Eötvös Loránd Tudományegyetem

Informatikai Kar

Catan társasjáték

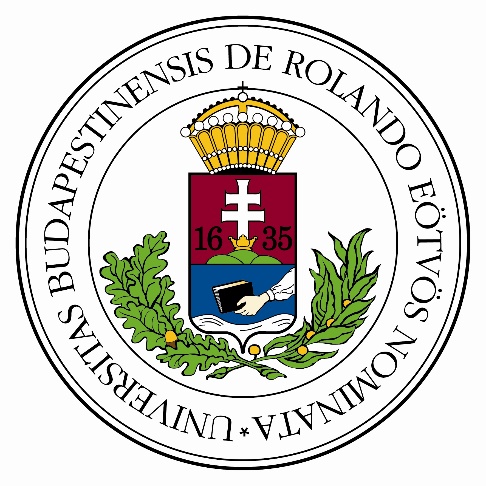
Szakdolgozat

*Készítette: Konzulens:*

Fodor Kristóf XXX

RFXQHM, titulus,

Programtervező informatikus BSc –B szakirány tanszék



Budapest

2023

Tartalomjegyzék

[1. Bevezetés 3](#_Toc135681651)

[2. Felhasználói dokumentáció 4](#_Toc135681652)

[2.1. A játékfelület áttekintése 4](#_Toc135681653)

[2.2. A játék futtatásához szükséges gépigény, programok 4](#_Toc135681654)

[2.3. A játék kezelőfelülete 4](#_Toc135681655)

[2.3.1. Regisztráció és bejelentkezés 4](#_Toc135681656)

[2.3.2. Váróterem 5](#_Toc135681657)

[2.3.3. Játék kezdete 8](#_Toc135681658)

[2.3.4. Egy játékos köre 9](#_Toc135681659)

[2.3.5. Fejlesztéskártyák 11](#_Toc135681660)

[2.3.6. Rabló 12](#_Toc135681661)

[2.3.7. Kereskedés 13](#_Toc135681662)

[2.3.8. Játék vége 14](#_Toc135681663)

[3. Fejlesztői dokumentáció 16](#_Toc135681664)

[3.1. Felhasznált technológiák a fejlesztéshez 16](#_Toc135681665)

[3.1.1. IntelliJ IDEA 16](#_Toc135681666)

[3.1.2. Kotlin 16](#_Toc135681667)

[3.1.3. Ktor 16](#_Toc135681668)

[3.1.4. Kotest 16](#_Toc135681669)

[3.1.5. Gson 16](#_Toc135681670)

[3.1.6. Hibernate 16](#_Toc135681671)

[3.1.7. TornadoFX 17](#_Toc135681672)

[3.1.8. Gradle 17](#_Toc135681673)

[3.2. Áttekintés, programok felépítése rövid leírása 17](#_Toc135681674)

[3.2.1. Kliens 17](#_Toc135681675)

[3.2.2. Szerver 17](#_Toc135681676)

[3.2.3. SDK (Software Development Kit) 17](#_Toc135681677)

[3.3. SDK 17](#_Toc135681678)

[3.3.1. Map osztály 17](#_Toc135681679)

[3.3.2. Map generálása 18](#_Toc135681680)

[3.3.3. Tile osztály 19](#_Toc135681681)

[1.1.1 Osztályok diagramjia 21](#_Toc135681682)

[3.4. Kliens 21](#_Toc135681683)

[3.4.1. Kliens -> Szerver kommunikáció 22](#_Toc135681684)

[3.5. Szerver 24](#_Toc135681685)

[4. Befejezés 25](#_Toc135681686)

# Bevezetés

A mai világban nagyon sokan szeretnek társasjátékozni a gyerekek és a felnőttek körében is. A 2019-ben kitörő COVID-járvány miatt a szociális kapcsolatok megszűntek, mindenki távolságot tartott, bezárkózott otthonába. Ezért sajnos a baráti társaságok, nem egy háztartásban élő rokonok közötti társasjátékos alkalmak száma a pandémia idején gyakorlatilag nullára csökkent. Ebben az időszakban terjedtek el a különböző online felületek, ahol minden biztonságosan a saját otthonából játszhatott különböző társasjátékokkal. Ha a játék során valamilyen platformon videóhívásban vannak a résztvevők, akkor az elég jól meg tudja közelíteni a személyes találkozókat.

A szakdolgozatom témájának a Catan társasjátékot választottam. Ez a játék volt az, amely rabul ejtett, bevezetett a modern társasjátékok világába. A Catan társasjátékot Klaus Teuber tervezte, és 1995-ben adta ki először Németországban a Kosmos cég[[1]](#footnote-1). A világ számos nyelvére lefordították, ezáltal nagyon sok ember által kedvelt játékká vált. Ez bizonyítja az is, hogy több díjat is nyert a játék. Ezek közül a legkiemelkedőbbet rögtön a kiadás évében nyerte el, a társasjáték világ egyik legrangosabb díját, a Spiel des Jahres díjat[[2]](#footnote-2).

A Catan egy olyan stratégiai játék, amely során egy hatszöglapkákból álló lakatlan szigetet próbálnak a játékosok belakni településekkel, városokkal, utakkal oly módon, hogy lehetőleg minél kedvezőbben férjenek hozzá az építkezéshez szükséges nyersanyagokhoz, ezáltal minél több győzelmi pontot szerezzenek.

A szakdolgozatnak nem témája a társasjáték pontos szabályainak ismertetése. Az 5. kiadás hivatalos szabálykönyve[[3]](#footnote-3) alapján készült a program.

TODO Ide szépen le kell írni tömören milyen formában valósítottad meg a programot.

A következőkben először a fejlesztői majd a felhasználói dokumentációt szeretném bemutatni.

# Felhasználói dokumentáció

## A játékfelület áttekintése

A program több játékos számára biztosít lehetőséget a Catan társasjáték játszárára. A program felhasználói felületén a feliratok angol nyelvűek.

A program kezdőoldalán tudnak a játékosok regisztrálni vagy bejelentkezni. A bejelentkezett felhasználók az alábbi lehetőségek válnak elérhetővé:

* asztal készítése,
* rendelkezésre álló asztalok kilistázása,
* elérhető asztalokhoz való csatlakozása,
* kilépés egy olyan asztaltól, amihez a felhasználó csatlakozott, de mégsem kíván részt venni a játékban.

Amint egy asztal adminisztrátora elindítja a játékot, az asztal eltűnik a rendelkezésre álló játékok listájából, és a csatlakozott felhasználók számára elindul a játék.

## A játék futtatásához szükséges gépigény, programok

TODO

## A játék kezelőfelülete

### Regisztráció és bejelentkezés

A képen képernyőkép, szöveg látható

Automatikusan generált leírás

.. ábra Regisztráció és bejelentkezés

A Register gomb megnyomásával a játékosok új felhasználói fiókot tudnak létrehozni. Sikeres regisztráció után azonnal átnavigál a program a bejelentkező felületre.

A Login gombra kattintva a felhasználók a már létező fiókjukba tudnak bejelentkezni. Mindkét felületen a felhasználónév és a jelszó megadása kötelező. TODO Van bármi megkötés a jelszó formájára? Ha valami hiányzik, legyen egy visszajelző üzenet, hogy minden mező kitöltése kötelező. A register felületen pedig, hogy már létező felhasználó.

### Váróterem

Ezen a felületen lehet létrehozni asztalokat, azokhoz csatlakozni, azokból kilépni.

A képen képernyőkép, szöveg, képernyő, Téglalap látható

Automatikusan generált leírás

.. ábra Váróterem

A Refresh gombbal megfrissíthető az elérhető asztalok felülete. Minden nyitott asztalnál látszik, hogy ki az asztal adminisztrátora (korona jelenik meg a neve előtt), illetve az is, hogy mely játékosok csatlakoztak már az asztalhoz. Az alábbi képen az első asztal adminisztrátora Peter, és Sara már csatlakozott a játékhoz. Mary is készített egy játékot, de az ő asztalához még senki sem lépett be.

A képen képernyőkép, szöveg, képernyő, Téglalap látható

Automatikusan generált leírás

.. ábra Elérhető asztal

A Catan alapjátéka 3-4 játékosra szól. Ha már csatlakozott 4 játékos egy asztalhoz, többet már nem engedélyez a játék felülete. TODOLegyen egy visszajelzés valahogy, hogy tele van! Az asztal adminisztrátora nem tudja elindítani a játékot, amíg nem csatlakozik hozzá legalább két játékospartner.

Ha egy játékos csatlakozott egy asztalhoz, látja, hogy rajta kívül kik vannak jelen ennél a játéknál. Amennyiben nem szeretne ebben a játékban részt venni, a Leave gombra kattintva el tudja hagyni az asztalt.

A képen képernyőkép, szöveg, képernyő, számítógép látható

Automatikusan generált leírás

.. ábra Egy játékos csatlakozott Peter asztalához

Az asztal készítője (adminisztrátora) a Start gomb megnyomásával tudja elindítani a játékot.

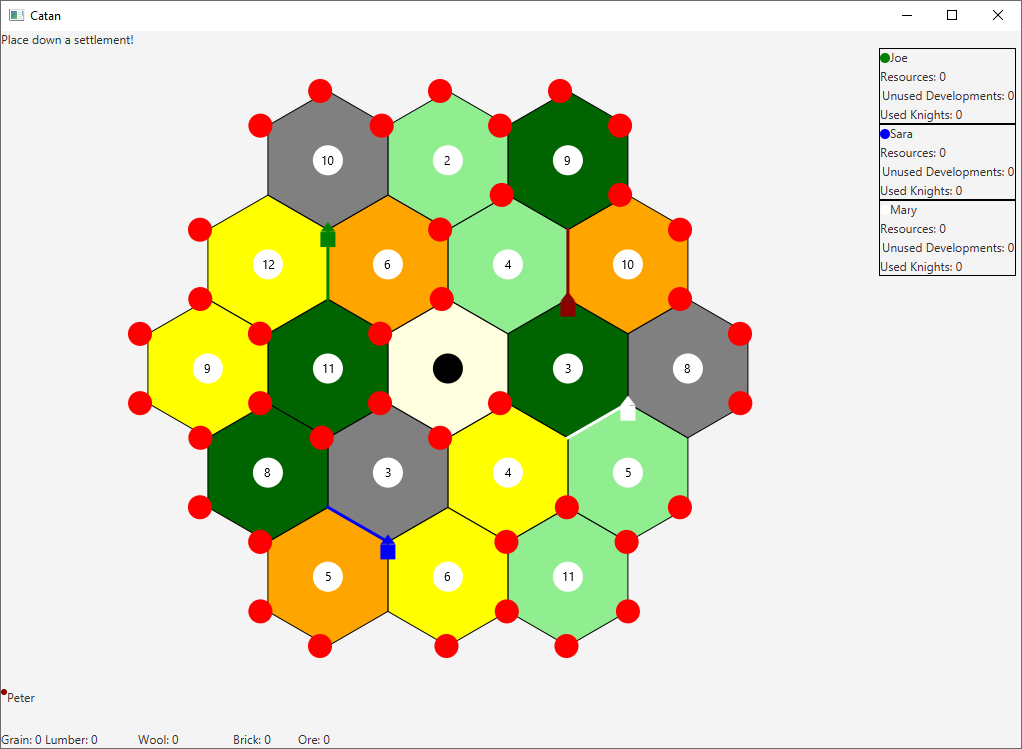
A képen képernyőkép, szöveg, számítógép, szoftver látható

Automatikusan generált leírás

.. ábra Az asztal adminisztrátorának felülete

### Játék kezdete

Miután az asztal tulajdonosa elindítja a játékot, minden felhasználó ablakában megjelenik a játékos tábla. A tábla mezőinek elrendezése a játékszabályban ajánlott módon történik. A kezdőjátékos az asztal adminisztrátora, utána csatlakozási sorrendben következnek a játékosok.



.. ábra Peter lehelyezi a második települését (settlement).

Az ablak bal felső sarkában látható mindig az aktuális játékos számára, hogy mit kell tennie a körében. Jobb oldalon láthatóak a többi játékosról publikus információk: összesen mennyi nyersanyagkártyája (Resources), fel nem használt fejlesztéskártyája (Unused development) és kijátszott lovagkártyája (Knights) van. A saját információit minden játékos a képernyő alján látja.

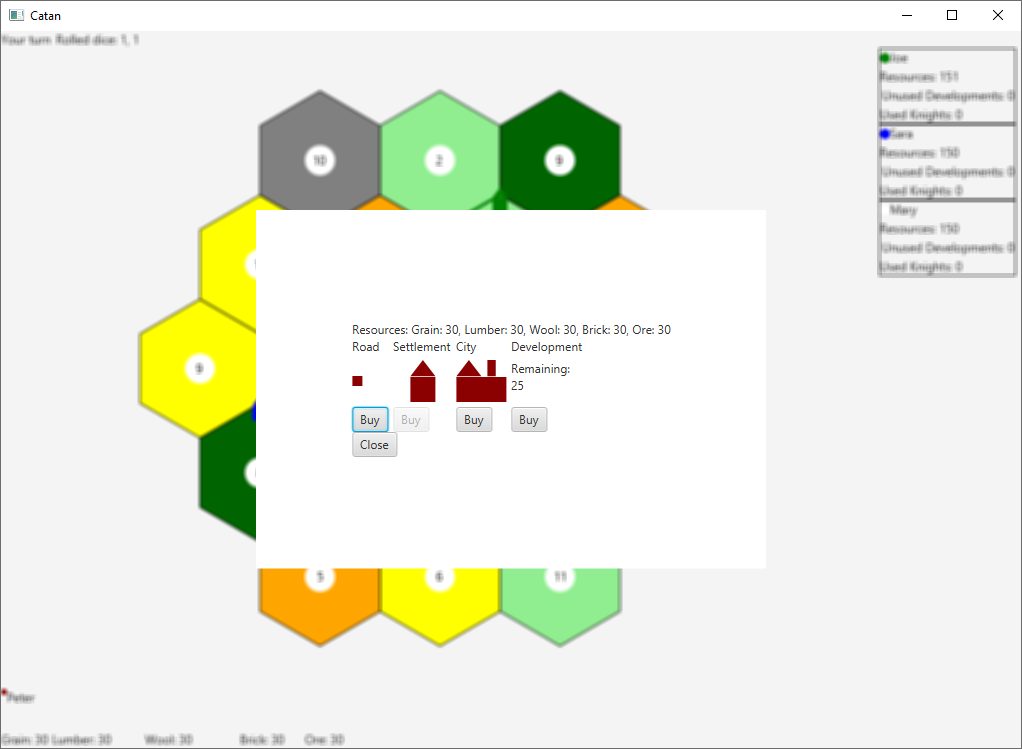
A játék elején minden játékos a játékossorrendnek megfelelően lehelyezi a településeit (settlement) és a hozzájuk kapcsolódó utakat (road). A fenti képen látható módon piros körök jelzik azokat a helyeket, ahova az aktuális játékos teheti a települését.

### Egy játékos köre

Miután minden játékos elhelyezte mindkét települését és útját, elkezdődik a játék. A kör elején automatikusan dob a játék a dobókockával, melynek eredménye a bal felső sarokban játható minden játékos számára. A nyersanyagok a dobás függvényében azonnal kiosztásra kerülnek.

A képernyő alján megjelenő gombokkal tudja az aktív játékos végrehajtani a lépéseit. A Buy gomb megnyomásával vásárolhat utat, települést, fejlesztéskártyát vagy fejlesztheti fel a települését várossá. Az ablak felső sorában látható, hogy mennyi az elérhető nyersanyaga van a játékosnak. Ha valamilyen terméket nem tud megvásárolni a játokos, mert nem tudja lehelyezni, vagy nincs elegendő nyersanyaga rá, akkor a vásárlás gomb nem aktív. Ha út, település vagy város vásárlását választotta a játékos, de mégsem szeretné megtenni, akkor a képernyő nem kiemelt részeire kattintva visszavonhatja a vásárlást.

Ha meggondolta magát, mégsem szeretne vásárolni, akkor a Close gombbal becsukhatja az ablakot.



.. ábra Vásárlás

Ha a játékos úgy határoz, hogy a települését várossá bővíti, akkor a vásárlás ablakban a város alatti Buy gomb megnyomása után a pályán választhatja ki a fejlesztendő települést. A választható településeket szürke kör jelzi.

A képen képernyőkép, diagram, pixel látható

Automatikusan generált leírás

.. ábra Település fejlesztése várossá.

### Fejlesztéskártyák

A megvásárolt fejlesztéskártyákat a Developments gombbal érhetjük el. Minden típusnál jelzi a program, hogy hány jelenleg használható kártyája van a játékosnak, hány használható fel később[[4]](#footnote-4), illetve hányat használt eddig.

A képen szöveg, képernyőkép, diagram, szoftver látható

Automatikusan generált leírás

.. ábra Fejlesztéskártya ablak

A fenti ábrán látható, hogy az aktuális játékosnak van 5 kijátszható lovagkártyája, egyet húzott ebben a fordulóban, és egyet már felhasznált korábbi körökben. Van egy kijátszható Year of Plenty kártyája is, illetve van két pontkártyája, amely a játék végéig rejtve maradnak a többi játékos elől. Amint egy fejlesztéskártyát kijátszik, a Use gombok inaktívvá válnak.

### Rabló

Ha egy játékos a köre elején hetest dob, a rablót köteles elmozdítani. Ehhez a kívánt mező számkorongjára kell kattintani. Ha olyan mezőre lépteti a rablót, amely körül több játékosnak is van települése vagy városa, egy felugró ablakban kiválaszthatja az aktív játékos, hogy kitől kíván véletlenszerűen egy nyersanyagkártyát ellopni.

A képen szöveg, képernyőkép, diagram, szoftver látható

Automatikusan generált leírás

.. ábra Rablás célpontjának kiválasztása.

### Kereskedés

Ha egy játékosnak nincsen elegendő nyersanyaga, akkor a bankkal kereskedhet. A Maritime trade gomb megnyomásával kiválaszthatja, hogy melyik nyersanyagból cserél be négyet egy általa választott másik típusú nyersanyagkártyára. A bal oldali legördülő menüből ki kell választani, miből ad be a játékos négyet a bankba, majd a másodikban meg kell adni mit kér cserébe. A tranzakció a Four to One gomb megnyomásával megy végbe.

A képen képernyőkép, szöveg, diagram, szoftver látható

Automatikusan generált leírás

.. ábra Kereskedés a bankkal.

### Játék vége

A játék azonnal véget ér, ha az aktuális játékos eléri a 10 pontot. Egy felugró ablak jelzi az összes résztvevő számára, hogy a játékot ki nyerte, és visszatérhetnek a váróterembe.

A képen képernyőkép, szöveg, diagram, Grafikai szoftver látható

Automatikusan generált leírás

.. ábra Játék vége

# Fejlesztői dokumentáció

Ez a fejezet a szakdolgozat programjait fogja részletezni fejlesztői szemszögből, hogy megértse a működését.

## Felhasznált technológiák a fejlesztéshez

### IntelliJ IDEA

Az IntelliJ IDEA a JetBrains készítette Java és Kotlin programozók számára készített IDE, ami specifikusan JVM alapú nyelvekhez lett fejlesztve. Képes használni Gradle-t és Maven-t is, van egy beépített debugger.

### Kotlin

A Kotlin egy JVM alapú, statikusan erősen típusos, magas szintű, JetBrains által fejlesztett programozási nyelv. Egyik nagy előnye, hogy egyszerre lehet használni Kotlin-t és Java-t is ugyan abban a kódbázisban, ezért lehetséges az, hogy egy meglévő kódbázis fokozatosan áll át Java-ról Kotlin-ra.

### Ktor

A Ktor az egy aszinkron alapú kommunikációs könyvtár, ami képes http készíteni/kiszolgálni, vagy akár Socket/WebSocketeken keresztül kommunikálni. Ktor-t az elejétől kezdve úgy fejlesztették, hogy Kotlinnal legyen használva, egyszerűen lehet hozzá készíteni egyedi plugin-okat. Ktor segítségével van megoldva a kliens és szerver közti socket kommunikáció.

### Kotest

A Kotest egy teszt keretrendszer Kotlinban írt programok tesztelésére készült. Több féle tesztelési stílust támogat, GIVEN WHEN THEN stílusú teszteket lehet vele készíteni, vagy csak sima egyszerű funckiót futtat le.

### Gson

A Gson egy nyílt forráskódú Java könyvtár, amellyel Java Object-eket lehet Json-nel reprezentálni, és Json által reprezentált objektumot lehet Java Object-é alakítani. Ezt a könyvtárt használtam arra, hogy a kliens és a szerver tudjon egymással kommunikálni.

### Hibernate

A Hibernate egy ORM (Object Relation Mapping) eszköz, ami képes arra, hogy Java object-et egy adatbázisban lévő elemhez rendel. Képes adatbázisba menteni, kiolvasni és változtatni.

### TornadoFX

A TornadoFX egy kotlin wrapper a Java FX-hez, ami megkönnyíti kotlinban fejlesztett ablakos alkalmazások fejlesztését.

### Gradle

TODO

## Áttekintés, programok felépítése rövid leírása

A projekt több különböző alkalmazásból áll mindegyiknek megvan a saját feladata

### Kliens

Egy grafikus felület, ami megjeleníti a szervertől kapott információkat a képernyőt, a felhasználó ezzel kommunikál a szerverrel.

### Szerver

Kiszolgálja a klienst vagy klienseket a tőlük kapott információkkal. A szerverhez egyszerre több kliens is csatlakozhat.

### SDK (Software Development Kit)

A kliens és szerver közös komponenseit tárolja, megkönnyíti a fejlesztés olyan szempontból, hogy közös komponenseket egy helyen kell csak megváltoztatni.

## SDK

Az SDK-nak a feladata, hogy a közös komponenseket tárolja, Kotlin-ban lehetőség van arra, hogy olyan osztályokat használjunk fel másik projektekből. Így nincs szükség ugyanazt az osztályt több különböző helyen deklarálni, ezzel csökkentve a kódbázis méretét, és így nem keletkezhet olyan hiba, hogy a kliens és a szerver máshogy implementálja ugyanazokat az osztályokat. Ha egy osztályt módosítani, vagy létre kell hozni akkor azt csak egy projektbe kell.

### Map osztály

A catan társas játékban a pálya hatszögekből áll, ezért nem lehet egyszerűen ábrázolni, a Map osztály felel azért, hogy a generált pályán a mezők (Tile-ok) helyes helyen legyenek egymáshoz képest, és ezeket a Tile-okat össze kapcsolja, a Tile-oknak a 6 sarkára rakjon egy Vertex osztályt, és ezeket a Vertexeket összekösse Edge osztállyal. Opció, hogy egy előre elkészített pályát generáljon le, ahol a Tile-okhoz hozza van rendelve a dobott szám és a típusa.

Alternatívan lehet Map osztály elő állítani egy MapDto-ból ami tartalmazza tömören a Tile-okat és ugyan azt a pályát állítsa elő, mint ha manuálisan generáltuk le.

### Map generálása

A pálya generálása két lépésből áll

* Mezők generálása
* Mezők és csúcsok össze kötése

Mezők generálásához le kell generálni alap esetben 19 Tile-t. A Tile-nak az id-ja két részből áll, az elején egy prefix ’’T’’ és a prefix után a suffix-ja, pedig ahányadik generált Tile. Például az első Tile ’’T0’’ második ’’T1’’ .

Majd a második lépésben ezeket a Tile-okat kell összekötni. Ezt úgy teszem meg, hogy végig iterálok tiles tömbön, és a jelenlegi tilehoz hozzá kötöm a tőle jobbra, jobbra lefele és balra lefele lévő Tile-okat. Ezeket a Tile-okat úgy kapom meg, hogy a jelenlegi Tile id-ját átalakítom Coordinate-té és „getCoordBottomRight / getCoordBottomLeft / getCoordRight” metódusok segítségével megkapom a relatív koordinátát. A kapott koordinátát átváltam toTile() segítségével Tile-lá, és az attachTileToSide segítségével össze tudom őket kötni.

Függvények

|  |  |
| --- | --- |
| generateTiles(numberOfRows: Int, isGeneric: Boolean) | Legenerálja az összes Tile-t, ha az isGeneric true, akkor beállít egy előre készített pályát. Pálya legelső és utolsó sora 3 nagyságú |
| attachAllTiles() | Az összes Tile-t ami ebben a Map-ben található össze kapcsolja az oldalai mentén. |
| addEdges | A tile-okon található vertexeket összekötjük egymással |
| toDto(): MapDto | Összeállít egy olyan MapDto-t amiből össze lehet állítani ugyanezt a Map objektumot |
| attachTileToSide(  attachToThis: Tile,  bottomRight: Tile,  right: Tile,  topRight: Tile,  topLeft: Tile,  left: Tile,  bottomLeft: Tile,  ) | Az attachToThis-hez oda köti a Tile-okat, olyan irányból amilyen irányból megadjuk. |
| idToCordinate(id: Int) | A tile ID-jából össze állít egy kordinátát ami egy pár számból áll, a pár első része a sorszámra vonatkozik, a második része a sorban lévő helyére. |
| getCoordBottomRight(cord: Coordinate): Coordinate | Vissza adja a ’’cord’’-hoz képest a jobbra lent lévő kordinátát |
| getCoordBottomLeft(cord: Coordinate) : Coordinate | Vissza adja a ’’cord’’-hoz képest a balra lent lévő kordinátát |
| getCoordRight(cord: Coordinate) : Coordinate | Vissza adja a ’’cord’’-hoz képest a jobbra lévő kordinátát |

### Tile osztály

Ez az osztály reprezentálja a Catan mezőit. Catan egy mezője hatszögből áll. Az osztályban van egy vertices lista ami 6 Vertex elemet tartalmaz, az 1. ábrán látható, hogy a lista elemei melyik csúcsot reprezentálják.

A képen diagram, tervezés látható

Automatikusan generált leírás

. ábra

Függvények

|  |  |
| --- | --- |
| connectVertices(edges: List<Edge>) | Összeköti a Vertex-eket a fent ábra alapján majd a keletkezett éleket a paraméterként megadott tömbbe helyezi |

#### Player osztály

Ez az osztály felel egy aktív játékban lévő játékos reprezentálására.

Függvények

|  |  |
| --- | --- |
| Függvény név és leírása | Működése |
| calculateMyPoints(): Int | Kiszámolja a játékos pontjait, beleszámítva a ki nem játszott pont kártyákat. |
| toDto(hidden: Boolean): PlayerDto | Átalakítja a Player osztályt egy PlayerDto osztállyá, amit a szerver tovább tud küldeni. A hidden paraméter, ha igaz akkor olyan dto-t ad vissza amin a játékos kártyái és nyersanyagai elvannak tüntetve |
| refreshFromDto(dto: PlayerDto) | A paraméterül kapott dto alapján felfrissíti a játékos adatait. |

### Osztályok diagramjia

A képen szöveg, képernyőkép, Betűtípus, dokumentum látható

Automatikusan generált leírás

## Kliens

A Kliens egy grafikus felület ami a szervertől kapott információkat jeleníti meg olyan formában, hogy a felhasználó láthassa és ezen keresztül kommunikáljon a szerverrel.

A Kliensnek a szerver irányába lévő kommunikációt 2 részre lehet bontani, játékon kívüli és játékon belüli.

### Játékon kívüli kommunikáció

A képen szöveg, képernyőkép, kör, Betűtípus látható

Automatikusan generált leírás

* Regisztráció: A felhasználó készíthet egy új felhasználót, amivel betud majd jelentkezni.
* Bejelentkezés: A felhasználó bejelentkezhet egy már regisztrált felhasználóval.
* Játék szoba létrehozása: Létre hozhatsz egy várakozó szobát, amibe betud több játék csatlakozni.
* Elérhető várakozó szobák megtekintése: Letudod a szervertől kérni a jelenleg elérhető várakózó szobákat
* Belépés várakozó szobába: Egy elérhető várakozó szobába betudsz lépni
* Játék elindítása: A várakozóban, a szoba tulajdonosa elindíthatja a játékot, és a szobában lévő játékosok átkerülnek a játékba.

### Játékon belüli kommunikáció

A játékon belüli kommunikációnak két része van, mielőtt minden játékos lerakta a maga faluit és útjait, és miután mindenki lerakta.

#### Mielőtt mindenki mindent lerakott

A képen szöveg, képernyőkép, Betűtípus, fekete látható

Automatikusan generált leírás

Játék elején a kliensnek ki kell választania egy csúcsot, ahova letesz egy falut, és egy élt, ami ehhez a csúcshoz kapcsolódik. Majd ezt a csúcs és él kombinációt küldi tovább a szervernek. Ha minden játékos letett 2 falut, és élet azután elindul ténylegesen a játék.

#### Miután mindenki mindent lerakott

A képen képernyőkép, szöveg, diagram, tervezés látható

Automatikusan generált leírás

Miután elindult a játék ténylegesen a játékos 5 különböző dolgot tud csinálni:

* Passzolás: Elküldi a szervernek, hogy végzett a játékos a körével
* Tengeri csere: Játékos kiválaszt egy nyersanyagot, amiből 4 vagy annál többje van a játékosnak, és átcseréi egy attól különböző nyersanyagra.
* Fejlesztés kijátszása:
  + Útépítés: játékos kiválaszt két utat, amit megszeretne építeni betartva a szabályokat, és elküldi a szervernek.
  + Monopólium:
  + Találmány:
  + Lovag:
* Rabló áthelyezés:
* Vásárlás:
  + Út:
  + Falu:
  + Város:
  + Fejlesztés:

## Szerver

# Befejezés

Hiányzik: tengeri kereskedelem, kereskedés a játékosok között, rabló lefelezi a kártyát sok esetén.

Ide lehet leírni, hogy hogyan készítenéd el azokat a részeket, amik eddig nem történtek meg.

1. <https://www.catan.com/catan> (Utolsó megtekintés: 2023.04.15. 17:00) [↑](#footnote-ref-1)
2. <https://www.spiel-des-jahres.de/en/spiele/die-siedler-von-catan/> (Utolsó megtekintés: 2023.04.15. 16:45) [↑](#footnote-ref-2)
3. <https://www.catan.com/sites/default/files/2021-06/catan_base_rules_2020_200707.pdf> (Utolsó megtekintés: 2023.04.15. 17:15) [↑](#footnote-ref-3)
4. Egy kör alatt legfeljebb egy fejlesztéskártya használható fel. Az aktuális körben vásárolt fejlesztéskártya nem játszható ki (kivétel a pontkártya, ha ezzel nyer a játékot.) [↑](#footnote-ref-4)