



SZAKDOLGOZAT

Játékfejlesztés mobil platformra

Készítette

Vasas Dániel Viktor
programtervező informatikus BSC

Témavezető

Dr. Tajti Tibor Gábor
egyetemi docens

EGER, 2023

Tartalomjegyzék

1. Bevezetés	5
1.1. A téma relevanciája a 21. században	5
1.2. Háttér-információ	5
1.3. Célkitűzés	6
1.4. Célközönség	7
1.5. Személyes motiváció	7
1.5.1.	7
2. Felhasználói dokumentáció	9
2.1. Letöltés és telepítés	9
2.2. Játék Funkciói	9
2.3. Játék indítása	10
2.4. A sakktábla	10
2.5. A bábuk kezdőpozíciója	11
2.6. A bábuk lépései	11
2.6.1. Király	12
2.6.2. Vezér vagy néven a királynő	12
2.6.3. Bástyá	12
2.6.4. Futó	12
2.6.5. Huszár vagy néven a ló	12
2.6.6. Gyalog vagy más néven a paraszt	13
2.6.7. Az ütékenységzerről	13
2.6.8. A játék célja avagy a ki a győztes	13
2.6.9. A sakk játékokkal kapcsolatos fogalmak	13
2.6.10. A Regisztráció	14
2.6.11. A Bejelentkezés	14
2.6.12. Játzsma kezdése	14
3. Fejlesztői dokumentáció	15
3.1. Rendszerkövetelmények	15
3.1.1. Minimális rendszerkövetelmények	15
3.1.2. Ajánlott rendszerkövetelmények	15

3.2.	Rendszerfejlesztés	15
3.2.1.	Platform rendszer kiválasztása	16
3.3.	Rendszer-architektúra	16
3.4.	Követelmények elemzése	17
3.5.	osztályhierarchia	17
3.6.	Activity-k a programban	17
3.6.1.	Main	17
3.6.2.	Bejelentkezés	17
3.6.3.	Regisztráció	17
3.6.4.	Játék activity	17
3.6.5.	vs gép nézetválasztó/hostoló	17
3.6.6.	vs online személy	17
3.6.7.	online játékoskereső/ kihívó szoba	17
3.6.8.	kihívás elfogadása	17
3.6.9.	Hogyan Játszunk/Szabályok	17
3.6.10.	fejlesztőkről, írd nekünk e-mail megadása menüpont mellé vagy ebbe	17
3.7.	Osztálystruktúra	17
3.7.1.	Piece abstract osztály	17
3.7.2.	King osztály	20
3.7.3.	Queen osztály	20
3.7.4.	Bishop osztály	20
3.7.5.	Knight osztály	20
3.7.6.	Rook osztály	20
3.7.7.	Pawn osztály	20
3.7.8.	ChessBoard osztály	20
3.7.9.	ChessView osztály	20
3.7.10.	ChessGameController osztály	20
3.7.11.	ChessSquare osztály vagy chessfield	20
3.7.12.	Color osztály	20
3.7.13.	Player osztály	21
3.7.14.	AIPlayer osztály	21
3.7.15.	Move osztály	21
3.7.16.	User osztály	21
4.	Rendszerfejlesztés	22
4.1.	Környezet és eszközök	22
4.2.	Felhasználói felület tervezése és implementálása	22
4.3.	Kliens-szerver kapcsolat	24

4.4.	Képernyőtervek	24
5.	Tesztelés	26
5.0.1.	Funkcionális tesztelés	26
5.0.2.	Hibakeresés és hibajavítás	26
5.0.3.	Felhasználói visszajelzések figyelembevétele	26
6.	Összefoglalás és jövőbeli fejlesztési lehetőségek	27
6.1.	Összefoglalás	27
6.2.	Projekt eredményeinek értékelése	28
6.3.	Jövőbeli fejlesztési lehetőségek	28
	Irodalomjegyzék	29

1. fejezet

Bevezetés

A fejlesztett program egy francia sakk alkalmazás ami egy Android platformon futó játék, amely lehetővé teszi a felhasználók számára, hogy francia sakk szabályrendszerben játsszanak a mobil/okos eszközeiken gép vagy akár másik személy ellen. Az alkalmazás Java nyelven íródott, a fejlesztés Android Stúdió fejlesztői környezetben valósult meg.

1.1. A téma relevanciája a 21. században

Napjainkban nagyon fontos a modern 21. századi emberek számára, hogy a számukra korábban megszokott játékok, társasjátékok elérhetőek legyenek okos eszközeiken, visszaadva a régi idők kellemes élményeit, a klasszikus játékokat. A piacon fellelhető hasonló programok után kutatva bár található francia sakk applikáció, de az elérhető választék nem kellően magas minőségű, hogy a felhasználóknak minőségi szórakozást nyújtson. A tervezett applikáció a felhasználóbarát kezelőfelülettel és menürendszerrel továbbá a magyar nyelvi funkcióval hiányt tölt be a mobil alkalmazások között.

1.2. Háttér-információ

A francia sakknak[1] nevezett játékot főként azok kedvelik, akik a sakk bonyolult stratégiáját nem, csak a lépéseket ismerik. A játék pontos eredete és kialakulása nem egyértelműen dokumentált, mivel a francia (vagy leüttetési) sakk gyakran informális, nem hivatalos játék formájában kerül játszásra. Ezért a játék történetéről és kialakulásáról nincs sok hivatalos információ vagy forrás. A sakk (majdnem) minden szabálya érvényes (kezdőállás, lépés- és ütésmódok):

1. Az nyer, akinek először elfogynak a bábuai, az összes báb.
2. Ha a lépésen levő játékos ütni tud, köteles is. Ha többféleképpen üthet, választhat ezek közül. Ezt ütékényszernek[2] hívjuk.

3. A királyt is ugyanúgy le lehet és kell ütni, mint a többi bábút. A sakkadás[3] fogalma nem létezik. Így bármit léphetünk akkor is, ha királyunk ütésben van.
4. Az en passant ütést[4] és/vagy a sáncolást általában megegyezés szerint megtiltják vagy engedélyezik. Jelen programban mindkét speciális lépés tiltott.
5. Patt esetén megegyezés szerint patt vagy a kevesebb bábuval rendelkező fél nyel. Jelen programban ez döntetlen.

Vagyis a játék célja megváltozott, immár nem az ellenfél királyának "leütése"/mattolása a cél, hanem az összes saját figura "feláldozása". További szabályok amelyeket érdemes megemlíteni, mert a játékot befolyásolhatják:

1. Semmilyen figurával nem léphetünk olyan mezőre, amelyen saját másik figura áll.
2. Amennyiben ellenséges figura áll egy mezőn, ahova lépni tudunk, úgy leüthetjük azt.
3. Gyalog átváltozáskor, ha az adott gyalog elérte a túloldalt kérhető király is. (Minden figura kérhető, kivéve gyalog)

A játékszabályokról bővebben a *wikipedia* oldalon tájékozódhat angol nyelven, magyarul pedig a *wikipedia magyar* felületén.

1.3. Célkítűzés

A jelen francia sakk alkalmazás célja az, hogy lehetővé tegye a felhasználók számára a francia sakk szabályrendszere szerinti játékot Android platformon. Az alkalmazás fejlesztése során a következő célkitűzéseket tűztük ki:

1. **Felhasználóbarát felület:** Az alkalmazásnak intuitív és könnyen kezelhető felhasználói felülettel kell rendelkeznie, amely lehetővé teszi a játékosok számára a sakkjáték könnyű és zökkenőmentes játékot. A cél az, hogy még a kezdő játékosok is könnyen megtalálják a szükséges funkciókat és könnyedén navigáljanak a különböző menüpontok között.
2. **Francia sakk szabályainak implementálása:** Az alkalmazásnak meg kell valósítania a francia sakk szabályrendszerét a játékmenet során. Ez magában foglalja a bábuk mozgásának korlátozásait, a speciális szabályokat (pl. gyalog átalakulás), valamint a játék befejezését bábok elfogyása esetén, illetve a patt szituációk kezelését.

3. **Egyjátékos mód:** Az alkalmazásnak lehetőséget kell biztosítania a felhasználóknak arra, hogy egyedül játszassanak a gép ellen. Ebben a játékmódban a játékosnak lehetősége van különböző nehézségi szintek közül választani, és a számítógéppel játszva fejleszthesse a sakkstratégiáját.
4. **Többjátékos mód:** Az alkalmazásnak támogatnia kell a két játékos közötti interakciót. Ez lehetőséget ad a felhasználóknak, hogy egymás ellen játszanak két külön eszközön keresztül az applikációban az internet segítségével.
5. **Esztétikus megjelenés:** Az alkalmazásnak szép vizuális kinézetet kell biztosítania, beleértve a játékmező megjelenését, a bábukat és a felhasználói felületet is. A cél az, hogy az alkalmazás kellemes vizuális élményt nyújtson a felhasználók számára.

1.4. Célközönség

Az applikáció által megcélzott közösség az 1995 előtt született korosztály. Előzetes felméréseim alapján ez a korosztály aki fogékony hasonló játékokra, az ettől fiatalabb korosztályt már nem lehet megfogni hasonló klasszikusnak mondható játékokkal. Természetesen az új generáció tagjai között is akad akinek az érdeklődését felkeltheti az alkalmazás, de arányaiban lényegesen kevesebb, mint az idősebb generációkból. A fejlesztés során a célközönség igényeire optimalizálódott az applikáció.

1.5. Személyes motiváció

Gyerekkorom óta szeretem és játszom a (hagyományos) sakkot, régi célom volt egy saját sakk játék elkészítése. Viszont a hagyományos sakk fejlesztést hamar elvetettem három ok miatt. Főleg amiatt, mert a piac már telített, jobbnál jobb programok találhatók, a felhasználók igényeit kellően magas színvonalon lefedik a jól kiválasztott applikációk. Másodsorban a hagyományos sakktól nagyobb kihívásnak gondolom a francia sakk implementációját. A harmadik oka, pedig az, hogy ezzel előkészítem egy jövőbeli projektem, ahol sakk gépet fogok készíteni, olyan célból, hogy összemérjem/-versenyeztessem más sakk gépekkel.

1.5.1.

A fejlesztés során inspirálódtam az alábbiakból:

- Gerard Taylor: LEARN OBJECT ORIENTED PROGRAMMING sakk youtube lejátszási-listából. [6]

- A programkód megtervezésében és a dokumentáció elkészítésében segítségemre volt Dr. Kusper Gábor és Dr. Radványi Tibor Jegyzet a project labor című tantárgyhoz jegyzete. [7]
- A dokumentáció Tex Stúdióban való elkészítéséhez a Dr. Tómacs Tibor: \LaTeX kézikönyvet használtam. [8]

2. fejezet

Felhasználói dokumentáció

2.1. Letöltés és telepítés

Az alkalmazás letöltése és telepítése egyszerűen elvégezhető a Google Play Áruházból az Android eszközökön. A telepítés lépései a következők:

- Nyissa meg a Google Play Áruházat az Android eszközén. Ehhez kattintson a készüléken található alkalmazás ikonra vagy a főképernyőn található Play Áruház ikonra. [plusz ide egy képet betallózni](#)
- A keresősávban írd be az alkalmazás nevét, majd kattints a keresés gombra. Beírandó szöveg: **AndroidFrencChessApplication** Ezután az alkalmazás megjelenik a keresési eredmények között.
- Az alkalmazás részletes oldalán kattintson a "Telepítés" gombra. Ezzel elindul a telepítési folyamat. Kérjük várjon türelemmel amíg az alkalmazás rövid idő alatt feltelepül a készülékre.
- Miután az alkalmazás telepítése befejeződött, megjelenik a "Telepítve" vagy "Megnyitás" gomb. Kattintson a "Megnyitás" gombra a játék azonnali elindításához. Későbbi indításhoz az alkalmazásikon megnyitása szükséges a készüléken.

Ezután az alkalmazás készen áll a használatra. Amennyiben a Google Play Áruházban elérhető frissítések állnak rendelkezésre, az alkalmazás automatikusan frissülhet az eszközödön, vagy értesítést kaphatsz az új frissítésekről.

2.2. Játék Funkciói

A felhasználók lehetőséget kapnak a francia sakk játék lejátszására az alkalmazásban. Az alapvető szabályokat és mozgási lehetőségeket a hagyományos sakkhöz képest a

francia sakk szabályai szerint valósítottuk meg. Az alkalmazás a következő játék funkciókkal rendelkezik:

- Játék a gép ellen: A felhasználók játszhatnak a gép ellen, ahol az alkalmazás AI ellenfelet biztosít, különböző nehézségi szintekben.
- Online játék más felhasználókkal: Az alkalmazás lehetőséget nyújt online játékokra más felhasználókkal is. A felhasználók regisztrálhatnak fiókokat, és ezután kihívhatják egymást online partikra az alkalmazáson keresztül a játékos kihívása felületen.

2.3. Játék indítása

Képernyőfotókkal röviden elmutogatni, hogy a kezdőképernyőről, hogyan tudunk úgy navigálni, h elinduljon az alkalmazásunk mindkét módban.

A játék két módot valósít meg. Egyjátékos módot, ami játék a gép ellen. Továbbá a többjátékos módot, ami valós játékosok ellen játszódik. A játék többjátékos (online más felhasználó ellen)üzemmódjának használatához szükség van internetkapcsolatra. A játék betöltésekor az üdvözlőképernyő fogadja a felhasználót. Innen könnyedén egy választó felületre navigálja a felhasználót, ahol:

- elindítható a gép elleni offline játék
- Játékszabályokat/ hogyan játszunk menüt tekintheti meg
- Bejelentkező oldalra navigálás, ahol bejelentkezhetsz, további funkciók eléréséhez
- Kilépés az alkalmazásból lehetőség

2.4. A sakktábla

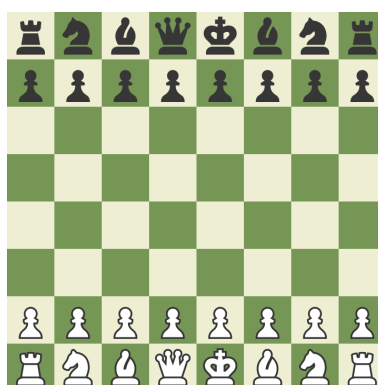
- A tábla akkor van megfelelően beállítva, ha a bal alsó sarokban sötét, a jobb alsóban pedig világos színű mező van.
- Vízszintesen 1-től 8-ig, arab számokkal jelölt sorok, függőlegesen A–H betűkkel azonosított oszlopok vannak.
- A játék kezdetén világosnak és sötétnek ugyanannyi figurája van: 1–1 király, 1–1 vezér (alternatív neve: „királynő”), 2–2 bástya , 2–2 huszár (alternatív neve: „ló”), 2–2 futó és 8–8 gyalog (alternatív neve: „paraszt”). A bábokat a 2.1-es ábrán tekintheti meg.



2.1. ábra. A sakkbábuk szemléltetése

2.5. A bábuk kezdőpozíciója

- Az ellentétes színű sakkfigurák a tábla két oldalára kerülnek két-két sorban, azaz az egyik oldalon csak a fekete, a másik oldalon csak a fehér bábuk találhatók.
- A világos szín összes figurája az 1. és 2. soron, sötét pedig a 7. és 8. soron helyezkedik el kezdéskor.
- A 2. és 7. sorban végig gyalogok kezdenek, fehér a 2. sötét a 7. sorban
- Az 1. sorban a fehér tisztek, a 8. sorban pedig a sötét tisztek helyezkednek el
- A két szélén(A és H oszlopokban) a 2-2 bástya kezd
- A bástyáktól 1 mezővel beljebb a 2-2 huszár kezd
- A huszárok mellett befelé található a 2-2 futó
- Közöttük pedig a király és a királynő(vagy vezér). Az utóbbi áll mindig a saját színén, míg a király az ellentétes színű mezőt foglalja el.



2.2. ábra. A sakktábla felállítása

2.6. A bábuk lépései

A figurák francia sakk lépéseit a lenti oldalon kipróbálhatja:

URL: <https://zoldsakk.hu/game.php?try&v=suicide>

- Lépésnek nevezzük, amikor egy bábu az aktuális mezőjét elhagyva új mezőre mozog.
- Ha a célmezőn másik színű bábu áll, úgy ezt levesszük a pályáról vagyis leütjük.
- Fontos: egyik figura sem léphet olyan helyre, ahol vele azonos színű bábu már áll!

2.6.1. Király

- Csupán egyetlen mezőt léphet, de az bármely irányba(Függőlegesen, vízszintesen vagy átlósan).

2.6.2. Vezér vagy néven a királynő

- Bármely irányba korlátlan számú mezőt léphet(Függőlegesen, vízszintesen vagy átlósan), de nem ugorhat át másik bábót.

2.6.3. Bástya

- Függőleges és vízszintes irányba korlátlan számú mezőt léphet(másképp: jobbra és balra, illetve fel és lefelé mozoghat), de nem ugorhat át másik bábót.

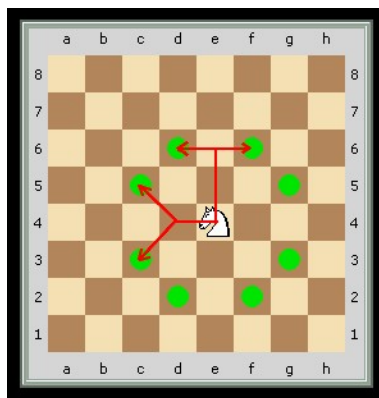
2.6.4. Futó

- Átlósan korlátlan számú mezőt(egész játék alatt csak ugyanazon színű mezőkre léphet, mint amilyen színű mezőn a bábu kezdett), de nem ugorhat át másik bábót.

2.6.5. Huszár vagy néven a ló

- A huszár speciális mozgást hajthat végre, úgynevezett "L" alakban lép. Pontosabban 2 eset létezik:
 - VAGY 2 mezőt lép vízszintesen valamelyik irányba, majd függőlegesen 1-et
 - VAGY 1 mezőt lép a vízszintes tengelyen, majd 2-t a függőlegesen

A 2.3-as ábra a huszár lehetséges lépéseit szemlélteti, az ábrán 8 helyre is léphet a huszár.



2.3. ábra. A huszár lépése

2.6.6. Gyalog vagy más néven a paraszt

- Csak előre felé mozoghat 1 mezőt (az ellenfél bábjaival kiinduló pozíciója felé, kivéve a kezdő lépést, amikor akár 2 mezőt is léphet az adott sorban). Egy lépésben 1 mezőt mozoghat előre felé, vagy ütés esetében átlósan előre hajtva végre az ütést (lépést). A gyalog ha eléri a túloldali alapvonalat, akkor a gyalog "átváltozik" [5] egy az aktuális játékos által választott tisztre.

2.6.7. Az ütéskényszerről

Ütéskényszerről beszélünk, ha az aktuális játékos legalább egy figurájának az úgynevezett tűzvonalát (ütésvonalát) elállja a másik szín bábja. Ekkor az aktuális játékosnak kötelező végrehajtania egy ellenséges bábu leütését. Amennyiben több lehetőség is van az ütésre, úgy a körön lévő játékos eldöntheti melyiket hajtja végre. Ezáltal előfordulhat, hogy körökön át a játékosoknak a kötelező lépéseik közül kell 1-1-et végrehajtania.

2.6.8. A játék célja avagy a ki a győztes

- Az a játékos győz akinek (előbb) elfogy az összes bábuja. Ezt bővebben a **Háttér-információ** részben tárgyaljuk.
- A játékot fehér kezdi, 1 lépés végrehajt, majd sötét következik, szintén 1-et lép. Ilyen módon a játékosok felváltva mozgatnak egy bábót a körökben, addig amíg a játék véget nem ér. Ekkor az egyik játékos összes bábuja elfogyott vagy patthelyzet alakult ki.

2.6.9. A sakk játékokkal kapcsolatos fogalmak

A sakkozással kapcsolatos fogalmakat az alábbi oldalon tekintheti meg:

URL: <http://www.kisbiro.hu/alkonyvtarak/sakkszotar/>

2.6.10. A Regisztráció

A regisztráció egyszerűen történik. Meg kell adni egy felhasználónevet(nem lehet már foglalt), továbbá egy email címet és egy jelszót.

2.6.11. A Bejelentkezés

A regisztrációt követően a bejelentkezés is egyszerű és értelemszerű. A már meglévő email cím vagy felhasználónév megadása mellett a jelszó megadását kéri a felület, ezután koppintásra sikeresen bejelentkeztünk.

2.6.12. Játzsma kezdése

- A gép elleni játékhoz a játék gép elleni funkciót kell kiválasztani, ami offline is elérhető.
- Valós játékos elleni meccset pedig az online játék indítása menüpont alatt kezdeményezhetünk.

3. fejezet

Fejlesztői dokumentáció

3.1. Rendszerkövetelmények

3.1.1. Minimális rendszerkövetelmények

nem a felhasználóiba kellene??

Operációs rendszer:	Android 14
Processzor:	Legalább 1,5 GHz teljesítményű processzor.
Memória (RAM):	Legalább 1 GB RAM.
Tárhely:	Legalább 200 MB szabad tárhely az alkalmazás telepítéséhez.
Kijelző:	Legalább 480x800 képpont felbontású érintőképernyő.

3.1.2. Ajánlott rendszerkövetelmények

Operációs rendszer:	Legújabb verziójú Android(jelenleg Android 14).
Processzor:	Legalább 2,0 GHz teljesítményű processzor.
Memória (RAM):	Legalább 2 GB RAM.
Tárhely:	Legalább 500 MB szabad tárhely.
Kijelző:	Legalább 720x1280 képpont felbontású érintőképernyő.
Hálózati kapcsolat*:	Ajánlott stabil és gyors internetkapcsolat.

*Hálózati kapcsolat: Ajánlott stabil és gyors internetkapcsolat. Az alkalmazás néhány funkciója és lehetőségei internetkapcsolatot igényelhetnek, például online játézmák más játékosok ellen funkció vagy a frissítések letöltése.

3.2. Rendszerfejlesztés

Az alkalmazás fejlesztéséhez szükséges követelmények a következők:

1. Java SDK 8 vagy újabb verzió

2. Operációs rendszer: Android Stúdió 4.0 vagy újabb verzió
3. Android SDK 6.0 vagy újabb verzió
4. + a teszteléshez az Android Stúdió beépített emulátora
5. **webszerverről majd ami azzal kell 1-2 pont +ban**

Az programok már telepítve voltak a fejlesztésre használt gépre. Az Android Stúdiót az xy helyről lehet letölteni. AZ SKD-t pedig innen xy Az emulátor beállítása Huawei Mate 20 Pro környezetre optimalizálva.

3.2.1. Platform rendszer kiválasztása

Az Android és az iOS két népszerű mobil operációs rendszer, amelyeket különböző okok miatt választhatunk, amikor mobilalkalmazásokat fejlesztünk. Én is ezen két operációs rendszer között kellett döntsek, az választásom Androidra esett, mégpedig:

1. Android rendszer a világszerte a legelterjedtebb mobil operációs rendszer, nagyobb piaci részesedéssel rendelkezik, mint az iOS. Vagyis Android platformon nagyobb közönséghez érhet el az alkalmazás.
2. Az Android applikációk "kiadása" ingyenes, míg IOS-en ez költségekkel járna, így a kis költségvetéssel rendelkező vagy ingyenes alkalmazást célszerűbb Android platformra fejleszteni.
3. Az Android platform részletes dokumentációval és eszközkészlettel rendelkezik, amely megkönnyíti az alkalmazásfejlesztést.
4. Az Android fejlesztői közösség nagyon aktív. Rengeteg online fórum és dokumentáció áll rendelkezésre, ahol tapasztalt fejlesztők megoszthatják tudásukat és segítséget nyújthatnak. Elakadás esetében könnyebb a továbbhaladás, mint IOS-re történő fejlesztés esetében.
5. Saját magam évek óta Androidos eszközöket használok és elégedett vagyok, így erre esett a választásom.

3.3. Rendszer-architektúra

Az alkalmazás fejlesztéséhez a Model-View-Controller (MVC) architektúris mintát . Az MVC modell segítségével elkülönítjük az adatok (model), a felhasználói felület (view) és az üzleti logika (controller) rétegeit. Ez a megközelítés lehetővé teszi a könnyű karbantarthatóságot és bővíthetőséget, továbbá letisztult kódot eredményez.

3.4. Követelmények elemzése

lehet nem is kellene...

3.5. osztályhierarchia

osztályhierarchia h vannak összekötve arról kép

3.6. Activity-k a programban

3.6.1. Main

3.6.2. Bejelentkezés

3.6.3. Regisztráció

3.6.4. Játék activity

3.6.5. vs gép nézetválasztó/hostoló

3.6.6. vs online személy

3.6.7. online játékoskereső/ kihívó szoba

3.6.8. kihívás elfogadása

3.6.9. Hogyan Játshunk/Szabályok

3.6.10. fejlesztőkről, írj nekünk e-mail megadása menüpont mellé vagy ebbe

3.7. Osztálystruktúra

Az alkalmazás a következő osztályokból épül fel.

3.7.1. Piece abstract osztály

Ez az absztrakt osztály reprezentálja egy sakkbábút, és tartalmazza az általános műveleteket, például a lépések ellenőrzését. Az egyes bábuk saját osztályainak ősoosztálya, a bábok osztályai ebből az osztályból származnak.

```
1 package chess.pieces;  
2
```

```

3  import chess.board.Board;
4  import chess.game.Move;
5
6  import java.util.ArrayList;
7  import java.util.List;
8
9  public abstract class ChessPiece {
10
11      protected int xPosition;
12      protected int yPosition;
13      protected Color color;
14
15
16      public ChessPiece() {
17      }
18
19      public ChessPiece(int xPosition, int yPosition, Color color)
20          ↪ {
21          this.xPosition = xPosition;
22          this.yPosition = yPosition;
23          this.color = color;
24      }
25
26      public abstract boolean isAttacking(int x, int y, Board
27          ↪ board);
28
29      public abstract boolean isWhite();
30
31      public abstract String getImageFileName();
32
33      public abstract boolean isValidMove(int x, int y, Board
34          ↪ board);
35
36      public List<Move> getLegalMoves(Board board) {
37          List<Move> legalMoves = new ArrayList<>();
38
39          for (int x = 0; x < 8; x++) {
40              for (int y = 0; y < 8; y++) {
41                  if (isValidMove(x, y, board)) {
42                      legalMoves.add(new Move(xPosition, yPosition
43                          ↪ , x, y));
44                  }
45              }
46          }
47
48          return legalMoves;
49      }
50

```

```

46
47     public abstract String getName();
48
49     public int getXPosition() {
50         return xPosition;
51     }
52
53     public void setXPosition(int xPosition) {
54         this.xPosition = xPosition;
55     }
56
57     public int getYPosition() {
58         return yPosition;
59     }
60
61     public void setYPosition(int yPosition) {
62         this.yPosition = yPosition;
63     }
64
65     public Color getColor() {
66         return color;
67     }
68
69     public void setColor(Color color) {
70         this.color = color;
71     }
72
73     public static boolean isPathBlocked(int fromX, int fromY,
74         ↪ int toX, int toY, Board board) {
75         if (toX < 0 || toX > 7 || toY < 0 || toY > 7) {
76             return true;
77         }
78         int dx = Integer.compare(toX, fromX);
79         int dy = Integer.compare(toY, fromY);
80
81         int x = fromX + dx;
82         int y = fromY + dy;
83
84         while (x != toX || y != toY) {
85             if (board.getPiece(x, y) != null) {
86                 return true;
87             }
88             x += dx;
89             y += dy;
90         }
91         return false;

```

```

92     }
93
94     public static ChessPiece getPiece(int x, int y, Board board)
95         ↪ {
96         return board.getBoard()[x][y];
97     }

```

3.7.2. King osztály

3.7.3. Queen osztály

3.7.4. Bishop osztály

3.7.5. Knight osztály

3.7.6. Rook osztály

3.7.7. Pawn osztály

3.7.8. ChessBoard osztály

Ez az osztály reprezentálja a sakktáblát és tartalmazza az egyes mezők állapotát és bábukat.

3.7.9. ChessView osztály

Ez az osztály a sakktábla grafikus megjelenítésért felelős.

3.7.10. ChessGameController osztály

Ez az osztály felelős a játéklógika kezeléséért, beleértve a lépések ellenőrzését és a játékállás frissítését.

3.7.11. ChessSquare osztály vagy chessfield

mezőt reprezentáló osztály

3.7.12. Color osztály

Az osztály reprezentálja egy sakkbábu színét (fehér vagy fekete). Lehetőséget nyújt az osztályoknak és metódusoknak a színek kezelésére és összehasonlítására.

3.7.13. Player osztály

Ez az osztály reprezentálja egy játékost a sakk alkalmazásban.

3.7.14. AIPlayer osztály

Az osztály a számítógépes játékos irányításáért felelős. Feladata a következő lehetséges lépés kiválasztása adott állapotban. Ehhez különböző algoritmusokat használhat, mint például a minimax algoritmust alfa-béta nyesés optimalizálással.

3.7.15. Move osztály

Egy lépést reprezentál.

3.7.16. User osztály

Egy felhasználót reprezentál. A User osztály a felhasználók menedzselésével kapcsolatos feladatokat segít ellátni. Tulajdonságai: - id: azonosító - nick: felhasználónév - pass: jelszó - email: a felhasználó e-mail címe -score: pontszám a felhasználó aktuális élő pontszáma -registrydate: a felhasználó regisztrálásának dátuma

4. fejezet

Rendszerfejlesztés

4.1. Környezet és eszközök

1. **Nyelv:** Mivel Androidra készült az alkalmazás, ezért a választott nyelv a Java programozási nyelv. Java egy objektumorientált nyelv, amely széles körben használatos az Android platformon. A Java nyelv használatával könnyen hozzáférhetünk az Android API-hoz és fejleszthetünk interaktív és megbízható alkalmazásokat. A java tartalmaz minden szükséges eszközt ami a fejlesztéshez szükséges. Ezen felül a java egy megbízható és általam is jól ismert nyelv.
2. **Környezet:** A francia sakk alkalmazás fejlesztéséhez Android Stúdió-t választottam, amely egy integrált fejlesztői környezet (IDE). Az Android Stúdió ingyenesen letölthető és tartalmazza az Android fejlesztéshez szükséges eszközöket, például az Android SDK-t (Software Development Kit) és az emulátorokat a teszteléshez, ezek miatt kiváló választás. Az emulátorok segítségével könnyedén ellenőrizhetjük az alkalmazás helyes működését különböző Android verziókon és készülékeken.

4.2. Felhasználói felület tervezése és implementálása

A felhasználói felület tervezésénél fontos szempont volt, hogy a felhasználó könnyen el tudjon igazodni a játék különböző részei között.

Implementációs kérdés. A mezőnek legyen-e figura attribútuma, vagy a figura ismerje, hogy melyik mezőn tartózkodik, esetleg a sakktáblához legyen hozzákötve a figura? Az egyszerű implementálhatóság miatt az első verziót választotta, vagyis a Square(mező) objektum rendelkezik majd egy Piece(figura) attribútummal. Ezáltal leegyszerűsödik a figurák kezelése. Összegezve, a sakktábla(8x8) minden mezője egy Square(mező) objektumot valósít meg, ahol a Square objektum ismeri, hogy a mezőn van-e figura éppen, illetve, igen esetében melyik figura. A Square osztály aggregációs kapcsolatban van a Piece(+gyeremek) osztályokkal.

1. **Felhasználói felület tervezése:** Mivel egy játék szoftverről beszélünk, fontos a kinézete, hiszen ez a játékelmény szintjét emeli. Ezt a felhasználói felület tervezésénél figyelembe vettem. A felhasználói felület egyszerű kezelhetősége és átláthatósága egyaránt fontos. Fontos, hogy az applikáció megőrizze a struktúráját, ne essen szét a felület 1-1 hibás rész miatt.
2. **Algoritmusok és logika implementálása**

A Min-Max algoritmusról

A **minimax** algoritmus egy döntési fa alapú keresési algoritmus, amelyet leggyakrabban stratégiai játékokban használnak, például sakktáblákon vagy sakkvariánsokban. Az algoritmus a játékfában keresztül bejárja a lehetséges lépéseket, kiértékeli azokat a játékosok szemszögéből, majd visszalép a fa gyökeréhez, hogy meghatározza a legjobb lépést. A **minimax** algoritmus a játékfát teljes mélységben bejárja, így nagyobb fák esetén sok időt igényelhet. Ezt érdemes optimalizálni.

Az **alfa-béta nyesés** egy hatékony technika az optimalizáláshoz, miszerint az értéktérket csökkenti azáltal, hogy kizárja azokat a lehetséges ágakat a döntési fából[9], amelyek már bizonyítottan nem adnak jobb eredményt. Az alfa-béta vágás gyakran használatos játékprogramozásban, különösen az olyan játékokban, mint a sakk, ahol a gépnek meg kell találnia a legjobb lehetséges lépéseket a győzelemhez. A játékfák nagyok lehetnek, és a gépi algoritmusok számára nehéz lehet az összes lehetőséget végigjárni, ezért az alfa-béta vágás használata lehetővé teszi a hatékonyabb keresést és a jobb teljesítményt. Az alfa-béta vágás működése az, hogy a gép a játédfa különböző ágain haladva folyamatosan kiértékeli az egyes pozíciókat, és tárolja a legjobb és legrosszabb eredményeket. Az "alfa" érték a legjobb érték az ágakon, amelyeket a gép már kiértékelt, míg a "béta" érték a legrosszabb érték az ágakon, amelyeket az ellenfél kiértékelt. A gép a keresés során állandóan frissíti ezeket az értékeket, és elhagyja azokat az ágakat, amelyek bizonyítottan nem vezetnek jobb eredményekhez. Az alfa-béta nyesés egy hatékony algoritmus, amely jelentősen csökkentheti a keresési időt és a memóriahasználatot az optimalizálás során.

- a) A minmax[10] algoritmus az AI osztályában kerül megvalósításra.
- b) Az algoritmus pszeudokódját[11] a 4.1-es ábrán tekintheti meg.
- c) Az minmax algoritmus alfa-béta nyeséssel működését magyar nyelven az alábbi youtube-ra mutató URL-en tekintheti meg.

URL: https://www.youtube.com/watch?v=vSXF-beEdko&ab_channel=OsmanOmar

Határozzuk meg az Alfa és Béta paramétereket.

1. Az alfa a legjobb érték, amit a maximalizáló jelenleg garantálhat ezen a szinten vagy afölött.
2. A béta a legjobb érték, amelyet a minimalizáló jelenleg garantálhat ezen a szinten vagy az alatt.

4.1. ábra. Pszeudokód | Minimax algoritmus alfa-béta nyeséssel,

```
1 function minimax(node, depth, isMaximizingPlayer, alpha, beta):
2
3     if node is a leaf node :
4         return value of the node
5
6     if isMaximizingPlayer :
7         bestVal = -INFINITY
8         for each child node :
9             value = minimax(node, depth+1, false, alpha, beta)
10            bestVal = max( bestVal, value)
11            alpha = max( alpha, bestVal)
12            if beta <= alpha:
13                break
14        return bestVal
15
16     else :
17         bestVal = +INFINITY
18         for each child node :
19             value = minimax(node, depth+1, true, alpha, beta)
20            bestVal = min( bestVal, value)
21            beta = min( beta, bestVal)
22            if beta <= alpha:
23                break
24        return bestVal
```

Esetleg erről egy matekos ábra is?? Esetleg erről egy matekos ábra is?? Esetleg erről egy matekos ábra is??

4.3. Kliens-szerver kapcsolat

Többjátékos üzemmódban a kliens kapcsolódik a szerverhez, emiatt oda kell rá figyel-nünk, hogy a szerver esetleges leállása ne befolyásolja a kliens tevékenységét. Leállítás esetén megfelelően lekezelje ezt a kliens. működését. blablabla blablabla blablabla blablabla

4.4. Képernyőtervek

Az kifejlesztett applikációnak igyekeztem egységes kinézetet elkészíteni.

Az applikációnak 7 db képernyője van, ezek az alábbiak:

1. Kezdőképernyő (menü nézet)
2. Bejelentkezés (menü nézet)
3. Regisztráció (menü nézet)
4. Bejelentkezés utáni felület(menü nézet)
5. Hogyan játszunk felület (menü nézet)
6. Játék vs gép (sakktáblás nézet)
7. Játék vs online játékos [kihívó nézet](sakktáblás nézet)
8. Játék vs online játékos [kihívott nézet](sakktáblás nézet)

Az online játékos elleni funkció csak bejelentkezett felhasználók számára érhető el.

5. fejezet

Tesztelés

A fejlesztés kezdeti szakaszában saját magam teszteltem a programot az AI ellen. Később minden osztályhoz megfelelő tesztosztályt írtam, ami ellenőrizte az osztály helyességét. Miután az online játék is elérhető volt, két eszközön manuálisan teszteltem. Ezután a kész programot Play Áruházba feltöltöttem és egy szűkebb ismerősi kör segítségét kértem a teszteléshez. A felmerülő problémák mind javításra kerültek, ez természetesen nem zárja ki, hogy az alkalmazásban nincs hiba.

elvégzett manuális teszt- bábok a saját lépéseiket végrehajtják-e báb tud-e hibásan lépni etcetc...

5.0.1. Funkcionális tesztelés

5.0.2. Hibakeresés és hibajavítás

5.0.3. Felhasználói visszajelzések figyelembevétele

beszámoló a visszajelzésekről a teszt időszak alatt

6. fejezet

Összefoglalás és jövőbeli fejlesztési lehetőségek

6.1. Összefoglalás

A szakdolgozat első része a projecthez kapcsolódó információkat és háttérrel tárgyalja, majd ezt követi a felhasználói dokumentáció, ami a játékosok számára nyújt segítséget. Ezután a fejlesztői dokumentáció majd a tesztelés részek jönnek, ami már fejlesztőknek szánt rész. A fejlesztői dokumentációban részletesen vizsgáljuk a program által használt osztályokat, képernyőket és algoritmusokat. A fejlesztőknek szóló rész a feladat tervezésének és implementálásának menetét is kifejti. A tesztelés részben a már elkészült alkalmazásban kerestünk meglévő hibákat, majd a hiba feltárása után a problémás rész javítása valósult meg.

A szakdolgozat célja egy Android platformon működő francia sakk alkalmazás megvalósítása volt. A fejlesztés során sikerült létrehozni a játékkeret vagy pályát amin a játék lezajlik. Létrehoztam továbbá a figurák és a játékmenethez szükséges osztályokat. Ezeken felül implementáltam a netes játékot kliens-szerver kommunikációval. A program központi eleme a sakktábla, a figurák és a bábuk mozgatása, emiatt ezen részek kialakítására nagy hangsúlyt fektettem. Az elkészült alkalmazás jól működik, kellemes kikapcsolódást nyújt a felhasználóknak, akik pont azt kapják amit várnak. Egy letisztult felülettel rendelkező szép és igényes megjelenésen keresztül játszhatnak ezen speciális sakkvariánssal.

Személyes véleményem szerint elmondható, hogy az alkalmazás nagyon jól sikerült, ezt a külső tesztelők visszajelzése megerősíti. AZ előzetes terveket jól visszatükrözi.

6.2. Projekt eredményeinek értékelése

lehet inkább külön pontba, elvégzett munka és a fejlesztői doksiba Elsőként a játékfejlesztés mobil platformra témakört jártam körbe, majd a többszemélyes játékok és a gép elleni játékok témakörben gyűjtöttem ismereteket. Ezután telepítettem a minden szükséges programot és elvégeztem a beállításukat. Ezekből sokat tanultam. Majd a program struktúrájának kialakításról tájékoztam. Ezek után megterveztem a felhasználói felület kinézetét. Ezt követte a játéktábla, a bábok és a játéklogika implementálása.

A project megvalósítása során sokat tanultam, illetve nagyon élveztem az ezzel eltöltött időt.

6.3. Jövőbeli fejlesztési lehetőségek

1. Bejelentkezés Google felhasználóval megvalósítása
2. Hagyományos sakk implementálása, hogy a felhasználó választhasson, ha épp azt játszana
3. A Patt szabályra kapcsoló bevezetése **igen** esetében pattnál döntetlen, **nem** esetében pattnál az nyer akinek kevesebb a bábja a pályán.
4. Sáncolás és en passzant lépések implementálása, kapcsoló bevezetése, hogy a felhasználó igénye szerint használjuk vagy sem az új szabályokat.
5. Új grafikus felület létrehozása(pálya színének váltása, pálya forgatásának lehetősége)
6. Kitüntetések funkció létrehozása(vagyis a játékosok bizonyos sikerek után medálokat kapnának, pl 50 megnyert meccs után stb.)
7. AI fejlesztése, új választható fokozatok implementálása
8. Játék mentése és betöltése funkciók megvalósítása
9. Gépi segítség nyújtás(kötelező lépés mutatása)
10. Kereskedelmi/fizetős verzió létrehozása
11. Reklámok beépítése(pl.: meccs utáni lépéslista lehívásakor előfeltétel a reklám videó megtekintése)

irodalomjegyzéket és a fogalomjegyzéket szétválasztani, kell külön fogalomjegyzék is

Irodalomjegyzék

- [1] FRANCIA SAKK: A francia sakk szabályait az alábbi linken megtekintheti.
URL: https://hu.wikipedia.org/wiki/Francia_sakk
- [2] ÜTÉSKÉNYSZER: Az aktuális játékos ha egy bábuval üthetne, akkor kötelező ütnie. Ez azt jelenti, hogy a játékos nem hagyhatja figyelmen kívül az ütési lehetőséget, több lehetőség esetén, választhat.
- [3] SAKKADÁS: Sakkadásnaknak tekintjük az alap sakkjátékban, ha a királyunk bármely ellenséges báb által támadható, azaz az ellenséges báb rá tudna lépni a következő lépésében a király aktuális mezőjére. Ez a fogalom a francia sakkban nem létezik.
- [4] EN PASSANT: Az en passant egy különleges lépés a sakkban, olyan ütés, amelyre akkor kerülhet sor, ha az egyik fél gyalogja kettőt előrelépve kikerülné az ellenfél gyalogjának átlós ütését. Az egyetlen eset a sakkban, amikor egy gyalog nem az átlósan előtte álló, hanem a mellé lépő figurát ütheti ki. Ekkor az átló mentén lép a gyalog és az áthaladó ellenséges figurát leüti.
- [5] GYALOG ÁTVÁLTOZÁS: Az átalakulás vagy átváltozás a sakkban a gyalog különleges képessége. Amennyiben a gyalog eléri az ellenfél alapsorát (tehát a világos a 8. sort, a sötét pedig az 1. sort), a gyalog a játékot tisztként folytatja. A játékos szabadon eldöntheti, hogy az új tiszt vezér, bástya, futó vagy huszár legyen-e (francia sakk esetében király is lehet).
- [6] GERARD TAYLOR: LEARN OBJECT ORIENTED PROGRAMMING
youtube videósorozat 2023.05.22. dátumkor még elérhető volt.
- [7] KUSPER GÁBOR: *Jegyzet a projekt labor című tárgyhoz*, Eger, 2012.
- [8] TÓMÁCS TIBOR: *L^AT_EX* kézikönyv, Eger, 2023.
A könyv szabadon letölthető az alábbi linkről:
URL: <https://tomacstibor.uni-eszterhazy.hu/tananyagok/LaTeX.pdf>
- [9] DÖNTÉSI FA: A döntési fa egy olyan, a döntéshozatalban használt grafikus modell, amit az optimális tevékenység határoz meg olyan esetekben, amikor több választási

lehetőség is rendelkezésre áll, és a kimeneteik bizonytalanok. A diagram arról kapta a nevét, hogy egy faágra hasonlít. A döntési fák matematikailag gráfok.

- [10] MINMAX ALGORITMUS: URL: https://hu.wikipedia.org/wiki/Minimax_elv
- [11] PSZEUDOKÓD: A pszeudokód az algoritmusok és általában az eljárások leírására használt mesterséges formális nyelv, amely változókból és állandó jelentésű kulcsszavakból áll, és hasonlít a számítógépes programozási nyelvekre. A pszeudokódok minden elemi lépésnek egy-egy szót, jelet, elnevezést feleltetnek meg, ezáltal alkalmasak az elemi lépések és így az egész algoritmus leírására.

Nyilatkozat

Alulírott *Vasas Dániel Viktor*, büntetőjogi felelősségem tudatában kijelentem, hogy az általam benyújtott, *Játékfejlesztés mobil platformra* című szakdolgozat önálló szellemi termékem. Amennyiben mások munkáját felhasználtam, azokra megfelelően hivatkozom, beleértve a nyomtatott és az internetes forrásokat is.

Aláírással igazolom, hogy az elektronikusan feltöltött és a papíralapú szakdolgozatom formai és tartalmi szempontból mindenben megegyezik.

Eger, 2023. május 16.

aláírás