**Eötvös Loránd Tudományegyetem**

**Informatikai Kar**

**Informatikatudományi Intézet**

**Programozáselmélet és Szoftvertechnológia Tanszék**

Szakdolgozat címe

Szerző: Témavezető:

Ábrahám Erik Név

Programtervező informatikus BSc. beosztás, titulus

**Budapest, 2025**

Ide kerül a hivatalos témabejelentő lap.

Tartalomjegyzék

[1. Bevezetés 1](#_Toc87521353)

[2. Felhasználói dokumentáció 2](#_Toc87521354)

[3. Fejlesztői dokumentáció 3](#_Toc87521355)

[4. Összefoglalás és további fejlesztési lehetőségek 4](#_Toc87521356)

[5. Irodalomjegyzék 5](#_Toc87521357)

[6. Melléklet 6](#_Toc87521358)

# Bevezetés

A számítógépes játékfejlesztés a szoftverfejlesztés egyik legsokoldalúbb és legizgalmasabb területe, amely egyesíti az algoritmikus gondolkodást, a vizuális megjelenítést, valamint a felhasználói élmény optimalizálásának kihívásait. A szakdolgozatom célja egy népszerű társasjáték, a *Bang!* digitális, asztali változatának megvalósítása Java nyelven. A projekt során egy önállóan futtatható, grafikus felhasználói felülettel rendelkező alkalmazást készítettem, amely hűen követi az eredeti játék szabályait, és lehetőséget biztosít egyjátékos (AI elleni) játékra.

A *Bang!* játék különlegessége, hogy a játékosok rejtett szerepek mentén, eltérő győzelmi feltételekkel versenyeznek, így a játékmenet változatos, és taktikai döntéseket igényel. A program célja ennek a játékmechanikának a pontos, de ugyanakkor élvezetes számítógépes adaptációja. A felhasználó egy grafikus felületen keresztül irányíthatja a karakterét, kártyákat húzhat, kijátszhat, támadhat vagy védekezhet, a játék során pedig a mesterséges intelligencia vezérelte ellenfelek a saját logikájuk szerint próbálják elérni győzelmüket.

A fejlesztés során a Java programozási nyelvet és a Swing grafikus felületkezelő könyvtárat választottam, mert ezek jól illeszkednek az asztali alkalmazásokhoz, illetve a Java objektumorientált szemlélete ideálissá teszi az összetettebb játéklogika kezelésére is. A játék fejlesztése során nagy hangsúlyt fektettem a rétegzett architektúra kialakítására, a kód újrafelhasználhatóságára és a tesztelhetőségre.

A dolgozat célja, hogy bemutassa a játék működését a felhasználó szemszögéből, majd részletesen feltárja a szoftverfejlesztési folyamatokat, az architektúrát, a program szerkezetét és a megvalósítás során használt technológiákat. Az utolsó fejezetben a játék teszteléséről és a továbbfejlesztési lehetőségekről esik szó.

# Felhasználói dokumentáció

## A játék rövid leírása

A Bang! egy vadnyugati tematikájú, szerepalapú társasjáték, amelyet digitális formában valósítottam meg Java nyelven. A játék során a játékosok rejtett szerepek szerint versenyeznek egymással: Sheriff, Renegát, Banditák vagy Sheriff-helyettes szerepben. Minden játékos célja eltérő, ami izgalmas és taktikus játékmenetet eredményez.

Az alkalmazás egy asztali (desktop) környezetben futó, grafikus felhasználói felülettel ellátott Java program, amely lehetővé teszi a játék egyjátékos módját. Az egyjátékos módban a felhasználó számítógép által irányított (bot) játékosok ellen játszik, akik egyszerű mesterséges intelligenciával rendelkeznek, és saját, előre meghatározott stratégiájuk mentén cselekednek.

A program a valódi társasjáték szabályait követi, beleértve a kártyák húzását, kijátszását, a különféle támadó és védekező kártyák használatát, valamint az életerő (HP) kezelését. A játékosok karaktereket választanak, amelyek mindegyike egyedi képességgel rendelkezik, ez tovább fokozza a stratégiai lehetőségeket.

A digitális verzió célja, hogy a társasjáték hangulatát és mechanikáját a számítógépen keresztül is elérhetővé tegye, lehetőséget adva gyakorlásra, játékelemzésre, vagy egyszerű szórakozásra akkor is, ha nincs kéznél elég ember egy fizikai játékpartihoz.

## Játékszabályok bemutatása

A Bang! egyjátékos változatában a felhasználó egy választott, vagy véletlenszerű szerepű karaktert irányít, és mesterséges intelligencia által vezérelt ellenfelekkel (botokkal) mérkőzik meg. A játék célja az, hogy a játékos túlélje az összes ellenség támadását, és győzelemre vezesse karakterét azáltal, hogy minden ellenfelet legyőz.

### Szerepkörök

A játék elején a játékos kiválaszthat egy tetszőleges szerepet, vagy rábízhatja a programra, hogy osszon ki neki egy véletlenszerű szerepet (Sheriff, Bandita, Renegát, Sheriff-helyettes). Az összes többi szerep botokra van kiosztva, és a játékos a játék kezdetén nem tudja, ki milyen szerepben van. A szerepkörök céljai:

* **Sheriff:** Minden Banditát és a Renegátot le kell győznie.
* **Banditák:** A Sheriff kiiktatása.
* **Renegát:** Úgy kell kiiktatnia a Sheriffet, hogy az összes Bandita már kiesett.
* **Sheriff-helyettes:** A Sheriff túlélésének biztosítása, a Banditák és a Renegát legyőzése.

### A kör menete

A játék körökre osztott. Egy körben minden játékos (bot is) végrehajtja a következő lépéseket:

1. **Kártyahúzás** – A kör elején minden[[1]](#footnote-2) játékos két kártyát húz a pakliból.
2. **Kártyák kijátszása** – A játékos tetszőleges számú kártyát kijátszhat, a szabályoknak megfelelően (pl. egyszerre csak egy Bang! kártya játszható ki).
3. **Kártyadobás** – A kör végén a játékos annyi kártyát tarthat a kezében, ahány életereje (HP) van. A fölösleget el kell dobni.

### Kártyatípusok

A játékban különböző típusú kártyák találhatók, melyek a harc és védekezés alapját képezik.

* **Bang!** – Támadókártya, amely sebzést okoz a célpontnak, ha nem védekezik.
* **Nem talált!** – Védekezőkártya, amely semlegesíti a Bang! hatását.
* **Sör** – Gyógyít 1 HP-t, de csak ha a játékos nem maximális életerőn van.
* **Gatling** – Minden ellenfelet egyszerre támad.
* **Indiánok!** – Mindenki egy Bang! kártyával védekezhet, vagy sebzést szenved el.
* **Pánik** – Elvesz egy kártyát a célpont kezéből vagy asztaláról.
* **Börtön** – Egy játékos a következő körében csak akkor játszhat, ha sikeresen "kiszabadul" (meghatározott feltétellel).

### Karakterek

A játékban minden karakter egyedi képességgel rendelkezik. A játék kezdetén a játékos kiválaszthat egy karaktert (vagy választhatja a véletlenszerű kiosztást), és a botokhoz is véletlenszerű karakterek kerülnek. Néhány példa karakterképességekre:

* **Jesse Jones:** Más játékostól húzhat kártyát a pakli helyett.
* **Slab the Killer:** Leakott Bang! kártyájára csak két Nem talált! kártyával lehet védekezni.
* **El Gringo:** Ha sebzést szenved, húzhat egy kártyát a támadótól.

Minden karakterképesség más és más, így érdemes ezek köré építeni a taktikánkat.

### Győzelmi feltételek

A Sheriff és a Sheriffhelyettesek egy csapatot alkotnak, de közülük a Sheriff a legfontosabb, mert a Banditák és a Renegát győzelmi feltétele is a Sheriff kieséséhez kapcsolódik. A Banditák akkor nyernek, ha még játékban van legalább 1 Bandita és a Sheriff kiesik. A Renegát akkor nyer, a Banditák közül mind kiesett és azután esik ki a Sheriff. A Sheriff és csapata akkor győzedelmeskedik, ha kiesik az összes Bandita és a Renegát, miközben a Sheriff még játékban van. A Sheriffhelyettesek nem befolyásolják egyik szerep győzelmi feltételét sem.

### Speciális szabályok

* A játékban nincs emberi játékostárs – mindenki más bot.
* Csak a Sheriff kiléte köztudott.
* A botok mesterséges intelligenciája egyszerű szabályok szerint működik, és elsősorban a kártyák típusára és lehetőségeikre reagálnak.

## A Bang! letöltése, telepítése

## Hardver- és szoftverigény

Az alkalmazás célja, hogy egy modern, könnyűsúlyú, egyjátékos asztali játékélményt nyújtson. A játék nem igényel jelentős számítási kapacitást vagy speciális eszközöket, így a legtöbb, Java-t futtatni képes számítógépen zökkenőmentesen használható.

### Ajánlott hardverkonfiguráció

* **Processzor**: 2 GHz-es, négymagos processzor vagy gyorsabb
* **Memória (RAM)**: 4 GB vagy több
* **Tárhely**: ~180 MB szabad lemezterület
* **Képernyőfelbontás**: 1920×1080 vagy magasabb, a kényelmes megjelenítéshez
* **Egér**: szükséges a játék futtatásához

### Szoftverkövetelmények

* **Operációs rendszer**:
  + Windows 10
* **Java futtatókörnyezet (JRE)**: Java SE 17 vagy újabb verzió

### Egyéb megjegyzések

* Az alkalmazás nem igényel internetkapcsolatot a használathoz.
* Nem szükséges adminisztrátori jogosultság a futtatásához.
* Mivel a játék csak egyjátékos módban működik, nincs hálózati vagy multiplayer komponens, így különösen kis erőforrásigényű.
* Mivel a játék angolul van, ezért minimális angol tudás szükséges.

## Játékmenet

### A játék áttekintése

A játék egy digitális adaptációja a klasszikus Bang! társasjátéknak, kifejezetten egyjátékos módra fejlesztve. A játékos a sheriff szerepét veszi fel, és célja, hogy a mesterséges intelligencia által irányított ellenfeleket legyőzze a játék szabályainak megfelelően. A játék teljes mértékben egérrel vezérelhető, és a felhasználói felület intuitívan vezeti végig a játékost a körökön.

A játék körökre oszlik, és minden körben a játékos vagy a mesterséges intelligencia által vezérelt ellenfelek lépnek. Minden játékos körében kártyákat húz, kijátszik, támad vagy védekezik, és végül eldobja a felesleges kártyákat.

### Új játék indítása

A program elindításakor a főmenü jelenik meg, ahol kettő lehetőség közül lehet választani:

* **New game (Új játék indítása)**
* **Exit (Kilépés)**

Az **New game** gomb megnyomásával megnyílik a *„Game settings”* (játék beállítások) ablak, ahol a játékos kiválasztja, hogy mennyi játékos legyen (2-7). A **Start Game** (játék indítása) gomb megnyomásával megjelen a *„Players settings”* (játékos beállítások) ablak. Ebben a felhasználó kiválaszthatja:

* melyik player (játékos) szeretne lenni
* a saját, és a botok karakterét (vagy választhat véletlenszerű karaktert)
* a saját, és a botok szerepeit (szintén lehetőség van véletlenszerű beállításokra)

Az ellenfelek minden esetben mesterséges intelligencia által vezérelt botok. A konfiguráció beállítása után a **Start Game** gombbal indítható el a játék.

### A játéktér felépítése

A játék elindulása után megjelenik a fő játéktér, amely több vizuális komponensből épül fel. A felület kialakítása Swing alapon történt, és minden elem világosan elkülönül, hogy segítse a játékos tájékozódását. A főbb elemek a következők:

#### **Játékos lapjai (alsó sáv)**

Ez a szekció a játékos kézben tartott lapjait jeleníti meg:

* **Kézben tartott kártyák**: mindegyik kártya egyedi képe jelenik meg, ha a játékos föle viszi az egeret, akkor megnő, és kijelölhető.

#### **Ellenfelek (felső sáv)**

Az ellenfelek, azaz a botok, külön panelen helyezkednek el. Mindegyik ellenfél esetén látható:

* A nevük, aktuális, és maximális HP értékük.
* Egy kártya borítóval felfelé, benne egy fehér kör és benne a kézben tartott kártyáinak száma.
* Az aktív, asztalra lerakott kártyáik képpel felfele (felszerelések, fegyverek).

A panelek dinamikusan igazodnak a tartalomhoz, hogy a karakternevek és HP-értékek teljesen látszódjanak.

#### **Középső vezérlőpanel**

Ez a szekció a játékos asztalon lévő lapjait jeleníti meg:

* **Kijátszott, aktív kártyák**: mindegyik kártya egyedi képe jelenik meg, nem lehet velük interakcióba lépni közvetlenül, csupán azt az információt adják át, hogy a kártya aktív és befolyásolj a játékos képességeit.

Itt található a játékmenetet vezérlő főbb elemek:

* **„Következő kör”** gomb: ezzel adható át a kör a következő játékosnak.
* **Célpontválasztó legördülő menü**: amelyben kijelölhető, hogy melyik ellenfelet kívánjuk megtámadni.
* **Kijátszott kártyák megjelenítése**: az asztalra kijátszott lapok itt is megjelenhetnek átmenetileg.

#### **Jobb oldali log panel**

* **Dobott lapok**: a „Discard Pile” jelzéssel jelenik meg az utolsó eldobott lap neve.
* **Aktuális/maximális életerő: a játékos aktuális és maximális életereje.**
* **Szerep: a játékos játékbeli szerepe.**
* **Karakter: a játékos karakterének a neve.**
* **Játéknapló**: itt látható az összes fontos esemény (pl. ki támadott kit, ki dobott lapot).
* **„Play this card” (kártya kijátszása) gomb: ezzel tudjuk kijátszani az aktuálisan kijelölt kártyát.**
* **„Discard this card” (kártya eldobása) gomb: ezzel tudjuk eldobni az aktuálisan kijelölt kártyát.**
* **„Next turn” (következő kör)** gomb: ezzel adható át a kör a következő játékosnak.
* **Célpontválasztó legördülő menü**: amelyben kijelölhető, hogy melyik ellenfelet kívánjuk megtámadni.

A felület célja, hogy világos, egyszerűen értelmezhető legyen, és megkönnyítse a játékos számára a döntéshozatalt.

### A körök működése és fázisai

A játék körökre osztva zajlik. Minden kör egy játékos (jelenleg mindig az emberi játékos) teljes körét jelenti, amely során végrehajthat különböző műveleteket. A botok automatikusan, a megírt mesterséges intelligencia logikája alapján hajtanak végre akciókat a saját köreikben. Ezek a fázisok automatikusan követik egymást a játék során. Az emberi játékos minden akcióját vizuálisan is támogatja az alkalmazás, míg a botok automatikusan és minimális késleltetéssel lépnek a saját logikájuk alapján.

Egy kör három fő fázisból áll:

#### Kártyahúzási fázis

* A játékos a kör elején **automatikusan húz két lapot** a húzópakliból.
* Ha a húzópakli kifogy, a dobópakli újra megkeverésre kerül és húzópaklivá válik.

#### Akciófázis

Ez a fázis a leghosszabb, itt történik a játék fő része. A játékos a kezében tartott kártyákat különböző célokra használhatja:

* **Felszerelés kijátszása**:  
  A játékos kijátszhat olyan kártyákat, mint a „Hordó” (védelmi mechanizmus), „Musztáng” (megnövelt távolság), „Távcső” (csökkenti az ellenfél távolságát) vagy fegyverek (amelyek meghatározzák a játékos lőtávolságát). Ezek az asztalra kerülnek és onnantól aktívak.
* **Támadás**:  
  A „Bang!” típusú kártyával a játékos egy ellenfelet támadhat meg, ha az lőtávolságban van. A célpont kiválasztása a célpontválasztó segítségével történik.  
  Egy körben alapesetben csak egy Bang! kártya játszható ki, kivéve, ha karakterképesség vagy más kártya ezt felülírja.
* **Különleges kártyák**:  
  Bizonyos kártyák minden ellenfelet érintenek („Gatling”, „Indiánok”), vagy más játékos lapját célozzák („Pánik”, „Cat Balou”). A játék automatikusan értelmezi ezek hatását.
* **Segéd kártyák**:Néhány kártya csak az aktuális játékosra van hatással, például gyógyítja („Sör”), vagy új lapokat huzat a játékossal („Postakocsi”, „Wells Fargo”).
* **Nem talált!** kártya („Missed!”):  
  Nem akcióként, hanem reakcióként használható. Amikor egy ellenfél megtámad minket, ezzel hárítható a támadás.

#### Dobás fázis

* A kör végén a játékosnak el kell dobnia annyi lapot, hogy a kezében lévő kártyák száma ne haladja meg az aktuális életerőpontjainak számát.
* A felesleges kártyák automatikusan dobópakli tetejére kerülnek.

### Győzelmi feltételek és a játék vége

A játékos és a botok közötti küzdelem célja az egyéni **győzelmi feltétel** teljesítése, amelyet a játékos szerepköre határoz meg. A játék indulásakor minden résztvevő véletlenszerűen (vagy felhasználói választással) kap egy szerepet. A szerepek titkosak, kivéve a Sheriff szerepét, amely mindig ismert.

#### A lehetséges szerepkörök és céljaik:

* **Sheriff**  
  Célja: Minden Bandita (Outlaw) és a Renegát likvidálása.  
  Ha a Sheriff túléli a játékot, és már csak ő és a Sheriff-helyettesek maradnak, akkor a Sheriff győz.
* **Outlaw (Bandita)**  
  Célja: Megölni a Sheriffet.  
  Ha a Sheriff meghal, és legalább egy Bandita életben van, akkor a Banditák nyernak.
* **Renegát**  
  Célja: Ő legyen az utolsó túlélő játékos.  
  Előbb el kell távolítania minden más játékost, utoljára a Sheriffet. Ha a Sheriff előtt bárki más túléli, a Renegát automatikusan veszít.
* **Deputy (Sheriff-helyettes)**  
  Célja: Megvédeni a Sheriffet, és segíteni a Banditák és a Renegát kiiktatásában.  
  Ha a Sheriff győz, a Sheriff-helyettesek is nyernek, de ha a Sheriff meghal, ők is elveszítik a játékot.

#### A játék vége

A játék vége automatikusan bekövetkezik, amikor valamelyik győzelmi feltétel teljesül. A játék végi képernyő kiírja a győztesek szerepét, valamint lezárja az aktív játékmenetet.

Az egyjátékos mód során a játékos egy előre beállított számú bot ellen küzd. A botok mesterséges intelligencia logikát követnek – támadnak, védekeznek, és megfelelő időben használják a kártyáikat. A játék vége után lehetőség van új játék indítására a főmenü segítségével.

# Fejlesztői dokumentáció

## A szoftver rövid leírása, specifikáció

A szoftver egy Java nyelven készült, asztali (desktop) környezetben futtatható egyjátékos kártyajáték, amely a Bang! nevű társasjáték digitális adaptációja. A játék célja a klasszikus szerepalapú játékmenet és a kártyahasználat élményének átültetése egy számítógépes környezetbe, a lehető leghitelesebb módon, miközben a felhasználó botok ellen játszik.

A projekt célja egy oktatási célú demonstrációs alkalmazás megvalósítása, amely bemutatja a Java objektumorientált programozási eszköztárát, a Swing felhasználói felületépítést, valamint az egyszerű AI logika integrálását.

Főbb funkciók:

* Egyjátékos mód botok ellen.
* Véletlenszerű vagy felhasználó által választott karakterek és szerepek.
* Kártyák kezelése: kézben tartott, asztalra helyezett, és eldobott kártyák.
* Játékosinterakciók: támadás, védekezés, különleges akciók.
* Botlogika: minden bot szerepkörtől és kártyáktól függően dönt.
* Felhasználói felület: grafikus ablakos megjelenítés Java Swing segítségével.
* Mentés/betöltés előkészítve, de még nem teljesen implementált.

A szoftver működésének rövid összefoglalása:

A játék elindítása után a felhasználó a főmenüből új játékot indíthat, ahol kiválaszthatja a játékosok számát (önmagán kívül csak botokat), valamint beállíthatja a karaktereket és szerepeket (vagy véletlenszerűen oszthatja ki). Ezután a játékosok egymás után hajtanak végre köröket, kártyákat húznak, kijátszanak, támadnak, védekeznek. A játék automatikusan véget ér, ha valaki eléri a saját szerepkörének győzelmi feltételét.

A szoftver fejlesztése során elsődleges szempont volt a modularitás, a karbantarthatóság, valamint az objektumorientált szemlélet, különösen a karakterek, kártyák és botok kezelésében.

## Követelményelemzés

### Funkcionális követelmények

* Új játék indítása:  
  A felhasználó új játékot indíthat a főmenüből, kiválaszthatja a karaktereket és szerepeket, vagy véletlenszerűen generált értékeket kérhet.
* Egyjátékos mód botok ellen:  
  A felhasználó egyedül játszik gépi ellenfelekkel, akik automatikusan, saját logikájuk szerint hajtanak végre lépéseket.
* Kártyakezelés:
  + Húzás a pakliból a kör elején.
  + Kártyák kijátszása (pl. Bang!, Missed!, Beer, stb.).
  + Kártyák asztalra helyezése (felszerelés kártyák, mint fegyverek vagy védekező tárgyak).
  + Kártyák eldobása, ha szükséges.
* Botok automatikus működése:  
  A botok minden körükben automatikusan végrehajtják a következő műveleteket: húzás, felszerelés lerakása, támadás, védekezés, védekező tárgyak használata, stb.
* Játék végállapotának kezelése:  
  Automatikus győztes megállapítása, ha valamelyik szerepkör eléri a győzelmi feltételeit.
* Felhasználói felület biztosítása:  
  Swing-alapú grafikus felület, ahol:
  + A játékos látja a kezében lévő kártyákat.
  + Látja az asztalon lévő lapokat.
  + Látja az ellenfelek asztalra tett lapjait és életerejét.
  + Kattintással kártyákat és célpontokat választhat.

### Nem funkcionális követelmények

* Platformfüggetlenség:  
  A program Java futtatókörnyezetet (JRE 17+) igényel, így Windows, macOS és Linux operációs rendszereken is futtatható.
* Egyszerű telepíthetőség:  
  A projekt futtatásához elegendő az IntelliJ IDEA vagy egyéb Java IDE, vagy akár parancssorból történő futtatás a Main osztályból.
* Felhasználóbarát grafikus felület:  
  A Swing alapú GUI átlátható, logikus elrendezésű, biztosítja a kényelmes kezelést, még akkor is, ha több ellenfél van.
* Stabilitás és hibatűrés:  
  A szoftver megfelelően kezeli az olyan hibákat, mint például az üres húzópakli vagy a nem választható célpontok.
* Bővíthetőség:  
  Az objektumorientált felépítés biztosítja, hogy később könnyen lehessen új kártyákat, karaktereket vagy szabálymódosításokat hozzáadni.

Tervezés

A szoftver architektúrája

A Bang! játék megvalósítása egy háromrétegű architektúra szerint történt:

Prezentációs réteg (UI):

Swing alapú grafikus felhasználói felület (BangGameUI, MainMenuUI stb.), amely felelős a felhasználói interakciók kezeléséért.

Üzleti logika réteg (BL):

A játékmenet szabályait, a körök kezelését, a támadások, védekezések, győzelmi feltételek érvényesítését valósítja meg (GameLogic, Bot osztályok).

Adatkezelési réteg (Model):

A játék modelljeit tartalmazza, mint például a játékosokat (BaseModel leszármazottai), a kártyákat (Card és leszármazottai), a paklikat és az állapotokat.

Ez a rétegzett szerkezet biztosítja a fejlesztés átláthatóságát, az egyes rétegek független módosíthatóságát és a könnyebb karbantartást.

Use-case diagram

Íme egy egyszerű Use-case diagram szöveges leírása, a konkrét rajzot is elkészítem majd PlantUML-ben később:

Felhasználó:

Új játék indítása

Karakter és szerep választása

Kör végrehajtása (kártyahúzás, kártyakijátszás)

Botok lépéseinek megvárása

Játék mentése (előirányzott)

Játék betöltése (előirányzott)

Kilépés a játékból

Rendszer (Botok):

Automatikus húzás

Kártyák automatikus kijátszása

Automatikus támadás és védekezés

Győzelem/veszteség kiértékelése

Osztálydiagram

PlantUML segítségével az osztálydiagramot is elkészítjük, de először itt szövegesen bemutatom a főbb osztályokat:

Main: a program belépési pontja.

BangGameUI: fő játékfelület kezelője.

MainMenuUI: főmenü kezelője.

PlayerSettingsUI: játékosok beállítását végző felület.

BaseModel: a játékosok (emberi és bot) ősosztálya.

Bot: a gépi játékosokat megvalósító osztály.

GameLogic: a játékmenet vezérlője.

Card: absztrakt kártyaosztály.

WeaponCard, BeerCard, BangCard, stb.: konkrét kártyatípusok.

Deck: húzó- és dobópaklik kezelője.

CardLabel, HiddenCardLabel: kártyák megjelenítését segítő UI-elemek.

## Megvalósítás

### Használt technológiák

A projekt megvalósítása során az alábbi technológiákat és eszközöket használtam:

* Programozási nyelv: Java 21
* Fejlesztőkörnyezet: IntelliJ IDEA Community Edition
* Grafikus felhasználói felület: Java Swing könyvtár
* Verziókezelés: Git, GitHub
* Dokumentáció: PlantUML (diagramokhoz), Microsoft Word (írásos dokumentáció)
* Build-rendszer: A projekt nem használ külön buildrendszert (nem Maven, nem Gradle), közvetlenül IntelliJ projektként jött létre.

A Java Swing keretrendszer biztosítja a teljes grafikus felhasználói felületet (GUI), amelyen keresztül a játékos a játékot vezérli.

A Git verziókezelővel a fejlesztési folyamat során rendszeres mentéseket készítettem, hogy a fejlesztési szakaszok könnyen nyomon követhetők legyenek, és szükség esetén korábbi állapotokra visszatérhessek.

## Projekt szerkezete, CI/CD

### Projekt szerkezete

A projekt egy egyszerű, IntelliJ IDEA által kezelt, tiszta Java projekt, amely nem használ külön buildrendszert (pl. Maven, Gradle). A projekt fájlszerkezete áttekinthető, a főbb csomagok logikailag jól elkülönítve tartalmazzák a különböző rétegekhez tartozó osztályokat:

/src

├── banggame

│ ├── bl (Üzleti logika - Business Logic)

│ ├── model (Adatmodellek, entitások)

│ ├── ui (Felhasználói felület - User Interface)

│ └── Main.java (A program belépési pontja)

* A banggame.model csomag tartalmazza a játék alapvető adatszerkezeteit: játékosok, kártyák, szerepkörök stb.
* A banggame.bl (business logic) csomag kezeli a játékmenet fő szabályait és folyamatát (pl. körök kezelése, kártyák húzása, támadások feldolgozása).
* A banggame.ui csomag tartalmazza a grafikus felhasználói felület összes elemét: panelek, ablakok, kezelőgombok.
* A Main.java osztályban történik a program inicializálása és a felhasználói felület elindítása.

### CI/CD folyamat

A projekt jelenleg nem használ automatikus CI/CD (Continuous Integration/Continuous Deployment) rendszert.

A fejlesztési workflow a következőképpen alakult:

* A forráskód módosításai helyileg történtek IntelliJ IDEA fejlesztőkörnyezetben.
* A változások rendszeres mentése Git verziókezeléssel, GitHub távoli repositoryba.
* A projekt buildelése és futtatása közvetlenül az IntelliJ IDEA eszközeivel történt.
* A tesztelés manuálisan zajlott a felhasználói felületen keresztül.

Lehetőség van jövőbeli bővítésekre, például GitHub Actions integrálására automatikus buildeléshez, unit tesztek futtatásához, vagy CI/CD pipeline kialakításához.

## Kódbázis felépítése, könyvtárszerkezet, forrásfájlok

A projekt forráskódja átlátható szerkezetben helyezkedik el, követve az MVC (Model-View-Controller) architektúra alapelveit. A csomagstruktúra célja, hogy elkülönítse a különböző funkcionális területeket: a megjelenítést, az üzleti logikát és az adatmodelleket.

Főkategóriák

banggame.model – Adatmodellek

banggame.bl – Üzleti logika

banggame.ui – Felhasználói felület

Main.java – Belépési pont

1. banggame.model

Ez a csomag tartalmazza az összes osztályt, amelyek a játék adatainak tárolásáért felelősek. Itt találhatók a következők:

BaseModel – Az összes játékos közös ősszülője. Tartalmazza az életerőt, kézben lévő kártyákat, asztali kártyákat stb.

Card – Egy általános kártyát reprezentáló osztály.

Role – Enum típusú osztály a szerepkörök (Sheriff, Outlaw, Renegade, Deputy) kezelésére.

CharacterType – Enum típus a különböző karakterek definiálására (Jesse Jones, Bart Cassidy stb.).

Ezek az osztályok főként adatszerkezetek, getterekkel, setterekkel, és minimális üzleti logikával.

2. banggame.bl

Az üzleti logika réteg, amely a játékmenet szabályait és folyamatait kezeli. A legfontosabb elemei:

GameLogic – A játék fő vezérlője. Kezeli a köröket, a játékosok lépéseit, kártyák használatát, győzelem ellenőrzését.

Bot – Az egyszerű mesterséges intelligencia, amely a botok körének automatikus lejátszását vezérli.

GameState – A játék pillanatnyi állapotát tároló osztály, például melyik játékos van soron.

Ez a réteg tartalmaz minden fontos szabályt és belső mechanizmust, amik a Bang! játékmenetét szimulálják.

3. banggame.ui

A felhasználói felületet megvalósító osztályok találhatóak itt, Swing alapokon. Főbb elemek:

BangGameUI – A fő ablak, amely tartalmazza az összes vizuális elemet: játékos panelek, kártyák, logpanel, gombok.

CardLabel, HiddenCardLabel – Különleges JLabel-ek, amelyek a kártyák megjelenítését (látható / rejtett módon) kezelik.

GameSetupUI – Az új játék indításakor a karakter- és szerepkörválasztó ablak.

MainMenuUI – A főmenü, ahol új játék indítható vagy a programból kiléphetünk.

Ezek az osztályok felelősek az eseménykezelésért is (pl. gombkattintásokra reagálás).

4. Main.java

A belépési pont, amely inicializálja az alkalmazást. Feladata:

A program elindítása (main() metódus).

A főmenü (MainMenuUI) megnyitása.

Általános jellemzők

Kódolási stílus: A kód következetes névkonvenciókat követ (camelCase metódusok, PascalCase osztályok).

Megjegyzések: A fontosabb metódusok előtt rövid kommentárok találhatók, amelyek segítik a megértést.

Egyszerűsített függőségek: A projekt kizárólag a Java SE beépített csomagjait használja (pl. javax.swing, java.util).

# Összefoglalás és további fejlesztési lehetőségek

# Irodalomjegyzék

# Melléklet

1. Néhány karakter speciális képessége ezt felülírja. Lásd: 2.2.4 Karakterek [↑](#footnote-ref-2)