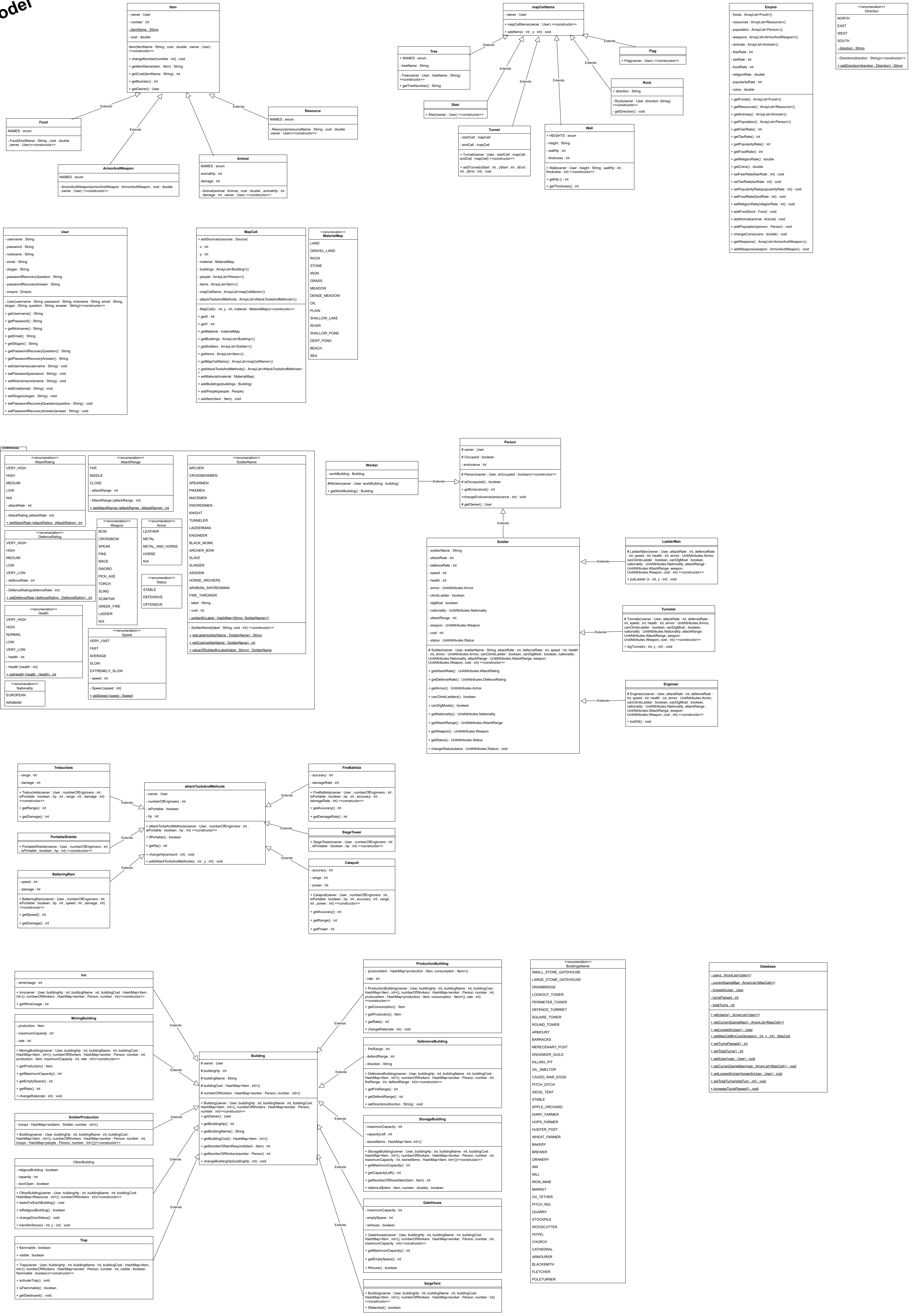


Model



LoginMenu
- controller : LoginMenuController
- scanner : Scanner
+ run() : void
+ enterSignupMenu() : void
+ loginUser(matcher : Matcher) : void
+ forgetPassword(matcher : Matcher) : void
+ isPasswordCorrect(matcher : Matcher) : void

SignupMenu
- controller : SignupMenuController
+ run(scanner : Scanner) : void
+ enterLoginMenu() : void
+ createUser(matcher : Matcher) : void
+ pickQuestion(matcher : Matcher) : void

MainMenu
- controller : MainMenuController
+ run(scanner : Scanner) : void
+ logout() : void
+ enterProfileMenu() : void
+ startNewGame(matcher : Matcher) : void

BuildingMenu
- controller : BuildingMenuController
+ run(scanner : Scanner) : void
+ dropBuilding(matcher : Matcher) : void
+ selectBuilding(matcher : Matcher) : void
+ createUnit(matcher : Matcher) : void
+ repair() : void

TradeMenu
- controller : TradeMenuController
+ run(scanner : Scanner) : void
+ tradeRequest(matcher : Matcher) : void
+ tradeList() : void
+ tradeAccept(matcher : Matcher) : void
+ tradeHistory() : void

ShopMenu
- controller : ShopMenuController
+ run(scanner : Scanner) : void
+ showPriceList() : void
+ buyItem(matcher : Matcher) : void
+ sellItem(matcher : Matcher) : void

ProfileMenu
- controller : ProfileMenuController
+ run(scanner : Scanner) : void
+ changeUsername(matcher : Matcher) : void
+ changeNickname(matcher : Matcher) : void
+ changePassword(matcher : Matcher) : void
+ changeEmail(matcher : Matcher) : void
+ changeSlogan(matcher : Matcher) : void
+ removeSlogan() : void
+ displayHighScore() : void
+ displayRank() : void
+ displaySlogan() : void
+ displayProfile() : void

UnitMenu
- controller : UnitController
+ run(scanner : Scanner) : void
+ selectUnit(matcher : Matcher) : void
+ moveUnit(matcher : Matcher) : void
+ patrolUnit(matcher : Matcher) : void
+ setUnitMood(matcher : Matcher) : void
+ attackEnemy(matcher : Matcher) : void
+ airAttack(matcher : Matcher) : void
+ pourOil(matcher : Matcher) : void
+ digTunnel(matcher : Matcher) : void
+ buildSurroundingEquipment(matcher : Matcher) : void
+ disbandUnit() : void
+ digDitch(matcher : Matcher) : void
+ fillDitch(matcher : Matcher) : void
+ turnOff() : void

GameMenu
- controller : GameMenuController
+ run(scanner : Scanner) : void
+ chooseMap(matcher : Matcher) : void
+ nextTurn() : void
+ showMap(matcher : Matcher) : void
+ enterEmpireMenu() : void
+ enterBuildingMenu() : void
+ enterMapMenu() : void
+ enterUnitMenu() : void
+ enterShopMenu() : void
+ enterTradeMenu() : void

MapMenu
- controller : MapMenuController
+ run(scanner : Scanner) : void
+ showMap(x : int, y : int) : void
+ moveMap(matcher : Matcher) : void
+ showDetails(matcher : Matcher) : void
+ setTutorial(matcher : Matcher) : void
+ clearBlock(matcher : Matcher) : void
+ dropRock(matcher : Matcher) : void
+ dropTree(matcher : Matcher) : void
+ dropBuilding(matcher : Matcher) : void
+ dropUnit(matcher : Matcher) : void

EmpireMenu
- controller : EmpireMenuController
+ run(scanner : Scanner) : void
+ showPopularity() : void
+ showPopularityFactors() : void
+ showFoodList() : void
+ selfFoodRate(matcher : Matcher) : void
+ showFoodRate() : void
+ setTaxRate(matcher : Matcher) : void
+ showTaxRate() : void
+ selfFearRate(matcher : Matcher) : void
+ showFearRate() : void

controller

SignupMenuCotroler
+ enterLoginMenu() : SignupMenuMessages
+ createUser(username : String , password : String , passwordConfirm : String , nickname : String , email : String , slogan : String) : SignupMenuMessages
+ pickQuestion(questionNumber : int , answer : String , answerConfirm : String) : SignupMenuMessages
+ captcha() : String
+ randomPasswordGenerator() : String
+ randomSloganGenerator() : String
+ validEmail(email : String) : boolean
+ validPassword(password : String) : boolean
+ validUsername(username : String) : boolean

GameMenuController
+ showMap(x : int , y : int) : GameMenuMessages
+ createCustomMap(width : int , length : int) : GameMenuMessages
+ nextTurn() : GameMenuMessages
+ applyDamages() : void
+ changePopularity() : void
+ changePopulation() : void
+ giveFood() : void
+ getTax() : void
+ changeResources() : void
+ handleUnemployedPopulation() : void
+ soldiersFight() : void
+ showTaxRate() : int
+ buildingFight() : void
+ chooseMap() : GameMenuMessages

UnitMenuController
- selectedUnit : Unit
+ turnOn() : UnitMenuMessages
+ rfidChn(x : int , y : int) : UnitMenuMessages
+ digOichn(x : int , y : int) : UnitMenuMessages
+ disbandUnit() : UnitMenuMessages
+ buildSurroundingEquipment(equipmentName : String) : UnitMenuMessages
+ digTunnel(x : int , y : int) : UnitMenuMessages
+ pourOil(direction : String) : UnitMenuMessages
+ airAttack(x : int , y : int) : UnitMenuMessages
+ attackEnemy(x : int , y : int) : UnitMenuMessages
+ setUnitMood(x : int , y : int , state : String) : UnitMenuMessages
+ patrolUnit(x1 : int , y1 : int , x2 : int , y2 : int) : UnitMenuMessages
+ moveUnit(x : int , y : int) : UnitMenuMessages
+ selectUnit(x : int , y : int) : UnitMenuMessages
+ getSelectedUnit() : Unit

ShopMenuController
+ sellItem(itemName : String , itemAmount : int) : ShopMenuMessages
+ buyItem(itemName : String , itemAmount : int) : ShopMenuMessages
+ showPriceList() : String

LoginMenuController
+ isPasswordCorrect(matcher : Matcher) : LoginMenuMessages
+ forgetPassword() : LoginMenuMessages
+ loginUser(username : String , password : String) : LoginMenuMessages

MainMenuController
+ startNewGame(users : String , turnsCount : int) : MainMenuMessages

BuildingMenuController
- selectedBuilding : Building
+ repair() : BuildingMenuMessages
+ createUnit(type : String , count : int) : BuildingMenuMessages
+ selectBuilding(x : int , y : int , type : String) : BuildingMenuMessages
+ dropBuilding(x : int , y : int , type : String) : BuildingMenuMessages
+ getSelectedBuilding() : Building

EmpireMenuController
+ showFearRate() : int
+ selfFearRate(rateNumber : int) : EmpireMenuMessages
+ setTaxRate(rateNumber : int) : EmpireMenuMessages
+ showFoodRate() : int
+ selfFoodRate(rateNumber : int) : EmpireMenuMessages
+ showFoodList() : int
+ showPopularityFactors() : String
+ showPopularity() : int

MapMenuController
+ dropUnit(x : int , y : int , type : String , count : int) : MapMenuMessages
+ dropBuilding(x : int , y : int , type : String) : MapMenuMessages
+ dropTree(x : int , y : int , type : String) : MapMenuMessages
+ dropRock(x : int , y : int , direction : String) : MapMenuMessages
+ clearBlock(x : int , y : int) : MapMenuMessages
+ setTexture(x : int , y : int , type : String) : MapMenuMessages
+ setTexture(x1 : int , y1 : int , x2 : int , y2 : int , type : String) : MapMenuMessages
+ showDetails(x : int , y : int) : MapMenuMessages
+ moveMap(directions : String) : MapMenuMessages
+ showMap(x : int , y : int) : MapMenuMessages

ProfileMenuController
+ displayProfile() : String
+ displaySlogan() : ProfileMenuMessages
+ displayRank() : int
+ displayHighScore() : int
+ removeSlogan() : void
+ changeSlogan(slogan : String) : ProfileMenuMessages
+ changeEmail(email : String) : ProfileMenuMessages
+ changePassword(oldPassword : String , newPassword : String) : ProfileMenuMessages
+ changeNickname(nickname : String) : ProfileMenuMessages
+ changeUsername(username : String) : ProfileMenuMessages

TradeMenuController
+ tradeHistory() : String
+ tradeAccept(id : int , message : String) : TradeMenuMessages
+ tradeList() : String
+ tradeRequest(resourceType : String , resourceAmount : int , price : int , message : String) : TradeMenuMessages

Enum

<<enumeration>> SignupMenuCommands
CREATE_USER
ENTER_LOGIN_MENU
+ regex : String
+ SignupMenuCommands(regex : String) : << constructor >>
+ getMatcher(input : String , regex : SignupMenuCommands) : Matcher

<<enumeration>> SignupMenuMessages
SUCCESS
BLANK_FIELD
INVALID_USERNAME
USERNAME_EXISTS
WEAK_PASSWORD
PASSWORD_DoesNT_Match
EMAIL_EXISTS
INVALID_EMAIL

<<enumeration>> EmpireMenuCommands
SHOW_POPULARITY_FACTORS
SHOW_POPULARITY
SHOW_FOOD_LIST
SET_FOOD_RATE
SHOW_FOOD_RATE
SET_TAX_RATE
SHOW_TAX_RATE
SET_FEAR_RATE
SHOW_FEAR_RATE
BACK
+ regex : String
+ EmpireMenuCommands(regex : String) : << constructor >>
+ getMatcher(input : String , regex : ProfileMenuCommands) : Matcher

<<enumeration>> EmpireMenuMessages
SUCCESS
INVALID_NUMBER

<<enumeration>> LoginMenuCommands
USER_LOGIN
FORGET_PASSWORD
ENTER_SIGNUP_MENU
+ regex : String
+ LoginMenuCommands(regex : String) : << constructor >>
+ getMatcher(input : String , regex : LoginMenuCommands) : Matcher

<<enumeration>> LoginMenuMessages
SUCCESS
WRONG_PASSWORD
USERNAME_DoesNT_Exists

<<enumeration>> ProfileMenuCommands
CHANGE_USERNAME
CHANGE_NICKNAME
CHANGE_PASSWORD
CHANGE_EMAIL
CHANGE_SLOGAN
REMOVE_SLOGAN
DISPLAY_HighSCORE
DISPLAY_RANK
DISPLAY_SLOGAN
DISPLAY_PROFILE
BACK
+ regex : String
+ ProfileMenuCommands(regex : String) : << constructor >>
+ getMatcher(input : String , regex : ProfileMenuCommands) : Matcher

<<enumeration>> ProfileMenuMessages
SUCCESS
INCORRECT_PASSWORD
SAME_PASSWORD
WEAK_PASSWORD
PASSWORD_DoesNT_Match
INVALID_EMAIL
SLOGAN_DoesNT_Exist
BLANK_FIELD

<<enumeration>> BuildingMenuCommands
DROP_BUILDING
SELECT_BUILDING
CREATE_UNIT
REPAIR
BACK
+ regex : String
+ BuildingMenuCommands(regex : String) : << constructor >>
+ getMatcher(input : String , regex : ProfileMenuCommands) : Matcher

<<enumeration>> BuildingMenuMessages
SUCCESS
X_OUT_OF_BOUNDS
Y_OUT_OF_BOUNDS
INVALID_TYPE
INAPPROPRITE_TEXTURE
CELL_IS_FULL
CELL_IS_EMPTY
OPPONENT_BUILDING
INVALID_NUMBER
INSUFFICIENT_STORAGE
NOT_ENOUGH_CROWD
INSUFFICIENT_STONE
OPPONENT_SOLDIER_AROUND

<<enumeration>> UnitMenuCommands
SELECT_UNIT
MOVE_UNIT
PETROL_UNIT
SET_UNIT_MODE
ATTACK_ENEMY
AIR_ATTACK
POUR_OIL
DIG_TUNNEL
BUILD_SURROUNDING_EQUIPMENT
DISBAND_UNIT
DIG_DITCH
BURN_OIL
FILL_DITCH
BACK
+ regex : String
+ UnitMenuCommands(regex : String) : << constructor >>
+ getMatcher(input : String , regex : ProfileMenuCommands) : Matcher

<<enumeration>> UnitMenuMessages
SUCCESS
X_OUT_OF_BOUNDS
Y_OUT_OF_BOUNDS
DoesNT_INCLUDE_UNIT
NOT_TRAVERSABLE
DISTANCE_OUT_OF_BOUNDS
INVALID_MODE
INVALID_DIRECTION
ENGINEER_REQUIRED
INVALID_TYPE_OF_SELECTED_UNIT
INSUFFICIENT_SOURCE
INVALID_EQUIPMENT_NAME
INAPPROPRIATE_TEXTURE
OIL_SMELTOR_DoesNT_Exist
OIL_SMELTOR_EMPTY
ATTACK_OPPONENT_UNIT
FILL_OPPONENT_DITCH
NO_UNIT_SELECTED

<<enumeration>> MainMenuCommands
LOGOUT
ENTER_PROFILE_MENU
START_NEW_GAME
+ regex : String
+ MainMenuCommands(regex : String) : << constructor >>
+ getMatcher(input : String , regex : ProfileMenuCommands) : Matcher

<<enumeration>> MainMenuMessages
SUCCESS
USERNAMES_DoesNT_Exist

<<enumeration>> MapMenuCommands
SHOW_MAP
MOVE_MAP
SHOW_DETAILS
SET_TEXTURE
CLEAR_BLOCK
DROP_ROCK
DROP_TREE
DROP_BUILDING
DROP_UNIT
EXIT
+ regex : String
+ MapMenuCommands(regex : String) : << constructor >>
+ getMatcher(input : String , regex : ProfileMenuCommands) : Matcher

<<enumeration>> MapMenuMessages
SUCCESS
X_OUT_OF_BOUNDS
Y_OUT_OF_BOUNDS
INVALID_DIRECTION
INVALID_TYPE
CELL_IS_FULL
INAPPROPRIATE_TEXTURE

<<enumeration>> TradeMenuCommands
TRADE_REQUEST
TRADE_LIST
ACCEPT_TRADE
TRADE_HISTORY
BACK
+ regex : String
+ TradeMenuCommands(regex : String) : << constructor >>
+ getMatcher(input : String , regex : ProfileMenuCommands) : Matcher

<<enumeration>> TradeMenuMessages
SUCCESS
INVALID_RESOURCE_TYPE
INSUFFICIENT_RESOURCE_AMOUNT
INVALID_AMOUNT
INVALID_PRICE
ID_DoesNT_Exists

<<enumeration>> GameMenuCommands
CHOOSE_GAME_MAP
SHOW_MAP
DEFINE_MAP_SIZE
ENTER_EMPIRE_MENU
ENTER_BUILDING_MENU
ENTER_UNIT_MENU
ENTER_TRADE_MENU
ENTER_SHOP_MENU
+ regex : String
+ GameMenuCommands(regex : String) : << constructor >>
+ getMatcher(input : String , regex : ProfileMenuCommands) : Matcher

<<enumeration>> GameMenuMessages
SUCCESS
INVALID_SIZE
INVALID_MAP_NUMBER

<<enumeration>> ShopMenuCommands
SHOW_PRICE_LIST
BUY_ITEM
SELL_ITEM
BACK
+ regex : String
+ ShopMenuCommands(regex : String) : << constructor >>
+ getMatcher(input : String , regex : ProfileMenuCommands) : Matcher

<<enumeration>> ShopMenuMessages
SUCCESS
INVALID_ITEM_NAME
ITEM_DoesNT_Exists
INVALID_ITEM_AMOUNT
NOT_ENOUGH_COIN
NOT_ENOUGH_SPACE