

Pannon Egyetem

Műszaki Informatikai Kar

Rendszer- és Számítástudományi Tanszék

Mérnökinformatikus MSc

DIPLOMAMUNKA

"Free to play, run to win" játék Androidra

Nyitrai Tamás

Témavezető: Dr. Hegyháti Máté

2016

KÖSZÖNETNYILVÁNÍTÁS

Meg szeretném köszönni édesanyámnak támogatását és folyamatos bátorítását. Továbbiakban köszönettel tartozom témavezetőmnek, Dr. Hegyháti Máténak, aki ötleteivel és tanácsaival végig a helyes úton tartott.

Továbbá szeretném megköszönni Hollósi Tamásnak a dream-iso-android elkészítőjének a segítségét, akihez bátran fordulhattam, ha bármi kérdésem volt a játékmotorral kapcsolatban. Valamint köszönöm Böröndi Evelinnek a grafikák elkészítésében vállalt segítségét.

TARTALMI ÖSSZEFOGLALÓ

Tartalmi összefoglaló...

Kulcsszavak: dream-iso-droid, Android, testmozgás, szerepjáték

ABSTRACT

Angol tartalmi összefoglaló...

Keywords: dream-iso-droid, Android, sports, RPG game

Tartalomjegyzék

1. Bevezetés	6
2. Hasonló fejlesztések, szerepjátékokról általánosan	8
2.1. A videójátékok története	8
2.2. Sportjátékok	9
2.3. Az RPG játékokról általánosságban	10
3. Követelmények, technológiák	12
3.1. Követelmények	12
3.2. Felhasznált technológiák	14
4. Fejlesztői dokumentáció	16
4.1. Architektúra	16
4.2. A játék indítása	17

Ábrák jegyzéke

3.1. Játékmotorok összehasonlítása	15
4.1. A program architektúrája	17
4.2. Az OAuth protokoll absztrakt működési ábrája	18

1. fejezet

Bevezetés

Az okos-telefonok térhódítása miatt már hazánkban is a lakosság fele rendelkezik valamilyen okos eszközzel, ez a szám pedig a jövőben egyre csak növekedni fog. A mindennapi élet megkönnyítésére rengetegféle alkalmazás születik napról napra. Egyetlen eszközön olvashatunk újságot, tudakozódhatunk a közlekedésről, vehetünk ebédet magunknak, vagy unaloműzőként játszhatunk. Manapság minden korosztály talál kedvére való játékot, legyen szó akár ingyenes akár fizetős verzióról. Napjainkban igencsak elterjedtek az olyan játékok ahol a felhasználók interakciókba léphetnek egymással. Ezek túlnyomó többsége az úgynevezett micro-paymentekre alapszik, ahol a játékosok csekély összegekért cserébe előnyökhöz, könnyítésekhez juthatnak. A „free to play, pay to win” kifejezést azokra a játékokra szokták használni, ahol az előbb említett vásárlások nélkül képtelenség megnyerni a játékot, mert túlzottan befolyásolják a játékosok fejlődését.

Egyre több ember életéből hiányzik napjainkban a rendszeres testmozgás. Sokan választják a séta és a bicikli helyett az autós vagy a tömegközlekedést, főleg kényelmi szempontokból. A technológia fejlődésével pedig egyre több olyan eszköz jön létre, amik az emberek életének kényelmesebbé tételét szolgálja. Ha rendelkezünk okostelefonnal, ma már a nagybevásárlást is el tudjuk intézni pár kattintással otthonról. Az ilyen alkalmazások célja az volt, hogy kényelmesebbé tegyék mindennapjainkat, nem azt, hogy elkényelmesítsenek minket. Az, hogy egyre több időt töltünk ezen eszközök előtt, csak súlyosbítja rendszeres testmozgás hiányát. Sajnálatos módon a felhasználók nagy része nem elégszik már meg annyival, hogy a mozgás jól tesz az egészségének, szükség lehet valamilyen fajta ösztönző módszerre. Ezek többféle módon is megnyilvánulhatnak, léteznek különböző virtuális díjazások, sportolásért járó pontok, amik később tárgynyerményekre válthatóak, sőt vannak tényleges pénzzel való díjazások is.

Célom a két trend összekapcsolása oly módon, hogy a játékban a gyorsabb fejlődést nem pénzkifizetéssel, hanem sportolással lehet kiváltani.

A 2. fejezetben ismertetem az elkészülő játék háttérét, bemutatok különböző játéktípusokat.

A 3. fejezetben ismertetem a program tervezett funkcionalitását és követelményeit. Illetve kitérek a játék egy fontos alapelemére az általam választott dream-iso-droid játékmotorra.

A 4. fejezetben bemutatom az elkészült alkalmazást felhasználói szempontból.

Az 5. fejezetben részletesen kifejtem a megvalósított játék fő alkotóelemeit, és azok implementációját.

[maradék fejezet]

2. fejezet

Hasonló fejlesztések, szerepjátékokról általánosan

A fejezetben részletesen bemutatásra kerülnek a fejlesztett játék elkészítését befolyásoló tényezők, és hátterük.

2.1. A videójátékok története

Videójátékoknak nevezzük azokat a típusú játékokat, ahol a játékosok egy felhasználói felületen keresztül lépnek interakcióba a játékkal. Története az 1950-es évekre vezethető vissza, ebben az évtizedben kezdte el foglalkoztatni az embereket az az ötlet, hogy a szórakozás élményét elektromos eszközök segítségével ériék el. Az első videójáték egy asztalitenisz szimulátor volt, 1958-ban készítette el William Higinbotham és a „Tennis for Two” nevet viselte. A játék egy asztali analóg számítógépre készült el, és egy oszcilloszkópot használt a megjelenítésre. Az első olyan játék, amelyet már egy kontroller ősenek nevezhető eszközzel irányítottak 1962-ben készült el, és a Spacewar nevet kapta. Itt két űrhajót irányíthatott a játékos, a másik fél letorpedózása volt. A Spacewar változataként jelent meg a világ első pénzbedobós játéktermi gépe a Computer Space. Ennek ellenére ez a játék feledésbe merült, a népszerűséget az 1972-ben megjelent PONG játék szerezte meg. A játék annyira népszerű volt, hogy 3 év múlva az otthoni verziója is elkészült a játéknak. Ezt követően megindult a játéktermi gépek térhódítása a világon. Az Atari mellett a Sega és a Midway cégek is belekezdtek a saját játékgépeik fejlesztésébe. Ekkoriban mindenki be akart szállni a játékgyártásba, emiatt gyorsan sok videójátékot termeltek ki a vállalatok. Az 1980-as évek elején a sok silány minőségű játék miatt halottnak nyilvánították a videójáték piacot, mert a emberek nem kívántak rossz minőségű játékokat venni, ennek következtében csökkent a keresletük. Az első olyan videójátékot, ami történetet mesél el a Nintendo cégnek köszönhetjük.

A játék az 1981-ben kiadott Donkey Kong volt, ahol a ma is híres Mario karakterével kellett megmenteni a bajbajutott lányt. 1985-től a videójáték piac újból felemelkedett a Nintendo Entertainment System nevű otthoni játékkonzolnak köszönhetően. Egymás után jelennek meg különböző vállalatok játéakai, ezek között található volt stratégiai, verekedős továbbá kalandozós játékok is. Az 1990-es évektől kezdve folyamatosan jelennek meg a felfejlesztett játékkonzolok. A hozzájuk készült játékok nem csak a fajtájukban térnek el, hanem különböző megjelenésükkel és irányítási rendszerükkel egyedivé varázsolják a játékelményt. A közeljövőben elérhetővé válik az a játékforma, amikor a játékos érzékeit becsapják a képek és a hangok, és teljesen úgy fogja érezni, mintha belecsöppent volna a játék világába.

2.2. Sportjátékok

Napjainkban nagy népszerűségnek örvendenek az olyan játékok, ahol a játékosok interakcióba tudnak lépni egymással, ez lehet együttműködés vagy egymás elleni versengés is. Mivel a versenyszellem velünk született tulajdonságunk, ezért a felhasználók minden tőlük telhetőt megtesznek azért, hogy a ranglista élére kerülhessenek. Erre alapozva sok játékkészítő nagyon kreatívan határozta meg a fejlődés árát. Az alábbiakban pár olyan játékot mutatok be, amelyeknél ez az ár a testmozgás.

Zombies, Run!

A Zombies, Run! egy olyan népszerű futó alkalmazás, ahol a felhasználó egy zombi-apokalipszisbe csöppen bele túlélőként. Feladata, hogy minél több zsákmányt szerezzen futás közben, amivel egy bázist kell folyamatosan fejlesztenie, hogy az ott állomásozó túlélőket biztonságban tudhassa. A felhasználó kezdetben elindítja az alkalmazást, majd a futás közben elkezdődik a játék történetének mesélése. Ez nem folyamatos, a felhasználó tud közben zenét hallgatni.

Futás közben találkozhatunk zombikkal, melyeket vagy a futás közben talált ellátmányért cserébe rázhatunk le, vagy egy rövid ideig 20%-kal gyorsabban kell futni. Ez a folyamatos változás a futás közben egyfajta intervallum edzésnek felel meg, melynek igen sok előnyös tulajdonsága van. Ugyanakkor az alkalmazás önmagában követi nyomon a felhasználó sportolási tevékenységét, a témában elkészítendő játék ezzel szemben a már leportolt tevékenységeket jutalmazná, melyet külső sport nyomkövető alkalmazásokból kapna meg. Más eltérés is mutatkozik a témámtól, mégpedig a játék stílusa. Míg a tervezett játékom szerepjáték jellegű, a Zombies, Run! gyűjtögetős játék, mely főleg történet központú.

Tep

Meg kell továbbá említeni a Tep-et, amely magyar fejlesztésű. A Tep szintén motivációs sport-nyomkövető alkalmazás, mely a valós teljesítmények után ad jutalmat a játékban. A játék stílusa a népszerű Tamagotchi játékhöz hasonló, azaz egy virtuális állatkát kell gondoznunk mindennaposan. A kapott jutalmakat beválthatjuk a virtuális állatunk részére különböző étel, ital és dekoratív elemre is. Ez az alkalmazás is különálló sport-nyomkövetőként működik, azaz nem külső forrásból szerzi be az adatokat, ugyanakkor össze lehet kötni hordozható eszközökkel, a Fitbit-tel és a Jawbone eszközökkel. Annak ellenére, hogy a játék motivációs célt szolgál, a felhasználó kevésbé van ösztönözve a sportolásra, ugyanis ha nem sportol folyamatosan, az egyetlen változás, ami bekövetkezik, hogy ha az állatkát „simogatjuk”, akkor nem csóválja a farkát és éhezik az állat. Ezzel szemben az általam készített játék esetén amennyiben a felhasználó nem sportol, nem lesz képes fejlődni a játékban.

Pokémon Go

A Pokémon Go az RPG játékok egy speciális fajtába az MMORPG-be tartozik. A játék kizárólagosan csak mobil eszközre készült abból az okból kifolyólag, hogy közvetett vagy közvetlen módon sportolásra vagy legalább mozgásra ösztönözze az embereket. Emiatt szükség volt arra, hogy az eszköz, amin játsszanak, mobilis legyen. Mindezek mellett a játék félig a valóságban félig pedig a virtuális világban játszódik. A pontos pozíciónkat megjeleníti a térképen, ami a valós világ útjaira, épületeire alapszik. Annyiban viszont eltér, hogy a játék egyes elemeit például a pokémonokat (kitalált állatszerű lények) a virtuális világban a térképre helyezi, majd a felhasználónak a való világban fizikailag oda kell jutnia hozzá, hogy elkaphassa.

2.3. Az RPG játékokról általánosságban

A következő alfejezetben az RPG-t (role playing game), vagyis szerepjátékot fogom bemutatni. Ezek a fajta játékok arra épülnek, hogy a felhasználó egy karakter „szerepébe” bújik bele, őt irányítva végzi el a feladatokat, kalandozik a világban. Eredete az ókorba vezethető vissza, elődjének tekinthetők a különböző harci játékok amelyek az ütközetek szimulálására szolgáltak. Az első szerepjáték az 1974-ben megjelent Dungeons & Dragons volt, ami hasonló szabályrendszerrel rendelkezett, mint a napjainkban megjelenő RPG-k. Előnyük, hogy sok ember számára elérhetőek, ugyanis a játékhoz dobókockákra, papírra és képzelőerőre van szükség. A mesélőnek kinevezett személy vezeti végig a kalandozáson a többi szereplőt. Minden játékoshoz tartozik egy karakter, aki fölött rendelkezhet, illetve a karakterlapján

vezetheti a statisztikákat és jellemzőket. Egy elvégzett feladat vagy küldetés után különböző jutalmakat kaphatnak a karakterek, amelyek fejlődésük során egyre erősebbek lesznek, emiatt pedig sikerül elérniük a játék elején kitűzött céljukat.

Nem kellett sok idő ahhoz, hogy az RPG meghódítsa a számítógépes közeget is. Az 1970-es évek közepe után sorra jelentek meg a többfelhasználós kalandjátékok, amik a szerepjátékok szabályait követve nyújtottak szórakozási lehetőséget azoknak, akik rendelkeztek internetkapcsolattal. Ennek a mintájára napjainkban már nem csak webes felületen érhetőek el a hasonló típusú játékok, hanem az okostelefonok terjedésével, már mobil felületen is. A grafikai kártyák fejlődése következtében az utóbbi két játékforma grafikai elemek felhasználásával szimulálja a különböző akciót, amelyek a régi típusú szerepjátékban a képzeletre voltak bízva. Továbbá egy jelentős különbség még, hogy míg az eredeti szerepjátékokban a harcok kimenetele többnyire a szerencsén múlik, addig az online játékoknál különböző képletek és algoritmusok segítségével számolják ki, egy egy támadás mértékét. Feltehetőleg a komplexebb harcrendszer miatt alakult ki több változata a küzdelem lebonyolításának. Egy népszerű formája a „turn based” vagyis körökre osztott összecsapás. A játékos karaktere és az ellenfél felváltva támad, minden fél a saját körében dönt arról, hogy milyen cselekvést fog végrehajtani. Ez az akció lehet támadás, öngyógyítás vagy esetleg menekülési kísérlet. Ezen kívül vannak szimulált harcot implementáló játékok, ahol a harc kimenetele időközben nem befolyásolható. Itt az összecsapás az ellenfél és a saját karakter tulajdonságpontjainak a felhasználásával kerül kiszámításra. A játékos a végeredményt látja csak, hogy sikerült-e legyőzni az ellenséget vagy sem.

Továbbá különbséget tehetünk abban, hogy az online felületen játszható játékok hosszú távon tudnak szórakozási lehetőséget biztosítani, nem szükséges a játékosok folyamatos jelenléte. Az online szerepjátékokban jellemzően nincs kitűzött végcél, a felhasználók kalandoznak, fejlődnek és egyre erősödő ellenfeleket győznek le. Ha a játékban nincsenek maximálisan elérhető értékek definiálva, akkor csak a játékos kitartása és eltökéltsége szab határt a játék végének. A cél egy olyan játék megvalósítása volt, ami hosszú távon képes ösztönözni a felhasználót a sportolásra.

3. fejezet

Követelmények, technológiák

A fejezetben szó esik a szoftverrel szemben támasztott követelményekről, valamint a felhasznált programok technológiák kerülnek bemutatásra.

3.1. Követelmények

Az alkalmazással szemben különféle követelményeket támasztok, melyeknek mindenképp meg kell felelnie, melyek a következők:

- Motiválás:

Első és legfontosabb követelmény, hogy képes legyen az embereket ösztönöznie a sportolásra. Ezt minél érdekesebb és izgalmasabb játékmekanikai elemekkel kívánom elérni.

- Kiterjeszthetőség:

Az elkészítendő játék nem különálló tracker alkalmazásként fog működni, önmagában nem lesz képes mérni a sporttevékenységeket. Minden esetben más tracker szolgáltatások által elmentett tevékenységet fog lekérni az alkalmazás. Emiatt fontos, hogy minél több alkalmazástól tudjon lementett sporttevékenységeket lekérni. Amiatt is fontos lenne minél több tracker alkalmazás támogatása, mivel azon felhasználók, akik már régebb óta sportolnak, ne kelljen a számukra bevált sport nyomkövető szolgáltatást lecserélni.

- Érdeklődés fenntartás

Miután a játék felkeltette a felhasználó figyelmét, el kell érni, hogy tovább játsszon vele. Ahogy a korábbiakban említettem, a játékmekanikai elemekkel és különböző kihívásokkal szeretném megvalósítani, melyeket lejjebb fejtek ki bővebben.

- Gyors, sok eszközön, ezt majd ki kell fejteni....

Kifejtés..

Funkcionális követelmények

Az alábbiakban a főbb funkcionális követelmények kerülnek bemutatásra.

- A játéknak képesnek kell lennie csatlakozni sport-nyomkövető alkalmazásokhoz. Lehetőleg minél több alkalmazást kell támogatnia a játéknak, mivel így nagyobb lehet az elérhető potenciális játékos-közösség is.
- A minél nagyobb számú támogatottság elérése érdekében az alkalmazásnak mindenképp támogatnia kell az OAuth szabványt. A legtöbb, ha nem minden sport tracker alkalmazás ezt a szabványt használja a külső alkalmazásokkal való kapcsolódásra. Amikor csatlakozni szeretnénk az adott profilunkhoz külső alkalmazásból, meg kell adni az engedélyt, hogy az alkalmazás mely adatainkhoz férjen hozzá.
- Csatlakozás után az alkalmazásnak le kell töltenie a felhasználó legújabb sport tevékenységeit. Erőforrás takarékoság szempontjából először meg kell bizonyosodni, hogy van-e új tevékenység. Törekedni kell, hogy a felhasználóhoz tartozó összes adatot csak az első csatlakozás alkalmával, vagy más eszközön való bejelentkezés esetén töltsük le.
- Ha a felhasználó nem csatlakozik más sport tracker alkalmazáshoz, akkor bejelentkezés esetén kell megbizonyosodni, hogy van-e csatlakoztatott sport trackerek esetén történt-e új tevékenység felvitel. Amennyiben igen, úgy csak ezeket az új tevékenységeket kell letölteni.
- A letöltött adatokat az Androidos eszközökön kell tárolni. Erre azért van szükség, hogy a későbbiekben a régebbi eseményeket, vagy a már jutalmazott tevékenységekért ne adjunk újra bónuszt. Ennél a pontnál szembe kell nézni a ténnyel, hogy a különböző sport trackerek bizonyos adatokat másként tárolnak, vagy teljesen hiányoznak. Emiatt létre kell hozni egy olyan általános adatbázis táblát, amelyben minden olyan adatot tárolunk, amelyekért jutalmat akarunk osztani a játékos számára. A későbbiekben támogatottságot nyerő sport trackereknek így valamilyen módon szolgáltatnia kell legalább azokat az információkat, amik ebben az adattáblában kapnak helyet. Az adattípusuk különbözhet, és amennyiben egy bizonyos adatot nem szolgáltat, de más adatokból származtatni lehet, úgy az nem okozhat akadályt.
- Bejelentkezés után az újonnan letöltött adatok alapján a felhasználó staminát (kitartást) kap. Egy játékosnak maximum 100 staminája lehet. A kapott stamina mennyisége összhangban kell lennie ezzel a maximális értékkel, a játékos szintjével, és a tevékenységekben szereplő adatok nagyságával. Azaz az alacsony és magas szintű felhasználóknak is egyaránt élvezetesnek kell maradnia a játéknak, nem szabad se túl sokat, se túl keveset kapni. Túl sok stamina esetén nagyon könnyen haladhatna a felhasználó a játékban, így

egy idő után beleunna, túl kevés esetén viszont a folyamatosan túl nagy kihívást jelentő és csak nagy megerőltetést jelentő tevékenységek szintén ugyanezt a hatást érnék el.

- A jutalomként megkapott staminát a felhasználó a játékos fejlődésére használhatja fel különböző módokon. Az egyik ilyen mód a világban való "barangolás", ami közben szörnyek támadhatnak a játékosra, amelyeket legyőzve játékbeli pénzt és tapasztalati pontot kap a játékos. A másik mód küldetések vállalása, amelyet a felhasználónak kell ténylegesen sportolva teljesíteni, és csak a teljesítése után kapja meg az érte járó játékbeli jutalmat.
- A játékos ezen kívül rendelkeznie kell tulajdonságokkal is, melyek a szörnyek elleni csatában segíthetnek számára. Tulajdonságot növeli szintlépéssel vagy valamilyen kirívó sportteljesítményért cserébe lenne érdemes megengedni.
- További tárgyakat is érdemes lenne megvalósítani a játékos számára, melyek védelemmel vagy támadóerővel növelhetnék a játékos erejét.

3.2. Felhasznált technológiák

Játékmotorok

Játékmotornak nevezzük a játékok - legyen az akár számítógépre vagy konzolra készült – azon részét, amely a program alapjául szolgáló technológiát adja. Szerepe, hogy megkönnyítse a fejlesztést illetve segítségével több platformon is futtatható lesz a játék.

A fejlesztés megkezdése előtt több fajta játékmotort is megvizsgáltam abból a célból, hogy kiválasszam a legmegfelelőbbet a diplomamunkám elkészítéséhez.

A fő szempontom az volt, hogy ingyenesen elérhető legyen, illetve illeszkedjen a választott játéktípus játékmenetéhez.

A két legnépszerűbb motorral kezdtem az ismerkedést, a Unity és az Unreal engine-ekkel. Mivel a programomat Android platformon terveztem elkészíteni, amit Java nyelven kell implementálni, ezek a motorok pedig a C++ nyelvet támogatják, így nem lehet közvetlenül Java nyelven használni őket ezért hamar kiestek. Méretük alapján túl nagynak is bizonyultak volna egy ilyen kisebb méretű projekthez. A következő játékmotor, amit megvizsgáltam a HexEngine volt, amit Szabó László készített el MSc diplomamunkájaként. Ez a motor kifejezetten körökre osztott játékokra lett kifejlesztve, amivel az általam írt játék is rendelkezik, viszont a játéktér hatszögű blokkokra van osztva, amivel megbonyolította volna a közlekedést a játékon belül.

A választásom így Hollósi Tamás által készített dream-iso-droid játékmotorra esett, amit témavezetőm ismertetett meg velem. Mivel készítője elérhető közelségben volt, ezért könnyebben sikerült megismerkednem a motor nyújtotta funkciókkal.

	Unity	Unreal	HexEngine	dream-iso-droid
<i>nézet</i>	3D	3D	2D izometrikus	2D izometrikus
<i>platform</i>	Több	Több	Több	Android
<i>nyelv</i>	C++, C#	C++, UnrealScript	C++	Java

3.1. ábra. Játékmotorok összehasonlítása

A dream-iso-droid egy olyan speciális játékmotor, ami kifejezetten Android platformra készült és a két dimenziós izometrikus nézetet támogatja. Ez a két funkciója pontosan megfelelt az elvárásaimnak, amit a játékmotor felé támasztottam, aminek segítségével fejleszteni szerettem volna az RPG játékomat.

4. fejezet

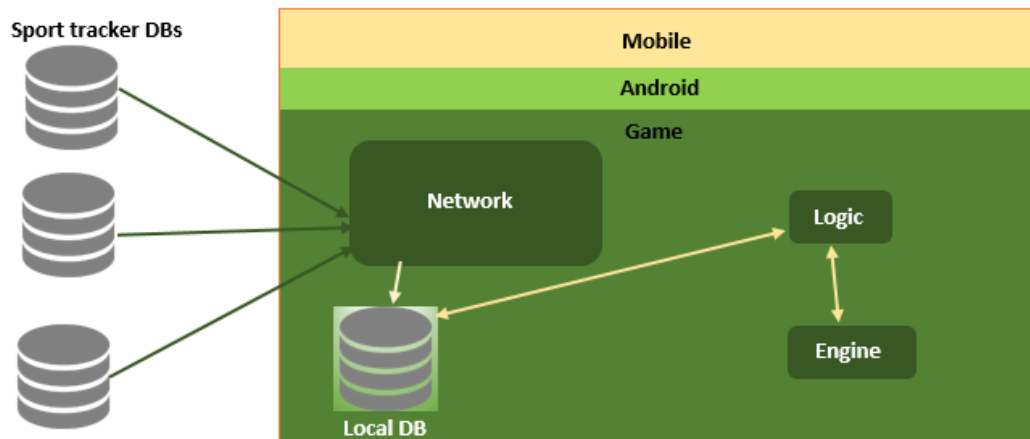
Fejlesztői dokumentáció

Az első fejezetben bemutatásra került egy alapgondolat, és az ebben megfogalmazódott problémára megoldásként elkészítettem egy Androidos szerepjátékot. A játék fő célja az, hogy az embereket valamilyen testmozgásra sarkallja. Ha képesek vagyunk ilyen módon motiválni az embereket, akkor egy nagy lépést tettünk a probléma megszűnéséért.

Ahhoz, hogy a játék beteljesítse a célját a harmadik fejezetben lefektetett követelményeknek mindenképp meg kell felelnie. A játék megtervezése ezeknek a követelményeknek a figyelembe vételével történt. A tervezés során ügyelnem kellett, hogy kiküszöböljem a hasonló játékok egyik legnagyobb hátrányát, azaz az általam készített játéknak nem szabad önálló nyomkövető alkalmazásként funkcionálnia. Ezek után ki kellett választanom egy olyan játéktípust amely felkelti és meg is tartja a felhasználó figyelmét. Továbbiakban fontos volt olyan grafikai megjelenítő keretrendszert találnom amely a játék stílusjegyeinek megvalósítását nagyban támogatja a választott platformon, amelyek az Androidos operációs rendszerrel rendelkező okos-eszközök.

4.1. Architektúra

Az 4.1. ábrán látható a játék architektúrája. Két fő részre bontható fel, egy megjelenítő egységre és egy logikai egységre. A megjelenítő egység elkészítéséhez Hollósi Tamás által készített dream-iso-droid nevű keretrendszert használtam fel. A játékban megjelenő grafikai elemek túlnyomó részét ez kezeli. A logikai egység feladata a megtervezett szabályok és játékmechanikai elemek felügyelete. Ez az egység több kisebb modulra bontható fel, melyeknek mind megvan a saját jól elkülöníthető feladata. Az egység ezeket a modulokat kezeli, és a belső működésükbe nem szól bele, elfedve azokat. Elkerülhetetlen, hogy két modul kommunikáljon egymással, ezt is a logikai egység szabályozza. A két egység szoros együttműködésüként valósul meg a játék. A továbbiakban ezeket az egységeket, és a modulok megvalósítását mutatom be bővebben.



4.1. ábra. A program architektúrája

4.2. A játék indítása

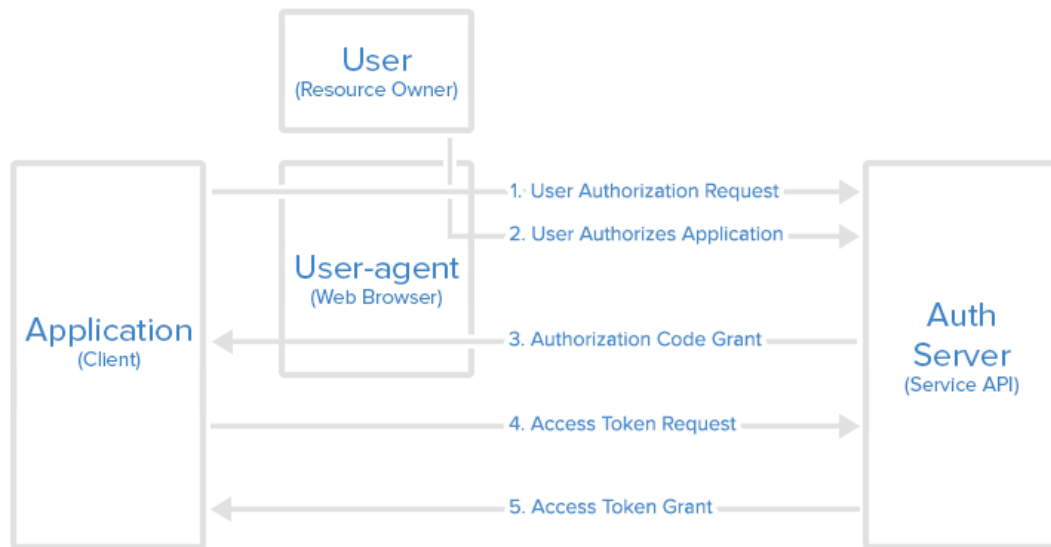
Az alkalmazás indítása után a felhasználót egy egyszerű menü fogadja. A menü tetején található a játék logója, melyet a játékban található több grafikai elemmel együtt Börönd Evelin hallgatótársam készített el számomra. Itt két lehetőség tárul a felhasználó elé, képes csatlakozni új sport-nyomkövető szolgáltatásokhoz, vagy elkezdhet játszani. A csatlakoztatott szolgáltatások egy egyszerű listában tárolódnak soronként, ami tartalmazza a nevét, és egy gombot, amely elindítja a csatlakozási folyamatot.

Csatlakozás nyomkövető alkalmazáshoz

Minden, nagyobb felhasználói bázissal rendelkező weboldalnál fontos, hogy a felhasználók adatait bizalmasan kezeljék, kívülről érkező illetéktelen hozzáférési kéréseket ne szolgálják ki. Ez nem csak adatok felvitelére vagy meglévők módosítására vonatkozik, hanem a jelenlegi információk megtekintését is sokszor biztonságosan kell kezelni. Az előbb röviden felvázolt szempontokat teljesítő protokollt használnak szinte minden hasonló portálon. Ennek a protokollnak a neve OAuth [?], melynek legfrissebb verziója a 2.0-s.

Az 4.2. ábrán látható módon működik a protokoll. Ha egy alkalmazás el akar érni olyan adatot ami bizalmasan van kezelve, ahhoz előbb engedélyt kell kérnie hozzá. Az alkalmazás továbbirányítja a felhasználót az adott szolgáltatás felületére - többnyire valamilyen webböngészőbe -, ahol engedélyt tud adni számára. Ekkor a felhasználónak be kell jelentkeznie a fiókjába, és megadni a kért engedélyeket. Az engedély megadása után a hitelesítő szerver egy megerősítő kódot juttat el az alkalmazás számára. Az alkalmazás ezt a megerősítő kódot tudja "elcserélni" a szerverrel egy tokenre. A későbbi adatelérés alkalmával minden kéréshez csatolnia kell az alkalmazásnak ezt a tokenet. A szerver ezt a tokenet vizsgálva tudja eldönteni,

Authorization Code Flow



4.2. ábra. Az OAuth protokoll absztrakt működési ábrája

hogy a kért információhoz a felhasználó engedélyt adott-e. Ez a folyamat implementálása sok biztonsági és hibakezelési kérdést vet fel, így úgy döntöttem, hogy egy már kész könyvtárat használok hozzá.

A választott könyvtáram a ScribeJava [?], mely a protokoll több verzióját is támogatja és számos szolgáltatáshoz már kész API-val rendelkezik. Az általam integrált két sportnyomkövető nem voltak elkészítve, így ezeket nekem kellett megvalósítanom. Szerencsére a könyvtár úgy lett kialakítva, hogy minden, a protokollt használó szolgáltatáshoz egységesen lehessen API-t készíteni.

A Runkeeper nyomkövetőhöz elkészített API legfontosabb része a következőként néz ki:

```
1  private static final String AUTHORIZATION_URL = "https://runkeeper.com/
2  apps/authorize?client_id=%s&response_type=code&redirect_uri=%s";
3  private static final String ACCESS_TOKEN_URL = "https://runkeeper.com/apps
4  /token";
5  @Override
6  public String getAccessTokenEndpoint() { return ACCESS_TOKEN_URL; }
7  @Override
8  public String getAuthorizationUrl(OAuthConfig config) {
9      Preconditions.checkNotNull(config.getCallback(), "Must provide a
10     valid url as callback.");
11     final StringBuilder sb = new StringBuilder(String.format(
12     AUTHORIZATION_URL, config.getApiKey(), OAuthEncoder.encode(config.
13     getCallback())));
14     // ...
15     return sb.toString();
16 }
```

Mivel ez a kommunikáció hálózati tevékenységgel jár, nem történhet az Android fő program-szálán. Egyrészt ha a fő szálon folya ez a kommunikáció, az alkalmazás nem tudna tovább futni, amíg hitelesítési folyamat be nem fejeződik. Ez akár több másodpercbe is telhet a hálózati körülményeket figyelembe véve, így addig az alkalmazás blokkolódna. A felhasználói élmény miatt ez nem megengedhető, így ezt a folyamatot a háttérben kell elvégezni, hogy az alkalmazás zavartalanul futhasson tovább. Az Android SDK-ban több beépített lehetőség segítségével is meg tudjuk valósítani ezt:

- Normál Java szálak
- AsyncTask
- IntentService

Ezeknek a lehetőségeknek megvan a maguk előnye és hátránya. A sima Java szálak használata széles körben elterjedt, de nagyobb, bonyolultabb programszerkezet mellett használatuk nehézkes. A következő választási lehetőség az Android SDK-ban bemutatott AsyncTask osztály használata. Ennek segítségével könnyedén indíthatunk háttérben futó kódrészleteket. Az osztályon belül felülírható metódusok, amelyek az adott tevékenység elején, közben, vagy a feladata végeztével hívódnak meg. Ez a fajta megoldás sokkal kötetlenebb, viszont ahogy az előző pontban taglalt sima szálhoz hasonlóan, ha az adott Activity, amelyikből el lett indítva háttérbe kerül, az operációs rendszer meg tudja szakítani a folyamat futását. Az *IntentService* esetén ez nem történik meg, tovább egyrészt nincs *Activity*-hez kötve, másrészt a kéréseket egy sorba teszi, amelynek a későbbieknek még fontos szerepe lesz.

MELLÉKLET

A mellékelt CD könyvtárszerkezete