doodle jump

Szerzők:
Halász Botond
Gergely Zsolt

SAPIENTIA
ERDÉLYI MAGYAR
TUDOMÁNYEGYETEM

Vezető tanár:
Dr.Szántó Zoltán

Szoftver rendszerek tervezése 2020

Tartalomjegyzék

1. Bevezetés

Minden ember kerül néha olyan helyzetbe, hogy várnia kell valakire, unatkozik vagy akár kedve van egy kis kikapcsolódásra, ilyenkor egy játéknak nagyon fontos szerepe van, mivel kikapcsolódik az ember, úgy érzékeli, hogy gyorsabban telik az idő és játszás közben fejlesztheti a logikai gondolkodást, reflexeit, koordinációs képességet, megtornáztatja az agyát különböző feladatok megoldásával. Manapság nagyon elterjedtek a telefonos játékok, mivel a telefon mindig a felhasználó közelében van, nagyon gyorsan el tudja indítani rajta kívánt applikációját, ezek mellett amikor dolga akad akkor egyszerűen szünetelteti alkalmazását és lezárja telefonját, majd amikor ismét van rá ideje újra képes folytatni ahol abbahagyta.

Pont ezekre a pillanatokra kínál a mi applikációnk egy lehetőséget, melynek révén képes a felhasználó egy élményekben dús játékban részt venni, melynek folyamán nagymértékben fejlődhet a felhasználó koordinációs képessége, reflexei, mindezek mellett egy kikapcsolódást nyújt amellyel élménydúsan töltheti az idejét. A fent említett előnyök miatt döntöttünk úgy hogy egy telefonos játékot fogunk fejleszteni olyan formában amellyel akkor is játszhatnak a játékosok, hogyha nem tudnak internetes kapcsolathoz csatlakozni.

A játékot egy általunk ismert régi játék mintájára építettük fel, amellyel a fejlesztői csapatunk minden tagja nagyon sokat játszott amikor megjelent és a régi élményeket szerettük volna felidézni vele.

1.1. Rövid összefoglaló a játékról

A játékban a játékos feladata az, hogy minél magasabbra jusson a pályán, mivel a magassággal arányos pontszámot kap a játék során a felhasználó.

1.2. A játék logikai menete

A felfelé ugrálás érdekében a felhasználó bizonyos szigetekre kell rá lépjen amelyekre ha rálép a karakter újból szökik egyet. A felfelé haladás viszont nem annyira egyszerű, ahogy egyre magasabbra jut a felhasználó, a játék egyre nehezedik, felgyorsul, érzékenyebbé válik a karakter irányítása, ezek mellett megjelennek nehezítő tényezők, viszont segítő tényezők is, ezek kihasználásával kell a felhasználónak minél jobb eredményt elérnie.

A játék menetét a játékos szüneteltetheti, információkat olvashat a játékról, irányítási funkciót válthat, majd folytathatja a játékot ahol abbahagyta.

A játékot a felhasználó játszhatja bejelentkezve illetve anélkül, a bejelentkezett felhasználónak azok az előnyei, hogy régebben elért pontszámokat illetve más felhasználók által elért pontszámokat is láthatnak.

A játéknak akkor lesz vége, amikor a játékos leesik illetve ha egy szörnyeteghez hozzáér.

2. Célkitűzések

Az applikáció tervezése közben számos célt tűztünk ki magunk elé amelyeket úgy gondolunk, hogy fontos az szoftver működése szempontjából illetve a jó felhasználói élmény eléréséhez szükségesek.

Ezen célok közé tartozik az, hogy lehetősége legyen a felhasználónak úgy is használni az applikációt, hogy nem jelentkezik be és úgy is ha be szeretne jelentkezni és nyomon szeretné követni előző adatait.

Szerettük volna azt, hogy ha bejelentkezik a felhasználó lássa a saját pontszámait amelyeket az előző játékokban elért, illetve lássa a toplistát is, tehát más játékosok pontszámát is, hogy ők milyen eredményeket értek el.

Céljaink között szerepelt az, hogy a játék menete végtelenszerű legyen, tehát a felhasználó addig tudjon haladni a játékban ameddik csak képes, ne legyen benne egy korlát, amelynél már a játéknak vége szakad.

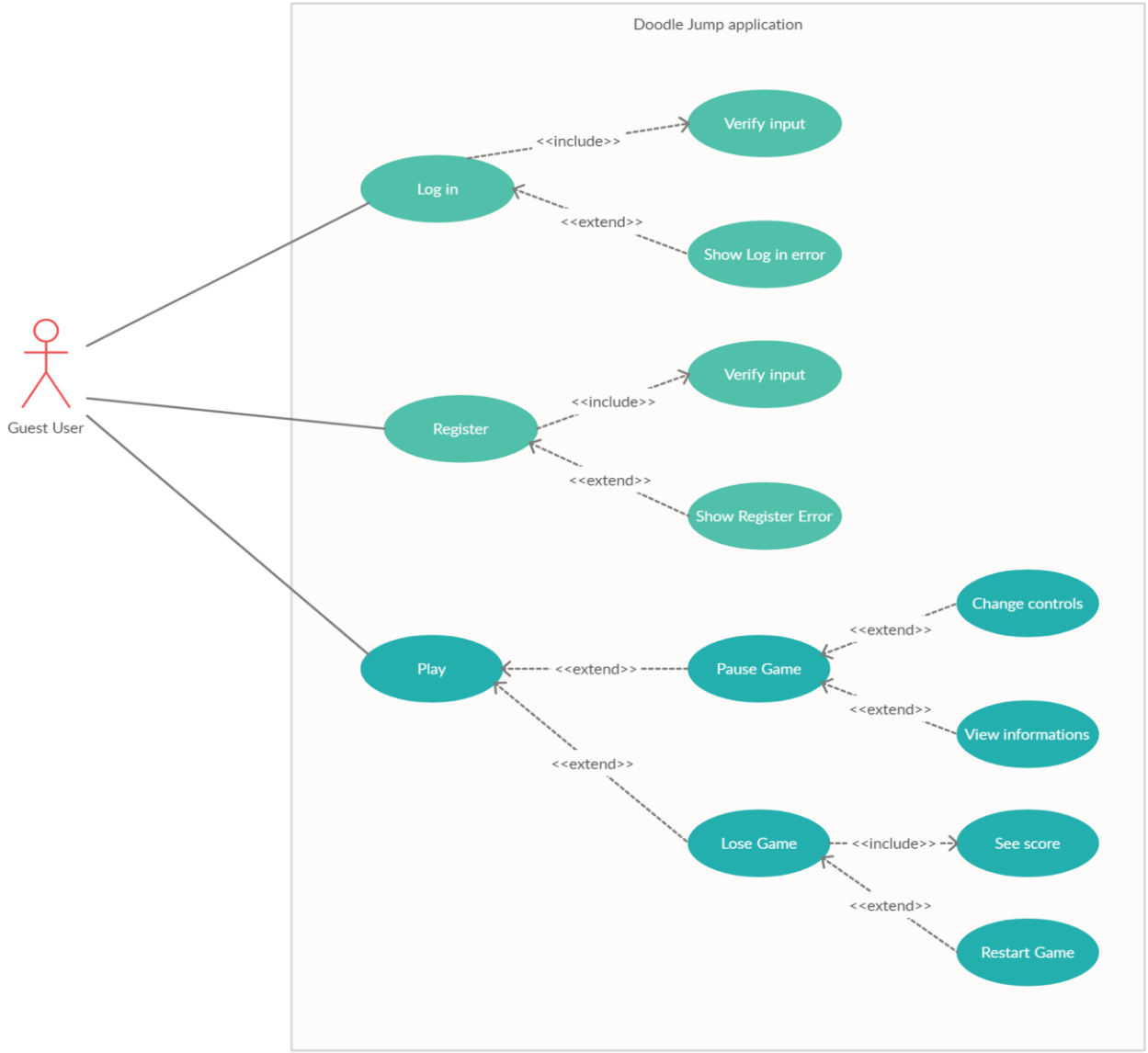
Az alkalmazásban lehetősége nyílna a felhasználónak a játék során szüneteltetni a játékmenetet, ezáltal csökkentve a feszültséget abból a szempontból, hogy ha a készüléket leteszi kezéből akkor bizonyosan véget ér a játék, ezek mellett információkat tudjon olvasni arról hogy a játékban bizonyos elemeknek mik a szerepük.

Fontosnak tartottuk azt is, hogy lehessen a játékot kétféleképpen is irányítani, úgy, hogy hozzáérünk a kijelzőhöz illetve úgy, hogy a telefon gyroscope-ját használva a telefon döntésével irányítjuk a karaktert a megfelelő irányba, ezt a játékos tudja majd a játékmenet közben kiválasztani.

További céljaink között szerepelt az is, hogy a játékmenet ne tűnjön monotonnak, tartsuk fennt a felhasználó érdeklődését, ezért használjunk fel olyan elemeket, amelyek érdekessé teszik a játékot illetve odafigyelést igényelnek a felhasználó részéről. Ennek érdekében a játék folyamatosan nehezedik és vannak ellenfelek amelyeket el kell pusztítani különben ha hozzáérünk a játék véget ér.

3.Követelmény specifikáció

3.1. Felhasználói követelmények



1. Ábra: Use-case diagram

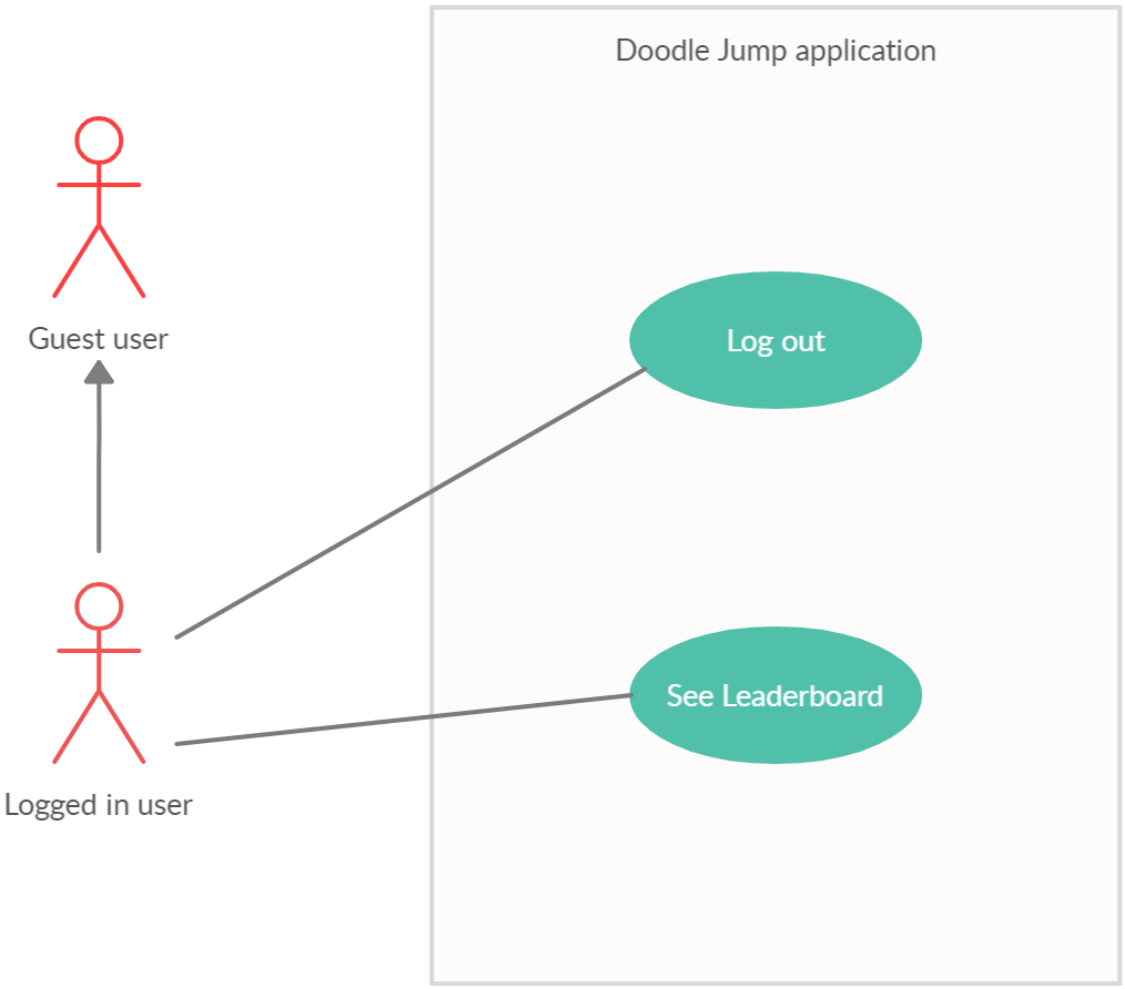
Az applikációnak két fajta szerepköre van, két féle felhasználó lehet. Egy be nem jelentkezett felhasználó (vendég felhasználó, lásd a fenti ábrán) és egy bejelentkezett felhasználó (2. Ábra: Use-case diagram).

A vendég felhasználó funkcionalitásai :

- Bejelentkezés: a felhasználó bejelentkezhet a már meglévő fiókjával, ha helyesek a megadott fiók adatai
- Regisztráció: a felhasználó létrehozhat új fiókot kedve szerint, ha a megadott adatok megfelelnek a kritériumoknak
- Játék indítása: a játék elindításával a felhasználó belecsöppen az alkalmazás lényegébe, kedve szerint játszhat, újraindíthatja a játékot game over esetén, megnézheti a jelenleg játszott pontszámát, szünetelteti a játékot, kiléphet a játékból

- a játék szüneteltetésével elolvashat némi információt a játék menetéről, illetve átválthat ‘gyroscope’ módba
- játék kilépésnél visszatérünk a fő menüre, mivel nem voltunk bejelentkezve, így az eddig szerzett pontszámokat se menti le - azonnali bejelentkezéssel elmenti a pontszámokat

2. Ábra: Use-case diagram2. Ábra: Use-case diagram2. Ábra: Use-case diagram2. Ábra: Use-case diagram



2. Ábra: Use-case diagram

Bejelentkezett felhasználó funkcionalitásai:

- Rendelkezik minden vendég felhasználó funkcionalitással
- Kijelentkezés
- Felhasználói felület: - belépést nyer erre a felületre
  - megnézheti az eddigi pontszámait rangsorolva
  - láthatja az összes felhasználó pontszámát rangsorolva, megkeresheti a saját legmagasabb pontszámának elhelyezkedését
- Minden további pontszám automatikusan felkerül az adatbázisba és a pontszámok rangsorolva lesznek

3.2. **Rendszer követelmények**

3.2.1. **Funkcionális követelmények**

A játék indításakor a menü oldalon találja magát a felhasználó, ahol tud regisztrálni, be tud jelentkezni illetve új játékot tud elindítani.
A játék menete közben a felhasználó szüneteltetni tudja a játékot, újraindítani, információkat olvasni illetve irányítani a karakterét.

3.2.2. **Nem funkcionális követelmények**

- Telefon
- Android: 4.1+
- Api Level: 16(Jelly Bean)
- Internet Connection: H+ , 3G, 4G, 5G
- Free storage? ~ 32Mb
- Ram: 1GB
- Gyroscope

Az applikáció kizárólag telefonon működik, amelyen szükséges lennie minimum 4.1-es androidnak(16 level api), körülbelül 32 Mb tárhelyre van szüksége a telepítéséhez és 1 Gb ram-ra van szükség az akadásmentes futás érdekében.
Az internet kapcsolat csak bizonyos funkciók eléréséhez szükséges, mintpéldául a regisztráció, bejelentkezés és a leaderboard megtekintése.
A gyroscope szintén csak akkor szükséges, hogyha a felhasználó a játék folyamán átállítja az irányítást a gyroscope használatára.

4. Alkalmazás felépítése (Architektúrája)

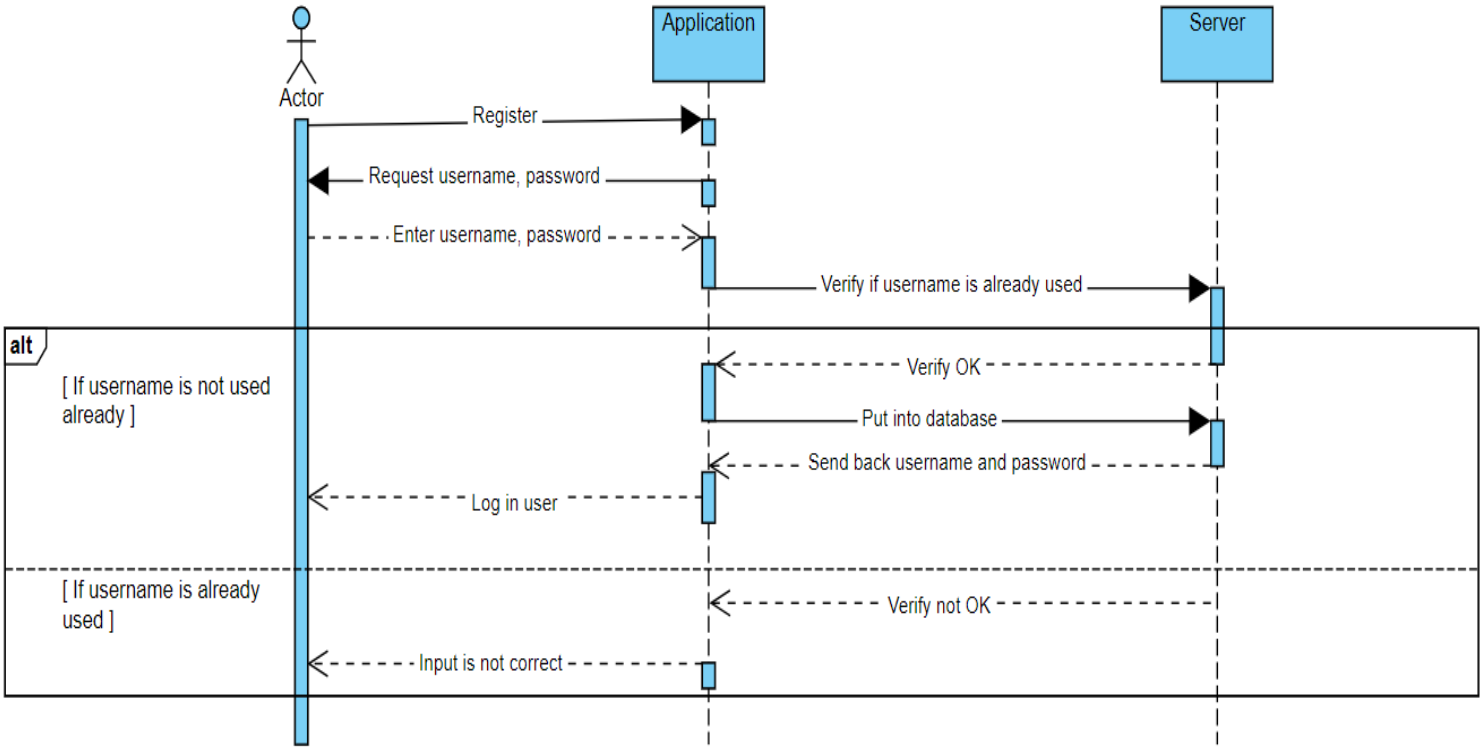
Az alkalmazás tervezésének első lépéseiben azt próbáltuk meghatározni, hogy a milyen funkciókkal szeretnénk, hogy bírjon az alkalmazásunk , ezek meghatározása során pedig sikerült elkülöníteni, hogy milyen felhasználó típusok lesznek. (1. Ábra: Use-case diagram, 2. Ábra: Use-case diagram)

Az applikációt két nagy részre bontottunk, eszerint a részek szerint osztottuk szét egymás közt a munkát is. Az egyik ilyen rész a back-end része az applikációnak, ez magába foglalja a szerver, adatbázisok létrehozását, a bejelentkezés és a regisztrációt. A másik modul pedig a játék megszervezését, a játékbeli logikát tartalmazza.

Ezek után minden egyes fő funkciót lebontottunk, illetve megnéztük, hogy mikor és hogyan teremt kapcsolatot a felhasználó az applikációval, illetve az applikáció a szerverrel, meghatároztunk a kommunikációkat közöttük.

4.1. Szekvencia diagramok

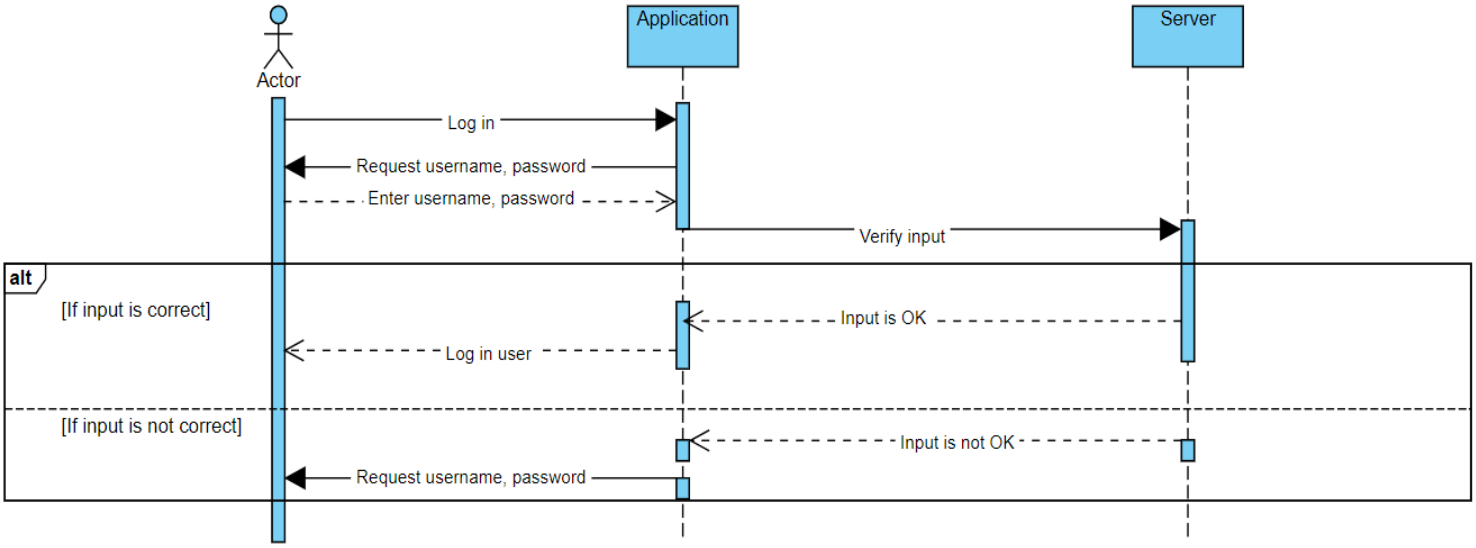
4.1.1. Regisztráció



3. Ábra

A regisztráció során a felhasználó, az ábrán “Actor” egy regisztrációs kérést küld az applikáció felé egy gomb lenyomásával, ezután az applikáció kér a felhasználótól egy felhasználónevet, jelszót illetve egy jelszó megerősítést, amelyet egy gomb lenyomásával elküld a felhasználó az alkalmazásnak, az applikáció csak akkor engedi meg az elküldést, hogyha kitöltött a felhasználó minden mezőt illetve rendelkezik a megfelelő karakterszámmal és karakter típusokkal. Ezután az applikáció leellenőrzi, hogy az adatbázisban szerepel-e ilyen nevű felhasználó, ha nem szerepel ilyen felhasználó akkor visszaküldi, hogy rendben van, a felhasználó betevődik az adatbázisba, majd az applikáció automatikusan bejelntkezteti a felhasználót a megadott adatok szerint. Ha van már ilyen felhasználó az adatbázisban akkor az applikáció visszakapja hogy nincs rendben az adat, az applikáció meg kimutatja a felhasználónak, hogy adjon meg más adatokat.

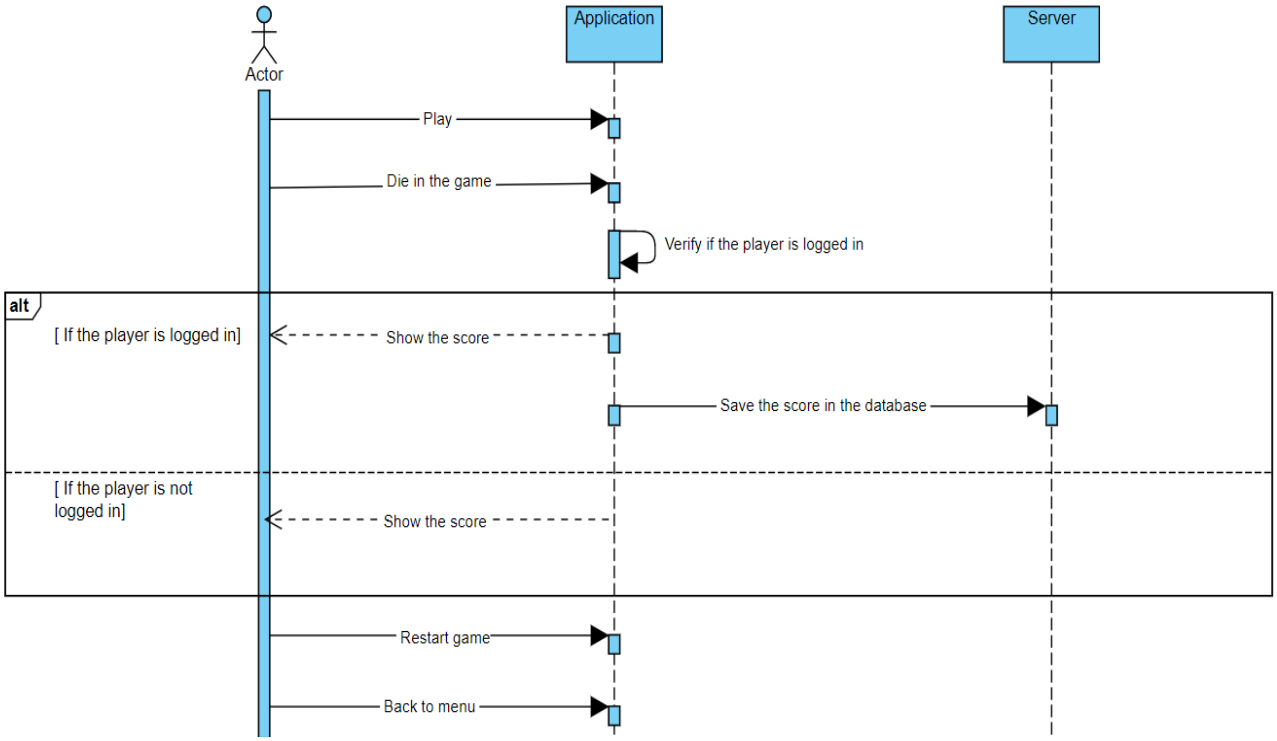
4.1.2. Bejelentkezés



4. Ábra

A bejelentkezés során a felhasználó egy bejelentkezési kérést küld az applikációnak egy gomb lenyomásával, amely kéri a felhasználót a felhasználónév illetve jelszó megadására. A kívánt adatok megadásával és egy gomb lenyomásával az alkalmazás lekérdezi a szervertől, hogy létezik-e ilyen felhasználó és kód páros az adatbázisban, hogyha létezik akkor az applikáció bejelentkezteti a felhasználót a megadott adatok szerint. Ha nem megfelelőek a megadott adatok, akkor kiírja az applikáció a felhasználónak, hogy nem helyesek a megadott adatok és újból kéri az adatokat.

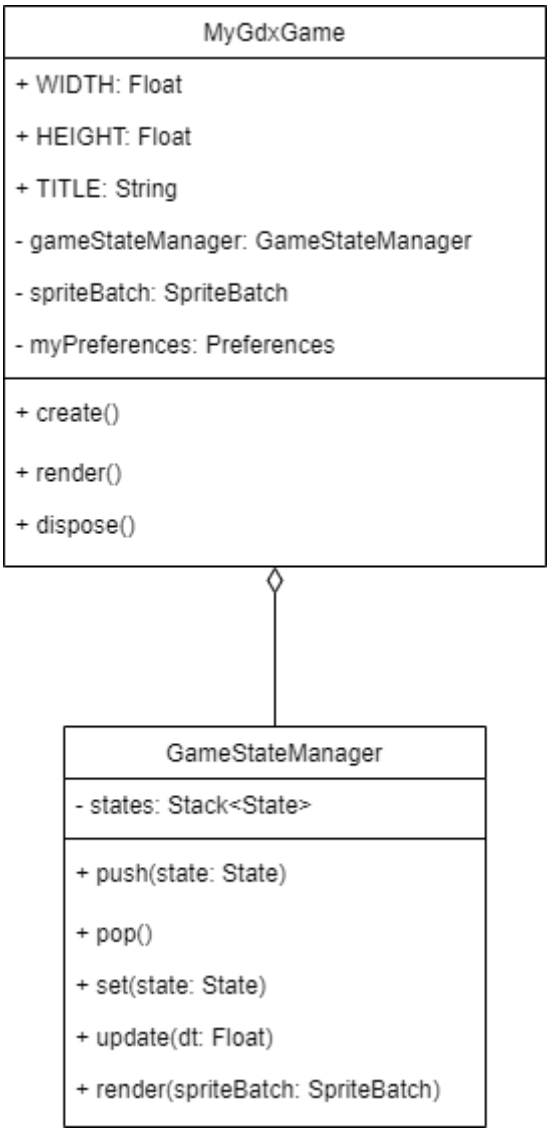
4.1.3 Játékmenet



5. Ábra

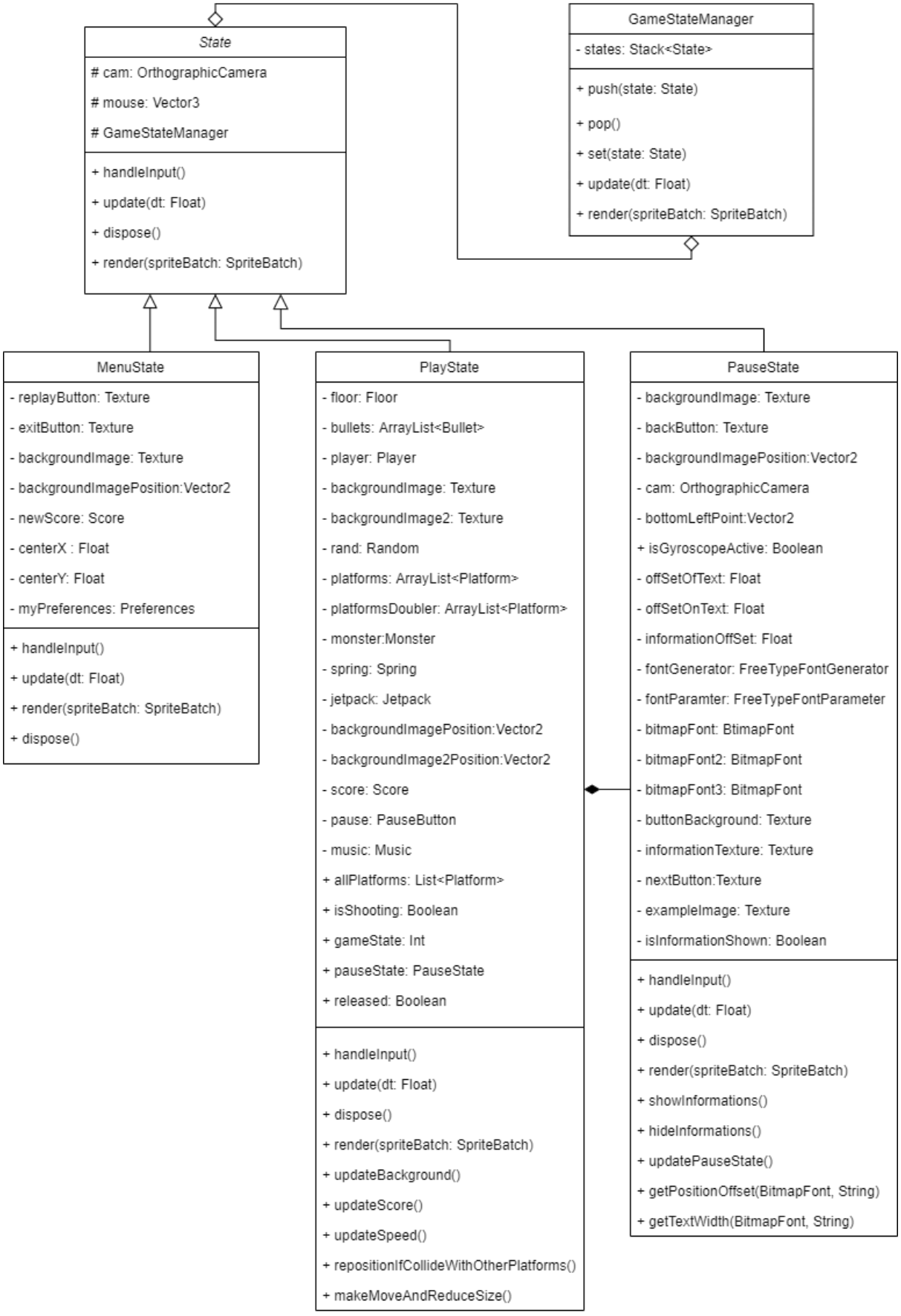
A felhasználó a “Play” gombra való kattintás során elindítja a játékot. Amikor a játék véget ér akkor az applikáció le ellenőrzi, hogy a felhasználó be van jelentkezve vagy sem, hogyha be van jelentkezve a felhasználó akkor megmutatja a felhasználónak a pontszámát és elmenti az adatbázisba a pontszámot a felhasználóhoz, ha nincs bejelentkezve akkor nem történik meg a mentés. Ezek után a felhasználó választhat hogy újrakezdi a játékot vagy visszatér a menübe.

4.2 Osztály diagramok



6. Ábra

A játék fő osztálya a MyGdxGame osztály, amelynek fő feladata létrehozni az elemeket, majd törölni azokat amikor véget ér az elemeinek a ciklusa. A legfőbb elem ebben az osztályban a GameStateManager, amely felel azért, hogy a felhasználó éppen milyen ablakot lát. Ebben a Managerben egy verem található amelybe “State”-eket lehet belerakni és kivenni belőle, mindig az látszik a felhasználó számára amely a verem tetején található.(6. Ábra)



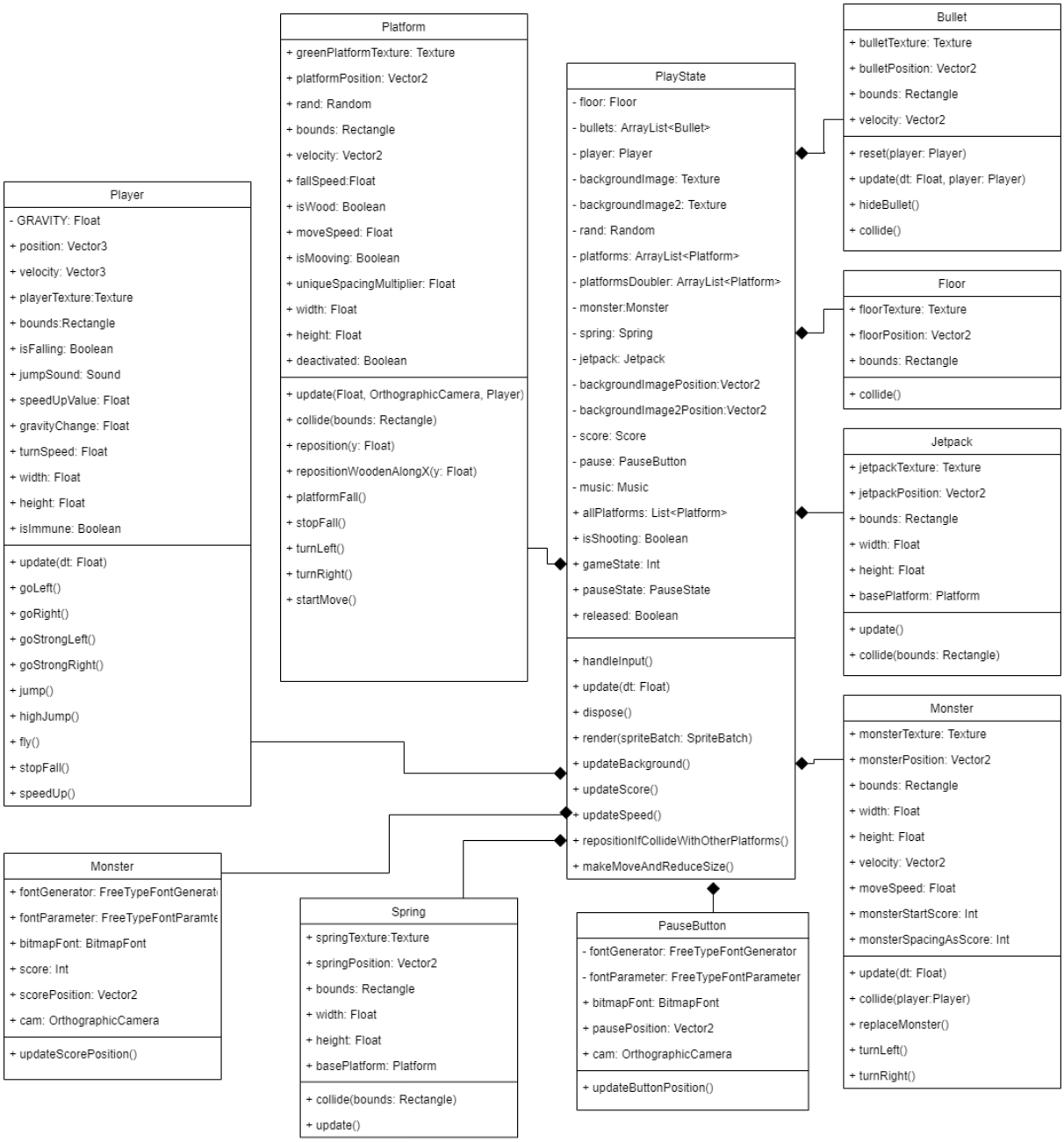
7. Ábra

A GameStateManager State típusú elemek vermét tartalmazza, ez a State egy absztrakt osztály, ebből származik három féle játékállapot(7. Ábra):

1. Menü állapot (Menu State)
2. Játék állapot (Play State)
3. Szünet állapot (Pause State)

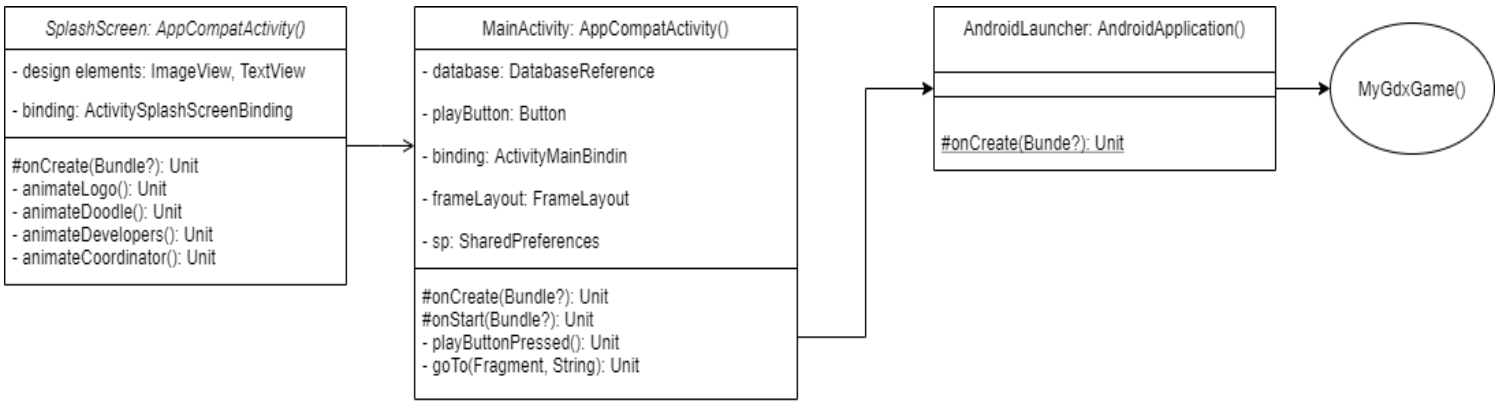
A Manager ezeknek az osztályoknak a példányait kezeli, teszi bele a verembe törli azokat.

Minden egyes állapotban megjelennek attribútumként azok az elemek amelyek kirajzolódnak a képernyőre, ezek mellett olyan segéd attribútumok és függvények amelyek ezeknek a megjelenéséért és interrakciójáért felelős.



8. Ábra

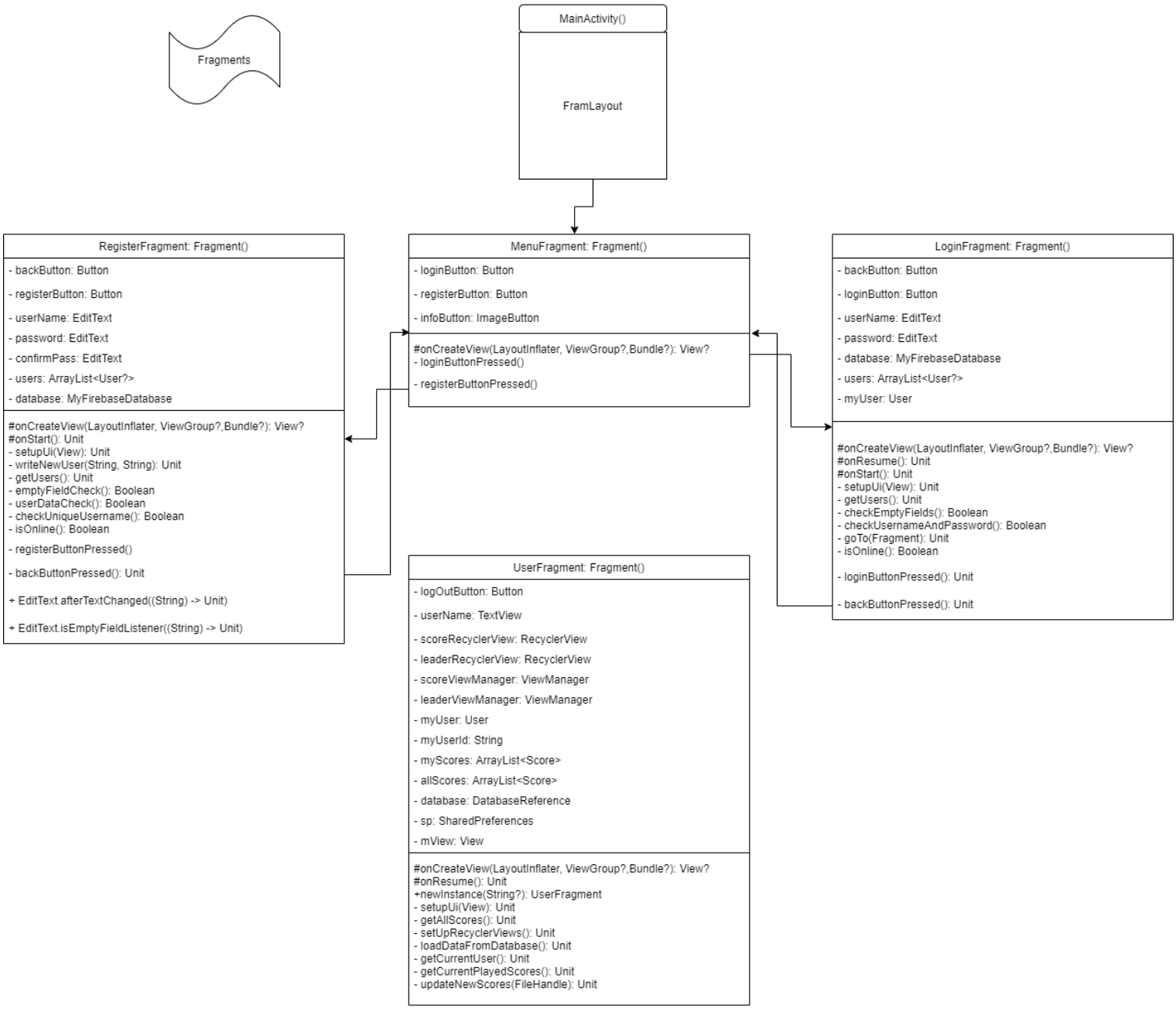
Az összes többi osztály a PlayState-hez kapcsolódik, mivel ez tartalmaz minden elemet, amely a játékban szerepel. (8. Ábra)



9. Ábra

Így néz ki az alkalmazás fő osztályainak diagramja (9. Ábra). Az alkalmazás indításával a egy kis animációnak lehetünk szemtanúi, ami nem csak egy egyszerű, szokványos “SplashScreen” hanem láthatunk némi információ az alkalmazás készítőiről is. Egy pár másodperces látványos animálás után átugrunk az alkalmazás menü részére, amit a “MainActivity” kezel. Ebben az osztályban fognak cserélődni a bejelentkezéshez, regisztrációhoz szükséges elemek. Lásd alább(10. Ábra).

Ugyanakkor ebben az osztályban található a játék indításához szükséges gomb, amit lenyomva egyből megkezdődik a játék, amit az “AndroidLauncher” oszály kezel.

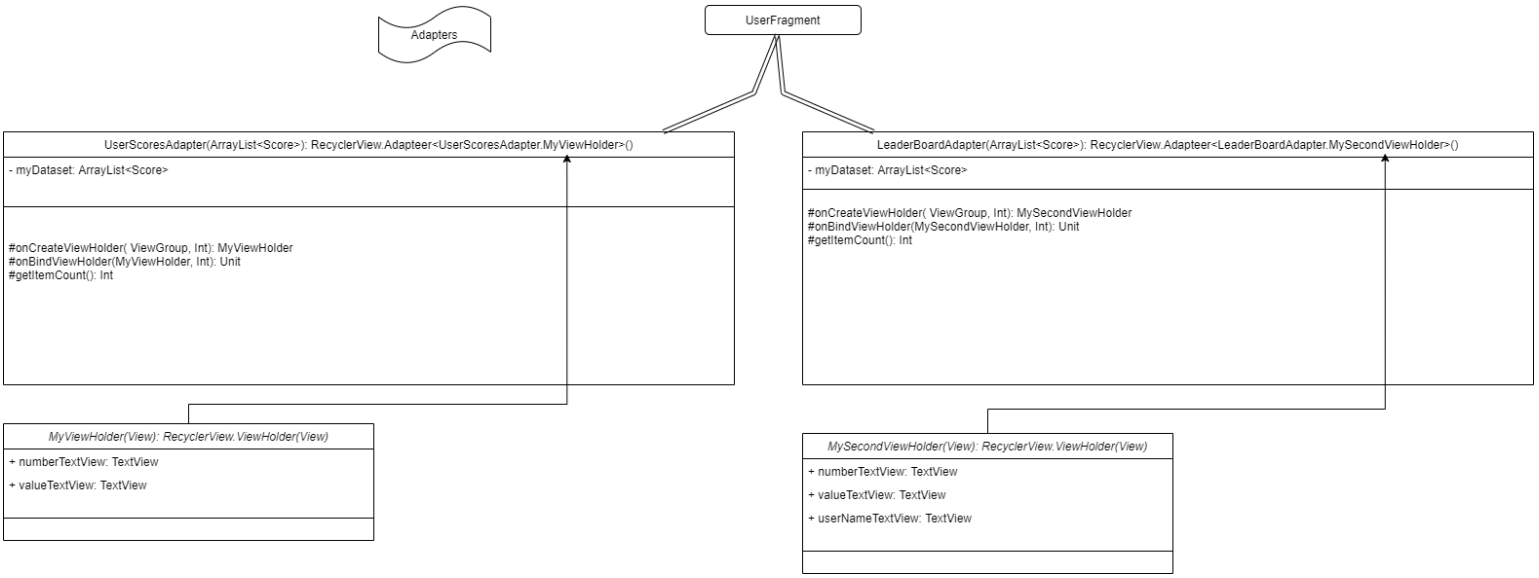


10. Ábra

Amint már mondtam, a “MainActivity” osztályban cserélődnek a különböző funkciókhoz szükséges osztályok (Fragmentek). A “MenuFragment” felelős azért, hogy navigálja át a felhasználót a LoginFragment, illetve RegisterFragment osztályba.

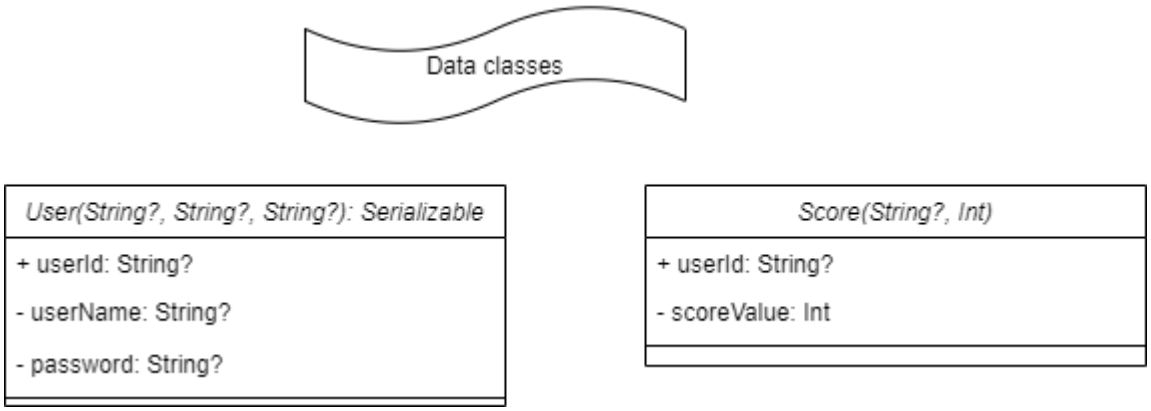
A LoginFragment felelős azért, hogy a felhasználó be tudjon jelentkezni a már meglévő fiókjával. A RegisterFragment pedig az új fiókok létrehozásáért felelős.

Mindkét osztály esetében, ha a felhasználó sikeresen bejelentkezett/regisztrált, akkor átugrunk a UserFragment osztályba, ahol a bejelentkezett fiókból kapunk adatokat. Ezek lehetnek: felhasználónév, eddigi pontszámok, mások pontszámai és a hozzájuk viszonyított helyezés, stb.



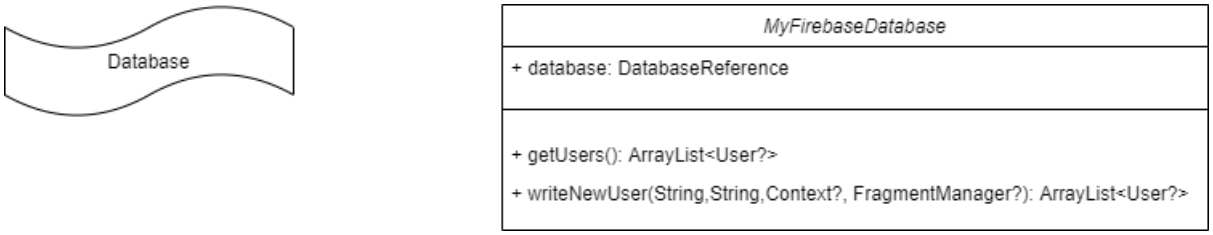
11. Ábra

A “UserFragment” osztályban két főbb osztály található, amik a felhasználó pontszámait és minden felhasználó pontszámainak rangsorolását és dinamikus, elő adatainak betöltéséért felelős. Ezek pedig a: UserScoresAdapter, LeaderBoardAdapter



12. Ábra

Két adat osztály segítségével történik meg az adatbázis adatok helyes tárolása. A User és Score osztályok megfelelnek egy egy tábla elemnek az adatbázisból.

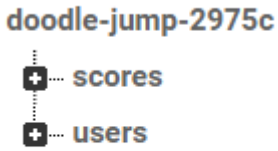


13. Ábra

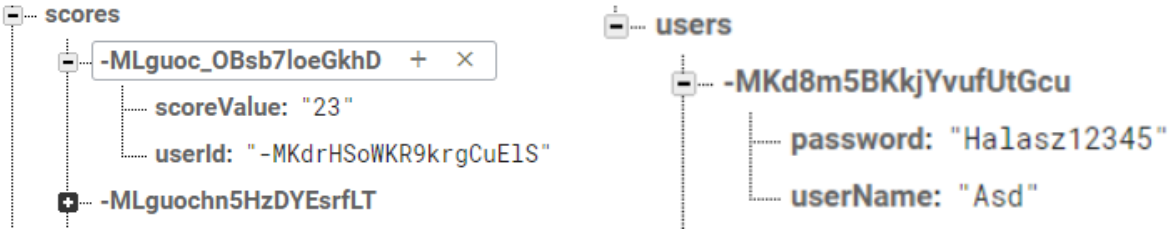
Kód szépsége és röviditése céljából készült a fenti osztály, ami az adatbázisból kéri és továbbítja az adatokat a már meglévő osztályokban (12. Ábra)

4.3 Adatbázis

A felhasználók adatainak tárolásához leginkább a mostanában feltörekvő Firebase adatbázist találtuk a legmegfelelőbbnek, mivel már ismertük ezért egyértelműen erre esett a választás. Azon belül is a ‘Realtime Databese’-t használtuk.



Így néz ki, amint látható két nagy táblája van, ez egy elem tartalmazó halmaz lényegében. A users táblában találhatóak a user elemek, illetve a scores -ban a score elemek. Lásd alább (14. ábra). Minden elem rendelkezik egy egyedi azonosító címmel.

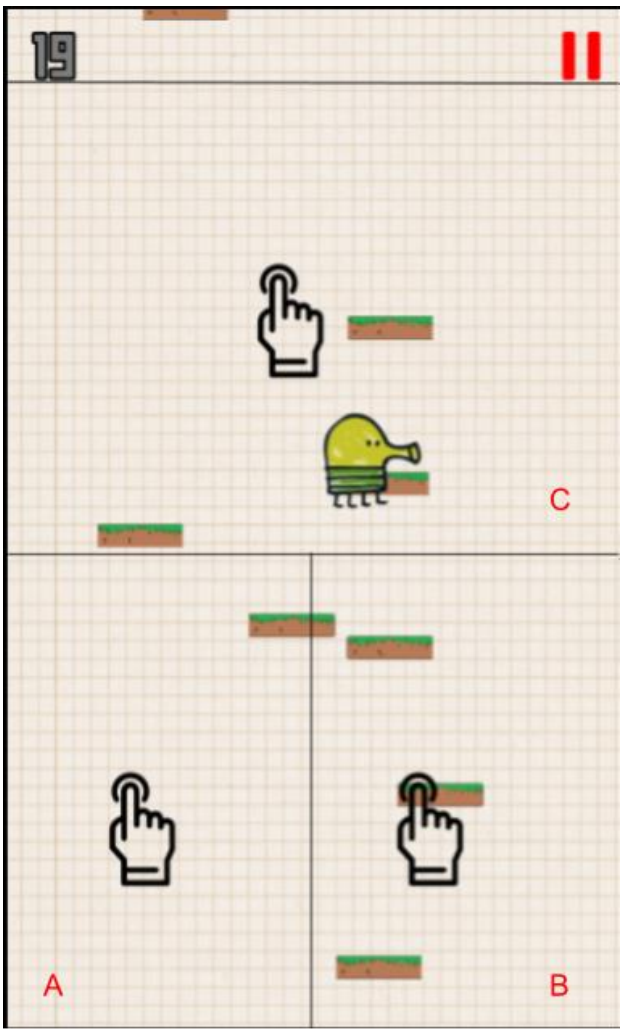


14. Ábra

5.Alkalmazás bemutatása

5.1 Játékmenet

A play gombra kattintva a felhasználó a játékba ugrik, ahol hozzáérve a képernyőhöz azonnal kezdődik a játék és a karakter elkezd ugrani.



15. Ábra

A játék során a játékosnak irányítania kell a karaktert a füves szigetek fele, azokra rálépve a karakter ismét ugrik és ezáltal jut a játékos minél fennebb.

A karakter irányítása a képernyő alsó felének a megnyomásával történik amelyet a 15. Ábra az A illetve B régió jelöl.

Az A régióra való érintést követően a karakterünk balra, a B régiót követően jobbra fog fokozatosan haladni.

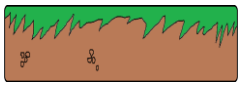
A C régióra való érintésre a karakterünk lőni fog, ezzel fogja tudni majd a szörnyeket elpusztítani.

A felfelé ugrálás során a karakter fokozatosan kap pontokat, amelyet a jobb felső sarokban láthat a játékos, amely valós időben változik a játék menete során.

A jobb felső sarokban található a szünet gomb amely tovább irányítja a felhasználót a szünet felületre, eközben az játék megáll, innen fogja tudni folytatni a játékot a visszatéréskor a felhasználó.

A játék pálya végtelen, tehát addig képes a játékosunk felfelé haladni ameddig csak bírja.

A szigetek megjelenési helye véletlenszerű, viszont mindig lesz olyan sziget amelyet a karakterünk képes elérni a fentebb jutás érdekében.



A játékmenet során a különböző nehézségekkel kell megvívnia a játékosnak a továbbhaladás érdekében:

- A legfőbb nehézség az, hogy ahogy a felhasználó fennebb halad,pontosabban minden 100 pontszám után nehezedik a játék tehát a játékmenet felgyorsul, érzékenyebb lesz a karakter irányítása.

- Bizonyos pontszámok elérése után ritkulnak a szigetek, tehát kevesebb sziget lesz amire tud majd a játékos lépni, bizonyos szigetek meg elkezdenek mozogni faltól falig, ezzel is nehezítve a játékos dolgát hogy ráirányítsa a karakterét.

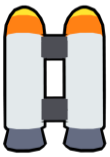
- Jelennek meg közben olyan szigetek amelyek fából vannak, ezek veszélyesek, mivel a karakterünk nem képes arról ismét elrugaszkodni és ha rálépik a játékosunk akkor leeshet és a játéknak vége.

- Legelőssző̋r 200 pontszámnál, majd ezután minden 500 elért pontszámnál a karakterünk találkozik egy szörnyeteggel amely 500 pontszám előtt áll egy helyben, majd az fölött mozog. Ezt a szörnyet kénytelen a játékosunk kilőni vagy kikerülni őt a továbbhaladás miatt. A szörnyeteget úgy is megölheti a karakterünk ha fentről szökik rá.

A játékos a játékmenet közben talál néhány elemet, amelyek segítik őt a továbbhaladásban:

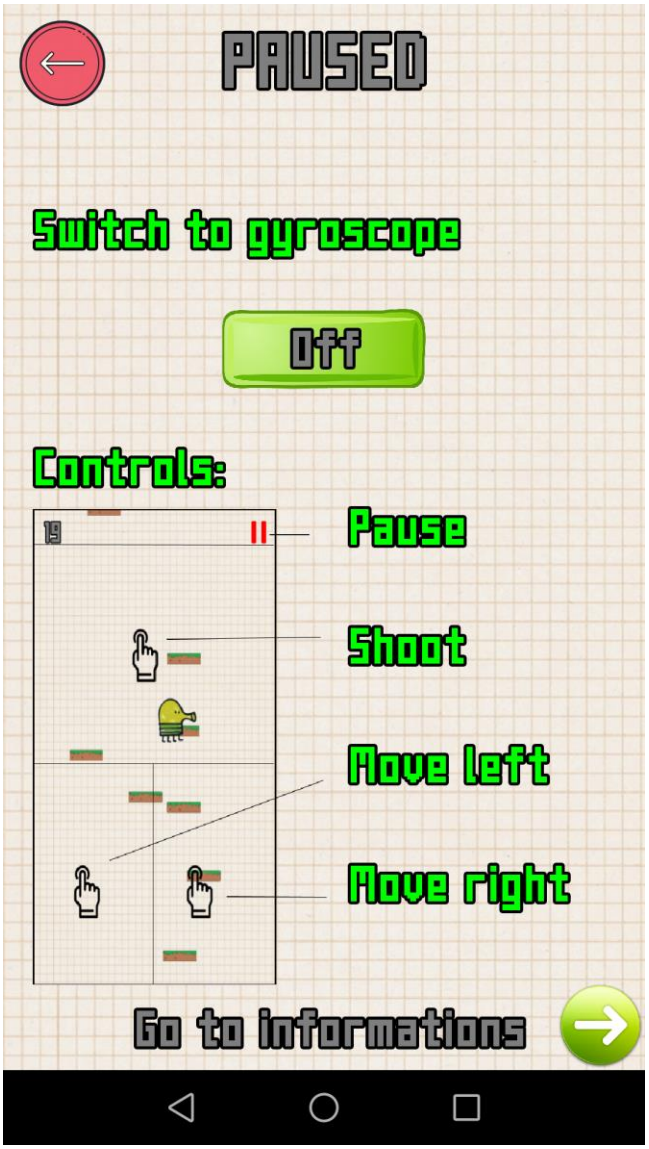
- Minden 200 pontszám után megjelenik az egyik szigeten egy rugó, amelyre rálépve nagy távokat ugorhat felfelé a karakterünk.

- Kezdetben 100 pontszám körül utána 1000 pontszám elérése után egy hát rakéta jelenik meg melynek segítségével nagyon sokat halad karakterünk felfele, és közben sebezhetetlen, nem tudja a szörny megölni őt miközben a rakéta a hátán van.



5.2. Szüneteltetés

A játékmenetben a szünet gombra kattintva érkezik a felhasználó a szüneteltetés felületére, itt olvashatja el, hogyan kell irányítani a karaktert illetve változtatni tudja az irányítás módját.



16. Ábra

A szünet ablak (16. Ábra) bal felső piros gombjára kattintva tud a felhasználó visszalépni a játékmenetre, folytatni a játékot ott ahol szüneteltette.

A szünet ablak közepén felül található egy gomb, amelyre rá kattintva tud a felhasználó váltani aközött, hogy a karaktert érintéssel működteti, vagy kihasználja a telefonja gyroscope-ját és döntéssel fogja irányítani tovább a karaktert.

Az ablak további részében egy ábrát láthat a felhasználó, amely megmutatja a kezdő játékosoknak, hogy hogyan tudnak elindulni, irányítani a karaktert, lőni illetve megállítani a játékmenetet.

A jobb alsó sarokban található egy gomb amely átirányít egy olyan ablakba amely további információkat mondd el nekünk a játék elemeivel kapcsolatban.

5.3. Információk

A szünet ablakon belül az információ gombra kattintva a felhasználó egy olyan ablakot nyithat meg, amelyben fel van sorakoztatva minden egyes elem amely a játékban megjelenik egy rövid tömör leírással. (17.Ábra)

Az információ ablak bal felső sarkában lévő gomb vissza irányítja a felhasználót a szünet ablakra.

17.Ábra

5.4. Játék vége

Ha a felhasználó által vezérelt karakter leesik a pályáról vagy valamilyen módon a karakter meghal, akkor érkezünk meg a játék végéhez.(18.Ábra)

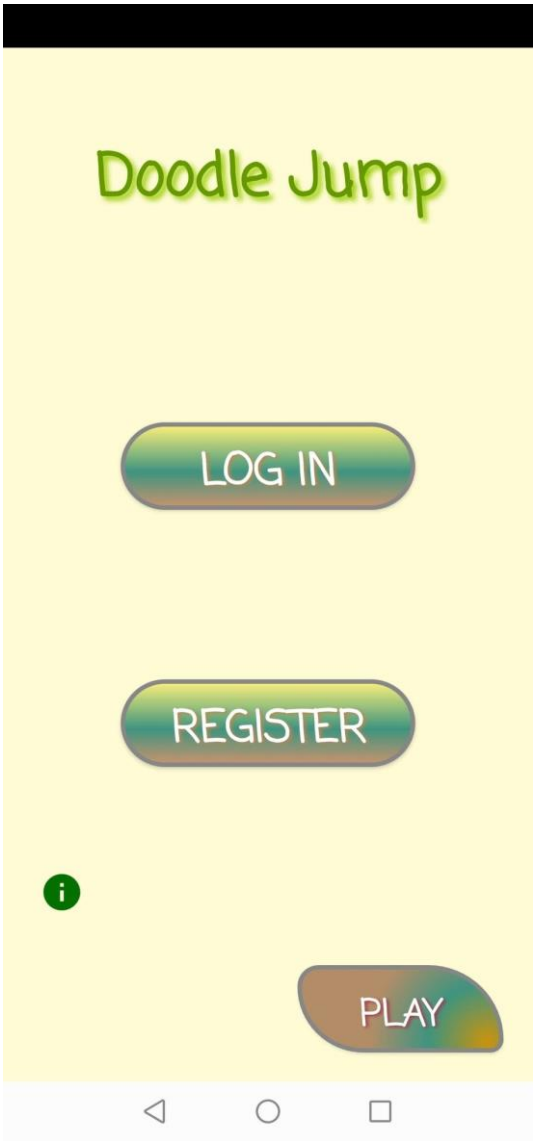
18.Ábra

Amint látható, a felhasználó megtekintheti az elért pontszámát és innen tovább két lehetősége van:

1. Megnyomja a középen található újraindítás gombot amivel megkezdődik egy új játék

2. A bal alsó sarokban található zöld gombbra nyomva visszaérkezünk az alkalmazás fő menüjébe.

5.5. Menü



19.Ábra

A SplashScreen animációjának befejeztével illetve a játékból való kilépéskor térhetünk át erre a felületre.(19.Ábra) Ha a felhasználó már be volt jelentkezve, akkor nem ide hanem a felhasználói felületre ugrunk automatikusan.

Több gomb is található ezen az oldalon, külömböző funkcionalitásokkal:

1. LOG IN: - átugrunk a bejelentkező oldalra (LoginScreen)
2. REGISTER: - átugrunk a regisztráló oldalra (RegisterScreen)
3. Info: - a kis zöld ‘i’ gomb lenyomásával egy fontos információt olvashatnak el a felhasználók: “Please login for saving your scores!”, ami annyit tesz, hogy “Kérjük jelentkezzen be a pontszámainak elmentéséhez”

- ez azért fontos, mert ha a felhasználó úgy játszik, hogy nincs bejelentkezve, akkor az addig játszott pontszámok elvesznek

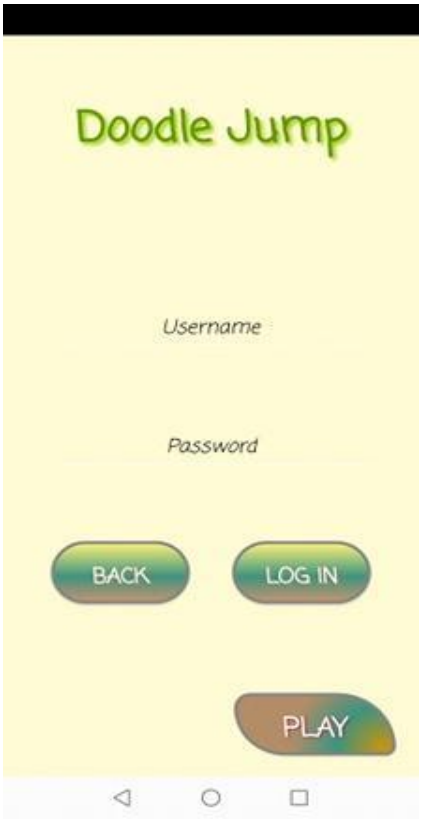
A jobb alsó sarokban található “PLAY” gomb mindenhol megtalálható, arra nyomva indul a játék

5.6. Loccsanás/Splash

Az előbb említett SplashScreen az animáció lejártával kis ikonokkal szemlélteti a projekten munkálkodó embereket, fejlesztőként Gergely Zsoltot, Halász Botondot és koordinátorként pedig Szántó Zoltánt.(20.Ábra)

20.Ábra

5.6. Bejelentkezés



21.Ábra

A felhasználónak két opciója van:

1. Emlékszik az adataira (Username, Password) és sikeresen bejelentkezik
2. Visszalép

Az felhasználónév és jelszó megadása után. A “login” gombra a háttérbe letesztelődik, hogy az adatok helyesek-e. Rossz adatok esetén a felhasználó szembesül a hibás adattal különböző hibaüzenetek által.

A játékot innen is elindíthatjuk de nem leszünk bejelentkezve!

22.Ábra

5.7 Regisztráció

Minden olyan felhasználónak amelyiknek még nincs fiókja, itt a helye!(22.Ábra)

Ez az felület felelős az új fiókok létrehozásáért.

Az adatok megadása után, ha minden megfelel a megadott kritériumoknak (egyedi felhasználónév, hossz: 5-10 karakter, minimum 8 karakteres jelszó,stb), akkor a felhasználó sikeresen létrehozhat egy új fiókot. Továbbá átugrunk a bejelentkezett felhasználók oldalára

23.Ábra

5.8 Bejelentkezett Felhasználó

Egy kis játszadozás után minden felhasználónak így fog kinézni a fiók oldala.

Látható a saját felhasználó neved, az eddig elért saját pontszámok sorrendben, illetve minden felhasználó pontszáma nagyságrend szerint sorrendben.

A különböző felhasználóknak lehetősége adódik egymás ellen versengeni a pontszámok alapján.

Az applikációt előzőleg használt bejelentkezett felhasználót azonnal erre az oldalra fogja dobni, automatikus bejelentkezés.

“LOG OUT” - kijelentkezés

6.Verziókövetés

Az applikáció elkészítéséhez GitHub verziókövető rendszert és a GitHub KanBan board-ot használtunk a könnyebb kezelhetőség érdekében, ennek segítségével tudtuk követni, hogy melyikünk hogyan haladt, milyen funkciókat valósított meg az alkalmazásban és, hogy még milyen elemeket kell leimplementálnunk.

A verziókövetésben azt a metódust követtük, hogy minden egyes új feature-t új branch-en valósítunk meg, amikor az sikeresen működik az applikációnk már meglévő részével, akkor azt beleolvasztjuk a késztermékbe így fokozatosan felépült az applikációnk. (24.Ábra)

7.Továbbfejlesztési lehetőségek

Az alkalmazás elnyerte tetszését annak a néhány embernek akinek megmutattuk és használták, viszont vannak még lehetőségek amelyekkel élvezetesebbé tudnánk tenni a játékot:

- További platform(sziget) típusok, amelyek különböző módon reagálnak amikor a játékos hozzájuk ér.
- Más pályák, témák választása.
- További szörnyek hozzáadása.
- Szörnyeknek adni képességeket.
- Több játékos üzemmód, amely során valós időben játszanánk ismerőseinkkel.
- Pontszámok megosztása ismerősöknek.
- Értesítések küldése, ha valaki megdönti rekordot.
- Bejelentkezés fejlesztése, email visszaigazolás

- Profile beállítások, jelszócsere, névcsere

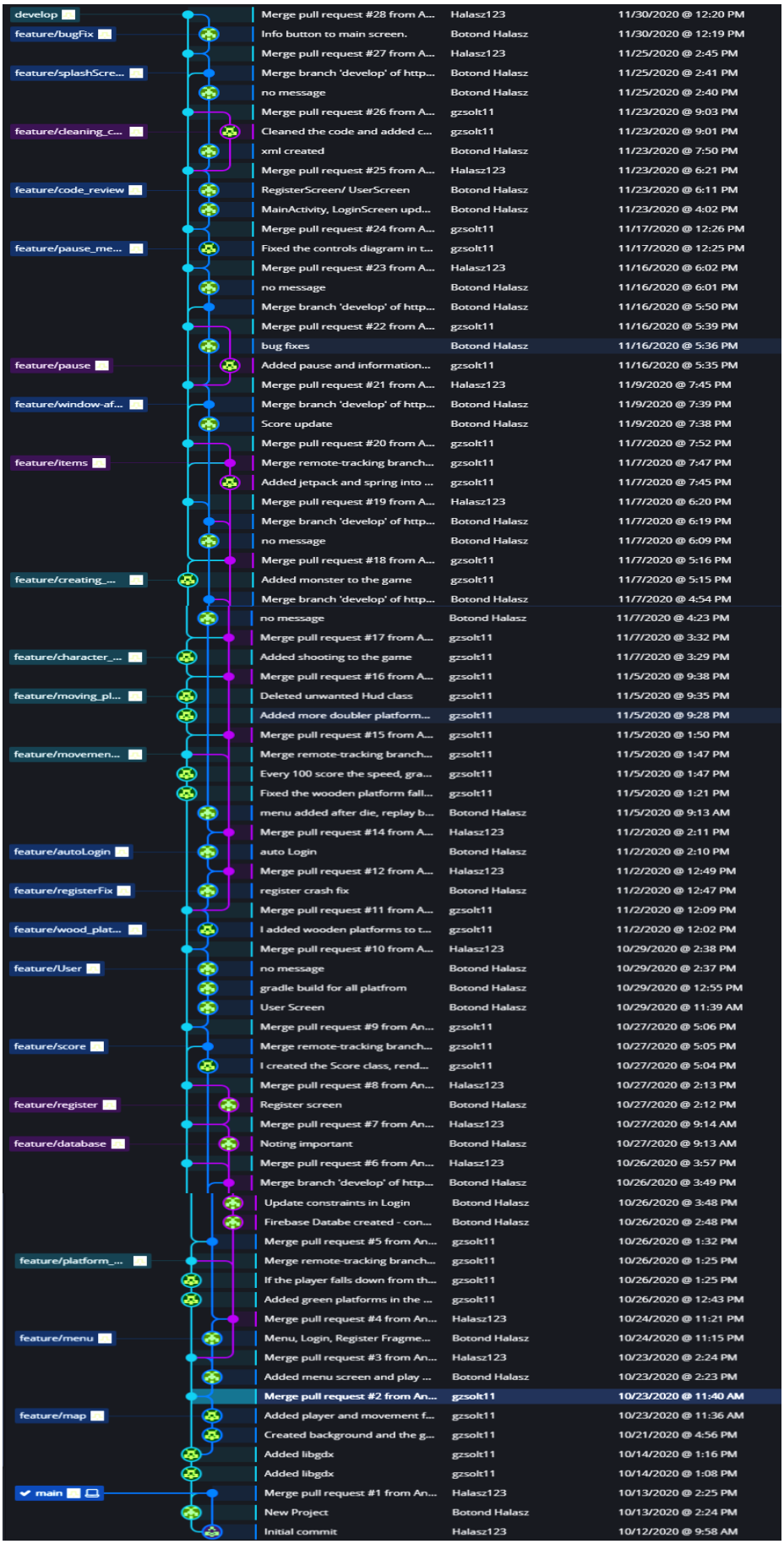
8.Összegzés

A projekt megvalósítása közben megtapasztaltuk, hogy milyen csapatban dolgozni, azt hogy hogyan tudjuk megoldani a munkamegosztást. Világos lett számunkra mennyire fontos a dolgokat megbeszélni, egymás véleményét kikérni bizonyos dolgok kapcsán így sokkal könnyebb volt haladni, ezáltal elkerültük a konfliktusokat.

Úgy gondoljuk tudás szempontjából is nagy mértékben gyarapodtunk, mivel mindketten értettünk az android programozáshoz de idáig még egyikünk sem csinált egy komolyabb játékot illetve egyikünk sem használta a LibGDX könyvtárat, amely segítségével egyszerűbben tudtuk megoldani a játék létrehozását. Mindketten megtanultuk, hogy mik a fő elemek ebben a könyvtárban, hogyan kell az elemekkel dolgozni és azt, hogy hogyan kell egy játék logikáját megvalósítani, mint például a gravitáció, mozgások, bizonyos felhasználói interakciók kezelése és a telefon által nyújtott eszközök használata mint például az érintőképernyő és a gyroscope.

9.Irodalomjegyzék

https://libgdx.badlogicgames.com/
https://gamefromscratch.com/libgdx-tutorial-series/
https://developer.android.com/docs
https://www.youtube.com/watch?v=DK1sGc4rOT4
https://www.youtube.com/watch?v=a8MPxzkwBwo&list=PLZm85UZQLd2SXQzsF-a0-pPF6IWDDdrXt
https://firebase.google.com/docs/android/setup/?gclid=EAIaIQobChMInKPWs7uy7QIVCJ3VCh0mRQf9EAAYASAAEgIfAfD_BwE


develop
feature/bugFix
feature/splashScre...
feature/cleaning_c...
feature/code_review
feature/pause_me...
feature/pause
feature/window-af...
feature/items
feature/creating_...
feature/character_...
feature/moving_pl...
feature/movemen...
feature/autoLogin
feature/registerFix
feature/wood_plat...
feature/User
feature/score
feature/register
feature/database
feature/platform_...
feature/menu
feature/map
main
Merge pull request #28 from A... Halasz123 11/30/2020 @ 12:20 PM
Info button to main screen. Botond Halasz 11/30/2020 @ 12:19 PM
Merge pull request #27 from A... Halasz123 11/25/2020 @ 2:45 PM
Merge branch 'develop' of http... Botond Halasz 11/25/2020 @ 2:41 PM
no message Botond Halasz 11/25/2020 @ 2:40 PM
Merge pull request #26 from A... gzsolt11 11/23/2020 @ 9:03 PM
Cleaned the code and added c... gzsolt11 11/23/2020 @ 9:01 PM
xml created Botond Halasz 11/23/2020 @ 7:50 PM
Merge pull request #25 from A... Halasz123 11/23/2020 @ 6:21 PM
RegisterScreen/ UserScreen Botond Halasz 11/23/2020 @ 6:11 PM
MainActivity, LoginScreen upd... Botond Halasz 11/23/2020 @ 4:02 PM
Merge pull request #24 from A... gzsolt11 11/17/2020 @ 12:26 PM
Fixed the controls diagram in t... gzsolt11 11/17/2020 @ 12:25 PM
Merge pull request #23 from A... Halasz123 11/16/2020 @ 6:02 PM
no message Botond Halasz 11/16/2020 @ 6:01 PM
Merge branch 'develop' of http... Botond Halasz 11/16/2020 @ 5:50 PM
Merge pull request #22 from A... gzsolt11 11/16/2020 @ 5:39 PM
bug fixes Botond Halasz 11/16/2020 @ 5:36 PM
Added pause and information... gzsolt11 11/16/2020 @ 5:35 PM
Merge pull request #21 from A... Halasz123 11/9/2020 @ 7:45 PM
Merge branch 'develop' of http... Botond Halasz 11/9/2020 @ 7:39 PM
Score update Botond Halasz 11/9/2020 @ 7:38 PM
Merge pull request #20 from A... gzsolt11 11/7/2020 @ 7:52 PM
Merge remote-tracking branch... gzsolt11 11/7/2020 @ 7:47 PM
Added jetpack and spring into ... gzsolt11 11/7/2020 @ 7:45 PM
Merge pull request #19 from A... Halasz123 11/7/2020 @ 6:20 PM
Merge branch 'develop' of http... Botond Halasz 11/7/2020 @ 6:19 PM
no message Botond Halasz 11/7/2020 @ 6:09 PM
Merge pull request #18 from A... gzsolt11 11/7/2020 @ 5:16 PM
Added monster to the game gzsolt11 11/7/2020 @ 5:15 PM
Merge branch 'develop' of http... Botond Halasz 11/7/2020 @ 4:54 PM
no message Botond Halasz 11/7/2020 @ 4:23 PM
Merge pull request #17 from A... gzsolt11 11/7/2020 @ 3:32 PM
Added shooting to the game gzsolt11 11/7/2020 @ 3:29 PM
Merge pull request #16 from A... gzsolt11 11/5/2020 @ 9:38 PM
Deleted unwanted Hud class gzsolt11 11/5/2020 @ 9:35 PM
Added more doubler platform... gzsolt11 11/5/2020 @ 9:28 PM
Merge pull request #15 from A... gzsolt11 11/5/2020 @ 1:50 PM
Merge remote-tracking branch... gzsolt11 11/5/2020 @ 1:47 PM
Every 100 score the speed, gra... gzsolt11 11/5/2020 @ 1:47 PM
Fixed the wooden platform fall... gzsolt11 11/5/2020 @ 1:21 PM
menu added after die, replay b... Botond Halasz 11/5/2020 @ 9:13 AM
Merge pull request #14 from A... Halasz123 11/2/2020 @ 2:11 PM
auto Login Botond Halasz 11/2/2020 @ 2:10 PM
Merge pull request #12 from A... Halasz123 11/2/2020 @ 12:49 PM
register crash fix Botond Halasz 11/2/2020 @ 12:47 PM
Merge pull request #11 from A... gzsolt11 11/2/2020 @ 12:09 PM
I added wooden platforms to t... gzsolt11 11/2/2020 @ 12:02 PM
Merge pull request #10 from A... Halasz123 10/29/2020 @ 2:38 PM
no message Botond Halasz 10/29/2020 @ 2:37 PM
gradle build for all platfrom Botond Halasz 10/29/2020 @ 12:55 PM
User Screen Botond Halasz 10/29/2020 @ 11:39 AM
Merge pull request #9 from An... gzsolt11 10/27/2020 @ 5:06 PM
Merge remote-tracking branch... gzsolt11 10/27/2020 @ 5:05 PM
I created the Score class, rend... gzsolt11 10/27/2020 @ 5:04 PM
Merge pull request #8 from An... Halasz123 10/27/2020 @ 2:13 PM
Register screen Botond Halasz 10/27/2020 @ 2:12 PM
Merge pull request #7 from An... Halasz123 10/27/2020 @ 9:14 AM
Noting important Botond Halasz 10/27/2020 @ 9:13 AM
Merge pull request #6 from An... Halasz123 10/26/2020 @ 3:57 PM
Merge branch 'develop' of http... Botond Halasz 10/26/2020 @ 3:49 PM
Update constraints in Login Botond Halasz 10/26/2020 @ 3:48 PM
Firebase Databe created - con... Botond Halasz 10/26/2020 @ 2:48 PM
Merge pull request #5 from An... gzsolt11 10/26/2020 @ 1:32 PM
Merge remote-tracking branch... gzsolt11 10/26/2020 @ 1:25 PM
If the player falls down from th... gzsolt11 10/26/2020 @ 1:25 PM
Added green platforms in the ... gzsolt11 10/26/2020 @ 12:43 PM
Merge pull request #4 from An... Halasz123 10/24/2020 @ 11:21 PM
Menu, Login, Register Fragme... Botond Halasz 10/24/2020 @ 11:15 PM
Merge pull request #3 from An... Halasz123 10/23/2020 @ 2:24 PM
Added menu screen and play ... Botond Halasz 10/23/2020 @ 2:23 PM
Merge pull request #2 from An... gzsolt11 10/23/2020 @ 11:40 AM
Added player and movement f... gzsolt11 10/23/2020 @ 11:36 AM
Created background and the g... gzsolt11 10/21/2020 @ 4:56 PM
Added libgdx gzsolt11 10/14/2020 @ 1:16 PM
Added libgdx gzsolt11 10/14/2020 @ 1:08 PM
Merge pull request #1 from An... Halasz123 10/13/2020 @ 2:25 PM
New Project Botond Halasz 10/13/2020 @ 2:24 PM
Initial commit Halasz123 10/12/2020 @ 9:58 AM

24.Ábra