Matematikai és Informatikai Intézet

Játékfejlesztés mobil platformra

Készítette
Vasas Dániel Viktor
programtervező informatikus BSC

Témavezető
Dr. Tajti Tibor Gábor
egyetemi docens

Eger, 2022

Tartalomjegyzék

1. fejezet

Bevezetés

A fejlesztett program egy francia sakk alkalmazás 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 nyelvei funkcióval hiányt tölt be a mobil alkalmazások között.

1.2. Háttér-információ

A francia sakknak 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 bábui, az összes báb.
2. Ha a lépésen levő játékos ütni tud, köteles is. Ha többféleképpen üthet, választhat ezek közül. Ezt ütéskényszernek[1] hívjuk.

3. A királyt is ugyanúgy le lehet és kell ütni, mint a többi bábut. A sakkadás[2] fogalma nem létezik. Így bármit léphetünk akkor is, ha királyunk ütésben van.

4. Az en passant ütést[3] 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. Célkitűzés

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 játszhassanak a gép ellen. Ebben a játékmódban a játékosnak lehetősége van különböző nehézségi szintek közül választani, és a számítógéppel játszva fejleszthesse a sakkstratégiáját.

4. **Egyjátékos 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 kialakítás:** Az alkalmazásnak esztétikus 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.

2. fejezet

Rendszerterv

2.1. Rendszerkövetelmények

Az alkalmazás fejlesztéséhez szükséges követelmények a következők:

1. Java SDK 8 vagy újabb verzió
2. Android Studio 4.0 vagy újabb verzió
3. Android SDK 6.0 (Marshmallow) vagy újabb verzió
4. + a teszteléshez az Android Stúdió beépített emulátora

2.2. Rendszerarchitektúra

Az alkalmazás fejlesztéséhez a Model-View-Controller (MVC) architekturális 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.

2.3. Követelmények elemzése

2.4. Osztálystruktúra

Az alkalmazás a következő osztályokat használja/ osztályokból épül fel.

2.4.1. Piece abstract osztály

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

```java
package chess.pieces;

import chess.board.Board;
import chess.game.Move;

import java.util.ArrayList;
import java.util.List;

public abstract class ChessPiece {

    protected int xPosition;
    protected int yPosition;
    protected Color color;


    public ChessPiece() {
    }

    public ChessPiece(int xPosition, int yPosition, Color color) {
        this.xPosition = xPosition;
        this.yPosition = yPosition;
        this.color = color;
    }

    public abstract boolean isAttacking(int x, int y, Board board);

    public abstract boolean isWhite();

    public abstract String getImageFileName();

    public abstract boolean isValidMove(int x, int y, Board board);

    public List<Move> getLegalMoves(Board board) {
        List<Move> legalMoves = new ArrayList<>();

        for (int x = 0; x < 8; x++) {
            for (int y = 0; y < 8; y++) {
                if (isValidMove(x, y, board)) {
                    legalMoves.add(new Move(xPosition, yPosition, x, y));
                }
            }
        }
```

```
43
44          return legalMoves;
45      }
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,
        ↪ int toX, int toY, Board board) {
74          if (toX < 0 || toX > 7 || toY < 0 || toY > 7) {
75              return true;
76          }
77          int dx = Integer.compare(toX, fromX);
78          int dy = Integer.compare(toY, fromY);
79
80          int x = fromX + dx;
81          int y = fromY + dy;
82
83          while (x != toX || y != toY) {
84              if (board.getPiece(x, y) != null) {
85                  return true;
86              }
87              x += dx;
88              y += dy;
```

```
89          }
90
91          return false;
92      }
93
94      public static ChessPiece getPiece(int x, int y, Board board)
            ↪ {
95          return board.getBoard()[x][y];
96      }
97  }
```

2.4.2. King osztály

2.4.3. Queen osztály

2.4.4. Bishop osztály

2.4.5. Knight osztály

2.4.6. Rook osztály

2.4.7. Pawn osztály

2.4.8. ChessBoard osztály

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

2.4.9. ChessGameController osztály

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

2.4.10. ChessSquare osztály vagy chessfield

mezőt reprezentáló osztály

2.4.11. Color osztály

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

2.4.12. Player osztály

Ez az osztály reprezentálja egy játékost a sakk alkalmazásban.

2.4.13. Move osztály

Egy lépést reprezentál.

piece abstract osztály:

3. fejezet

Rendszerfejlesztés

3.1. Környezet és eszközök

1. **Nyelv:**Mivel Androidra készülne 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.

3.2. Felhasználói felület tervezése és implementálása

1. **Felhasználói felület tervezése:**A tervezési fázisban először gondolkodjunk el az alkalmazás felhasználói felületének kinézetén és struktúráján. Gondoljuk át, hogy milyen elemekre van szükségünk a játék kezeléséhez, például a játéktábla, a bábuk, a gombok, stb. Fontos megtervezni az elemek elhelyezkedését és a felhasználóval való interakciójukat.

2. **Algoritmusok és logika implementálása**

4. fejezet

Felhasználói dokumentáció

4.1. Letöltés és telepítés

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

- Nyissa meg a Google Play Áruházat az Android eszközén. Ehhez koppintson 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 koppints a keresés gombra. Beírandó szöveg: **AndroidFrencChessApplication** Ezután az alkalmazás megjelenik a keresési eredmények között.
- Az alkalmazás részletes oldalán koppintson 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. Koppintson 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.

4.2. Játék Funkciói

A felhasználók lehetőséget kapnak a francia sakk játék lejátszására az alkalmazásban. Az alapvető szabályokat és mozgási lehetőségeket a hagyományos sakkhoz 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 játékra más felhasználókkal is. A felhasználók regisztrálhatnak fiókokat, és ezután kihívhatják egymást online partikra az alkalmazáson keresztül a játékos kihívása felületen.

4.3. Játék indítása

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

4.4. A bábuk lépései

Fontos: egyik figura sem léphet olyan helyre, ahol vele azonos színű báb már áll!

4.4.1. Király

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

4.4.2. Vezér vagy néven a királynő

- Bármely irányba korlátlan számú mezőt léphet.(Függőlegesen, vízszintesen vagy átlósan).

4.4.3. Bástya

- Függőleges és vízszintes irányba korlátlan számú mezőt léphet(másképp: jobbra és balra, illetve fel és lefelé mozoghat).

4.4.4. Futó

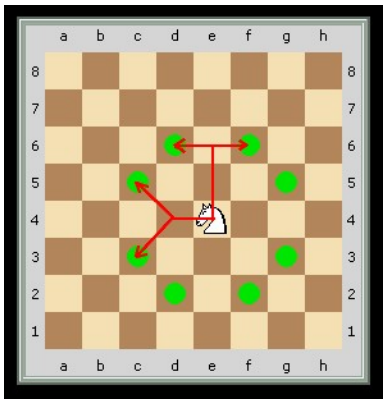
- Átlósan korlátlan számú mezőt(egész játék alatt csak ugyanazon színű mezőkre léphet, mint amilyen színű mezőn a bábu kezdett.)

4.4.5. 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 4.1-es ábra a huszár lehetséges lépéseit szemlélteti, az ábrán 8 helyre is léphet a huszár.



4.1. ábra. A huszár lépése

4.4.6. Gyalog vagy más néven a paraszt

– Bármely irányba korlátlan számú mezőt léphet.(Függőlegesen, vízszintesen vagy átlósan).

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, 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.

5.1. Funkcionális tesztelés

5.2. Hibakeresés és hibajavítás

5.3. Felhasználói visszajelzések figyelembevétele

6. fejezet

Összefoglalás és jövőbeli fejlesztési lehetőségek

6.1. Összefoglalás

6.2. Projekt eredményeinek értékelése

6.3. Jövőbeli fejlesztési lehetőségek

1. A Patt szabályra kapcsoló bevezetése **igen** esetében pattnál döntetlen, **nem** esetében pattnál az nyer akinek kevesebb a bábja a pályán.
2. Sáncolás és en passzant lépések implementálása
3. új grafikus felület létrehozása
4. AI fejlesztése
5. játék mentése és betöltése funkciók implementálása
6. gépi segítség nyújtás(kötelező lépés mutatása)

Irodalomjegyzék

[1] 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.

[2] 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.

[3] 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.

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.

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

Eger, 2023. május 16.

aláírás