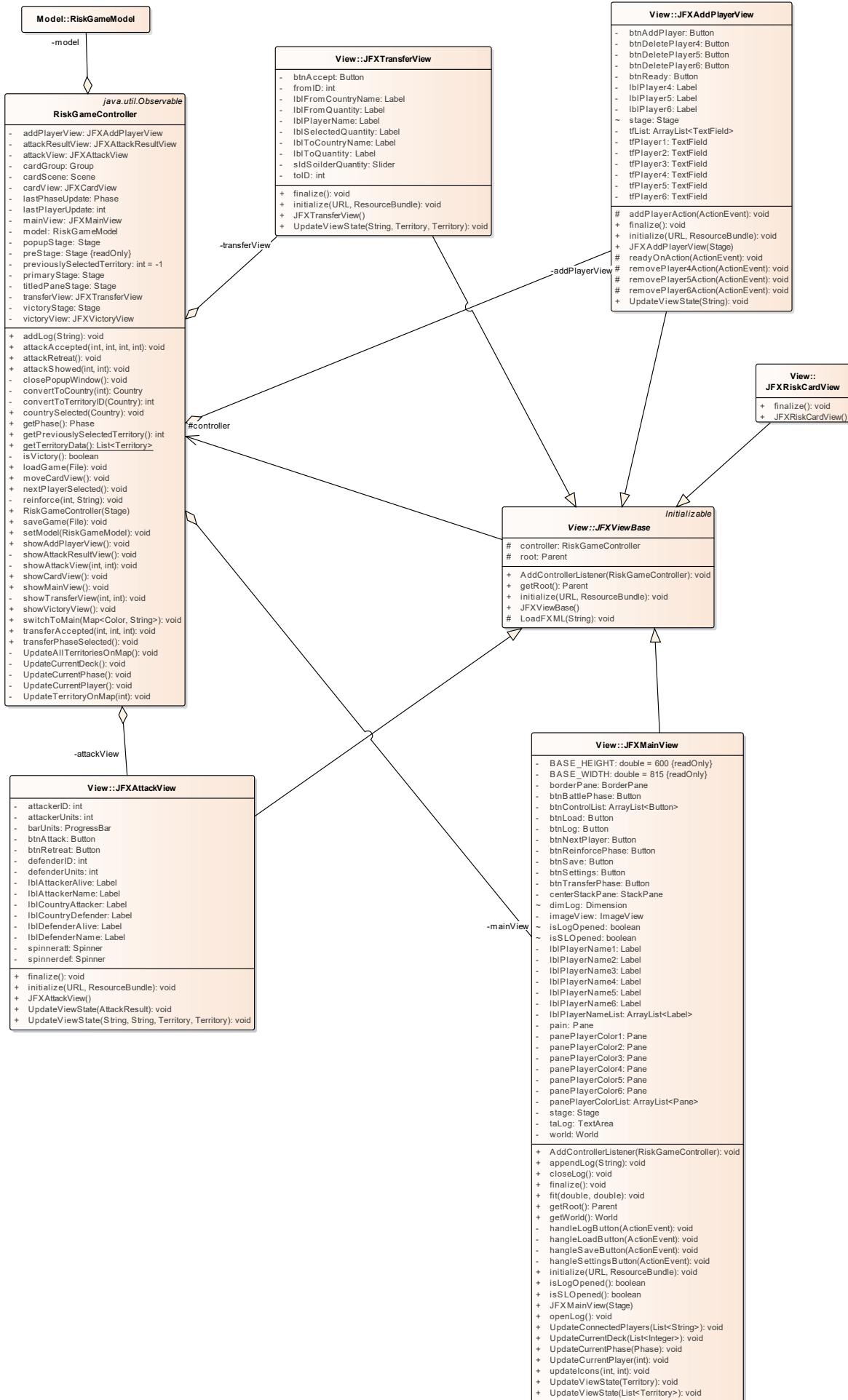


Figure 20: Controller

RiskGameModel

Class in package 'Model'

Az MVC architektura Model részét megvalósító osztálya.

RiskGameModel
Version 1.0 Phase 1.0 Proposed
Szabi created on 2017. 02. 21.. Last modified 2017. 11. 22.

OUTGOING STRUCTURAL RELATIONSHIPS

⬅ Aggregation from RiskGameModel to RiskGameController

[Direction is 'Source -> Destination'.]

INCOMING STRUCTURAL RELATIONSHIPS

➡ Aggregation from Map to RiskGameModel

[Direction is 'Source -> Destination'.]

➡ Aggregation from Deck to RiskGameModel

[Direction is 'Source -> Destination'.]

ATTRIBUTES

◆ calculatedCircleNumber : int Private

[Is static False. Containment is Not Specified.]

◆ capturedThisTurn : Boolean Private

[Is static False. Containment is Not Specified.]

◆ circlenumber : int Private

[Is static False. Containment is Not Specified.]

◆ currentPlayer : int Private

[Is static False. Containment is Not Specified.]

◆ deck : Deck Private

[Is static False. Containment is Not Specified.]

◆ hasTransferred : boolean Private

[Is static False. Containment is Not Specified.]

◆ map : Map Private

[Is static False. Containment is Not Specified.]

◆ miscnumber : int Private

ATTRIBUTES	
	[Is static False. Containment is Not Specified.]
◆ phase : Phase Private	[Is static False. Containment is Not Specified.]
◆ playersList : List<Player> Private	[Is static False. Containment is Not Specified.]
◆ r : Random Private = new Random()	[Is static False. Containment is Not Specified.]
◆ waitForUnitsTemp : Territory Private = new Territory[2]	[Is static False. Containment is Not Specified.]

ASSOCIATIONS	
✓ Association (direction: Source -> Destination)	
Source: Public (Class) RiskGameModel	Target: Private waitForUnitsTemp (Class) Territory
✓ Association (direction: Source -> Destination)	
Source: Public (Class) RiskGameModel	Target: Private playersList (Class) Player Cardinality: [0..*]
✓ Association (direction: Source -> Destination)	
Source: Public (Class) RiskGameModel	Target: Public deck (Class) Player

OPERATIONS	
◆ addPlayer (name : String , color : Color) : boolean Public	
Add player with name and color @return boolean	[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ addPlayerToPlayerList (player : Player) : void Public	
Adder for playerList	
Properties:	
throws = Exception	
ea_guid = {45684F69-D727-4875-BEF7-5232B4E9D1C9}	[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ addPlayerToPlayerList (map : java.util.Map<Color, String>) : void Public	
Pairs overload	

OPERATIONS	
Properties:	
throws = Exception	[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ attackTerritoryWithResult (defenderID : int , attackerID : int , defendUnits : int , attackUnits : int) : AttackResult Public	
Return the result of the attack If the attacker have more than 3 units, the fighting attacker units will reduce to 3 If the defender have more than 2 units, the fighting defender units will reduce to 2	
Properties:	
throws = Exception	[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ calculateCircle () : void Private	
Properties:	
throws = Exception	[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ checkAttackPossible (defender : Territory , attacker : Territory , defendUnits : int , attackUnits : int) : boolean Public	
Csak akkor tĂmadhat, ha a terĂlleten elegendL' egysĂge van. Igazzal tĂcr vissza, ha a tamadas lehetsĂges.	
@return boolean	
	[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ checkAttackPossible (defenderID : int , attackerID : int) : boolean Public	
Attack checking method	
@return	
	[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ checkIfCapturedAndConquer (defender : Territory) : boolean Public	
Capture and conquer checking method	
@return boolean	
Properties:	
ea_guid = {BFF4F3B6-2538-4778-9372-EAA3B468EDD8}	
	[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ checkIfTerritoryIsEmpty (territory : Territory) : boolean Public	
Empty territory checking method	
@return boolean	
Properties:	
ea_guid = {BE4F4E07-A4C4-4e91-9F00-EF1EEB65D26C}	
	[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ checkTransferPossible (from : int , to : int) : boolean Public	
Transfer checking method	
@return boolean	

OPERATIONS	
[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]	
◆ endBattlePhase () : boolean Public	
Battle @return boolean	
Properties: throws = Exception	[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ endTurn () : int Public	
Kor vege Returns next player's id @return int	
Properties: throws = Exception ea_guid = {F5F8D5D6-8997-43de-A5AD-DDDC647E775F}	[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ getCurrentPlayer () : int Public	
Getter for current player @return int	[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ getPhase () : Phase Public	
Getter for Phase @return phase	[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ getPlayerCards (id : int) : List<Integer> Public	[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ getPlayerName (id : int) : String Public	
Getter for Player's name @return String	[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ getPlayers () : List<String> Public	
Getter for Players list @return List<Player>	[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ getTerritory (id : int) : Territory Public	
Getter for Territory @return Territory	[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]

OPERATIONS
<p>◆ getUnitsLeftToReinforce (playerID : int) : int Public Getter for left units in reinforce phase <code>@return int</code> <code>[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]</code></p>
<p>◆ loadGame (file : File) : boolean Public Load data method <code>@return boolean</code> Properties: <code>throws = Exception</code> <code>[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]</code></p>
<p>◆ moveUnits (units : int) : boolean Public Move units method <code>@return boolean</code> Properties: <code>throws = Exception</code> <code>[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]</code></p>
<p>◆ nextPhase () : Phase Private Calculate the next phase <code>@return Phase</code> Properties: <code>throws = Exception</code> <code>[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]</code></p>
<p>◆ nextPlayer () : void Protected Initializes the next Player's turn. Calculates the amount of Units to place. Properties: <code>throws = Exception</code> <code>ea_guid = {806D8A15-D81F-4852-9BFA-E0988D68A2D8}</code> <code>[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]</code></p>
<p>◆ reinforce (territoryId : int) : boolean Public Reinforce method <code>@return boolean</code> Properties: <code>throws = Exception</code> <code>[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]</code></p>
<p>◆ RiskGameModel () : Public</p>

OPERATIONS	
The amount of Units left to Place in the Reinforcement Stage of the current Player's current turn.	[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
 saveGame (file : File) : boolean Public	
Save data method @return	
Properties: throws = IOException	[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
 selectStartingPlayer (list : List<Player>) : Player Public	
Random select the starting player @return Player	[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
 transfer (fromID : int , toID : int , units : int) : boolean Public	
@return boolean	
Properties: throws = Exception	[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
 transfer (from : Territory , to : Territory , units : int) : boolean Public	
Transfer method @return boolean	
Properties: throws = Exception ea_guid = {23034260-4F89-40e8-BC43-DD7B2D6023D5}	[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
 winGame (winner : int) : void Private	
	[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]

JFXAddPlayerView

Class in package 'View'

Ezen a View-en keresztul tortenik a leendo jatekosok hozzaadasa a jatekhoz.

JFXAddPlayerView
Version 1.0 Phase 1.0 Proposed
fodorad created on 2017. 03. 22.. Last modified 2017. 11. 22.
Extends JFXViewBase

OUTGOING STRUCTURAL RELATIONSHIPS

OUTGOING STRUCTURAL RELATIONSHIPS	
↳ Aggregation from JFXAddPlayerView to RiskGameController	[Direction is 'Source -> Destination'.]
↳ Generalization from JFXAddPlayerView to JFXViewBase	[Direction is 'Source -> Destination'.]
ATTRIBUTES	
◆ btnAddPlayer : Button Private	
Properties:	
annotations = @FXML	[Is static False. Containment is Not Specified.]
◆ btnDeletePlayer4 : Button Private	
Properties:	
annotations = @FXML	[Is static False. Containment is Not Specified.]
◆ btnDeletePlayer5 : Button Private	
Properties:	
annotations = @FXML	[Is static False. Containment is Not Specified.]
◆ btnDeletePlayer6 : Button Private	
Properties:	
annotations = @FXML	[Is static False. Containment is Not Specified.]
◆ btnReady : Button Private	
Properties:	
annotations = @FXML	[Is static False. Containment is Not Specified.]
◆ lblPlayer4 : Label Private	
Properties:	
annotations = @FXML	[Is static False. Containment is Not Specified.]
◆ lblPlayer5 : Label Private	
Properties:	
annotations = @FXML	[Is static False. Containment is Not Specified.]

ATTRIBUTES	
◆ lblPlayer6 : Label	Private
Properties:	
annotations = @FXML	[Is static False. Containment is Not Specified.]
◆ stage : Stage	Package
	[Is static False. Containment is Not Specified.]
◆ tfList : ArrayList<TextField>	Private
	[Is static False. Containment is Not Specified.]
◆ tfPlayer1 : TextField	Private
Properties:	
annotations = @FXML	[Is static False. Containment is Not Specified.]
◆ tfPlayer2 : TextField	Private
Properties:	
annotations = @FXML	[Is static False. Containment is Not Specified.]
◆ tfPlayer3 : TextField	Private
Properties:	
annotations = @FXML	[Is static False. Containment is Not Specified.]
◆ tfPlayer4 : TextField	Private
Properties:	
annotations = @FXML	[Is static False. Containment is Not Specified.]
◆ tfPlayer5 : TextField	Private
Properties:	
annotations = @FXML	[Is static False. Containment is Not Specified.]
◆ tfPlayer6 : TextField	Private
Properties:	
annotations = @FXML	[Is static False. Containment is Not Specified.]

OPERATIONS	
◆ addPlayerAction (event : ActionEvent) : void	Protected

OPERATIONS
<p>Properties: annotations = @FXML [Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]</p>
<p>◆ finalize () : void Public</p>
<p>Properties: throws = Throwable annotations = @Override [Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]</p>
<p>◆ initialize (location : URL , resources : ResourceBundle) : void Public</p>
<p>Properties: annotations = @Override [Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]</p>
<p>◆ JFXAddPlayerView (stage : Stage) : Public [Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]</p>
<p>◆ readyOnAction (event : ActionEvent) : void Protected</p>
<p>Properties: annotations = @FXML [Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]</p>
<p>◆ removePlayer4Action (event : ActionEvent) : void Protected</p>
<p>Properties: annotations = @FXML [Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]</p>
<p>◆ removePlayer5Action (event : ActionEvent) : void Protected</p>
<p>Properties: annotations = @FXML [Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]</p>
<p>◆ removePlayer6Action (event : ActionEvent) : void Protected</p>
<p>Properties: annotations = @FXML [Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]</p>
<p>◆ UpdateViewState (name : String) : void Public [Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]</p>

Class in package 'View'

Ezen a View-en keresztul jelölheti meg az egyik játékos a másik játékos (ellenfel) azon területet, amelyet tamadni szeretne.

JFXAttackView
Version 1.0 Phase 1.0 Proposed
Szabó Dávid created on 2017. 03. 08.. Last modified 2017. 11. 22.
Extends JFXViewBase

OUTGOING STRUCTURAL RELATIONSHIPS

↳ Generalization from JFXAttackView to JFXViewBase

[Direction is 'Source -> Destination'.]

↳ Aggregation from JFXAttackView to RiskGameController

[Direction is 'Source -> Destination'.]

ATTRIBUTES

◆ attackerID : int Private

[Is static False. Containment is Not Specified.]

◆ attackerUnits : int Private

[Is static False. Containment is Not Specified.]

◆ barUnits : ProgressBar Private

Properties:

annotations = @FXML

[Is static False. Containment is Not Specified.]

◆ btnAttack : Button Private

Properties:

annotations = @FXML

[Is static False. Containment is Not Specified.]

◆ btnRetreat : Button Private

Properties:

annotations = @FXML

[Is static False. Containment is Not Specified.]

◆ defenderID : int Private

[Is static False. Containment is Not Specified.]

◆ defenderUnits : int Private

[Is static False. Containment is Not Specified.]

◆ lblAttackerAlive : Label Private

ATTRIBUTES	
Properties:	
annotations = @FXML	[Is static False. Containment is Not Specified.]
◆ lblAttackerName : Label Private	
Properties:	
annotations = @FXML	[Is static False. Containment is Not Specified.]
◆ lblCountryAttacker : Label Private	
Properties:	
annotations = @FXML	[Is static False. Containment is Not Specified.]
◆ lblCountryDefender : Label Private	
Properties:	
annotations = @FXML	[Is static False. Containment is Not Specified.]
◆ lblDefenderAlive : Label Private	
Properties:	
annotations = @FXML	[Is static False. Containment is Not Specified.]
◆ lblDefenderName : Label Private	
Properties:	
annotations = @FXML	[Is static False. Containment is Not Specified.]
◆ spinneratt : Spinner Private	
Properties:	
annotations = @FXML	[Is static False. Containment is Not Specified.]
◆ spinnerdef : Spinner Private	
Properties:	
annotations = @FXML	[Is static False. Containment is Not Specified.]

OPERATIONS	
◆ finalize () : void Public	
Properties:	
throws = Throwable	

OPERATIONS	
annotations = @Override	[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ initialize (location : URL , resources : ResourceBundle) : void Public	
Properties:	
annotations = @Override	[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ JFXAttackView () : Public	[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ UpdateViewState (attackResult : AttackResult) : void Public	[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ UpdateViewState (defenderName : String , attackerName : String , defender : Territory , attacker : Territory) : void Public	[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]

JFXMainView

Class in package 'View'

Az MVC architektura View részet megvalósító osztálya. Ez a View a játék főkepernyője, ezen történik maga a játék (egysegek elhelyezése, csata stb.).

JFXMainView
Version 1.0 Phase 1.0 Proposed
Szabó Dávid created on 2017. 03. 01.. Last modified 2017. 11. 22.
Extends JFXViewBase

OUTGOING STRUCTURAL RELATIONSHIPS	
◀ Generalization from JFXMainView to JFXViewBase	[Direction is 'Source -> Destination'.]
◀ Aggregation from JFXMainView to RiskGameController	[Direction is 'Source -> Destination'.]

ATTRIBUTES	
◆ BASE_HEIGHT : double Private Const = 600	[Is static False. Containment is Not Specified.]
◆ BASE_WIDTH : double Private Const = 815	[Is static False. Containment is Not Specified.]
◆ borderPane : BorderPane Private	

ATTRIBUTES	
Properties: annotations = @FXML	[Is static False. Containment is Not Specified.]
◆ btnBattlePhase : Button Private	
Properties: annotations = @FXML	[Is static False. Containment is Not Specified.]
◆ btnControlList : ArrayList<Button> Private	[Is static False. Containment is Not Specified.]
◆ btnLoad : Button Private	
Properties: annotations = @FXML	[Is static False. Containment is Not Specified.]
◆ btnLog : Button Private	
Properties: annotations = @FXML	[Is static False. Containment is Not Specified.]
◆ btnNextPlayer : Button Private	
Properties: annotations = @FXML	[Is static False. Containment is Not Specified.]
◆ btnReinforcePhase : Button Private	
Properties: annotations = @FXML	[Is static False. Containment is Not Specified.]
◆ btnSave : Button Private	
Properties: annotations = @FXML	[Is static False. Containment is Not Specified.]
◆ btnSettings : Button Private	
Properties: annotations = @FXML	[Is static False. Containment is Not Specified.]
◆ btnTransferPhase : Button Private	
Properties:	

ATTRIBUTES	
annotations = @FXML	[Is static False. Containment is Not Specified.]
◆ centerStackPane : StackPane Private	
Properties:	
annotations = @FXML	[Is static False. Containment is Not Specified.]
◆ dimLog : Dimension Package	[Is static False. Containment is Not Specified.]
◆ imageView : ImageView Private	
Properties:	
annotations = @FXML	[Is static False. Containment is Not Specified.]
◆ isLogOpened : boolean Package	[Is static False. Containment is Not Specified.]
◆ isSLOpened : boolean Package	[Is static False. Containment is Not Specified.]
◆ lblPlayerName1 : Label Private	
Properties:	
annotations = @FXML	[Is static False. Containment is Not Specified.]
◆ lblPlayerName2 : Label Private	
Properties:	
annotations = @FXML	[Is static False. Containment is Not Specified.]
◆ lblPlayerName3 : Label Private	
Properties:	
annotations = @FXML	[Is static False. Containment is Not Specified.]
◆ lblPlayerName4 : Label Private	
Properties:	
annotations = @FXML	[Is static False. Containment is Not Specified.]
◆ lblPlayerName5 : Label Private	

ATTRIBUTES	
Properties: annotations = @FXML	[Is static False. Containment is Not Specified.]
◆ lblPlayerName6 : Label Private	
Properties: annotations = @FXML	[Is static False. Containment is Not Specified.]
◆ lblPlayerNameList : ArrayList<Label> Private	[Is static False. Containment is Not Specified.]
◆ pain : Pane Private	
Properties: annotations = @FXML	[Is static False. Containment is Not Specified.]
◆ panePlayerColor1 : Pane Private	
Properties: annotations = @FXML	[Is static False. Containment is Not Specified.]
◆ panePlayerColor2 : Pane Private	
Properties: annotations = @FXML	[Is static False. Containment is Not Specified.]
◆ panePlayerColor3 : Pane Private	
Properties: annotations = @FXML	[Is static False. Containment is Not Specified.]
◆ panePlayerColor4 : Pane Private	
Properties: annotations = @FXML	[Is static False. Containment is Not Specified.]
◆ panePlayerColor5 : Pane Private	
Properties: annotations = @FXML	[Is static False. Containment is Not Specified.]
◆ panePlayerColor6 : Pane Private	
Properties:	

ATTRIBUTES	
annotations = @FXML	[Is static False. Containment is Not Specified.]
panePlayerColorList : ArrayList<Pane> Private	[Is static False. Containment is Not Specified.]
stage : Stage Private	[Is static False. Containment is Not Specified.]
taLog : TextArea Private	
Properties:	
annotations = @FXML	[Is static False. Containment is Not Specified.]
world : World Private	[Is static False. Containment is Not Specified.]
OPERATIONS	
AddControllerListener (controller : RiskGameController) : void Public	
Properties:	
annotations = @Override	[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
appendLog (s : String) : void Public	[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
closeLog () : void Public	[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
finalize () : void Public	
Properties:	
throws = Throwable	
annotations = @Override	[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
fit (height : double , width : double) : void Public	[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
getRoot () : Parent Public	
Properties:	
annotations = @Override	[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
getWorld () : World Public	

OPERATIONS	
[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]	
<p>◆ handleLogButton (event : ActionEvent) : void Private</p> <p>Properties: annotations = @FXML [Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]</p>	
<p>◆ handleLoadButton (event : ActionEvent) : void Private</p> <p>Properties: annotations = @FXML [Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]</p>	
<p>◆ handleSaveButton (event : ActionEvent) : void Private</p> <p>Properties: annotations = @FXML [Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]</p>	
<p>◆ handleSettingsButton (event : ActionEvent) : void Private</p> <p>Properties: annotations = @FXML [Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]</p>	
<p>◆ initialize (location : URL, resources : ResourceBundle) : void Public</p> <p>Properties: annotations = @Override [Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]</p>	
<p>◆ isLogOpened () : boolean Public</p> <p>[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]</p>	
<p>◆ isSLOpened () : boolean Public</p> <p>[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]</p>	
<p>◆ JFXMainView (stage : Stage) : Public</p> <p>[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]</p>	
<p>◆ openLog () : void Public</p> <p>[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]</p>	
<p>◆ UpdateConnectedPlayers (playerList : List<String>) : void Public</p> <p>[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]</p>	
<p>◆ UpdateCurrentDeck (cards : List<Integer>) : void Public</p>	

OPERATIONS	
TODO:	[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ UpdateCurrentPhase (phase : Phase) : void Public	[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ UpdateCurrentPlayer (playerID : int) : void Public	
@param player	[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ updateIcons (attacker : int, defender : int) : void Public	[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ UpdateViewState (territory : Territory) : void Public	
@param territories	
Properties:	
ea_guid = {0E811D72-EC4E-47ad-8B46-C39084EAD8F7}	[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ UpdateViewState (territories : List<Territory>) : void Public	[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]

JFXRiskCardView

Class in package 'View'

JFXRiskCardView
Version 1.0 Phase 1.0 Proposed
Oliver created on 2017. 03. 28.. Last modified 2017. 11. 22.
Extends JFXViewBase

OUTGOING STRUCTURAL RELATIONSHIPS	
↳ Generalization from JFXRiskCardView to JFXViewBase	[Direction is 'Source -> Destination'.]

OPERATIONS	
◆ finalize () : void Public	
Properties:	
throws = Throwable	
annotations = @Override	[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ JFXRiskCardView () : Public	[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]

OPERATIONS**JFXTransferView***Class in package 'View'*

Ezen a View-en valaszthatja ki a jatekos, hogy a sajat teruleten belül hova helyezi az egységeit.

JFXTransferView
Version 1.0 Phase 1.0 Proposed
Szabó Dávid created on 2017. 03. 08.. Last modified 2017. 11. 22.
Extends JFXViewBase

OUTGOING STRUCTURAL RELATIONSHIPS

↳ Generalization from JFXTransferView to JFXViewBase

[Direction is 'Source -> Destination'.]

↳ Aggregation from JFXTransferView to RiskGameController

[Direction is 'Source -> Destination'.]

ATTRIBUTES

◆ btnAccept : Button Private

Properties:

annotations = @FXML

[Is static False. Containment is Not Specified.]

◆ fromID : int Private

[Is static False. Containment is Not Specified.]

◆ lblFromCountryName : Label Private

Properties:

annotations = @FXML

[Is static False. Containment is Not Specified.]

◆ lblFromQuantity : Label Private

Properties:

annotations = @FXML

[Is static False. Containment is Not Specified.]

◆ lblPlayerName : Label Private

Properties:

annotations = @FXML

[Is static False. Containment is Not Specified.]

◆ lblSelectedQuantity : Label Private

ATTRIBUTES	
Properties: annotations = @FXML	[Is static False. Containment is Not Specified.]
◆ lblToCountryName : Label Private Properties: annotations = @FXML	[Is static False. Containment is Not Specified.]
◆ lblToQuantity : Label Private Properties: annotations = @FXML	[Is static False. Containment is Not Specified.]
◆ sldSoilderQuantity : Slider Private Properties: annotations = @FXML	[Is static False. Containment is Not Specified.]
◆ toID : int Private	[Is static False. Containment is Not Specified.]
OPERATIONS	
◆ finalize () : void Public Properties: annotations = @Override throws = Throwable	[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ initialize (location : URL , resources : ResourceBundle) : void Public Properties: annotations = @Override	[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ JFXTransferView () : Public	[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ UpdateViewState (player : String , from : Territory , to : Territory) : void Public	[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]

JFXViewBase

Class in package 'View'

JFXViewBase
Version 1.0 Phase 1.0 Proposed
Oliver created on 2017. 05. 04.. Last modified 2017. 11. 22.
Extends Initializable

INCOMING STRUCTURAL RELATIONSHIPS

- ⇒ Generalization from JFXMainView to JFXViewBase
[Direction is 'Source -> Destination'.]
- ⇒ Generalization from JFXAttackView to JFXViewBase
[Direction is 'Source -> Destination'.]
- ⇒ Generalization from JFXTransferView to JFXViewBase
[Direction is 'Source -> Destination'.]
- ⇒ Generalization from JFXRiskCardView to JFXViewBase
[Direction is 'Source -> Destination'.]
- ⇒ Generalization from JFXAddPlayerView to JFXViewBase
[Direction is 'Source -> Destination'.]

ATTRIBUTES

- ◆ controller : RiskGameController Protected
[Is static False. Containment is Not Specified.]
- ◆ root : Parent Protected
[Is static False. Containment is Not Specified.]

ASSOCIATIONS

- ✍ Association (direction: Source -> Destination)
Source: Public (Class) JFXViewBase Target: Protected controller (Class)
RiskGameController

OPERATIONS

- ◆ AddControllerListener (controller : RiskGameController) : void Public
[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
- ◆ getRoot () : Parent Public
[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
- ◆ initialize (arg0 : URL , arg1 : ResourceBundle) : void Public

OPERATIONS	
Properties:	
annotations = @Override	[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ JFXViewBase () : Public	[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ LoadFXML (resourceName : String) : void Protected	[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]

RiskGameController

Class in package 'Controller'

Az MVC architektura Controller részet megvalósító osztálya. A Controller a Modell és a View kozotti kapcsolatert felelől. Rajta keresztül hajtoddnak végre a View-on történt változások, melyek a Model-t érintik, illetve a Model a Controller-en keresztül módosítja a View-t.

RiskGameController
Version 1.0 Phase 1.0 Proposed
Szabó Dávid created on 2017. 03. 01.. Last modified 2017. 11. 22.
Extends java.util.Observable

INCOMING STRUCTURAL RELATIONSHIPS	
⇒ Aggregation from JFXAttackView to RiskGameController	[Direction is 'Source -> Destination'.]
⇒ Aggregation from JFXAddPlayerView to RiskGameController	[Direction is 'Source -> Destination'.]
⇒ Aggregation from JFXTransferView to RiskGameController	[Direction is 'Source -> Destination'.]
⇒ Aggregation from RiskGameModel to RiskGameController	[Direction is 'Source -> Destination'.]
⇒ Aggregation from JFXMLMainView to RiskGameController	[Direction is 'Source -> Destination'.]

ATTRIBUTES	
◆ addPlayerView : JFXAddPlayerView Private	[Is static False. Containment is Not Specified.]

ATTRIBUTES	
◆ attackResultView : JFXAttackResultView	Private [Is static False. Containment is Not Specified.]
◆ attackView : JFXAttackView	Private [Is static False. Containment is Not Specified.]
◆ cardGroup : Group	Private [Is static False. Containment is Not Specified.]
◆ cardScene : Scene	Private [Is static False. Containment is Not Specified.]
◆ cardView : JFXCardView	Private CardView stuffs [Is static False. Containment is Not Specified.]
◆ lastPhaseUpdate : Phase	Private [Is static False. Containment is Not Specified.]
◆ lastPlayerUpdate : int	Private [Is static False. Containment is Not Specified.]
◆ mainView : JFXMainView	Private View objects [Is static False. Containment is Not Specified.]
◆ model : RiskGameModel	Private [Is static False. Containment is Not Specified.]
◆ popupStage : Stage	Private [Is static False. Containment is Not Specified.]
◆ preStage : Stage	Private Const Stages [Is static False. Containment is Not Specified.]
◆ previouslySelectedTerritory : int	Private = -1 [Is static False. Containment is Not Specified.]
◆ primaryStage : Stage	Private [Is static False. Containment is Not Specified.]
◆ titledPaneStage : Stage	Private [Is static False. Containment is Not Specified.]

ATTRIBUTES	
◆ transferView : JFXTransferView Private	
private JFXRiskCardView cardView;	[Is static False. Containment is Not Specified.]
◆ victoryStage : Stage Private	[Is static False. Containment is Not Specified.]
◆ victoryView : JFXVictoryView Private	
VictoryView stuffs	[Is static False. Containment is Not Specified.]

ASSOCIATIONS	
↙ Association (direction: Source -> Destination)	
Source: Public (Class) JFXViewBase	Target: Protected controller (Class) RiskGameController

OPERATIONS	
◆ addLog (log : String) : void Public	
A főablak naplójába küld egy új bejegyzést	[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ attackAccepted (defender : int , attacker : int , defenderUnits : int , attackerUnits : int) : void Public	[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ attackRetreat () : void Public	[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ attackShowed (attacker : int , defender : int) : void Public	[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ closePopupWindow () : void Private	[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ convertToCountry (ID : int) : Country Private	[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ convertToTerritoryID (country : Country) : int Private	[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ countrySelected (country : Country) : void Public	

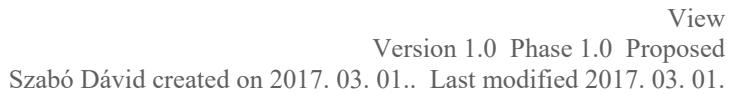
OPERATIONS	
	[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ getPhase () : Phase Public	[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ getPreviouslySelectedTerritory () : int Public	[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ getTerritoryData () : List<Territory> Public	[Is static True. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ isVictory () : boolean Private	[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ loadGame (loadFile : File) : void Public	[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ moveCardView () : void Public	[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ nextPlayerSelected () : void Public	[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ reinforce (territoryID : int , country : String) : void Private	
Properties:	
throws = Exception	
	[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ RiskGameController (stage : Stage) : Public	[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ saveGame (saveFile : File) : void Public	[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ setModel (model : RiskGameModel) : void Public	
Eltárolja a modellt és feliratkoztatja magát az eseményeire	
	[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ showAddPlayerView () : void Public	
Add player view	
	[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ showAttackResultView () : void Private	
	[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]

OPERATIONS	
◆ showAttackView (defenderID : int , attackerID : int) : void Private	
Attack View	[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ showCardView () : void Public	
Main View	[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ showMainView () : void Public	
Transfer View	[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ showTransferView (fromID : int , toID : int) : void Private	
Stage switch Pre -> Main	[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ showVictoryView () : void Public	
Transfer Accepted	[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ switchToMain (map : Map<Color, String>) : void Public	
Transfer Phase Selected	[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ transferAccepted (from : int , to : int , units : int) : void Public	
Update All Territories On Map	[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ transferPhaseSelected () : void Public	
Update Current Deck	[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ UpdateAllTerritoriesOnMap () : void Private	
Update Current Phase	[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ UpdateCurrentDeck () : void Private	
Update Current Player	[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ UpdateCurrentPhase () : void Private	
Update Territory On Map	[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ UpdateCurrentPlayer () : void Private	
Update Territory On Map	[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ UpdateTerritoryOnMap (territoryID : int) : void Private	

OPERATIONS

View

Package in package 'Class Model'



View diagram

Class diagram in package 'View'

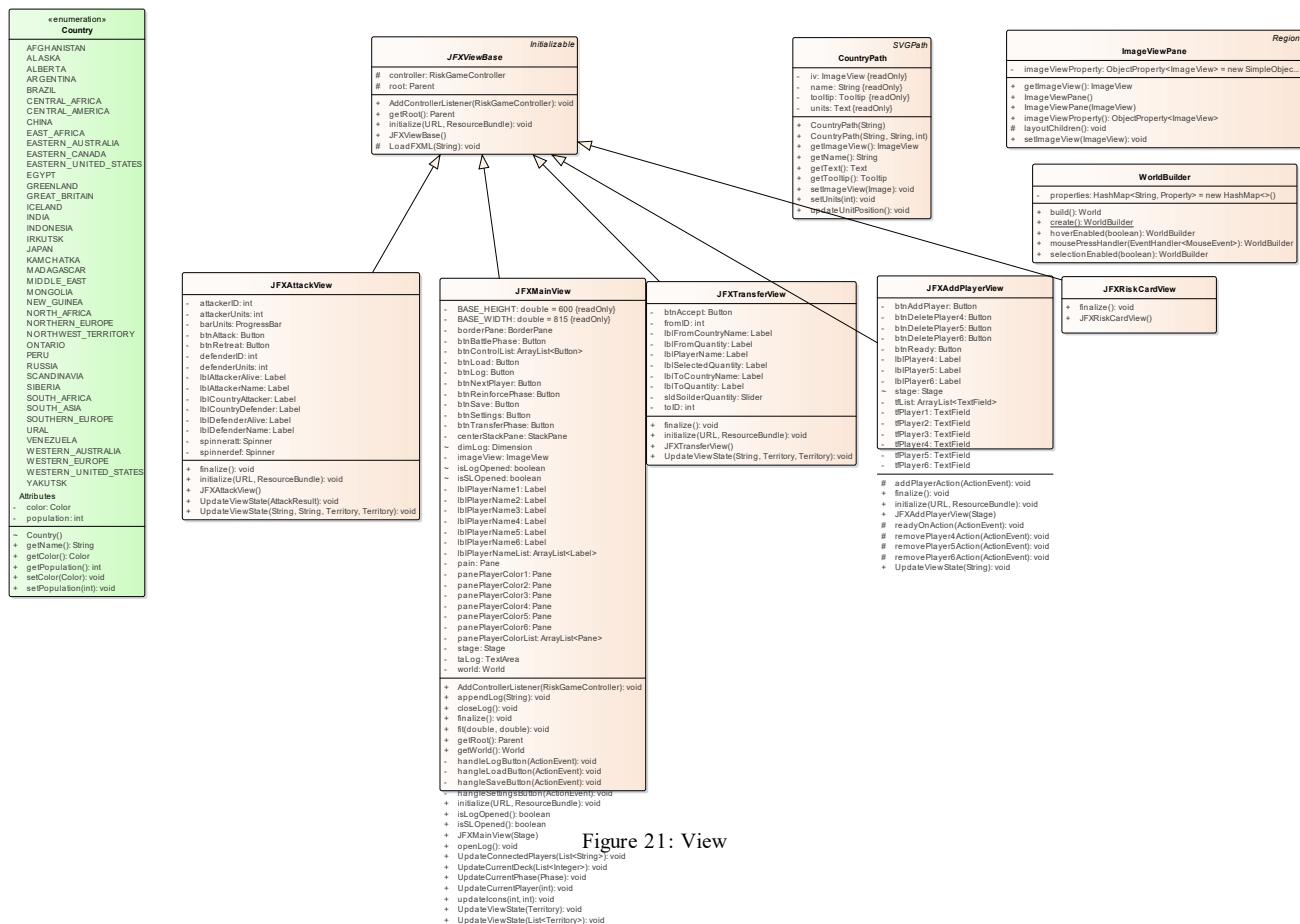


Figure 21: View

CountryPath

Class in package 'View'

SVG for countries



ATTRIBUTES

◆ iv : ImageView Private Const

ATTRIBUTES	
	[Is static False. Containment is Not Specified.]
◆ name : String Private Const	[Is static False. Containment is Not Specified.]
◆ tooltip : Tooltip Private Const	[Is static False. Containment is Not Specified.]
◆ units : Text Private Const	[Is static False. Containment is Not Specified.]
OPERATIONS	
◆ CountryPath (name : String) : Public	[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ CountryPath (name : String, content : String, unit : int) : Public	[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ getImageView () : ImageView Public	[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ getName () : String Public	[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ getText () : Text Public	[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ getTooltip () : Tooltip Public	[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ setImageView (img : Image) : void Public	[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ setUnits (u : int) : void Public	[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ updateUnitPosition () : void Public	[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]

ImageViewPane

Class in package 'View'

ImageViewPane

Version 1.0 Phase 1.0 Proposed

Oliver created on 2017. 09. 21.. Last modified 2017. 11. 22.

Extends Region

ATTRIBUTES

- ◆ imageViewProperty : ObjectProperty<ImageView> Private = new SimpleObjectProperty<ImageView>()

[Is static False. Containment is Not Specified.]

OPERATIONS

- ◆ getImageView () : ImageView Public

[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
- ◆ ImageViewPane () : Public

[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
- ◆ ImageViewPane (imageView : ImageView) : Public

[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
- ◆ imageViewProperty () : ObjectProperty<ImageView> Public

[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
- ◆ layoutChildren () : void Protected

Properties:
 annotations = @Override

[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
- ◆ setImageView (imageView : ImageView) : void Public

[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]

JFXAddPlayerView*Class in package 'View'*

Ezen a View-en keresztul tortenik a leendo jatekosok hozzaadasa a jatekhoz.

JFXAddPlayerView

Version 1.0 Phase 1.0 Proposed

fodorad created on 2017. 03. 22.. Last modified 2017. 11. 22.

Extends JFXViewBase

OUTGOING STRUCTURAL RELATIONSHIPS

- ◀ Aggregation from JFXAddPlayerView to RiskGameController

[Direction is 'Source -> Destination'.]

OUTGOING STRUCTURAL RELATIONSHIPS	
↳ Generalization from JFXAddPlayerView to JFXViewBase	[Direction is 'Source -> Destination'.]
ATTRIBUTES	
◆ btnAddPlayer : Button Private	
Properties:	
annotations = @FXML	[Is static False. Containment is Not Specified.]
◆ btnDeletePlayer4 : Button Private	
Properties:	
annotations = @FXML	[Is static False. Containment is Not Specified.]
◆ btnDeletePlayer5 : Button Private	
Properties:	
annotations = @FXML	[Is static False. Containment is Not Specified.]
◆ btnDeletePlayer6 : Button Private	
Properties:	
annotations = @FXML	[Is static False. Containment is Not Specified.]
◆ btnReady : Button Private	
Properties:	
annotations = @FXML	[Is static False. Containment is Not Specified.]
◆ lblPlayer4 : Label Private	
Properties:	
annotations = @FXML	[Is static False. Containment is Not Specified.]
◆ lblPlayer5 : Label Private	
Properties:	
annotations = @FXML	[Is static False. Containment is Not Specified.]
◆ lblPlayer6 : Label Private	
Properties:	
annotations = @FXML	

ATTRIBUTES	
	[Is static False. Containment is Not Specified.]
◆ stage : Stage Package	[Is static False. Containment is Not Specified.]
◆ tfList : ArrayList<TextField> Private	[Is static False. Containment is Not Specified.]
◆ tfPlayer1 : TextField Private	
Properties:	
annotations = @FXML	[Is static False. Containment is Not Specified.]
◆ tfPlayer2 : TextField Private	
Properties:	
annotations = @FXML	[Is static False. Containment is Not Specified.]
◆ tfPlayer3 : TextField Private	
Properties:	
annotations = @FXML	[Is static False. Containment is Not Specified.]
◆ tfPlayer4 : TextField Private	
Properties:	
annotations = @FXML	[Is static False. Containment is Not Specified.]
◆ tfPlayer5 : TextField Private	
Properties:	
annotations = @FXML	[Is static False. Containment is Not Specified.]
◆ tfPlayer6 : TextField Private	
Properties:	
annotations = @FXML	[Is static False. Containment is Not Specified.]

OPERATIONS	
◆ addPlayerAction (event : ActionEvent) : void Protected	
Properties:	
annotations = @FXML	[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]

OPERATIONS
<p>◆ finalize () : void Public</p> <p>Properties:</p> <p>throws = Throwable annotations = @Override</p> <p>[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]</p>
<p>◆ initialize (location : URL , resources : ResourceBundle) : void Public</p> <p>Properties:</p> <p>annotations = @Override</p> <p>[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]</p>
<p>◆ JFXAddPlayerView (stage : Stage) : Public</p> <p>[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]</p>
<p>◆ readyOnAction (event : ActionEvent) : void Protected</p> <p>Properties:</p> <p>annotations = @FXML</p> <p>[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]</p>
<p>◆ removePlayer4Action (event : ActionEvent) : void Protected</p> <p>Properties:</p> <p>annotations = @FXML</p> <p>[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]</p>
<p>◆ removePlayer5Action (event : ActionEvent) : void Protected</p> <p>Properties:</p> <p>annotations = @FXML</p> <p>[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]</p>
<p>◆ removePlayer6Action (event : ActionEvent) : void Protected</p> <p>Properties:</p> <p>annotations = @FXML</p> <p>[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]</p>
<p>◆ UpdateViewState (name : String) : void Public</p> <p>[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]</p>

JFXAttackView

Class in package 'View'

Ezen a View-en keresztul jelölheti meg az egyik játékos a másik játékos (ellenfel) azon területet, amelyet tamadni szeretne.

JFXAttackView
Version 1.0 Phase 1.0 Proposed
Szabó Dávid created on 2017. 03. 08.. Last modified 2017. 11. 22.
Extends JFXViewBase

OUTGOING STRUCTURAL RELATIONSHIPS

↳ Generalization from JFXAttackView to JFXViewBase [Direction is 'Source -> Destination'.]

↳ Aggregation from JFXAttackView to RiskGameController [Direction is 'Source -> Destination'.]

ATTRIBUTES

◆ attackerID : int Private [Is static False. Containment is Not Specified.]

◆ attackerUnits : int Private [Is static False. Containment is Not Specified.]

◆ barUnits : ProgressBar Private

Properties:

 annotations = @FXML [Is static False. Containment is Not Specified.]

◆ btnAttack : Button Private

Properties:

 annotations = @FXML [Is static False. Containment is Not Specified.]

◆ btnRetreat : Button Private

Properties:

 annotations = @FXML [Is static False. Containment is Not Specified.]

◆ defenderID : int Private

[Is static False. Containment is Not Specified.]

◆ defenderUnits : int Private

[Is static False. Containment is Not Specified.]

◆ lblAttackerAlive : Label Private

Properties:

 annotations = @FXML [Is static False. Containment is Not Specified.]

ATTRIBUTES	
◆ <code>lblAttackerName : Label</code>	Private
Properties:	
<code>annotations = @FXML</code>	[Is static False. Containment is Not Specified.]
◆ <code>lblCountryAttacker : Label</code>	Private
Properties:	
<code>annotations = @FXML</code>	[Is static False. Containment is Not Specified.]
◆ <code>lblCountryDefender : Label</code>	Private
Properties:	
<code>annotations = @FXML</code>	[Is static False. Containment is Not Specified.]
◆ <code>lblDefenderAlive : Label</code>	Private
Properties:	
<code>annotations = @FXML</code>	[Is static False. Containment is Not Specified.]
◆ <code>lblDefenderName : Label</code>	Private
Properties:	
<code>annotations = @FXML</code>	[Is static False. Containment is Not Specified.]
◆ <code>spinneratt : Spinner</code>	Private
Properties:	
<code>annotations = @FXML</code>	[Is static False. Containment is Not Specified.]
◆ <code>spinnerdef : Spinner</code>	Private
Properties:	
<code>annotations = @FXML</code>	[Is static False. Containment is Not Specified.]

OPERATIONS	
◆ <code>finalize () : void</code>	Public
Properties:	
<code>throws = Throwable</code>	
<code>annotations = @Override</code>	
	[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]

OPERATIONS	
◆ initialize (location : URL , resources : ResourceBundle) : void Public	
Properties:	
annotations = @Override	[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ JFXAttackView () : Public	[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ UpdateViewState (attackResult : AttackResult) : void Public	[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ UpdateViewState (defenderName : String , attackerName : String , defender : Territory , attacker : Territory) : void Public	[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]

JFXMainView

Class in package 'View'

Az MVC architektura View részet megvalósító osztálya. Ez a View a játék főkepernyője, ezen történik maga a játék (egysegek elhelyezése, csata stb.).

JFXMainView
Version 1.0 Phase 1.0 Proposed
Szabó Dávid created on 2017. 03. 01.. Last modified 2017. 11. 22.
Extends JFXViewBase

OUTGOING STRUCTURAL RELATIONSHIPS	
↳ Generalization from JFXMainView to JFXViewBase	[Direction is 'Source -> Destination'.]
↳ Aggregation from JFXMainView to RiskGameController	[Direction is 'Source -> Destination'.]

ATTRIBUTES	
◆ BASE_HEIGHT : double Private Const = 600	[Is static False. Containment is Not Specified.]
◆ BASE_WIDTH : double Private Const = 815	[Is static False. Containment is Not Specified.]
◆ borderPane : BorderPane Private	
Properties:	
annotations = @FXML	[Is static False. Containment is Not Specified.]

ATTRIBUTES	
◆ btnBattlePhase : Button	Private Properties: annotations = @FXML [Is static False. Containment is Not Specified.]
◆ btnControlList : ArrayList<Button>	Private [Is static False. Containment is Not Specified.]
◆ btnLoad : Button	Private Properties: annotations = @FXML [Is static False. Containment is Not Specified.]
◆ btnLog : Button	Private Properties: annotations = @FXML [Is static False. Containment is Not Specified.]
◆ btnNextPlayer : Button	Private Properties: annotations = @FXML [Is static False. Containment is Not Specified.]
◆ btnReinforcePhase : Button	Private Properties: annotations = @FXML [Is static False. Containment is Not Specified.]
◆ btnSave : Button	Private Properties: annotations = @FXML [Is static False. Containment is Not Specified.]
◆ btnSettings : Button	Private Properties: annotations = @FXML [Is static False. Containment is Not Specified.]
◆ btnTransferPhase : Button	Private Properties: annotations = @FXML [Is static False. Containment is Not Specified.]

ATTRIBUTES	
◆ centerStackPane : StackPane	Private
Properties:	
annotations = @FXML	[Is static False. Containment is Not Specified.]
◆ dimLog : Dimension	Package
	[Is static False. Containment is Not Specified.]
◆ imageView : ImageView	Private
Properties:	
annotations = @FXML	[Is static False. Containment is Not Specified.]
◆ isLogOpened : boolean	Package
	[Is static False. Containment is Not Specified.]
◆ isSLOpened : boolean	Package
	[Is static False. Containment is Not Specified.]
◆ lblPlayerName1 : Label	Private
Properties:	
annotations = @FXML	[Is static False. Containment is Not Specified.]
◆ lblPlayerName2 : Label	Private
Properties:	
annotations = @FXML	[Is static False. Containment is Not Specified.]
◆ lblPlayerName3 : Label	Private
Properties:	
annotations = @FXML	[Is static False. Containment is Not Specified.]
◆ lblPlayerName4 : Label	Private
Properties:	
annotations = @FXML	[Is static False. Containment is Not Specified.]
◆ lblPlayerName5 : Label	Private
Properties:	
annotations = @FXML	[Is static False. Containment is Not Specified.]

ATTRIBUTES	
◆ lblPlayerName6 : Label	Private
Properties:	
annotations = @FXML	[Is static False. Containment is Not Specified.]
◆ lblPlayerNameList : ArrayList<Label>	Private
	[Is static False. Containment is Not Specified.]
◆ pain : Pane	Private
Properties:	
annotations = @FXML	[Is static False. Containment is Not Specified.]
◆ panePlayerColor1 : Pane	Private
Properties:	
annotations = @FXML	[Is static False. Containment is Not Specified.]
◆ panePlayerColor2 : Pane	Private
Properties:	
annotations = @FXML	[Is static False. Containment is Not Specified.]
◆ panePlayerColor3 : Pane	Private
Properties:	
annotations = @FXML	[Is static False. Containment is Not Specified.]
◆ panePlayerColor4 : Pane	Private
Properties:	
annotations = @FXML	[Is static False. Containment is Not Specified.]
◆ panePlayerColor5 : Pane	Private
Properties:	
annotations = @FXML	[Is static False. Containment is Not Specified.]
◆ panePlayerColor6 : Pane	Private
Properties:	
annotations = @FXML	[Is static False. Containment is Not Specified.]

ATTRIBUTES	
◆ panePlayerColorList : ArrayList<Pane>	Private [Is static False. Containment is Not Specified.]
◆ stage : Stage	Private [Is static False. Containment is Not Specified.]
◆ taLog : TextArea	Private Properties: annotations = @FXML [Is static False. Containment is Not Specified.]
◆ world : World	Private [Is static False. Containment is Not Specified.]

OPERATIONS	
◆ AddControllerListener (controller : RiskGameController) : void	Public
Properties:	
annotations = @Override	[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ appendLog (s : String) : void	Public [Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ closeLog () : void	Public [Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ finalize () : void	Public Properties: throws = Throwable annotations = @Override [Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ fit (height : double, width : double) : void	Public [Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ getRoot () : Parent	Public Properties: annotations = @Override [Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ getWorld () : World	Public [Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]

OPERATIONS	
◆ handleLogButton (event : ActionEvent) : void Private	
Properties:	
annotations = @FXML	[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ handleLoadButton (event : ActionEvent) : void Private	
Properties:	
annotations = @FXML	[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ handleSaveButton (event : ActionEvent) : void Private	
Properties:	
annotations = @FXML	[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ handleSettingsButton (event : ActionEvent) : void Private	
Properties:	
annotations = @FXML	[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ initialize (location : URL , resources : ResourceBundle) : void Public	
Properties:	
annotations = @Override	[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ isLogOpened () : boolean Public	
	[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ isSLOpened () : boolean Public	
	[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ JFXMainView (stage : Stage) : Public	
	[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ openLog () : void Public	
	[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ UpdateConnectedPlayers (playerList : List<String>) : void Public	
	[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ UpdateCurrentDeck (cards : List<Integer>) : void Public	
TODO:	[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]

OPERATIONS	
◆ UpdateCurrentPhase (phase : Phase) : void Public	[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ UpdateCurrentPlayer (playerID : int) : void Public @param player	[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ updateIcons (attacker : int, defender : int) : void Public	[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ UpdateViewState (territory : Territory) : void Public @param territories	Properties: ea_guid = {0E811D72-EC4E-47ad-8B46-C39084EAD8F7} [Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ UpdateViewState (territories : List<Territory>) : void Public	[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]

JFXRiskCardView

Class in package 'View'

JFXRiskCardView
Version 1.0 Phase 1.0 Proposed
Oliver created on 2017. 03. 28.. Last modified 2017. 11. 22.
Extends JFXViewBase

OUTGOING STRUCTURAL RELATIONSHIPS	
↳ Generalization from JFXRiskCardView to JFXViewBase	[Direction is 'Source -> Destination'.]
OPERATIONS	
◆ finalize() : void Public Properties: throws = Throwable annotations = @Override	[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ JFXRiskCardView() : Public	[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]

JFXTransferView

Class in package 'View'

Ezen a View-en valaszthatja ki a jatekos, hogy a sajat teruleten belül hova helyezi az egységeit.

JFXTransferView
Version 1.0 Phase 1.0 Proposed
Szabó Dávid created on 2017. 03. 08.. Last modified 2017. 11. 22.
Extends JFXViewBase

OUTGOING STRUCTURAL RELATIONSHIPS

↳ Generalization from JFXTransferView to JFXViewBase

[Direction is 'Source -> Destination'.]

↳ Aggregation from JFXTransferView to RiskGameController

[Direction is 'Source -> Destination'.]

ATTRIBUTES

◆ btnAccept : Button Private

Properties:

annotations = @FXML

[Is static False. Containment is Not Specified.]

◆ fromID : int Private

[Is static False. Containment is Not Specified.]

◆ lblFromCountryName : Label Private

Properties:

annotations = @FXML

[Is static False. Containment is Not Specified.]

◆ lblFromQuantity : Label Private

Properties:

annotations = @FXML

[Is static False. Containment is Not Specified.]

◆ lblPlayerName : Label Private

Properties:

annotations = @FXML

[Is static False. Containment is Not Specified.]

◆ lblSelectedQuantity : Label Private

Properties:

annotations = @FXML

[Is static False. Containment is Not Specified.]

ATTRIBUTES	
◆ lblToCountryName : Label Private	
Properties:	
annotations = @FXML	[Is static False. Containment is Not Specified.]
◆ lblToQuantity : Label Private	
Properties:	
annotations = @FXML	[Is static False. Containment is Not Specified.]
◆ sldSoilderQuantity : Slider Private	
Properties:	
annotations = @FXML	[Is static False. Containment is Not Specified.]
◆ toID : int Private	[Is static False. Containment is Not Specified.]
OPERATIONS	
◆ finalize () : void Public	
Properties:	
annotations = @Override	
throws = Throwable	[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ initialize (location : URL , resources : ResourceBundle) : void Public	
Properties:	
annotations = @Override	[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ JFXTransferView () : Public	[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ UpdateViewState (player : String , from : Territory , to : Territory) : void Public	[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]

JFXViewBase

Class in package 'View'

JFXViewBase
Version 1.0 Phase 1.0 Proposed

INCOMING STRUCTURAL RELATIONSHIPS

⇒ Generalization from JFXMainView to JFXViewBase

[Direction is 'Source -> Destination'.]

⇒ Generalization from JFXAttackView to JFXViewBase

[Direction is 'Source -> Destination'.]

⇒ Generalization from JFXTransferView to JFXViewBase

[Direction is 'Source -> Destination'.]

⇒ Generalization from JFXRiskCardView to JFXViewBase

[Direction is 'Source -> Destination'.]

⇒ Generalization from JFXAddPlayerView to JFXViewBase

[Direction is 'Source -> Destination'.]

ATTRIBUTES

◆ controller : RiskGameController Protected

[Is static False. Containment is Not Specified.]

◆ root : Parent Protected

[Is static False. Containment is Not Specified.]

ASSOCIATIONS

✍ Association (direction: Source -> Destination)

Source: Public (Class) JFXViewBase

Target: Protected controller (Class)
RiskGameController

OPERATIONS

◆ AddControllerListener (controller : RiskGameController) : void Public

[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]

◆ getRoot () : Parent Public

[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]

◆ initialize (arg0 : URL , arg1 : ResourceBundle) : void Public

Properties:

annotations = @Override

[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]

OPERATIONS	
◆ JFXViewBase () : Public	[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ LoadFXML (resourceName : String) : void Protected	[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]

WorldBuilder

Class in package 'View'

Parent view object builder

WorldBuilder
Version 1.0 Phase 1.0 Proposed
fodorad created on 2017. 09. 21.. Last modified 2017. 11. 22.

ATTRIBUTES	
◆ properties : HashMap<String, Property>	Private = new HashMap<>() [Is static False. Containment is Not Specified.]
OPERATIONS	
◆ build () : World	Public Const [Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ create () : WorldBuilder	Public Const [Is static True. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ hoverEnabled (isEnabled : boolean) : WorldBuilder	Public Const [Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ mousePressHandler (eventHandler : EventHandler<MouseEvent>) : WorldBuilder	Public Const [Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ selectionEnabled (isEnabled : boolean) : WorldBuilder	Public Const [Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]

Country

Enumeration in package 'View'

Enumerator for territories' view objects

Country
Version 1.0 Phase 1.0 Proposed
fodorad created on 2017. 09. 21.. Last modified 2017. 11. 22.

ATTRIBUTES	
◆ AFGHANISTAN : Public	[Stereotype is «enum». Is static False. Containment is Not Specified.]
◆ ALASKA : Public	[Stereotype is «enum». Is static False. Containment is Not Specified.]
◆ ALBERTA : Public	[Stereotype is «enum». Is static False. Containment is Not Specified.]
◆ ARGENTINA : Public	[Stereotype is «enum». Is static False. Containment is Not Specified.]
◆ BRAZIL : Public	[Stereotype is «enum». Is static False. Containment is Not Specified.]
◆ CENTRAL_AFRICA : Public	[Stereotype is «enum». Is static False. Containment is Not Specified.]
◆ CENTRAL_AMERICA : Public	[Stereotype is «enum». Is static False. Containment is Not Specified.]
◆ CHINA : Public	[Stereotype is «enum». Is static False. Containment is Not Specified.]
◆ EAST_AFRICA : Public	[Stereotype is «enum». Is static False. Containment is Not Specified.]
◆ EASTERN_AUSTRALIA : Public	[Stereotype is «enum». Is static False. Containment is Not Specified.]
◆ EASTERN_CANADA : Public	[Stereotype is «enum». Is static False. Containment is Not Specified.]
◆ EASTERN_UNITED_STATES : Public	[Stereotype is «enum». Is static False. Containment is Not Specified.]
◆ EGYPT : Public	[Stereotype is «enum». Is static False. Containment is Not Specified.]
◆ GREENLAND : Public	[Stereotype is «enum». Is static False. Containment is Not Specified.]
◆ GREAT_BRITAIN : Public	

ATTRIBUTES	
	[Stereotype is «enum». Is static False. Containment is Not Specified.]
◆ ICELAND : Public	[Stereotype is «enum». Is static False. Containment is Not Specified.]
◆ INDIA : Public	[Stereotype is «enum». Is static False. Containment is Not Specified.]
◆ INDONESIA : Public	[Stereotype is «enum». Is static False. Containment is Not Specified.]
◆ IRKUTSK : Public	[Stereotype is «enum». Is static False. Containment is Not Specified.]
◆ JAPAN : Public	[Stereotype is «enum». Is static False. Containment is Not Specified.]
◆ KAMCHATKA : Public	[Stereotype is «enum». Is static False. Containment is Not Specified.]
◆ MADAGASCAR : Public	[Stereotype is «enum». Is static False. Containment is Not Specified.]
◆ MIDDLE_EAST : Public	[Stereotype is «enum». Is static False. Containment is Not Specified.]
◆ MONGOLIA : Public	[Stereotype is «enum». Is static False. Containment is Not Specified.]
◆ NEW_GUINEA : Public	[Stereotype is «enum». Is static False. Containment is Not Specified.]
◆ NORTH_AFRICA : Public	[Stereotype is «enum». Is static False. Containment is Not Specified.]
◆ NORTHERN_EUROPE : Public	[Stereotype is «enum». Is static False. Containment is Not Specified.]
◆ NORTHWEST_TERRITORY : Public	[Stereotype is «enum». Is static False. Containment is Not Specified.]
◆ ONTARIO : Public	[Stereotype is «enum». Is static False. Containment is Not Specified.]

ATTRIBUTES	
◆ PERU : Public	[Stereotype is «enum». Is static False. Containment is Not Specified.]
◆ RUSSIA : Public	[Stereotype is «enum». Is static False. Containment is Not Specified.]
◆ SCANDINAVIA : Public	[Stereotype is «enum». Is static False. Containment is Not Specified.]
◆ SIBERIA : Public	[Stereotype is «enum». Is static False. Containment is Not Specified.]
◆ SOUTH_AFRICA : Public	[Stereotype is «enum». Is static False. Containment is Not Specified.]
◆ SOUTH_ASIA : Public	[Stereotype is «enum». Is static False. Containment is Not Specified.]
◆ SOUTHERN_EUROPE : Public	[Stereotype is «enum». Is static False. Containment is Not Specified.]
◆ URAL : Public	[Stereotype is «enum». Is static False. Containment is Not Specified.]
◆ VENEZUELA : Public	[Stereotype is «enum». Is static False. Containment is Not Specified.]
◆ WESTERN_AUSTRALIA : Public	[Stereotype is «enum». Is static False. Containment is Not Specified.]
◆ WESTERN_EUROPE : Public	[Stereotype is «enum». Is static False. Containment is Not Specified.]
◆ WESTERN_UNITED_STATES : Public	[Stereotype is «enum». Is static False. Containment is Not Specified.]
◆ YAKUTSK : Public	[Stereotype is «enum». Is static False. Containment is Not Specified.]
◆ color : Color Private	[Is static False. Containment is Not Specified.]
◆ population : int Private	[Is static False. Containment is Not Specified.]

OPERATIONS	
◆ Country () : Package	[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ getName () : String Public	[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ getColor () : Color Public	[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ getPopulation () : int Public	[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ setColor (color : Color) : void Public	[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ setPopulation (population : int) : void Public	[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]

Model

Package in package 'Class Model'

Model
Version 1.0 Phase 1.0 Proposed
Szabi created on 2017. 02. 21.. Last modified 2017. 03. 01.

Model diagram

Class diagram in package 'Model'

Model
Version 1.0
Szabi created on 2017. 02. 21.. Last modified 2017. 09. 21.

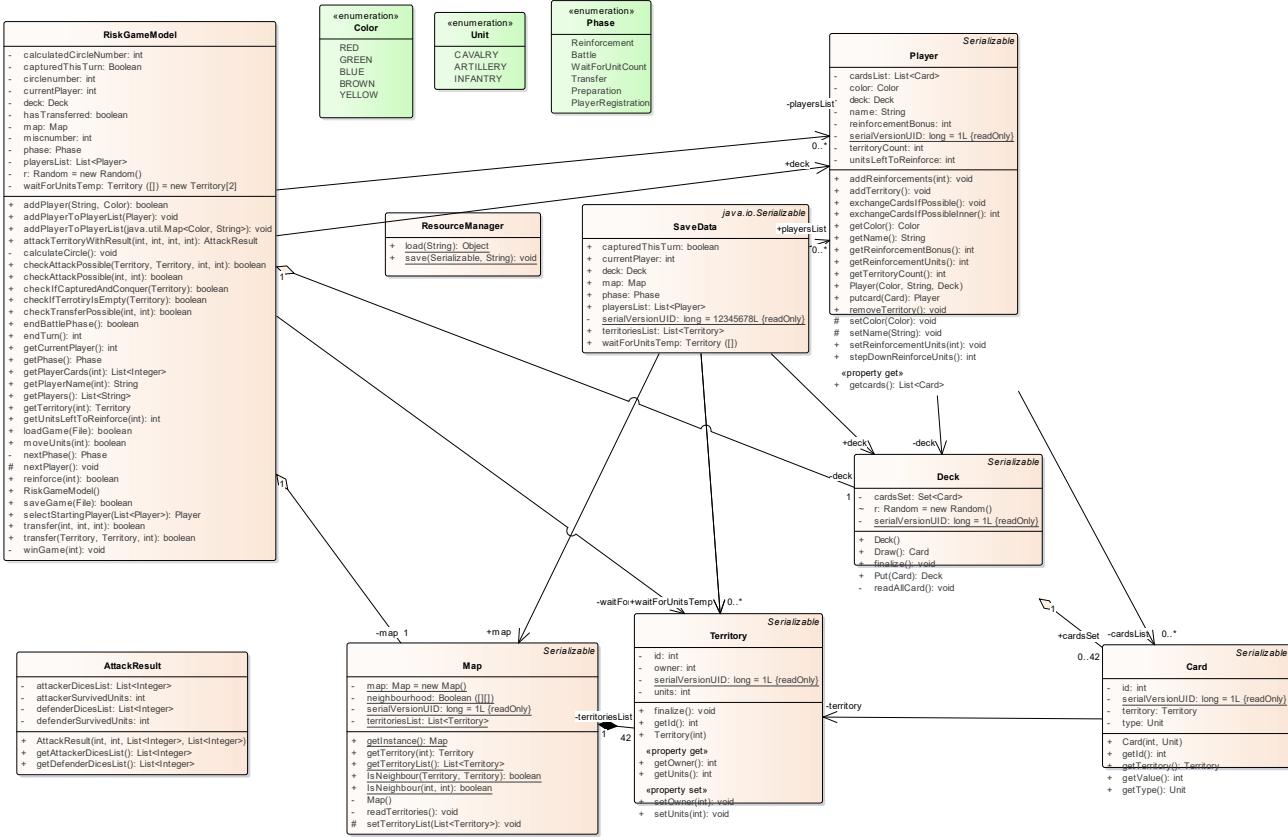


Figure 22: Model

AttackResult

Class in package 'Model'

A tamadas eredményet megadó független.

AttackResult
Version 1.0 Phase 1.0 Proposed
Szabó Dávid created on 2017. 03. 08.. Last modified 2017. 11. 22.

ATTRIBUTES

ATTRIBUTES	
◆ attackerDicesList : List<Integer>	Private [Is static False. Containment is Not Specified.]
◆ attackerSurvivedUnits : int	Private [Is static False. Containment is Not Specified.]
◆ defenderDicesList : List<Integer>	Private [Is static False. Containment is Not Specified.]
◆ defenderSurvivedUnits : int	Private [Is static False. Containment is Not Specified.]

OPERATIONS	
◆ AttackResult (attackerSurvivedUnits : int , defenderSurvivedUnits : int , attackerDicesList : List<Integer> , defenderDicesList : List<Integer>) :	Public [Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ getAttackerDicesList () :	List<Integer> Public [Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ getDefenderDicesList () :	List<Integer> Public [Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]

Card

Class in package 'Model'

Teruletkartyak adatait adja meg.

Card
Version 1.0 Phase 1.0 Proposed
Szabi created on 2017. 02. 28.. Last modified 2017. 11. 22.
Extends Serializable

OUTGOING STRUCTURAL RELATIONSHIPS	
↳ Aggregation from Card to Deck	[Direction is 'Source -> Destination'.]

ATTRIBUTES	
◆ id : int	Private [Is static False. Containment is Not Specified.]
◆ serialVersionUID : long	Private Const = 1L [Is static True. Containment is Not Specified.]

ATTRIBUTES	
◆ territory : Territory Private	[Is static False. Containment is Not Specified.]
◆ type : Unit Private	[Is static False. Containment is Not Specified.]
ASSOCIATIONS	
✓ Association (direction: Source -> Destination)	
Source: Public (Class) Card	Target: Private territory (Class) Territory
✓ Association (direction: Source -> Destination)	
Source: Public (Class) Player	Target: Private cardsList (Class) Card Cardinality: [0..*]
OPERATIONS	
◆ Card (id : int , type : Unit) : Public	[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ getId () : int Public	[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ getTerritory () : Territory Public	[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ getValue () : int Public	Return the card value Converted from the unit enum @return int
Properties: throws = Exception	[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ getType () : Unit Public	[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]

Deck

Class in package 'Model'

Deck

Version 1.0 Phase 1.0 Proposed
Szabi created on 2017. 02. 28.. Last modified 2017. 11. 22.

OUTGOING STRUCTURAL RELATIONSHIPS

↳ Aggregation from Deck to RiskGameModel

[Direction is 'Source -> Destination'.]

INCOMING STRUCTURAL RELATIONSHIPS

⇒ Aggregation from Card to Deck

[Direction is 'Source -> Destination'.]

ATTRIBUTES

◆ cardsSet : Set<Card> Private

[Is static False. Containment is Not Specified.]

◆ r : Random Package = new Random()

[Is static False. Containment is Not Specified.]

◆ serialVersionUID : long Private Const = 1L

[Is static True. Containment is Not Specified.]

ASSOCIATIONS

↙ Association (direction: Source -> Destination)

Source: Public (Class) SaveData

Target: Public deck (Class) Deck

↙ Association (direction: Source -> Destination)

Source: Public (Class) Player

Target: Private deck (Class) Deck

OPERATIONS

◆ Deck () : Public

[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]

◆ Draw () : Card Public

A kihuzott kartya eltunik a paklibol

Properties:

ea_guid = {A9BC1E05-34BF-47e7-B78F-61F3E6E58456}

[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]

◆ finalize () : void Public

Properties:

throws = Throwable

OPERATIONS	
annotations = @Override	[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ Put (card : Card) : Deck Public	
@param kartya kartya	[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ readAllCard () : void Private	
Read cards from territorycards.properties	[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]

Map

Class in package 'Model'

A jatek terkepet adja meg.

Map
 Version 1.0 Phase 1.0 Proposed
 Szabi created on 2017. 02. 21.. Last modified 2017. 11. 22.
 Extends Serializable

ELEMENTS OWNED BY Map	
█ Country : Enumeration	
OUTGOING STRUCTURAL RELATIONSHIPS	
↳ Aggregation from Map to RiskGameModel	[Direction is 'Source -> Destination'.]
INCOMING STRUCTURAL RELATIONSHIPS	
⇒ Aggregation from Territory to Map	[Direction is 'Source -> Destination'.]
ATTRIBUTES	
◆ map : Map Private = new Map()	[Is static True. Containment is Not Specified.]
◆ neighbourhood : Boolean Private	[Is static True. Containment is Not Specified.]
◆ serialVersionUID : long Private Const = 1L	[Is static True. Containment is Not Specified.]

ATTRIBUTES		
<ul style="list-style-type: none"> ◆ territoriesList : List<Territory> Private [Is static True. Containment is Not Specified.] 		
ASSOCIATIONS		
<ul style="list-style-type: none"> ↙ Association (direction: Source -> Destination) <table style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 50%;">Source: Public (Class) SaveData</td> <td style="width: 50%;">Target: Public map (Class) Map</td> </tr> </table>	Source: Public (Class) SaveData	Target: Public map (Class) Map
Source: Public (Class) SaveData	Target: Public map (Class) Map	
OPERATIONS		
<ul style="list-style-type: none"> ◆ getInstance () : Map Public [Is static True. Is abstract False. Is return array False. Is query False. Is synchronized False.] 		
<ul style="list-style-type: none"> ◆ getTerritory (id : int) : Territory Public [Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.] 		
<ul style="list-style-type: none"> ◆ getTerritoryList () : List<Territory> Public [Is static True. Is abstract False. Is return array False. Is query False. Is synchronized False.] 		
<ul style="list-style-type: none"> ◆ IsNeighbour (first : Territory , second : Territory) : boolean Public @param else @param masodik Properties: ea_guid = {E82D9C40-3357-491d-AF89-8DCD3370D3A2} [Is static True. Is abstract False. Is return array False. Is query False. Is synchronized False.] 		
<ul style="list-style-type: none"> ◆ IsNeighbour (fst : int , snd : int) : boolean Public [Is static True. Is abstract False. Is return array False. Is query False. Is synchronized False.] 		
<ul style="list-style-type: none"> ◆ Map () : Private TODO what is this number even [Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.] 		
<ul style="list-style-type: none"> ◆ readTerritories () : void Private [Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.] 		
<ul style="list-style-type: none"> ◆ setTerritoryList (territoriesList : List<Territory>) : void Protected [Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.] 		

Country

Enumeration owned by 'Map', in package 'Model'

Tmp solution. Sorry. I'm tired.

Country
Version 1.0 Phase 1.0 Proposed
Oliver created on 2017. 05. 18.. Last modified 2017. 11. 22.

ATTRIBUTES	
◆ AFGHANISTAN : Public	[Stereotype is «enum». Is static False. Containment is Not Specified.]
◆ ALASKA : Public	[Stereotype is «enum». Is static False. Containment is Not Specified.]
◆ ALBERTA : Public	[Stereotype is «enum». Is static False. Containment is Not Specified.]
◆ ARGENTINA : Public	[Stereotype is «enum». Is static False. Containment is Not Specified.]
◆ BRAZIL : Public	[Stereotype is «enum». Is static False. Containment is Not Specified.]
◆ CENTRAL_AFRICA : Public	[Stereotype is «enum». Is static False. Containment is Not Specified.]
◆ CENTRAL_AMERICA : Public	[Stereotype is «enum». Is static False. Containment is Not Specified.]
◆ CHINA : Public	[Stereotype is «enum». Is static False. Containment is Not Specified.]
◆ EAST_AFRICA : Public	[Stereotype is «enum». Is static False. Containment is Not Specified.]
◆ EASTERN_AUSTRALIA : Public	[Stereotype is «enum». Is static False. Containment is Not Specified.]
◆ EASTERN_CANADA : Public	[Stereotype is «enum». Is static False. Containment is Not Specified.]
◆ EASTERN_UNITED_STATES : Public	[Stereotype is «enum». Is static False. Containment is Not Specified.]
◆ EGYPT : Public	[Stereotype is «enum». Is static False. Containment is Not Specified.]

ATTRIBUTES
◆ GREENLAND : Public [Stereotype is «enum». Is static False. Containment is Not Specified.]
◆ GREAT_BRITAIN : Public [Stereotype is «enum». Is static False. Containment is Not Specified.]
◆ ICELAND : Public [Stereotype is «enum». Is static False. Containment is Not Specified.]
◆ INDIA : Public [Stereotype is «enum». Is static False. Containment is Not Specified.]
◆ INDONESIA : Public [Stereotype is «enum». Is static False. Containment is Not Specified.]
◆ IRKUTSK : Public [Stereotype is «enum». Is static False. Containment is Not Specified.]
◆ JAPAN : Public [Stereotype is «enum». Is static False. Containment is Not Specified.]
◆ KAMCHATKA : Public [Stereotype is «enum». Is static False. Containment is Not Specified.]
◆ MADAGASCAR : Public [Stereotype is «enum». Is static False. Containment is Not Specified.]
◆ MIDDLE_EAST : Public [Stereotype is «enum». Is static False. Containment is Not Specified.]
◆ MONGOLIA : Public [Stereotype is «enum». Is static False. Containment is Not Specified.]
◆ NEW_GUINEA : Public [Stereotype is «enum». Is static False. Containment is Not Specified.]
◆ NORTH_AFRICA : Public [Stereotype is «enum». Is static False. Containment is Not Specified.]
◆ NORTHERN_EUROPE : Public [Stereotype is «enum». Is static False. Containment is Not Specified.]
◆ NORTHWEST_TERRITORY : Public [Stereotype is «enum». Is static False. Containment is Not Specified.]

ATTRIBUTES
◆ ONTARIO : Public [Stereotype is «enum». Is static False. Containment is Not Specified.]
◆ PERU : Public [Stereotype is «enum». Is static False. Containment is Not Specified.]
◆ RUSSIA : Public [Stereotype is «enum». Is static False. Containment is Not Specified.]
◆ SCANDINAVIA : Public [Stereotype is «enum». Is static False. Containment is Not Specified.]
◆ SIBERIA : Public [Stereotype is «enum». Is static False. Containment is Not Specified.]
◆ SOUTH_AFRICA : Public [Stereotype is «enum». Is static False. Containment is Not Specified.]
◆ SOUTH_ASIA : Public [Stereotype is «enum». Is static False. Containment is Not Specified.]
◆ SOUTHERN_EUROPE : Public [Stereotype is «enum». Is static False. Containment is Not Specified.]
◆ URAL : Public [Stereotype is «enum». Is static False. Containment is Not Specified.]
◆ VENEZUELA : Public [Stereotype is «enum». Is static False. Containment is Not Specified.]
◆ WESTERN_AUSTRALIA : Public [Stereotype is «enum». Is static False. Containment is Not Specified.]
◆ WESTERN_EUROPE : Public [Stereotype is «enum». Is static False. Containment is Not Specified.]
◆ WESTERN_UNITED_STATES : Public [Stereotype is «enum». Is static False. Containment is Not Specified.]
◆ YAKUTSK : Public [Stereotype is «enum». Is static False. Containment is Not Specified.]

Player

Class in package 'Model'

A jatekos adatait adja meg.

Player
Version 1.0 Phase 1.0 Proposed
Szabi created on 2017. 02. 28.. Last modified 2017. 11. 22.
Extends Serializable

ATTRIBUTES

◆ cardsList : List<Card> Private
[Is static False. Containment is Not Specified.]

◆ color : Color Private
[Is static False. Containment is Not Specified.]

◆ deck : Deck Private
[Is static False. Containment is Not Specified.]

◆ name : String Private
[Is static False. Containment is Not Specified.]

◆ reinforcementBonus : int Private
Az előző körben beváltott kártyákból kapott bónusz, kézben tartott egységek száma
[Is static False. Containment is Not Specified.]

◆ serialVersionUID : long Private Const = 1L
[Is static True. Containment is Not Specified.]

◆ territoryCount : int Private
[Is static False. Containment is Not Specified.]

◆ unitsLeftToReinforce : int Private
[Is static False. Containment is Not Specified.]

ASSOCIATIONS

◆ Association (direction: Source -> Destination)

Source: Public (Class) Player

Target: Private cardsList (Class) Card
Cardinality: [0..*]

◆ Association (direction: Source -> Destination)

Source: Public (Class) Player

Target: Private deck (Class) Deck

ASSOCIATIONS	
 Association (direction: Source -> Destination)	
Source: Public (Class) SaveData	Target: Public playersList (Class) Player Cardinality: [0..*]
 Association (direction: Source -> Destination)	
Source: Public (Class) RiskGameModel	Target: Private playersList (Class) Player Cardinality: [0..*]
 Association (direction: Source -> Destination)	
Source: Public (Class) RiskGameModel	Target: Public deck (Class) Player
OPERATIONS	
 addReinforcements (troopCount : int) : void Public	
Hozzáadja a játékos következő körö erősítéséhez a beváltott kártyák értékét [Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]	
 addTerritory () : void Public	
Properties: ea_guid = {A36E97BD-9B28-4914-B74A-B242A054F681} [Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]	
 exchangeCardsIfPossible () : void Public	
Ha a játékosnak van 3 beváltható kártyája, ezeket elveszi tőle, majd a kártyákhoz tartozó erősítési értékkel tér vissza. Ha a beváltás nem sikeres, 0- val tér vissza.	
Properties: throws = Exception ea_guid = {8D5F72A0-C149-46e6-A2E7-D07DF6181D8B} [Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]	
 exchangeCardsIfPossibleInner () : int Public	
Ha a játékosnak van 3 beváltható kártyája, ezeket elveszi tőle, majd a kártyákhoz tartozó erősítési értékkel tér vissza. Ha a beváltás nem sikeres, 0- val tér vissza.	
Properties: throws = Exception [Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]	
 getcards () : List<Card> Public	
Properties: attribute_name = cards [Stereotype is «property get». Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]	

OPERATIONS	
◆ getColor () : Color	Public [Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ getName () : String	Public [Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ getReinforcementBonus () : int	Public [Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ getReinforcementUnits () : int	Public [Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ getTerritoryCount () : int	Public [Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ Player (color : Color , name : String , deck : Deck) : Public	 [Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ putcard (card : Card) : Player	Public Properties: ea_guid = {6D4E7A74-6F1A-40bd-94E6-623AD2DCE824} [Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ removeTerritory () : void	Public Properties: ea_guid = {451707C0-2406-4c42-981F-C36F9629E3A4} [Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ setColor (color : Color) : void	Protected [Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ setName (name : String) : void	Protected [Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ setReinforcementUnits (unit : int) : void	Public [Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ stepDownReinforceUnits () : int	Public [Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]

ResourceManager

Class in package 'Model'

A ResourceManager feladata a szerializált adat beolvasása és kiírása

ResourceManager

Version 1.0 Phase 1.0 Proposed

Oliver created on 2017. 09. 21.. Last modified 2017. 11. 22.

OPERATIONS

◆ load (fileName : String) : Object Public

Beolvasás metódus, univerzális

Properties:

throws = Exception

[Is static True. Is abstract False. Is return array False. Is query False. Is synchronized False.]

◆ save (data : Serializable , fileName : String) : void Public

Mentes metódus, univerzális

Properties:

throws = Exception

[Is static True. Is abstract False. Is return array False. Is query False. Is synchronized False.]

RiskGameModel

Class in package 'Model'

Az MVC architektura Model részét megvalósító osztalya.

RiskGameModel

Version 1.0 Phase 1.0 Proposed

Szabi created on 2017. 02. 21.. Last modified 2017. 11. 22.

OUTGOING STRUCTURAL RELATIONSHIPS

↳ Aggregation from RiskGameModel to RiskGameController

[Direction is 'Source -> Destination'.]

INCOMING STRUCTURAL RELATIONSHIPS

⇒ Aggregation from Map to RiskGameModel

[Direction is 'Source -> Destination'.]

⇒ Aggregation from Deck to RiskGameModel

[Direction is 'Source -> Destination'.]

ATTRIBUTES

◆ calculatedCircleNumber : int Private

[Is static False. Containment is Not Specified.]

ATTRIBUTES	
◆ capturedThisTurn : Boolean	Private [Is static False. Containment is Not Specified.]
◆ circlenumber : int	Private [Is static False. Containment is Not Specified.]
◆ currentPlayer : int	Private [Is static False. Containment is Not Specified.]
◆ deck : Deck	Private [Is static False. Containment is Not Specified.]
◆ hasTransferred : boolean	Private [Is static False. Containment is Not Specified.]
◆ map : Map	Private [Is static False. Containment is Not Specified.]
◆ miscnumber : int	Private [Is static False. Containment is Not Specified.]
◆ phase : Phase	Private [Is static False. Containment is Not Specified.]
◆ playersList : List<Player>	Private [Is static False. Containment is Not Specified.]
◆ r : Random	Private = new Random() [Is static False. Containment is Not Specified.]
◆ waitForUnitsTemp : Territory	Private = new Territory[2] [Is static False. Containment is Not Specified.]
ASSOCIATIONS	
✓ Association (direction: Source -> Destination)	Source: Public (Class) RiskGameModel Target: Private waitForUnitsTemp (Class) Territory
✓ Association (direction: Source -> Destination)	Source: Public (Class) RiskGameModel Target: Private playersList (Class) Player Cardinality: [0..*]
✓ Association (direction: Source -> Destination)	

ASSOCIATIONS	
Source: Public (Class) RiskGameModel	Target: Public deck (Class) Player
OPERATIONS	
<p>◆ addPlayer (name : String , color : Color) : boolean Public</p> <p>Add player with name and color @return boolean</p> <p>[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]</p>	
<p>◆ addPlayerToPlayerList (player : Player) : void Public</p> <p>Adder for playerList</p> <p>Properties: throws = Exception ea_guid = {45684F69-D727-4875-BEF7-5232B4E9D1C9}</p> <p>[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]</p>	
<p>◆ addPlayerToPlayerList (map : java.util.Map<Color, String>) : void Public</p> <p>Pairs overload</p> <p>Properties: throws = Exception</p> <p>[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]</p>	
<p>◆ attackTerritoryWithResult (defenderID : int , attackerID : int , defendUnits : int , attackUnits : int) : AttackResult Public</p> <p>Return the result of the attack If the attacker have more than 3 units, the fighting attacker units will reduce to 3 If the defender have more than 2 units, the fighting defender units will reduce to 2</p> <p>Properties: throws = Exception</p> <p>[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]</p>	
<p>◆ calculateCircle () : void Private</p> <p>Properties: throws = Exception</p> <p>[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]</p>	
<p>◆ checkAttackPossible (defender : Territory , attacker : Territory , defendUnits : int , attackUnits : int) : boolean Public</p> <p>Csak akkor tĂˇmadhat, ha a terĂĽleten elegendL' egysĂ©ge van. Igazzal tĂ�r vissza, ha a tamadas lehetsĂ©ges. @return boolean</p> <p>[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]</p>	
<p>◆ checkAttackPossible (defenderID : int , attackerID : int) : boolean Public</p> <p>Attack checking method @return</p> <p>[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]</p>	

OPERATIONS
<p>◆ checkIfCapturedAndConquer (defender : Territory) : boolean Public</p> <p>Capture and conquer checking method @return boolean</p> <p>Properties: ea_guid = {BFF4F3B6-2538-4778-9372-EAA3B468EDD8} [Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]</p>
<p>◆ checkIfTerritoryIsEmpty (territory : Territory) : boolean Public</p> <p>Empty territory checking method @return boolean</p> <p>Properties: ea_guid = {BE4F4E07-A4C4-4e91-9F00-EF1EEB65D26C} [Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]</p>
<p>◆ checkTransferPossible (from : int, to : int) : boolean Public</p> <p>Transfer checking method @return boolean [Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]</p>
<p>◆ endBattlePhase () : boolean Public</p> <p>Battle @return boolean</p> <p>Properties: throws = Exception [Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]</p>
<p>◆ endTurn () : int Public</p> <p>Kor vege Returns next player's id @return int</p> <p>Properties: throws = Exception ea_guid = {F5F8D5D6-8997-43de-A5AD-DDDC647E775F} [Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]</p>
<p>◆ getCurrentPlayer () : int Public</p> <p>Getter for current player @return int [Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]</p>
<p>◆ getPhase () : Phase Public</p> <p>Getter for Phase @return phase</p>

OPERATIONS	
[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]	
<p>◆ getPlayerCards (id : int) : List<Integer> Public [Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]</p>	
<p>◆ getPlayerName (id : int) : String Public Getter for Player's name <code>@return String</code> [Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]</p>	
<p>◆ getPlayers () : List<String> Public Getter for Players list <code>@return List<Player></code> [Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]</p>	
<p>◆ getTerritory (id : int) : Territory Public Getter for Territory <code>@return Territory</code> [Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]</p>	
<p>◆ getUnitsLeftToReinforce (playerID : int) : int Public Getter for left units in reinforce phase <code>@return int</code> [Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]</p>	
<p>◆ loadGame (file : File) : boolean Public Load data method <code>@return boolean</code> Properties: <code>throws = Exception</code> [Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]</p>	
<p>◆ moveUnits (units : int) : boolean Public Move units method <code>@return boolean</code> Properties: <code>throws = Exception</code> [Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]</p>	
<p>◆ nextPhase () : Phase Private Calculate the next phase <code>@return Phase</code> Properties:</p>	

OPERATIONS	
throws = Exception	[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ nextPlayer () : void Protected	Initializes the next Player's turn. Calculates the amount of Units to place.
Properties:	
throws = Exception	
ea_guid = {806D8A15-D81F-4852-9BFA-E0988D68A2D8}	[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ reinforce (territoryId : int) : boolean Public	Reinforce method
@return boolean	
Properties:	
throws = Exception	
	[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ RiskGameModel () : Public	The amount of Units left to Place in the Reinforcement Stage of the current Player's current turn.
	[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ saveGame (file : File) : boolean Public	Save data method
@return	
Properties:	
throws = IOException	
	[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ selectStartingPlayer (list : List<Player>) : Player Public	Random select the starting player
@return Player	
	[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ transfer (fromID : int, toID : int, units : int) : boolean Public	@return boolean
Properties:	
throws = Exception	
	[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ transfer (from : Territory, to : Territory, units : int) : boolean Public	Transfer method
@return boolean	

OPERATIONS	
Properties:	
throws = Exception	
ea_guid = {23034260-4F89-40e8-BC43-DD7B2D6023D5}	[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ winGame (winner : int) : void Private	[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]

SaveData

Class in package 'Model'

SaveData
Version 1.0 Phase 1.0 Proposed
Oliver created on 2017. 09. 21.. Last modified 2017. 11. 22.
Extends java.io.Serializable

ATTRIBUTES	
◆ capturedThisTurn : boolean Public	[Is static False. Containment is Not Specified.]
◆ currentPlayer : int Public	[Is static False. Containment is Not Specified.]
◆ deck : Deck Public	[Is static False. Containment is Not Specified.]
◆ map : Map Public	[Is static False. Containment is Not Specified.]
◆ phase : Phase Public	[Is static False. Containment is Not Specified.]
◆ playersList : List<Player> Public	[Is static False. Containment is Not Specified.]
◆ serialVersionUID : long Private Const = 12345678L	[Is static True. Containment is Not Specified.]
◆ territoriesList : List<Territory> Public	[Is static False. Containment is Not Specified.]
◆ waitForUnitsTemp : Territory Public	[Is static False. Containment is Not Specified.]

ASSOCIATIONS	
Association (direction: Source -> Destination)	
Source: Public (Class) SaveData	Target: Public waitForUnitsTemp (Class) Territory
Association (direction: Source -> Destination)	
Source: Public (Class) SaveData	Target: Public playersList (Class) Player Cardinality: [0..*]
Association (direction: Source -> Destination)	
Source: Public (Class) SaveData	Target: Public deck (Class) Deck
Association (direction: Source -> Destination)	
Source: Public (Class) SaveData	Target: Public territoriesList (Class) Territory Cardinality: [0..*]
Association (direction: Source -> Destination)	
Source: Public (Class) SaveData	Target: Public map (Class) Map

Territory

Class in package 'Model'

Az adott terulet adatait adja meg.

Territory
Version 1.0 Phase 1.0 Proposed
Szabi created on 2017. 02. 21.. Last modified 2017. 11. 22.
Extends Serializable

OUTGOING STRUCTURAL RELATIONSHIPS	
Aggregation from Territory to Map	[Direction is 'Source -> Destination'.]

ATTRIBUTES	
id : int Private	[Is static False. Containment is Not Specified.]
owner : int Private	[Is static False. Containment is Not Specified.]
serialVersionUID : long Private Const = 1L	[Is static True. Containment is Not Specified.]

ATTRIBUTES
<p>◆ units : int Private Constraints: Pozitiv : Invariant [Is static False. Containment is Not Specified.]</p>

ASSOCIATIONS
<p>↙ Association (direction: Source -> Destination) Source: Public (Class) SaveData Target: Public waitForUnitsTemp (Class) Territory</p>
<p>↙ Association (direction: Source -> Destination) Source: Public (Class) RiskGameModel Target: Private waitForUnitsTemp (Class) Territory</p>
<p>↙ Association (direction: Source -> Destination) Source: Public (Class) Card Target: Private territory (Class) Territory</p>
<p>↙ Association (direction: Source -> Destination) Source: Public (Class) SaveData Target: Public territoriesList (Class) Territory Cardinality: [0..*]</p>

OPERATIONS
<p>◆ finalize () : void Public Properties: throws = Throwable annotations = @Override [Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]</p>
<p>◆ getId () : int Public [Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]</p>
<p>◆ getOwner () : int Public Properties: attribute_name = owner ea_guid = {9C292B7F-DE75-4544-B341-0329C012F2A4} [Stereotype is «property get». Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]</p>
<p>◆ getUnits () : int Public Properties: attribute_name = units ea_guid = {B42ACE21-7893-47fa-95D0-7D42512D2D87}</p>

OPERATIONS	
[Stereotype is «property get». Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]	
◆ setOwner (newVal : int) : void Public	Properties: attribute_name = owner ea_guid = {2C8215FC-D727-46c6-80B7-AFEAFD60919E} [Stereotype is «property set». Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ setUnits (newVal : int) : void Public	Properties: attribute_name = units ea_guid = {517BD20B-8541-478a-A260-A03B6F3B6029} [Stereotype is «property set». Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ Territory (id : int) : Public	[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]

Color

Enumeration in package 'Model'

A szín adatokat adja meg.

Color

Version 1.0 Phase 1.0 Proposed

Szabi created on 2017. 02. 28.. Last modified 2017. 11. 22.

ATTRIBUTES	
◆ RED : Public	[Stereotype is «enum». Is static False. Containment is Not Specified.]
◆ GREEN : Public	[Stereotype is «enum». Is static False. Containment is Not Specified.]
◆ BLUE : Public	[Stereotype is «enum». Is static False. Containment is Not Specified.]
◆ BROWN : Public	[Stereotype is «enum». Is static False. Containment is Not Specified.]
◆ YELLOW : Public	[Stereotype is «enum». Is static False. Containment is Not Specified.]

Phase

Enumeration in package 'Model'

Phase
Version 1.0 Phase 1.0 Proposed
Oliver created on 2017. 09. 21.. Last modified 2017. 11. 22.

ATTRIBUTES	
◆ Reinforcement :	Public [Stereotype is «enum». Is static False. Containment is Not Specified.]
◆ Battle :	Public [Stereotype is «enum». Is static False. Containment is Not Specified.]
◆ WaitForUnitCount :	Public Terulet elfoglalasa utan meg kell varni az egysegek szamat, amiket az ujonnan elfoglalt teruletre akarunk mozgatni. [Stereotype is «enum». Is static False. Containment is Not Specified.]
◆ Transfer :	Public [Stereotype is «enum». Is static False. Containment is Not Specified.]
◆ Preparation :	Public [Stereotype is «enum». Is static False. Containment is Not Specified.]
◆ PlayerRegistration :	Public [Stereotype is «enum». Is static False. Containment is Not Specified.]

Unit

Enumeration in package 'Model'

Az egység adatait adja meg.

Unit
Version 1.0 Phase 1.0 Proposed
Szabi created on 2017. 02. 28.. Last modified 2017. 11. 22.

ATTRIBUTES	
◆ CAVALRY :	Public [Stereotype is «enum». Is static False. Containment is Not Specified.]
◆ ARTILLERY :	Public [Stereotype is «enum». Is static False. Containment is Not Specified.]
◆ INFANTRY :	Public [Stereotype is «enum». Is static False. Containment is Not Specified.]

User Interface Model

Package in package 'Model'

User Interface Model
Version Phase 1.0 Proposed
fodorad created on 2017. 03. 01.. Last modified 2017. 03. 01.

User Interface Model diagram

Custom diagram in package 'User Interface Model'

User Interface Model
Version 1.0
fodorad created on 2017. 03. 01.. Last modified 2017. 03. 01.

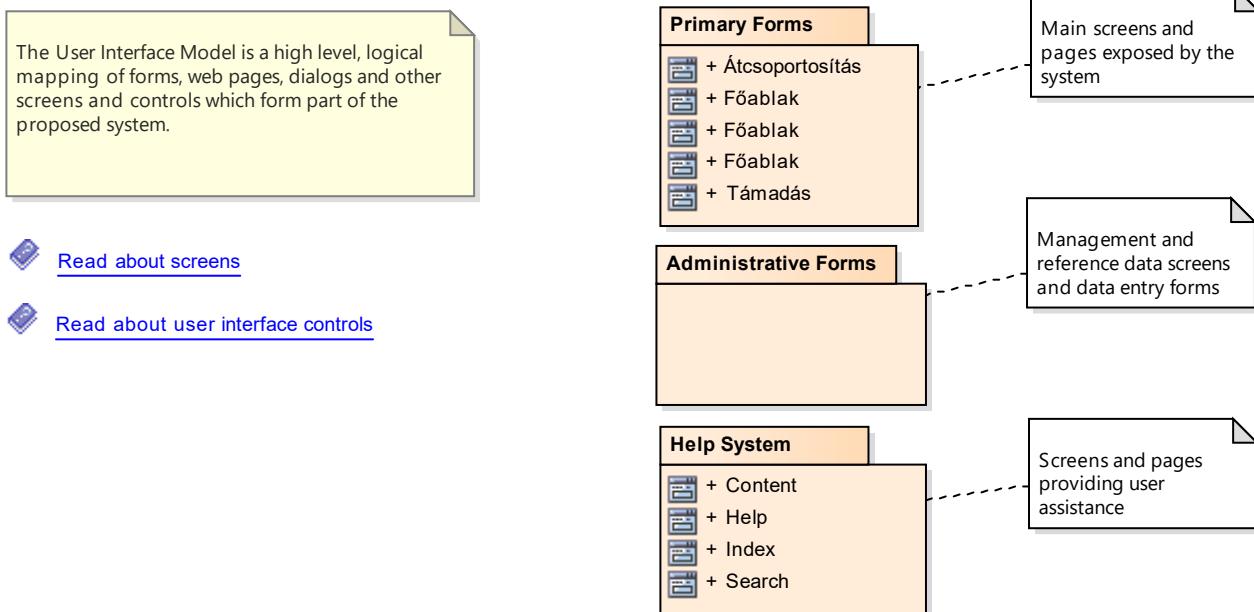


Figure 23: User Interface Model

<anonymous>

Note in package 'User Interface Model'

Screens and pages providing user assistance

<anonymous>
Version 1.0 Phase 1.0 Proposed
fodorad created on 2017. 03. 01.. Last modified 2017. 03. 01.
Extends

<anonymous>

Note in package 'User Interface Model'

The User Interface Model is a high level, logical mapping of forms, web pages, dialogs and other screens and controls which form part of the proposed system.

<anonymous>
Version 1.0 Phase 1.0 Proposed

fodorad created on 2017. 03. 01.. Last modified 2017. 03. 01.
Extends

<anonymous>

Note in package 'User Interface Model'

Management and reference data screens and data entry forms

<anonymous>
Version 1.0 Phase 1.0 Proposed
fodorad created on 2017. 03. 01.. Last modified 2017. 03. 01.
Extends

<anonymous>

Note in package 'User Interface Model'

Main screens and pages exposed by the system

<anonymous>
Version 1.0 Phase 1.0 Proposed
fodorad created on 2017. 03. 01.. Last modified 2017. 03. 01.
Extends

\$help://screen.htm

Text in package 'User Interface Model'

\$help://screen.htm
Version 1.0 Phase 1.0 Proposed
fodorad created on 2017. 03. 01.. Last modified 2017. 03. 01.
Alias Read about screens
Extends

\$help://user_interface_group.htm

Text in package 'User Interface Model'

\$help://user_interface_group.htm
Version 1.0 Phase 1.0 Proposed
fodorad created on 2017. 03. 01.. Last modified 2017. 03. 01.
Alias Read about user interface controls
Extends

Primary Forms

Package in package 'User Interface Model'

Primary Forms
Version 1.0 Phase 1.0 Proposed
fodorad created on 2017. 03. 01.. Last modified 2017. 03. 01.

Administrative Forms

Package in package 'User Interface Model'

Administrative Forms
Version 1.0 Phase 1.0 Proposed
fodorad created on 2017. 03. 01.. Last modified 2017. 03. 01.

Help System

Package in package 'User Interface Model'

Help System
Version 1.0 Phase 1.0 Proposed
fodorad created on 2017. 03. 01.. Last modified 2017. 03. 01.

Primary Forms

Package in package 'User Interface Model'

Primary Forms
Version 1.0 Phase 1.0 Proposed
fodorad created on 2017. 03. 01.. Last modified 2017. 03. 01.

Primary Forms diagram

Custom diagram in package 'Primary Forms'

Primary Forms
Version 1.0
fodorad created on 2017. 03. 01.. Last modified 2017. 03. 01.

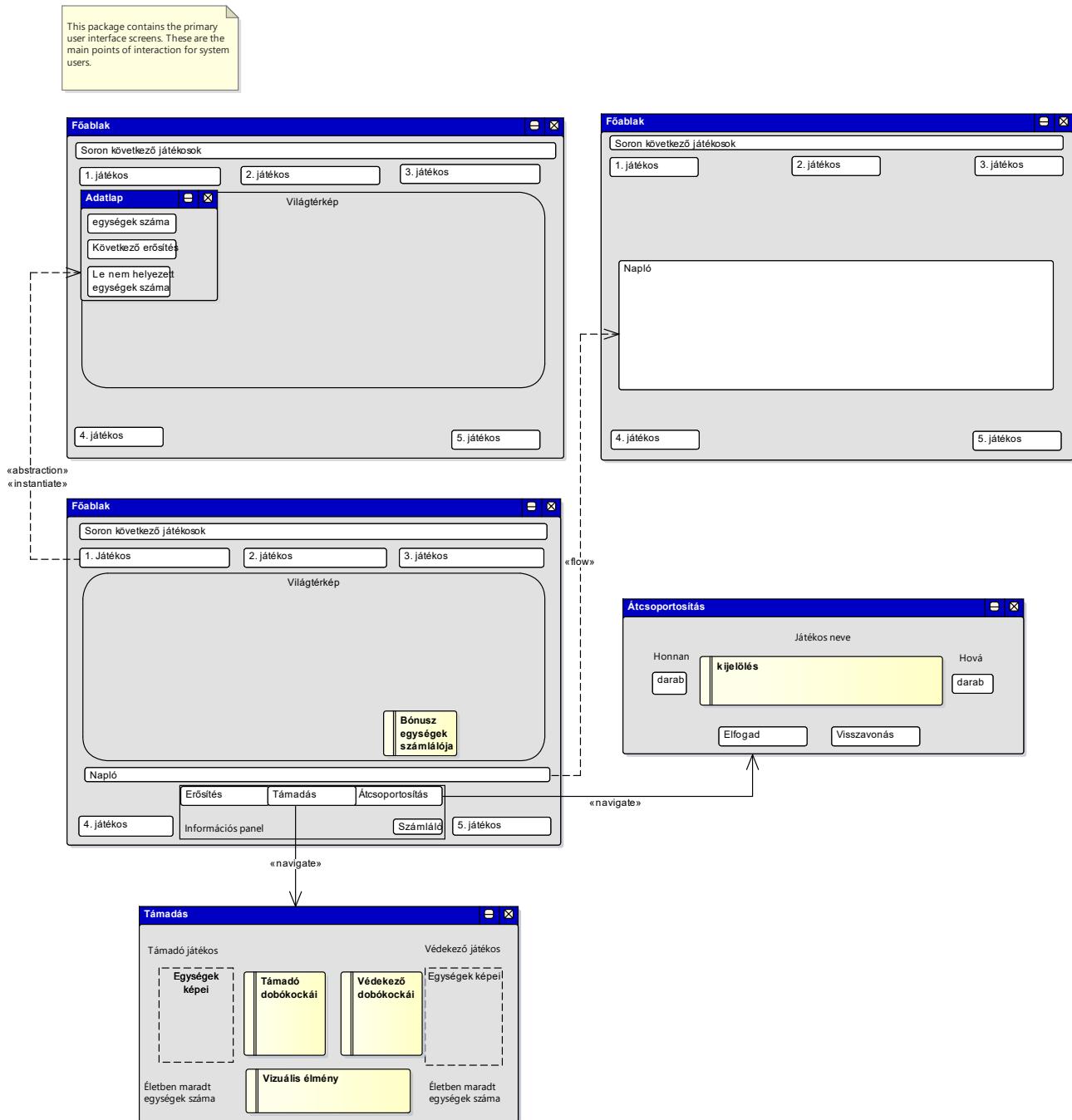


Figure 24: Primary Forms

<anonymous>

Boundary in package 'Primary Forms'

<anonymous>
Version 1.0 Phase 1.0 Proposed
fodorad created on 2017. 03. 01.. Last modified 2017. 03. 01.
Extends

Egységek képei

Boundary in package 'Primary Forms'

Egységek képei
Version 1.0 Phase 1.0 Proposed

fodorad created on 2017. 03. 01.. Last modified 2017. 03. 01.
Extends

Egységek képei

Boundary in package 'Primary Forms'

Egységek képei
Version 1.0 Phase 1.0 Proposed
fodorad created on 2017. 03. 01.. Last modified 2017. 03. 01.
Extends

Világtérkép

Boundary in package 'Primary Forms'

Világtérkép
Version 1.0 Phase 1.0 Proposed
fodorad created on 2017. 03. 01.. Last modified 2017. 03. 01.
Extends

Világtérkép

Boundary in package 'Primary Forms'

Világtérkép
Version 1.0 Phase 1.0 Proposed
fodorad created on 2017. 03. 01.. Last modified 2017. 03. 01.
Extends

<anonymous>

Note in package 'Primary Forms'

This package contains the primary user interface screens. These are the main points of interaction for system users.

<anonymous>
Version 1.0 Phase 1.0 Proposed
fodorad created on 2017. 03. 01.. Last modified 2017. 03. 01.
Extends

<anonymous>

Text in package 'Primary Forms'

Információs panel

<anonymous>
Version 1.0 Phase 1.0 Proposed
fodorad created on 2017. 03. 01.. Last modified 2017. 03. 01.
Extends

<anonymous>

Text in package 'Primary Forms'

Játékos neve

<anonymous>
Version 1.0 Phase 1.0 Proposed
fodorad created on 2017. 03. 01.. Last modified 2017. 03. 01.
Extends

<anonymous>

Text in package 'Primary Forms'

Honnan

<anonymous>
Version 1.0 Phase 1.0 Proposed
fodorad created on 2017. 03. 01.. Last modified 2017. 03. 01.
Extends

<anonymous>

Text in package 'Primary Forms'

Hová

<anonymous>
Version 1.0 Phase 1.0 Proposed
fodorad created on 2017. 03. 01.. Last modified 2017. 03. 01.
Extends

<anonymous>

Text in package 'Primary Forms'

Támadó játékos

<anonymous>
Version 1.0 Phase 1.0 Proposed
fodorad created on 2017. 03. 01.. Last modified 2017. 03. 01.
Extends

<anonymous>

Text in package 'Primary Forms'

Védekező játékos

<anonymous>
Version 1.0 Phase 1.0 Proposed
fodorad created on 2017. 03. 01.. Last modified 2017. 03. 01.
Extends

<anonymous>

Text in package 'Primary Forms'

Életben maradt egységek száma

<anonymous>
Version 1.0 Phase 1.0 Proposed
fodorad created on 2017. 03. 01.. Last modified 2017. 03. 01.
Extends

<anonymous>

Text in package 'Primary Forms'

Életben maradt egységek száma

<anonymous>
Version 1.0 Phase 1.0 Proposed
fodorad created on 2017. 03. 01.. Last modified 2017. 03. 01.
Extends

Főablak

Screen in package 'Primary Forms'

Főablak
Version 1.0 Phase 1.0 Proposed
fodorad created on 2017. 03. 01.. Last modified 2017. 03. 08.
Extends

1. játékos

GUIElement owned by 'Főablak', in package 'Primary Forms'

1. játékos
Version 1.0 Phase 1.0 Proposed
fodorad created on 2017. 03. 01.. Last modified 2017. 03. 01.
Extends

2. játékos

GUIElement owned by 'Főablak', in package 'Primary Forms'

2. játékos
Version 1.0 Phase 1.0 Proposed
fodorad created on 2017. 03. 01.. Last modified 2017. 03. 01.
Extends

3. játékos

GUIElement owned by 'Főablak', in package 'Primary Forms'

3. játékos
Version 1.0 Phase 1.0 Proposed
fodorad created on 2017. 03. 01.. Last modified 2017. 03. 01.
Extends

4. játékos

GUIElement owned by 'Főablak', in package 'Primary Forms'

4. játékos
Version 1.0 Phase 1.0 Proposed
fodorad created on 2017. 03. 01.. Last modified 2017. 03. 01.
Extends

5. játékos

GUIElement owned by 'Főablak', in package 'Primary Forms'

5. játékos
Version 1.0 Phase 1.0 Proposed
fodorad created on 2017. 03. 01.. Last modified 2017. 03. 01.
Extends

Napló

GUIElement owned by 'Főablak', in package 'Primary Forms'

Napló
Version 1.0 Phase 1.0 Proposed
fodorad created on 2017. 03. 01.. Last modified 2017. 03. 01.
Extends

Soron következő játékosok

GUIElement owned by 'Főablak', in package 'Primary Forms'

Soron következő játékosok
Version 1.0 Phase 1.0 Proposed
fodorad created on 2017. 03. 01.. Last modified 2017. 03. 01.
Extends

Főablak

Screen in package 'Primary Forms'

Főablak
Version 1.0 Phase 1.0 Proposed
fodorad created on 2017. 03. 01.. Last modified 2017. 03. 08.
Extends

1. játékos

GUIElement owned by 'Főablak', in package 'Primary Forms'

1. játékos
Version 1.0 Phase 1.0 Proposed
fodorad created on 2017. 03. 01.. Last modified 2017. 03. 01.
Extends

2. játékos

GUIElement owned by 'Főablak', in package 'Primary Forms'

2. játékos
Version 1.0 Phase 1.0 Proposed
fodorad created on 2017. 03. 01.. Last modified 2017. 03. 01.
Extends

3. játékos

GUIElement owned by 'Főablak', in package 'Primary Forms'

3. játékos
Version 1.0 Phase 1.0 Proposed
fodorad created on 2017. 03. 01.. Last modified 2017. 03. 01.
Extends

4. játékos

GUIElement owned by 'Főablak', in package 'Primary Forms'

4. játékos
Version 1.0 Phase 1.0 Proposed
fodorad created on 2017. 03. 01.. Last modified 2017. 03. 01.
Extends

5. játékos

GUIElement owned by 'Főablak', in package 'Primary Forms'

5. játékos
Version 1.0 Phase 1.0 Proposed
fodorad created on 2017. 03. 01.. Last modified 2017. 03. 01.
Extends

Soron következő játékosok

GUIElement owned by 'Főablak', in package 'Primary Forms'

Soron következő játékosok
Version 1.0 Phase 1.0 Proposed
fodorad created on 2017. 03. 01.. Last modified 2017. 03. 01.
Extends

Adatlap

Screen owned by 'Főablak', in package 'Primary Forms'

Adatlap
Version 1.0 Phase 1.0 Proposed
fodorad created on 2017. 03. 01.. Last modified 2017. 03. 01.
Extends

CONNECTORS

 **Abstraction** «instantiate» Source -> Destination

From: 1. Játékos : GUIElement, Public
To: Adatlap : Screen, Public

egységek száma

GUIElement owned by 'Adatlap', in package 'Primary Forms'

egységek száma
Version 1.0 Phase 1.0 Proposed
fodorad created on 2017. 03. 01.. Last modified 2017. 03. 01.
Extends

Következő erősítés

GUIElement owned by 'Adatlap', in package 'Primary Forms'

Következő erősítés
Version 1.0 Phase 1.0 Proposed
fodorad created on 2017. 03. 01.. Last modified 2017. 03. 01.
Extends

Le nem helyezett egységek száma

GUIElement owned by 'Adatlap', in package 'Primary Forms'

Le nem helyezett egységek száma
Version 1.0 Phase 1.0 Proposed
fodorad created on 2017. 03. 01.. Last modified 2017. 03. 01.
Extends

Főablak

Screen in package 'Primary Forms'

Főablak
Version 1.0 Phase 1.0 Proposed
fodorad created on 2017. 03. 01.. Last modified 2017. 03. 01.
Extends

Bónusz egységek számlálója

Change «Functional» owned by 'Főablak', in package 'Primary Forms'

Bónusz egységek számlálója
 Version 1.0 Phase 1.0 Proposed
 fodorad created on 2017. 03. 01.. Last modified 2017. 03. 01.

Change1

GUIElement owned by 'Főablak', in package 'Primary Forms'

Change1
 Version 1.0 Phase 1.0 Proposed
 fodorad created on 2017. 03. 01.. Last modified 2017. 03. 01.

1. Játékos

GUIElement owned by 'Főablak', in package 'Primary Forms'

1. Játékos
 Version 1.0 Phase 1.0 Proposed
 fodorad created on 2017. 03. 01.. Last modified 2017. 03. 01.
 Extends

CONNECTORS

 Abstraction «instantiate» Source -> Destination
 From: 1. Játékos : GUIElement, Public
 To: Adatlap : Screen, Public

2. játékos

GUIElement owned by 'Főablak', in package 'Primary Forms'

2. játékos
 Version 1.0 Phase 1.0 Proposed
 fodorad created on 2017. 03. 01.. Last modified 2017. 03. 01.
 Extends

3. játékos

GUIElement owned by 'Főablak', in package 'Primary Forms'

3. játékos
 Version 1.0 Phase 1.0 Proposed
 fodorad created on 2017. 03. 01.. Last modified 2017. 03. 01.
 Extends

4. játékos

GUIElement owned by 'Főablak', in package 'Primary Forms'

4. játékos
 Version 1.0 Phase 1.0 Proposed
 fodorad created on 2017. 03. 01.. Last modified 2017. 03. 01.
 Extends

5. játékos

GUIElement owned by 'Főablak', in package 'Primary Forms'

5. játékos
Version 1.0 Phase 1.0 Proposed
fodorad created on 2017. 03. 01.. Last modified 2017. 03. 01.
Extends

Átcsoportosítás

GUIElement owned by 'Főablak', in package 'Primary Forms'

Átcsoportosítás
Version 1.0 Phase 1.0 Proposed
fodorad created on 2017. 03. 01.. Last modified 2017. 03. 01.
Extends

Erősítés

GUIElement owned by 'Főablak', in package 'Primary Forms'

Erősítés
Version 1.0 Phase 1.0 Proposed
fodorad created on 2017. 03. 01.. Last modified 2017. 03. 01.
Extends

Napló

GUIElement owned by 'Főablak', in package 'Primary Forms'

Napló
Version 1.0 Phase 1.0 Proposed
fodorad created on 2017. 03. 01.. Last modified 2017. 03. 01.
Extends

Soron következő játékosok

GUIElement owned by 'Főablak', in package 'Primary Forms'

Soron következő játékosok
Version 1.0 Phase 1.0 Proposed
fodorad created on 2017. 03. 01.. Last modified 2017. 03. 01.
Extends

Számláló

GUIElement owned by 'Főablak', in package 'Primary Forms'

Számláló
Version 1.0 Phase 1.0 Proposed
fodorad created on 2017. 03. 01.. Last modified 2017. 03. 01.

Támadás

GUIElement owned by 'Főablak', in package 'Primary Forms'

Támadás
Version 1.0 Phase 1.0 Proposed
fodorad created on 2017. 03. 01.. Last modified 2017. 03. 01.
Extends

UI Control

GUIElement «combobox» owned by 'Főablak', in package 'Primary Forms'

UI Control
Version 1.0 Phase 1.0 Proposed
fodorad created on 2017. 03. 01.. Last modified 2017. 03. 01.
Extends

Támadás

Screen in package 'Primary Forms'

Támadás
Version 1.0 Phase 1.0 Proposed
fodorad created on 2017. 03. 01.. Last modified 2017. 03. 01.
Extends

Támadó dobókockái

Change «Functional» owned by 'Támadás', in package 'Primary Forms'

Támadó dobókockái
Version 1.0 Phase 1.0 Proposed
fodorad created on 2017. 03. 01.. Last modified 2017. 03. 01.

Védekező dobókockái

Change «Functional» owned by 'Támadás', in package 'Primary Forms'

Védekező dobókockái
Version 1.0 Phase 1.0 Proposed
fodorad created on 2017. 03. 01.. Last modified 2017. 03. 01.

Vizuális élmény

Change «Functional» owned by 'Támadás', in package 'Primary Forms'

Vizuális élmény
Version 1.0 Phase 1.0 Proposed
fodorad created on 2017. 03. 01.. Last modified 2017. 03. 01.

Átcsoportosítás

Screen in package 'Primary Forms'

Átcsoportosítás
Version 1.0 Phase 1.0 Proposed
fodorad created on 2017. 03. 01.. Last modified 2017. 03. 01.
Extends

kijelölés

Change «Functional» owned by 'Átcsoportosítás', in package 'Primary Forms'

kijelölés
Version 1.0 Phase 1.0 Proposed
fodorad created on 2017. 03. 01.. Last modified 2017. 03. 01.

darab

GUIElement owned by 'Átcsoportosítás', in package 'Primary Forms'

darab
Version 1.0 Phase 1.0 Proposed
fodorad created on 2017. 03. 01.. Last modified 2017. 03. 01.
Extends

darab

GUIElement owned by 'Átcsoportosítás', in package 'Primary Forms'

darab
Version 1.0 Phase 1.0 Proposed
fodorad created on 2017. 03. 01.. Last modified 2017. 03. 01.
Extends

Elfogad

GUIElement owned by 'Átcsoportosítás', in package 'Primary Forms'

Elfogad
Version 1.0 Phase 1.0 Proposed
fodorad created on 2017. 03. 01.. Last modified 2017. 03. 01.
Extends

UI Control

GUIElement owned by 'Átcsoportosítás', in package 'Primary Forms'

UI Control
Version 1.0 Phase 1.0 Proposed
fodorad created on 2017. 03. 01.. Last modified 2017. 03. 01.
Extends

Visszavonás

GUIElement owned by 'Átcsoportosítás', in package 'Primary Forms'

Visszavonás
Version 1.0 Phase 1.0 Proposed
fodorad created on 2017. 03. 01.. Last modified 2017. 03. 01.
Extends

Administrative Forms

Package in package 'User Interface Model'

Administrative Forms
Version 1.0 Phase 1.0 Proposed
fodorad created on 2017. 03. 01.. Last modified 2017. 03. 01.

Administrative Forms diagram

Custom diagram in package 'Administrative Forms'

Administrative Forms
Version 1.0
Oliver created on 2017. 03. 01.. Last modified 2017. 03. 01.

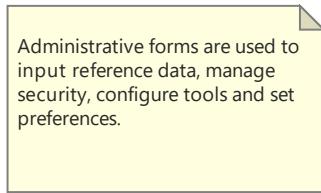


Figure 25: Administrative Forms

<anonymous>

Note in package 'Administrative Forms'

Administrative forms are used to input reference data, manage security, configure tools and set preferences.

<anonymous>
Version 1.0 Phase 1.0 Proposed
fodorad created on 2017. 03. 01.. Last modified 2017. 03. 01.
Extends

Help System

Package in package 'User Interface Model'

Help System
Version 1.0 Phase 1.0 Proposed
fodorad created on 2017. 03. 01.. Last modified 2017. 03. 01.

Help System diagram

Custom diagram in package 'Help System'

Help System
Version 1.0
Oliver created on 2017. 03. 01.. Last modified 2017. 03. 01.

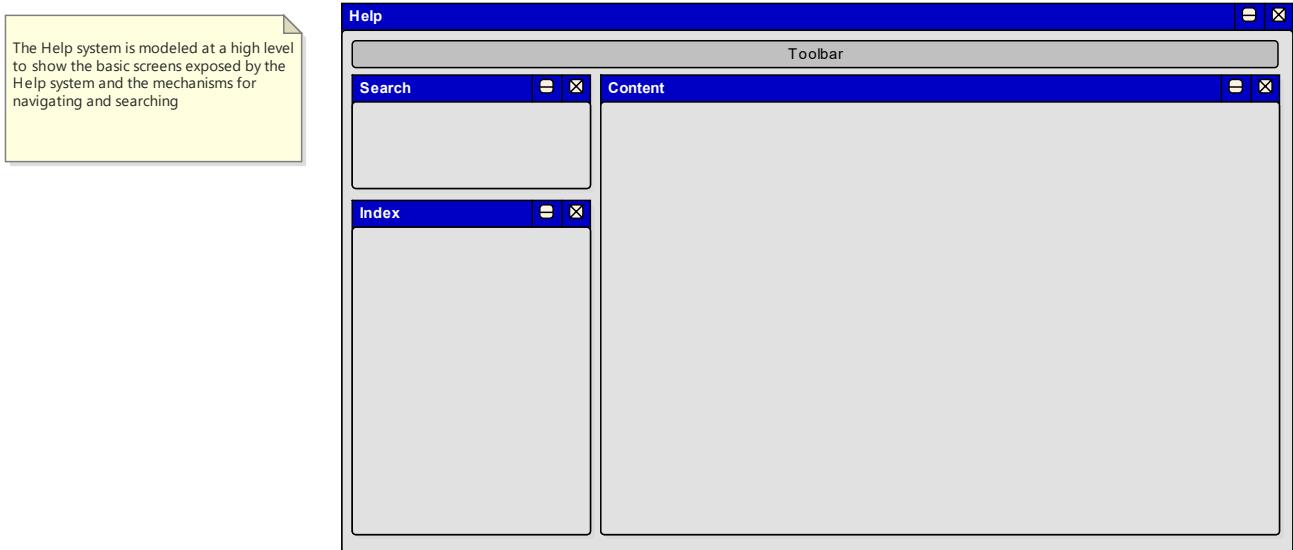


Figure 26: Help System

<anonymous>

Note in package 'Help System'

The Help system is modeled at a high level to show the basic screens exposed by the Help system and the mechanisms for navigating and searching

<anonymous>
Version 1.0 Phase 1.0 Proposed
fodorad created on 2017. 03. 01.. Last modified 2017. 03. 01.
Extends

Content

Screen in package 'Help System'

Content
Version 1.0 Phase 1.0 Proposed
fodorad created on 2017. 03. 01.. Last modified 2017. 03. 01.
Extends

Help

Screen in package 'Help System'

Help
Version 1.0 Phase 1.0 Proposed
fodorad created on 2017. 03. 01.. Last modified 2017. 03. 01.
Extends

Toolbar

GUIElement «button» owned by 'Help', in package 'Help System'

Toolbar
Version 1.0 Phase 1.0 Proposed
fodorad created on 2017. 03. 01.. Last modified 2017. 03. 01.
Extends

Index

Screen in package 'Help System'

Index
Version 1.0 Phase 1.0 Proposed
fodorad created on 2017. 03. 01.. Last modified 2017. 03. 01.
Extends

Search

Screen in package 'Help System'

Search
Version 1.0 Phase 1.0 Proposed
fodorad created on 2017. 03. 01.. Last modified 2017. 03. 01.
Extends