



M Ű E G Y E T E M 1 7 8 2

Budapesti Műszaki és Gazdaságtudományi Egyetem
Villamosmérnöki és Informatikai Kar

Kiss Gábor

ANDROID MÉRŐRENDSZER ERŐFORRÁS-KEZELÉSE

Diplomaterv

KONZULENS

Dr. Fehér Gábor

BUDAPEST, 2013

Tartalomjegyzék

Összefoglaló	4
Abstract.....	5
1 Bevezetés	6
2 Az erőforrás-kezelő rendszerkomponens	8
2.1 A DroidLab mérőrendszer	8
2.1.1 A rendszert építő eszközök	8
2.1.2 A rendszer felépítése.....	10
2.2 A kvóta rendszer	13
2.2.1 Erőforrás használat szabályozása egyéb rendszerekben	14
2.2.2 Szolgáltatás-csoportok és kvóták.....	19
3 Gamification	24
3.1 A gamification fogalma	24
3.2 Gamification kategóriák	25
3.2.1 Internal gamification.....	25
3.2.2 External gamification.....	26
3.2.3 Behavior change gamification	26
3.3 A gamification eszköztár	28
3.3.1 Az elemek hierarchikus piramisa.....	28
3.3.2 A „PBL hármas”	30
4 Összefoglalás.....	33
4.1 Eddig elvégzett feladatok.....	33
4.2 Második félévre maradt feladatok	33
Irodalomjegyzék.....	35

HALLGATÓI NYILATKOZAT

Alulírott **Kiss Gábor**, szigorló hallgató kijelentem, hogy ezt a diplomatervet meg nem engedett segítség nélkül, saját magam készítettem, csak a megadott forrásokat (szakirodalom, eszközök stb.) használtam fel. Minden olyan részt, melyet szó szerint, vagy azonos értelemben, de átfogalmazva más forrásból átvettem, egyértelműen, a forrás megadásával megjelöltem.

Hozzájárulok, hogy a jelen munkám alapadatait (szerző(k), cím, angol és magyar nyelvű tartalmi kivonat, készítés éve, konzulens(ek) neve) a BME VIK nyilvánosan hozzáférhető elektronikus formában, a munka teljes szövegét pedig az egyetem belső hálózatán keresztül (vagy hitelesített felhasználók számára) közzétegye. Kijelentem, hogy a benyújtott munka és annak elektronikus verziója megegyezik. Dékáni engedéllyel titkosított diplomatervek esetén a dolgozat szövege csak 3 év eltelte után válik hozzáférhetővé.

Kelt: Budapest, 2013. 05. 17.

.....
Kiss Gábor

Összefoglaló

Mindennapos életünk részévé váltak a különböző mobil készülékek. Ezek közül is kiemelt szerep jutott az okostelefonoknak, hiszen számos egyéb eszköz funkcionalitását átvette az évek során. Hála a technológia folytonos fejlődésének a gyártók a versenyképes készülék előállítása érdekében egyre változatosabb és nagyobb teljesítményű erőforrásokat integrálnak az eszközeikbe. Ennek ellenére a legtöbb felhasználó csak nagyon ritkán szokta készülékének maximális teljesítményét kihasználni egy konkrét időpillanatban.

Az ilyen szabad kapacitások kiaknázását szeretné megoldani az általunk kezdeményezett, DroidLabnak elkeresztelt mérőrendszer. A projektben a fő feladatomb volt a megfelelő erőforrás-kezelés kialakítása, melynek célja, hogy végje az adott eszköz tulajdonost az esetleges korlátlan mérések által okozott kellemetlenségektől. Ez a fő témája diplomatervemnek is.

Az erőforrás-szabályozás megvalósítása mellett rám hárult a feladat, hogy a gamification („játékosítás”) módszertanának megismerése után ajánlást tegyek a megfelelő felhasználó-bázis illetve mérési körülmények kialakítására.

Abstract

The mobile devices have a very important role in our lives. Cell phones are the most used multifunctional tool today. The device manufacturers are intergating the newest and most powerful moduls into their products, because they want to stay in the market. But the most user can't utilise these new resources effectively.

Our goal is to realize a testbed-like system, called Droidlab, which exploit these unused resources. As member of the team, my job was to create a resourse-management subsystem. This component's task is to protect the user and the device from the disadvantages of measurements (e.g.: performance decrease). This is the main subject of this work.

My secondary job was to study the gamification methods and use them to design a business-like modell, that assist the project in evolving a proper user base and measurement environment.

1 Bevezetés

Az Internet és az egyéb vezetékes illetve vezeték nélküli hálózatok teljes mértékben megváltoztatták életünket. Számos olyan szolgáltatás vált mindenki számára elérhetővé, melyről akár egy évszázada csak álmodni mertünk volna (például: kilométereket áthidaló videó hívások vagy vásárlás). A megnövekedett felhasználói igények szükségessé teszik a meglévő megoldások optimalizálását vagy újabb technológiák fejlesztését. Természetesen egy-egy újítás teljes körű bevezetése előtt számos tesztelési és finomítási lépésen kell végighaladnia az adott technológiának. E folyamat általában valamilyen szimulációval szokott indulni, mely során becsült értéket nyerhetünk az adott technológia egyes tulajdonságaival kapcsolatban (például: teljesítmény vagy skálázhatóság). A szimuláció során általában valamilyen leegyszerűsített modellt használunk, ami a valós környezet tényleges dinamizmusát nem tükrözi, így a szimuláció során kapott eredmények nem megfelelőek az adott technológia eredményességének bizonyítására.

Következő lépésként általában valamilyen emulációt szoktak használni a további vizsgálatok elvégzésére. Ebben az esetben már jóval komplexebb forgatókönyveket tudunk létrehozni, melyek sokkal jobban tükrözik az éles helyzetben elvárható eseményeket, így ez már megfelelőbb megközelítésnek mondható. Ennek ellenére számos olyan környezeti hatás (például aktuális felhasználók viselkedése, területi infrastruktúra) lehet, melyet még így sem tudunk kellőképp modellezni, ezért manapság egyre jobban elterjedtek az úgynevezett tesztrendszerek vagy idegen szóval kifejezve testbed-ek.

Egy ilyen tesztrendszer valamilyen kísérleti eszközök sokaságából áll, melyeken a valósághoz legközelebb eső méréseket tudjuk elvégezni. Az így kapott eredmények már alkalmasak a helytálló következtetések megállapítására. A tesztrendszerek sikerességét igazolja az is, hogy egyre több ilyen kezdeményezés indul el világszerte, például az ORBIT[1] (Open-Access Research Testbed for Next-Generation Wireless Networks) névre keresztelt vezeték nélküli hálózatot biztosító tesztrendszer, vagy a PlanetLab[2], mely többféle hálózati szolgáltatással kapcsolatban is képes tesztkörnyezetet biztosítani. Azt az eddigiek alapján beláthatjuk, hogy különböző tesztrendszerek pozitív hatással vannak a technológiai fejlődésre.

A mobil eszközök egyre fontosabbá válnak mindennapi életünkben. Az eszközgyártók folyamatosan növelik készülékeik teljesítményét, és ezeket az erőforrásokat a felhasználók nem használják ki teljes mértékben. Felmerült az ötlet, hogy e szabad kapacitásokat jó lenne valahogyan kutatási illetve fejlesztési célokra felhasználni. Az előbbi elképzelést továbbgondolva született meg egy olyan mérőrendszer megvalósításának ötlete, mely a már szóba hozott tesztrendszerekhez hasonlóan működik és az említett szabad erőforrásokat használja fel. A projektet DroidLabnak neveztük el.

Feladatom az előbb említett mérőrendszer megvalósításában a készülékek erőforrásainak egységekbe szervezése, ezen egységek (későbbiekben kvóták) kijánlása a felhasználóknak és végül a felhasználók által beállított mennyiségek betartatása a rendszerrel. E mellett feladatom még úgynevezett gamification módszereket felhasználva olyan megvalósítás megtervezése és részleges vagy teljes implementálása, mely a megfelelő felhasználó bázis kialakítását szolgálja, illetve motiválja az addigi felhasználókat minél több erőforrás megosztására.

A dolgozatom következő fejezetében bemutatom röviden a mérőrendszer egyes komponenseit illetve felépítését, ezt követően részletesen kitérek az eszközökkel kapcsolatos erőforrásokra hozzájuk tervezett kvótákra. A 3. fejezetben ismertetem a gamification fogalmát és a legfontosabb alkotó elemeit.

2 Az erőforrás-kezelő rendszerkomponens

Ebben a fejezetben bemutatom magát a mérőrendszert, majd beszámolok pár eddig megvalósított erőforrás-szabályozási módszerről, végezetül a kvóta rendszerrel kapcsolatos fontos információkat ismertetem.

2.1 A DroidLab mérőrendszer

Az általunk megvalósított rendszer célplatformja az Android lett. A többi, piacon levő alternatívával összehasonlítva az Android rendelkezett a feladat számára legkedvezőbb tulajdonságokkal. Ezek közül a legfontosabb három tulajdonságot emelném ki:

- nyílt forráskódú, jól dokumentált, nagy fejlesztői és felhasználói körrel rendelkezik
- jelenleg az egyik legelterjedtebb platform, megjelenése óta népszerűsége folyamatosan növekszik
- legváltozatosabb az eszköztára (telefon, tablet, tv, játék konzol...)

2.1.1 A rendszert építő eszközök

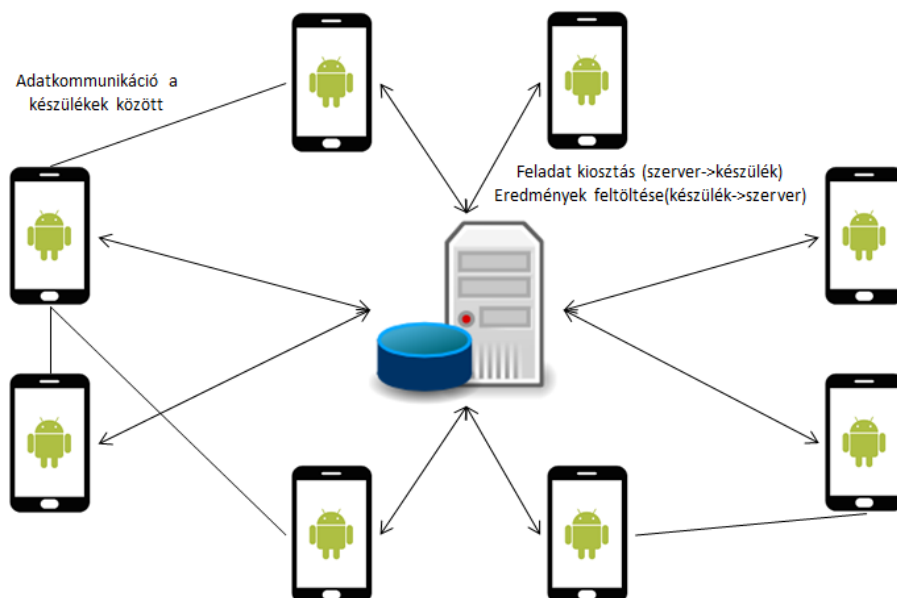
A DroidLab részben eltér a hagyományos tesztrendszerektől a hálózatot alkotó eszközök szempontjából. A megszokottól eltérően nekünk nem áll szándékunkban egy izolált, általában laboratóriumi környezetbe beágyazott eszközállományt kialakítani, hanem egyszerű eszköztulajdonosok (későbbiekben felhasználó) készülékeit szeretnénk erre a célra felhasználni néhány általunk írt alkalmazás segítségével. Ez a megoldás azért ideálisabb a hagyományos megvalósításnál, mert így nem kell külön az eszközöket megvásárolni a hálózat kialakításához, ami általában egy hasonló projekt költségvetésének nagy részét szokta jelenteni. Megoldásunk mellett szól az az érv is, hogy az így végzett mérések sokkal realisabbak lesznek, hiszen a mért adatok a mindennapos használat során születnek, nem pedig a már említett izolált környezetben.

Természetesen az előnyök mellett hátrányai is vannak ennek az elképzelésnek, hiszen sokkal kiszámíthatatlanabb lesz az adott hálózat, mivel a teljes irányítás az adott felhasználó kezében van. E mellett személyi jogokkal kapcsolatos gondok is

felmerülhetnek, a későbbi kellemetlenségek elkerülése végett a rendszer tervezésénél ezeket megfelelő súllyal figyelembe kell venni. Végezetül kezdeti nehézségként felmerülhet a megfelelő felhasználó bázis kiépítése is, erről még később részletesen is szó lesz (lásd: 3.2.2 External gamification).

A mérésekre használt készülékek mellett szükség van egy központi vezérlő egységre is. Ez az rendszerkomponens nem vesz közvetlen részt a mérések tényleges elvégzésében, legfőbb feladatai közé tartozik a készülékek és a hozzájuk tartozó adatok számon tartása, az elvégzendő feladatok kiosztása a rendelkezésre álló szabad eszközökre és az elvégzett mérések eredményeinek fogadása.

Az elhangzottak alapján a rendszer fizikai topológiájának egy vázlatos sémáját az 2.1. ábrán látható módon képzelhetjük el.



2.1. ábra: A rendszer fizikai topológiájának egy vázlatos sémája

Láthatjuk, hogy az összes mérőrendszert alkotó készüléknek kapcsolatban kell lennie a központi egységgel, viszont ennek a kapcsolatnak nem kell feltétlen állandó, „online” kapcsolatnak lennie. Elképzelhető olyan mérési szituáció, amely nem követeli meg ezt az helyzetet. Viszont a feladatok kiosztásánál és az adatok feltöltésénél mindenképpen kell valamilyen rádiós hálózati összeköttetés a megfelelő működés érdekében.

Az ábráról jól látszik, hogy lehetséges közvetlen kapcsolat kialakítani kettő vagy több mérőműszer között is. Erre általában csak valamilyen speciális mérés során lehet szükség, például Bluetooth erőforrás segítségével szeretnénk mérni átviteli sebességet.

A Bluetooth kommunikáció kialakításához legalább két hardware-sen támogatott eszköz kell, amiből következik, hogy ezt egy mérőműszer segítségével nem lehet megoldani. Ilyenkor van szükség az említett közvetlen kapcsolatra. Az ilyen mérések lebonyolítása sokkal nagyobb koordinációt igényel, mint azoké, melyeket egymástól független mérő eszközökön el lehet végezni.

2.1.2 A rendszer felépítése

A fizikai topológiához hasonlóan a szoftveres megvalósítást is két jól elkülöníthető részre oszthatjuk, egy szerverre és egy kliens alkalmazásra. A szerver a már említett központi vezérlő egységen fut és fő feladata a már bemutatott funkciók szoftveres megvalósítása, ezek közül is a legnagyobb hangsúlyt a kliens információk megfelelő karbantartására helyezve. Emellett még a mérőrendszert mérési célokra igénybevevő személyek (későbbiekben kutatók) is a szerver egy erre szolgáló felületén tudják a mérési feladataikat feltölteni. A tervek szerint a felhasználóknak is kialakítunk egy felületet, amin a saját adataikat és profiljukhoz tartozó információkat tudják megtekinteni, vagy egyes adatokat pótolni illetve módosítani. A szerver funkcióit a projekt kezdetekor egy PHP website valósította meg, de úgy lett kialakítva, hogy későbbiekben egyéb megoldásra (például Google Appengine¹) könnyen cserélhető legyen.

Magán a mérő eszközökön futtatott szoftvert tekintjük a már említett kliensnek. A kliens szinte minden készüléken futtatható, két elvárást támasztunk az eszközökkel szemben:

- rendelkezzenek hálózati hozzáféréssel
- érjék el a Google szolgáltatásait

Az előbbi magától értetődik, ha visszagondolunk a rendszert alkotó eszközök ismertetésénél leírtakra. Eza kritérium nagyon kevés esetben jelenthet problémát, az olcsóbb készülékekben is van Wi-Fi modul, mely segítségével vezeték nélküli hálózatra képes csatlakozni. Utóbbi elvárásra azért van szükség, mivel a klienshez tartozó alkalmazásokat a Google Play² digitális tartalomszolgáltatón keresztül szeretnénk

¹ <https://developers.google.com/appengine/>

² <https://play.google.com/store>

terjeszteni a könnyű elérés és frissítés érdekében, az online bolt által nyújtott ellenőrzési és hitelesítési lehetőségeket kihasználva.

Mivel az egyik fő szempont a mérőrendszer tervezésekor a felhasználók szabad döntési jogának biztosítása volt, ezért a kliens alkalmazást teljesen modulárisra terveztük, így három egymástól jól elkülöníthető részre bontottuk.

2.1.2.1 Plugin-ek

Az Android platform eléggé sajátos erőforrás kezelési mechanizmussal rendelkezik. Ennek lényege, hogy az adott erőforrást használni kívánó alkalmazásnak rendelkeznie kell a megfelelő hozzáférési joggal (idegen szóval permission-nel), hogy az adott funkció hibamentesen működjön. Ez még nem jelentene problémát, a gondot a hozzáférési jogok kezelése okozza. A felhasználó az adott alkalmazás megfelelő működéséhez szükséges jogokat csak a telepítési folyamat kezdeti szakaszában engedélyezheti, és mindenképpen el kell fogadni az összes igényelt kérelmet, hogy a telepítés sikeres legyen. Az engedélyek megtagadása a folyamat automatikus megszakításával egyenlő. Emellett a megadott jogokat a későbbiekben nincs lehetősége a felhasználónak visszavonni, csak az alkalmazás törlése által.

Beláthatjuk, hogy ez a megoldás nem felhasználóbarát. Mivel az általunk tervezett rendszer egy adott készülék majdnem minden lehetséges erőforrását szeretné kihasználni, így ha csak egy komplex alkalmazásba sűríténénk az összes funkciót az előbb ismertetett mechanizmus miatt minden szükséges jogot egy időben kellene engedélyezni a DroidLab futtatásához. Ez teljes mértékben megfosztaná a felhasználót annak eldöntésétől, hogy mely erőforrásokat hajlandó a rendelkezésünkre bocsájtani. Ennek kiküszöbölése érdekében olyan felépítés mellett döntöttünk, amely az egyes szolgáltatás-csoportokat pluginek-ként valósítja meg. Ez a kialakítás biztosítja az erőforrásokhoz definiált funkciók könnyű bővítési lehetőségét illetve a jövőbeli új erőforrások rendszerbe való integrálását.

Minden plugin önálló alkalmazásként képzelhető el, így csak az adott csoport működéséhez szükséges jogokat kell a felhasználónak biztosítania. Ezen kívül a felhasználóknak lehetőséget kínálunk az adott plugin által biztosított funkciók egyéni korlátozására is az erre a célra készített beállítási felületen.

Az egyes csoportok a későbbiekben részletesen bemutatásra kerülnek (lásd: 2.2.2-es fejezet).

2.1.2.2 Modulok

A kutatók által fejlesztett méréseket hívjuk moduloknak. A modulok a tesztelési fázisban Java nyelven készülnek. A moduloknak olyan futtatható kódrészleteknek kell lennie, melyek felett a készülékeken futó kliensnek teljes ellenőrzése van. Ezért a modulok nem kommunikálhatnak közvetlenül az Android rendszerrel, nem indíthatnak saját szálakat, nem iratkozhatnak fel rendszer-eseményekre.

A biztonsági követelményeknek a Java kód nem felel meg minden szempontból. Különböző trükkökkel a Java által biztosított biztonsági konténer kijátszható és tetszőleges kódrészlet futtatható lenne, többek közt olyan is, mely magát a klienst vezérelné, így teljes ellenőrzést gyakorolna a felhasználó eszköze fölött. Ezért a modulok számára egy második, saját magunk által tervezett biztonsági konténert akarunk kialakítani a Java által nyújtott lehetőségen belül, hogy a fentebb említett követelmények mindenképpen biztosíthatóak legyenek. Ezért a modulok programozási nyelvét az éles rendszerben magasabb szintű script nyelvre, valószínűleg a LUA-ra fogjuk cserélni, de lehet, hogy egy saját fejlesztésű nyelv fogja felváltani.

A modulok a plugin-ek funkcióit felhasználva végeznek méréseket. Ezek a mérések vagy periodikusak, vagy eseményvezéreltek, a mérendő tulajdonság vagy jelenség jellegéhez igazodva. A kétféle viselkedés a modul kódjában elkülönül. Elnevezési konvenciókkal biztosítjuk a kétféle működés megvalósítását.

A modulok számára a kliens állandó háttértárat biztosít a részeredmények tárolására, és egy naplózási lehetőséget az eredmények rögzítésére.

A kutatók megadják, hogy hány készüléken, milyen mennyiségű erőforrásra van szükségük, a vezérlő ezeket az igényeket párosítja a felajánlásokkal, majd küldi a modulokat a megfelelő eszközökre.

A modulok funkcionalitásukból adódóan nem rendelkeznek saját felhasználói felülettel, ennek ellenére lehetőséget biztosítunk az adott eszközön futtatott modulok listás megjelenítésére és egyes listaelemeket kiválasztva az adott felhasználó információt szerezhet a kiválasztott modulról, illetve szükség esetén akár meg is szakíthatja annak futását.

2.1.2.3 A keretrendszer

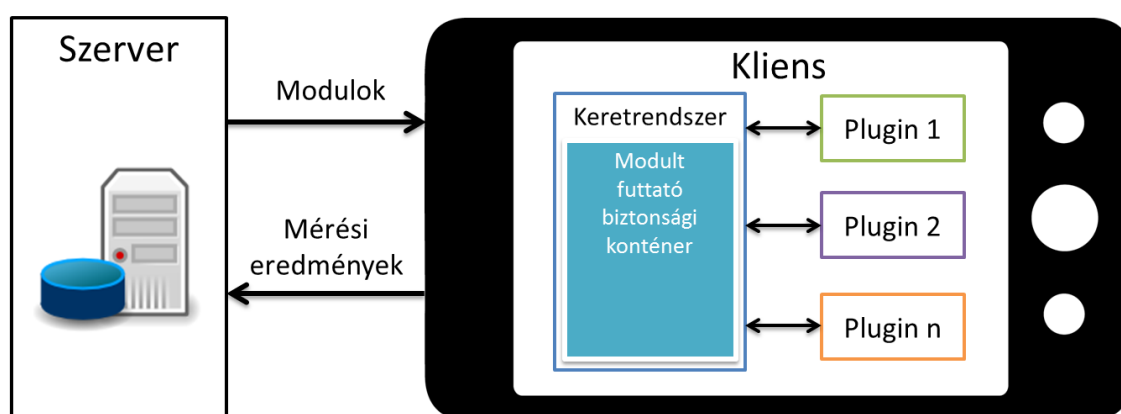
A DroidLab alapfunktionalitását megvalósító alkalmazást nevezzük keretrendszernek. A keretrendszer feladata a telepített plugin-ek megkeresése és igény

szerinti frissítése vagy letöltése. A keretrendszer tölti le, indítja el és állítja le a modulokat, tölti föl a mérési eredményeket.

Minden plugin definiálhat generikus eseményeket és generikus adatokat, amikre a keretrendszer feliratkozik, és az így kapott információkat összegyűjtve továbbítja a szerver felé.

A keretrendszer felülete általános híreket és statisztikákat közöl a rendszerről. Ezen a felületen tudnak a hálózat üzemeltetői kommunikálni a felhasználóval, felhívások, hibaüzenetek és egyéb visszajelzések közvetíthetők a felhasználók felé.

Ezek alapján a mérőrendszer szoftveres felépítésének egy vázlatos sémáját az 2.2. ábrán látható módon képzelhetjük el.



2.2. ábra: A mérőrendszer szoftveres felépítésének egy vázlatos sémája

2.2 A kvóta rendszer

Nyugodtan kijelenthetjük, hogy a mobil készülékek jelen technológia mellett sajnos még rendelkeznek néhány szűkös erőforrással. Ezek közül talán tápellátás az, ami a minden felhasználó számára kiemelt fontosságú. Könnyen belátható, hogy egy olyan kliens, amely az üzemidő drasztikus csökkentését eredményezné, az egész eszköz-hálózat kiépítését kivitelezhetetlenné tenné. Egy felhasználó sem lenne hajlandó ilyen tulajdonsággal rendelkező alkalmazás telepítésére. A felhasználók bizalmának elnyerése érdekében mindenképpen szükség van a mérőrendszerrel kapcsolatos korlátok létrehozására és annak biztosítására, hogy ezeket a rendszer ténylegesen be is tartja. Azt már láthattuk, az Android jogosultság kezelési rendszere több kívánnivalót hagy maga után, de sajnos a platform nem biztosít lehetőséget a telepített alkalmazások erőforrás-használatának korlátozására sem. E hiányosság pótlása az én fő feladatom.

Az eddig már kiderült, hogy a mérőrendszer használhatósága a felhasználók számától és az általuk felajánlott erőforrásoktól függ. Bár ez egy nem túl elterjedt modell, számos más területen érvényesülő, felhasználók szabad erőforrásait felhasználó projekt létezik, mely az elképzelés kivitelezhetőségét igazolja. Ezek közül talán a legismertebb a világsikerű SETI@home[3] projekt. A SETI@home egy tudományos kísérlet, amely az Internethez csatlakozó számítógépeket használja a földönkívüli élet kutatásához. A részvételhez mindössze egy ingyenes program futtatására van szükség, ami rádióteleszkóp-adatokat tölt le és elemez ki, mindezt kizárólag egy képernyőkímélő futtatása alatt. Így ténylegesen csak akkor használják az asztali gépek kapacitását, ha a tulajdonos már jó ideje nem használta azt komolyabb feladat végzésére. Ezzel a módszerrel sikerült a kezdetben használt, speciális szuperszámítógépet helyettesíteni résztvevő gépek szabad kapacitásával. Természetesen a célunk nem egy ilyen elosztott szuperszámítógép létrehozása a DroidLab segítségével (inkább egyfajta szenzorhálózatként tekintünk a mérőrendszerre), de a SETI@home sikere mindenképpen biztató a saját projektünk megvalósíthatósága szempontjából.

Mivel a DroidLab-hoz hasonló elképzelés még nem született a tudományos világban, ezért kénytelen voltam más jellegű rendszerekben megvalósított erőforrás-kezelést megvizsgálni az Android hiányosságát pótló megoldás keresése közben.

2.2.1 Erőforrás használat szabályozása egyéb rendszerekben

Mivel az Android platform Linux kernellel rendelkezik, ezért kézenfekvő volt ezen operációs rendszerrel kapcsolatos lehetőségeket megvizsgálni. A változatos felhasználási területeknek köszönhetően számos erőforrás szabályozási módszert sikerült találnom. Természetesen ezek a megoldások a mobil eszközök és az asztali gépek közötti különbségek miatt nem adhatnak tökéletes megoldást, de a közös erőforrásokkal kapcsolatban jó kiindulási alapot nyújthatnak.

2.2.1.1 setrlimit[4]

A `setrlimit()` egy olyan rendszerhívás, mely segítségével folyamatonkénti erőforrás korlátozást tudunk beállítani. Minden ilyen korlátozáshoz egy érték pár tartozik, mely egy névleges (soft limit) és egy tényleges (hard limit) határértékből áll. A soft limit értéket próbálja a kernel fenntartani, ez tulajdonképpen egyfajta intervallumot definiál az erőforrással kapcsolatban. A hard limit ennek az intervallumnak a felső korlátja, amit a rendszer semmiképp nem enged túllépni. Ezeket az értékpárokat egy

úgynevezett rlimit struktúrában tárolja a rendszer. A setrlimit segítségével számos erőforrás korlátját beállíthatjuk ezek közül a fontosabbak:

- RLIMIT_CORE: a folyamat által létrehozható visszaállítási fájl maximális mérete byte-okban
- RLIMIT_CPU: a folyamat által használható maximum processzor idő másodpercben kifejezve (csak soft limitként értelmezhető)
- RLIMIT_DATA: a folyamat által létrehozható összes adat maximális mérete byte-okban
- RLIMIT_FSIZE: a folyamat által létrehozható fájl maximális mérete byte-okban
- RLIMIT_NOFILE: a folyamat által használt fájl leírók maximális száma
- RLIMIT_STACK: a folyamathoz tartozó maximális stack mérete byte-okban
- RLIMIT_MEMLOCK: a folyamat által RAM-ba (Random Access Memory) zárható memória maximális mérete byte-ban
- RLIMIT_AS: a folyamathoz tartozó virtuális memória maximális mérete byte-ban

Maga a rendszerhívás a Linux 2.6.36-os verziója óta elérhető, előtte egy úgynevezett vlimit() függvény segítségével lehetett hasonló funkciót elérni.

2.2.1.2 cpulimit[5]

A cpulimit egy 2006-ban, Angelo Marletta nevezetű úriember által létrehozott projekt. Lényege, hogy egy egyszerű program segítségével a különböző folyamatok processzor használatát szabályozhatjuk. Az setrlimit-tel ellentétben itt nem a processzor időre adhatunk korlátot, hanem beállíthatunk egy százalékban kifejezett értéket a processzor használatra. Egy konkrét példán ezt bemutatva: ha van egy egymagos gépünk, amelyen egy folyamathoz a cpulimit segítségével beállítottuk, hogy 50%-os maximális kihasználtságot érhet el, akkor ebben az esetben az adott folyamat másodpercenként legfeljebb 500 milliszekundum processzor időt használhat fel. Ahány maggal rendelkezik az adott számítógép, annyiszor 100%-ot oszthatunk szét a folyamatok között (például egy négymagos gép esetén összesen 400%-ot).

Maga az alkalmazás teljes mértékben a rendszer felhasználói módjában fut, így a Linux ütemezőt egyáltalán nem zavarja. A program lényege, hogy a /proc könyvtárban

tárolt folyamat statisztikákat és a felhasználó által beállított értékeket figyelve az alkalmazás jelzések segítségével folyamatosan megállítja vagy folytatja az adott folyamatot.

A `cpulimit` alkalmazást minden 2.2 vagy e feletti Linux rendszeren használhatjuk.

2.2.1.3 quota[6]

A quota egy fájlrendszer-használati korlátozás, melyet felhasználónként vagy csoportonként érvényesíthetünk. A korlátokat két szempont szerint is meghatározhatjuk: korlátozhatjuk a lehetséges csomópontok (inode-k) számát, illetve meghatározhatjuk az egyes felhasználókhoz vagy csoportokhoz tartozó lemezblokkok számát is. Hasonlóan a `setrlimit`hez, itt is definiálhatunk `soft` illetve `hard` limiteket is. `Soft` limitet elérve a rendszer figyelmeztetést küld a felhasználónak, a `hard` limit elérése esetén a kernel nem engedi a további szabad hely használatát. A módszer hátránya, hogy csak az `ext2` típusú fájlrendszereken alkalmazható. A quota lehetőségét engedélyezni kell a kernelben, és csatolási pontonként a `/etc/fstab`-ban.

Az 1.3.8x-as Linux verzió óta a quota integrálva van a kernelben.

2.2.1.4 grsecurity[7]

A grsecurity tulajdonképpen olyan Linux kernel foltok (patch-ek) gyűjteménye, mely a Linux ismert biztonsági réseit hivatott javítani. GNU General Public License alatt adták ki és bárki számára ingyenesen elérhető.

A grsecurity több nagyobb komponensből épül fel, ezek közül számunkra az az egy fontos, amelyik egy teljes szerepalapú hozzáférési jogosultság szabályozó rendszert (Role-based access control (RBAC)) valósít meg. A RBAC előnye a hagyományos Unix hozzáférés-vezérlési listákhoz képest, hogy az egyes felhasználók és folyamatok jogosultságait a minimálisra tudja csökkenteni, így egy esetleges támadás esetén a támadónak sokkal kisebb esélye van sikerrel járni, mint normál esetben.

A RBAC-nek köszönhetően az erőforrás szabályozást is szerepekhez köthetjük. A grsecurity logikailag ugyanúgy működik, mint a `setrlimit` rendszerhívás. Ebben az esetben is `soft` és `hard` limiteket állíthatunk és a grsecurity is minden Linux erőforrást támogat. Két főbb eltérés van a `setrlimit`-el szemben:

- az erőforrások megnevezése: mint azt láthattuk alapvetően a Linux erőforrásokat az RLIMIT előtaggal használjuk általában. A grsecurity kapcsán ezt az előtagot RES-re cserélték (például a RLIMIT_CPU helyett RES_CPU megnevezéssel hivatkozhatunk a processzor időre).
- a setrlimit esetén a legtöbb esetben byte-ban illetve másodpercben adtuk meg az adott korlátokat, a grsecurity esetén már nagyobb mértékegységű értékeket is megadhatunk a megfelelő toldalék segítségével (például az 5K 5 kilobyte méretkorlátot jelent)

A projektet 2001 februárjában indították és az első kompatibilis kernelverzió a 2.4.1-es volt.

2.2.1.5 cgroups[8]

A cgroups (control groups) a Linux kernel egy olyan szolgáltatása, mely segítségével egyszerűen szabályozhatjuk és számlázhatjuk különböző folyamatcsoportok erőforrás-felhasználását. A cgroups segítségével különböző csoportokat hozhatunk létre, melyek hasonló tulajdonságokkal rendelkező folyamatokból állnak. A csoportokat a rendszer hierarchikus fa struktúrába rendezi. A gyermek csoportok öröklik a szülők számára beállított korlátokat. Természetesen lehetőség van további szűkítő korlátozások megadására is. A kernel több vezérlő segítségével menedzseli a csoportokkal kapcsolatos információkat. A projekt elsődleges célja egy egységes interfész létrehozása volt, mely a következő feladatok elvégzését támogatja:

- Erőforrások korlátozása (processzor idő, memória használat, háttértár IO)
- Priorizálás: adott folyamat csoportnak nagyobb használati jog biztosítása
- Számlázás: felhasznált erőforrások mérése például későbbi értékesítés céljából
- Kontrollálás: az adott rendszer felügyelete

A cgroups-t 2006-ban a Google mérnökei kezdték el fejleszteni. Először a 2.6.24-es kernel verzióba integrálták a megoldást, azóta számos új szolgáltatással és vezérlővel bővült a projekt.

2.2.1.6 PAM (Pluggable Authentication Modules) [9]

A PAM tulajdonképpen olyan megosztott könyvtárak sokasága, amely lehetővé teszi a helyi rendszergazda számára annak eldöntését, hogy milyen módszerrel történjen a felhasználói azonosítás a két érintett fél között (felhasználó és alkalmazás). A projekt célja egy ilyen funkciót megvalósító szoftver létrehozása és a hozzá szükséges biztonsági valamint azonosítási modulok fejlesztésének elkülönítése. A kezdeti funkciókat a későbbiekben bővítették, így már lehetséges a rendszerben levő felhasználók erőforrás szabályozása is hála a pam_limits modulnak.

A modul által megvalósított szabályozás eltér az eddig bemutatott megoldásoktól. Az előző projektekben az erőforrás korlátok megadása közvetlen módon történt, ezzel ellentétben a modul segítségével csak közvetve módosíthatunk. A modul segítségével megjelölhetünk egy alternatív limits.conf nevű konfigurációs fájlt, mely felírja az alapértelmezett korlátokat tartalmazó fájlban foglaltakat. Magát a régi fájlt nem módosítja a modul, így probléma esetén visszaállítható a beállítás az eredeti értékekre.

A PAM megvalósítás először a 3.0.4-es Red Hat Linuxban jelent meg 1996-ban. Később az X/Open UNIX részeként standardizálták. Mára számos más operációs rendszer is integrálta a különböző Linuxokon kívül (például a FreeBSD).

A 2.2.1-es fejezetben belül több, különböző Linux operációs rendszerekben használt erőforrás-szabályozás megoldást ismertettem. A projektek között több olyan van, amit a saját feladatunk számára is részleges, vagy némi átalakítás után teljes megoldást jelenthetne a felvázolt hiányossággal kapcsolatban. Viszont azt vegyük észre, hogy ezen megoldások mindegyike kernel szintű módosítást igényelne (kivéve a cpulimit-et, de az ezzel kapcsolatos problémát a 2.2.2.1 a) részében ismertetem). Elviekben kivitelezhető lenne valamelyik megoldás integrálása az Android rendszer kernelébe, de az eddigi tapasztalataim szerint az ilyen jellegű feladatok óriási szakmai tudást és erőforrás kapacitást igényelnek, a folyamat nagy valószínűséggel még a Diplomatervezés tárgy keretein is túlmutatna. Emellett, ha mégis sikerülne megoldást találni rövid időn belül, akkor is a DroidLab projekt szempontjából nem lenne használható, mivel a kernel módosítása mindenképpen egy saját ROM létrehozását vonná maga után, amit a mérésben résztvevő eszközökre telepíteni kellene. Egy ROM

csere egy egyszerű felhasználó számára komoly feladatnak minősül, így nem várható el, hogy a mérés kedvéért mindenki lecserélje készüléke firmware-t.

Következésképpen olyan megoldást kell találni a problémára, amely szoftveresen a kliensen belül kivitelezhető, és a felhasználónak nem okoz extra tevékenységet a használatának biztosítása.

2.2.2 Szolgáltatás-csoportok és kvóták

Az általunk tervezett erőforrást korlátozó megoldást kvóta rendszernek neveztük el. A kvóták testesítik meg a felhasználó által felajánlott erőforrásokat. A kvóták a plugin-ekre jellemzőek, lesznek olyan plugin-ek, melyekre egyáltalán nem vonatkozik kvóta és lesznek olyanok is, melyek többféle kvótát is definiálnak. A felhasználó az általa biztosítani kívánt kvóta mennyiségeket az erőforráshoz kapcsolódó plugin felhasználói felületén szabályozhatja, hasonlóan a már korábban említett funkcionális korlátozáshoz. A felhasználó döntésének megkönnyítése érdekében egyes erőforrásokhoz néhány tipikus értéket választunk ki. Így a felhasználók érezni fogják a kvóták nagyságrendjét és tudatos döntést tudnak hozni.

A modulok is rendelkeznek kvótákkal, ezeket azonban nem a felhasználó szabályozza. A központi vezérlő feladata, hogy a felhasználók által felajánlott kvótákhoz modulokat rendeljen.

A kvóta rendszer megvalósításához szükség volt a lehetséges használható erőforrások számba vételére, ezeknek plugin-ekbe rendezésére illetve a kvóták jellegének és gyakoriságának meghatározására (ahol ez szükséges). Úgy láttam megfelelőnek, hogy a legtöbb erőforrást egyéni plugin-ben kell megvalósítani, mivel így használható ki maximálisan a moduláris kialakításból származó előny.

2.2.2.1 Eszközmonitorozó plugin-ek

Az eszközmonitorozó plugin-ek közé tartoznak azok a megvalósított plugin-ek, melyeknek feladata, hogy az adott mérőeszköz fizikai állapotáról információt gyűjtsenek. A plugin-ekhez nem terveztem külön kvóták definiálását, de ha a későbbiekben esetleg problémát okoz a felhasználóknak e plugin-ek energia fogyasztása, akkor a lekérdezések gyakoriságára vonatkozó korlátozási lehetőséget fogunk biztosítani.

- a) **Processzor és memória-használat:** a plugin segítségével feljegyezhetővé válna a készülék processzor illetve memória kihasználtságának értéke. Mint azt láthattuk a 2.2.1 fejezetben, más Linux disztribúciókon lehetséges a processzorral kapcsolatban egyéb beállítások megadása is. Az Androidban is lehet hasonló processzor paramétereket állítani, viszont az ilyen beállítások érvényesítéséhez úgynevezett root jogosultsággal kell rendelni az adott alkalmazásnak. Root jogosultságot csak szoftveresen módosított készülékeken lehet engedélyezni. Ez a módosítás nagy valószínűséggel az átlag felhasználó készülékén nincs elvégezve, így az adott funkciók használhatatlanok lennének. Ezért az ilyen jellegű beállítási lehetőségek a projekt kezdeti szakaszában nem kerülnek implementálásra. Későbbiekben, ha lesz rá igény és kapacitás, akkor egy frissítés keretein belül pótoljuk őket (az ehhez szükséges kvótákkal együtt).
- b) **Akkumulátor statisztika:** az Androidnak külön menedzser osztálya van az akkumulátor számára. A plugin segítségével számos adatot tudhatunk meg a készülék akkumulátorával kapcsolatban (például a töltöttségi szint, vagy a töltés típusa). Ezeket az adatokat más plugin-ek által szolgáltatott információkkal párosítva hasznos következtetéseket vonhatunk le például a felhasználói szokásokkal kapcsolatban.

2.2.2.2 Lokációs és hálózati információkat szolgáltató plugin-ek

A mobil eszközök legnagyobb előnye a nevükben is benne levő mobilitás, hordozhatóság. Ennek köszönhetően ezek a készülékek folyton velünk vannak, ez alkalmassá teszi őket a lokációs és hálózati információk gyűjtésére. Mivel az adatok nyeréséhez használt eljárások viszonylag energiaigényesnek mondhatóak, ezért a lekérdező műveletek számát és gyakoriságát mindenképpen korlátozni kell az kvóta rendszer segítségével.

- a) **GPS (Global Positioning System):** GPS vevő ma már minden mobil eszközben megtalálható, mivel a vevő kedvező ára és kellően kis mérete ezt lehetővé teszi. Kezdetben GPS-t csak a helymeghatározásra alkalmazták, majd megjelentek a GPS alapú navigációs alkalmazások, jelenleg pedig az úgynevezett helyfüggő szolgáltatások kerülnek előtérbe. A vevő a Föld körül keringő műholdakkal kommunikál és a gyűjtött adatokból hozza létre a szükséges információt. Az általunk tervezett plugin elsősorban a készülék pozícióját és a műholdakkal

kapcsolatos információkat képes mérni az idő függvényében. Ezekből utána más származtatott adatokat is képesek vagyunk előállítani, például két rövid időn belül mért egymás utáni pozícióból lehetséges sebességet becsülni. Jelenlegi tervek szerint a plugin-hez tartozó kvóták számát napi limitben fogjuk meghatározni és a lekérdezések közt eltelt időt pedig percekben.

- b) **cella információ:** egy aktív mobil eszköz és a szolgáltató hálózata folytonos kommunikációban áll egymással. A kommunikáció során az adott cella számos információt szolgáltat a készülék számára (például térerő, operátor név, egyéb hálózati események). Az információk egy része automatikusan érkezik, a többit a készüléknek van lehetősége lekérdezni. A cellainformációk nem csak hálózati adatokat tartalmaznak, lehetséges például helymeghatározást is végezni ezek alapján. Bár ez nagy valószínűséggel nem lesz olyan pontos, mint egy GPS alapú, viszont elképzelhető olyan eset, amikor valamilyen ok folytán csak ez használható. Az általunk létrehozott plugin összes cella által szolgáltatott információ gyűjtésére alkalmas. Mivel az említett kommunikáció egyébként is része telekommunikációs szolgáltatás nyújtásának, ezért kisebb energia szükséges a használatához, mint a GPS esetén. Így a tervezett extra lekérdezések számát másodperces intervallumban tervezzük korlátozni.

2.2.2.3 Adatforgalmat bonyolító plugin-ek

Kétféle vezeték nélküli adatkommunikációt különböztetünk meg a mobil eszközökkel kapcsolatban. Vannak rövid hatótávolságú, kisebb adat sebességű megoldások illetve vannak nagy átviteli sebességű és hatótávolságú kommunikációt megvalósító szabványok. Amíg a rövid hatótávú megoldásoknál a maga a használat, ami legjobban energia igényes, addig a nagy hatótávolságú megoldások esetén maga a forgalmazott adat, aminek korlátokat kell szabni a kvóta rendszer segítségével.

- a) **hálózati kommunikáció:** a hálózati kommunikációért felelős plugin-ben a két nagy hatótávolságú szabvánnyal, 3G-vel és a Wi-Fi-vel kapcsolatos funkciókat valósítjuk meg. A két szabványt nem választottuk szét, mivel nagyon sok közös tulajdonsággal rendelkeznek. Viszont teljes mértékben nem kezelhetjük őket egyenlőként. A 3G-t a mai világban mobil internetnek is nevezzük, ezt a szolgáltatást külön igényelheti a felhasználó a szolgáltatójától, aki ezért külön díjat számláz fel. Ezért ennek az erőforrás használatának tökéletes szabályozása

az egyik kardinális pontja a kvóta rendszernek. A plugin-ben definiált kvóták elsősorban az adatforgalom méretének korlátozását szolgálják.

- b) **Bluetooth:** a Bluetooth egy 1994 óta létező, rövid hatótávolságú, adatcseréhez használt vezeték nélküli szabvány. Alkalmazásával számítógépek, mobiltelefonok (telefonkihangosítók) és egyéb készülékek között tudunk rádiós kapcsolatot létrehozni. A plugin segítségével e kapcsolatok tulajdonságait szeretnénk mérni. Az ilyen jellegű információk számos jelenleg is futó kutatással kapcsolatban (például késleltetés-toleráns hálózatok) jól hasznosíthatóak. A Bluetooth kapcsolat kiépülése több egymástól jól elkülöníthető fázisra bontható (például eszköz felderítés vagy párosítás). Ezek a fázisok külön funkciókban lesznek megvalósítva. Ezekhez a funkciókhoz definiálunk kvótákat, melyek az adott funkcióban végzendő tevékenység napi korlátját határozzák meg.
- c) **NFC (Near field communication):** a NFC, rövid hatótávú kommunikációs szabványgyűjtemény okostelefonok és hasonló (általában mobil) eszközök között, egymáshoz érintéssel vagy egymáshoz nagyon közel helyezéssel (maximum pár centiméter) létrejövő rádiós kommunikációra. NFC nagyon alacsony sebességű adatátvitelt tesz lehetővé, de a kapcsolat extrém gyorsan jön létre két NFC kompatibilis eszköz között. A plugin ebben az esetben is a kommunikáció adatait rögzítené. Mivel az adatátvitel jelenti az NFC-vel kapcsolatos szűk keresztmetszetet, ezért az átviteli fájl méretét kell korlátozni.

2.2.2.4 Felhasználó orientált plugin-ek

Ebbe a csoportba tartoznak azok a plugin-ek, melyek konkrétan a felhasználóval kapcsolatos adatok gyűjtését végzik. Ezeknél a plugin-eknél nem a különböző erőforrások használata jelenti a problémát, hanem a személyes adatok kezelése és védelme. A kliens kialakítása során az egyik elsődleges szempont a felhasználók anonimitásának biztosítása volt, ennek ellenére ezeknél a plugin-eknél a definiált funkciók korlátozásának megfelelő megvalósítása létfontosságú követelmény. Indokolt kvóta definiálását nem láttam szükségesnek ezeknél a plugin-eknél.

- a) **Telefónia:** telefónia alatt a felhasználó telekommunikációs szokásainak összességét értjük. A megvalósított plugin segítségével vizsgálhatjuk az adott telefonon folytatott hívások illetve SMS-k (Short Message Service)

gyakoriságát, vagy a két előbb említett esemény között eltelt időt, átlagos híváshosszakat különböző napszakokban,...

- b) **Alkalmazások:** a létrehozott plugin célja, hogy a mérőkészülékre telepített alkalmazásokról minden lehetséges információt kinyerjen (telepített alkalmazások száma, csomagok neve, hozzáférési jogok,...).
- c) **Kontaktok:** a definiált plugin a kontaktok és a készüléken használt közösségi hálózatokból nyert információkat gyűjti későbbi statisztikai és egyéb célú felhasználás céljából

2.2.2.5 Egyéb készülékfüggő szenzor plugin

Utolsó csoportként következzen az a speciális plugin, amely az eddig nem ismertetett szenzorok felhasználását valósítja meg. Ezek a szenzorok nem találhatók meg minden eszközben, általában készülék illetve gyártó függők. Az Android rendszer lehetőséget biztosít az ilyen szenzorok felderítésére. A felderítés után a kvóta rendszerrel már beállítható a megfelelő korlát mennyiség.

Ezzel végére értem a mérőrendszerrel kapcsolatos információk ismertetésének. A következő fejezet fő témája azon eszközök részletes ismertetése lesz, melyek segítségével a felhasználókkal kapcsolatos rendszerigények (részvétel és erőforrás-megosztás) rendelkezésre állását szeretnénk növelni.

3 Gamification

Ebben a részben ismertetem a gamification-nel kapcsolatos fogalmakat módszerben rejlő lehetőségeket.

3.1 A gamification fogalma

A gamification egy olyannyira új tudományág, hogy eddig még nem sikerült mindenki által elfogadott, általános definíciót megfogalmazni vele kapcsolatban. A leggyakrabban a következő leírással próbálják meghatározni a lényegét: gamification-ön azt a folyamatot értjük, mikor játékokra jellemző elemeket és játéktervezési mechanizmusokat használunk valamilyen nem játékhoz köthető környezetben[10]. Csak akkor beszélhetünk ténylegesen a gamification-ról, ha az előbbi leírásban elhangzottak mindegyike teljesül.

Mielőtt a módszer részleteivel is megismerkednénk érdemes tisztázni pár dolgot a gamification-nel kapcsolatban, mivel sokak számára nem világos mit is jelent pontosan e kifejezés. Gyakran keverik össze más játékokhoz köthető fogalmakkal. Először is a gamification nem szándékozik minden tevékenységet játékká változtatni, sőt számos szempontból pont az ellenkezőjét szeretné elérni. A tényleges cél, hogy az alkalmazott módszerek által egy adott cselekvés sokkal élvezetesebbé váljon a cselekvő számára, vagy sokkal nagyobb motivációt érezzen az adott tevékenység elvégzésére. A gamification alkalmazásáról van szó akkor is, ha egy unalmas feladat kapcsán sikerül egy olyan részfeladatot találni, melyen különböző játékelemeket alkalmazva izgalmasabbá válik a munka. Tulajdonképpen a gamification felfogható egy hatékony eszköztárként, mely segítségével különböző valós (általában üzleti vonatkozású) problémákat oldhatunk meg.

A leggyakrabban a gamification-t az úgynevezett komoly játékokkal (serious games) tévesztik össze. A két kifejezés közti legnagyobb különbség, hogy az előbbivel ellentétben az utóbbi ténylegesen egyfajta játékot takar, melyet valamilyen virtuális környezetben generált probléma megoldásának gyakorlására hoztak létre. Ezzel ellentétben a gamification esetén az adott problémákat a valóságban kell megoldani.

3.2 Gamification kategóriák

A gamification három fő kategóriára osztható attól függően, hogy mit szeretnénk elérni az alkalmazásával. Ez a három kategória a következő: internal („belső”), external („külső”) és behavior change („viselkedést változtató”) gamification.

3.2.1 Internal gamification

Internal gamification-ról általában valamilyen vállalat vagy szervezet kapcsán beszélhetünk, gyakran szokták vállalati gamification-nek is nevezni. Ebben az esetben a gamification célja a szervezeten belüli termelékenység javítása az alkalmazottak motiválása segítségével.

Az internal gamification-nek két fontos sajátossága van a másik két kategóriához képest. Az első, hogy a résztvevők már tagjai egy meghatározott csoportnak, ezáltal egy bizonyos szinten ismerik egymást, gyakran közös érdek vezérli őket. Erre jó példa a Microsoft Honosítás Minősítési Játék (Microsoft Language Quality Game[11]). Röviden összefoglalva: a cégen belüli hálózaton elérhetővé tették egy alkalmazást, mely segítségével a még fejlesztés alatt álló Windows7 különböző honosításaiban lehetett hibát keresni. A dologban az az érdekes, hogy az alkalmazottak önkéntesen vettek részt a hibák keresésében, pedig nem járt jutalom az extra munkáért. Ez azzal magyarázható, hogy az alkalmazottak érdeke is a jó minőségű szoftver kiadása, nemcsak a cégé. Az indítvány olyan sikeres lett, hogy egy hónap leforgása alatt több mint 4600 alkalmazott 6000-nél több hibát jelentett be, melyek közül több száz tényleges javítást igényelt.

A második jellegzetesség az, hogy a motivációs készítésnek összhangban kell lennie a cégen belüli menedzselési és jutalmazási rendszerrel. Az Microsoft példája azért is volt ennyire hatékony, mert a résztvevők nem erre a munkára voltak felbérelve. Tulajdonképpen mondhatjuk azt, hogy a céghez fűződő loyálisuk miatt vettek részt a hibakeresésben, nem pedig azért, mert a megélhetésük függött tőle. Természetesen más motiváció is létezik a munkáltató iránt táplált hűségén kívül, például a munkatársak közötti versengés vagy egy új képesség megtanulásának lehetősége.

A DroidLab számára az internal gamification a kvóták méretének növelésében játszik szerepet. Az általunk megvalósított megoldás célja a projektben résztvevő felhasználók ösztönzése arra, hogy az általuk korábban beállított értékeket növeljék.

3.2.2 External gamification

External gamification-t általában valamilyen marketing céllal létrehozott alkalmazások kapcsán szoktak alkalmazni. A cél leendő ügyfelek gyűjtése és meglévő vásárlók megtartása, márkához fűződő kötődésük növelése.

Az external gamification-re jó példa egy kaliforniai napilap, a Record Searchlight története. Az utóbbi időben a társadalom olvasási szokásai nagymértékben megváltozott, papír alapú újságok egyre jobban a háttérbe szorulnak és a digitális terjesztésű tartalmak kerülnek előtérbe. A Record Searchlight menedzsment részlege is belátta, hogy nem tud ez ellen az új trend ellen küzdeni és stratégiát váltott. Új célként kitűzte, hogy egy stabil olvasói közösséget épít ki magának egy saját weboldalt használva. A legnagyobb kihívást az jelentette, hogy azt oldalt ritkán látogató felhasználókat a közösség aktív és megbecsült tagjává tegyék. A probléma megoldásaként bevezettek egy úgynevezett jelvényrendszert. Bár elsőre nem úgy tűnhet, de a jelvényeknek nagyon jelentős motivációs hatása lehet. Három hónap alatt az oldal látogatottsága 25%-kal növekedett és a cikkekhez írt hasznos kommentek száma is gyarapodott. A példa tanulságát levonva az external gamification egy gazdag eszköztárat biztosít a felhasználói szokások megértéséhez és a motiváció befolyásolásához.

A tervezett mérőrendszerrel kapcsolatban is fontos az external gamification, hiszen létfontosságú a megfelelő felhasználói kör kialakítása a működés szempontjából. A DroidLab kapcsán mi is létrehozunk egy jelvényrendszert, ami teljes mértékben a mérőrendszerhez lesz alakítva. A megszerzett jelvényeket a felhasználó a tervek szerint saját készülékén is és a webes felületen is megtekintheti. Emellett terveink között van a közösségi oldalak bizonyos szintű integrációja is, mely segítségével az adott felhasználó közzé tehetné az elért eredményeit. Természetesen a jelvényrendszeren kívül számos más eszköze létezik az external gamification-nek, melyeket egy részét mi is alkalmazni fogjuk. A további lehetőségeket a 3.3-as fejezetben láthatnak.

3.2.3 Behavior change gamification

Behavior change gamification alkalmazása során arra törekszünk, hogy minél nagyobb társadalmi körökben új, hasznos viselkedési szokásokat terjesszünk el. Ilyen hasznos szokás alatt számos dolgot érthetünk, következzen néhány példa:

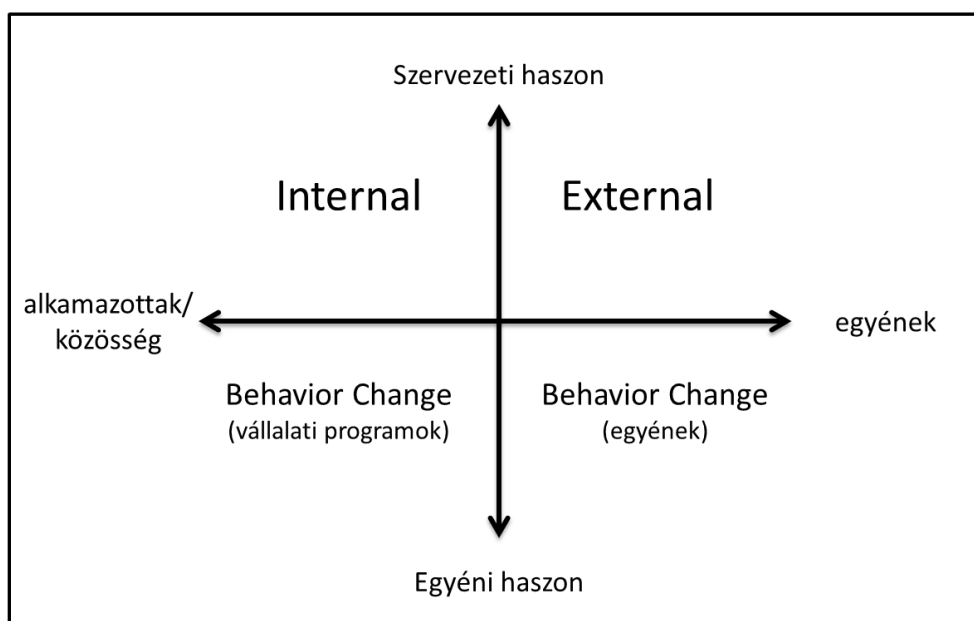
- egészséges életmód folytatása (egészséges étkezés, napi sportolás)

- új oktatási módszerek bevezetése
- hatékony megtakarítási technikák alkalmazása
- környezettudatos viselkedés elsajátítása

Hosszútávon ezeknek a változtatásoknak jelentős pozitív hatásai lehetnek. Az egészséges életmód elterjedése következtében feltehetőleg csökkenne a túlsúlyos emberek és az egészségügyi ellátásra szorulóknak száma, az oktatási reformok következtében növekedne az oktatás hatékonysága, a megfelelő pénzhasználat elsajátítása következtében tudatosabb pénzügyi döntések születnének és környezettudatosság elterjedésével csökkentene az egy főre jutó ökológiai lábnyom értéke. Az ilyen jellegű kezdeményeket az állam vagy valamilyen nonprofit szervezet szokta támogatni.

A projektünk szempontjából ez a gamification típus nem annyira releváns, mint az előző kettő. Nem áll szándékunkban a felhasználók bármilyen jellegű szokásait megváltoztatni. Esetleges hasznát akkor hozhat, ha az egész elképzelést kisebb csoportra vetítjük le és nem állandó, csak ideiglenes viselkedésbeli változást szeretnénk elérni (például a már említett Bluetooth mérés kapcsán). A projekt kezdeti szakaszában ezzel az eshetőséggel nem nagyon számolunk, viszont a megvalósítás úgy lesz kialakítva, hogy szükség szerint kisebb módosítások bevezetése mellett könnyen megvalósítható legyen.

Az fejezet zárásaként következzen egy ábra, melyen a különböző kategóriák egymással való kapcsolata látható (3.1. ábra):



3.1. ábra Gamification kategóriák közötti kapcsolatok

3.3 A gamification eszköztár

Ahogy az az eddig bemutatott példákból illetve az általam adott „definícióból” is megállapítható, a gamification sikeres használatának egyik feltétele a megfelelő játékelemek kiválasztása és helyes alkalmazása. Ezt még a folyamat kezdeti szakaszában érdemes megtenni, de a megfontolt döntés meghozásához ismerni kell a lehetőségeket. Az fejezet további részében az eszköztár bemutatása következik.

3.3.1 Az elemek hierarchikus piramisa

A játékelemeket három hierarchikusan egymásra épülő csoportra oszthatjuk. Minél nagyobb egy adott csoport szintje, annál kevesebb elemet tartalmaz.

3.3.1.1 Játék dinamizmusok

A játék dinamizmusok olyan játékra és a játékosra vonatkozó, időben változó folyamatok és módszerek, melyeknek köszönhetően a játék (vagy a játékosított tevékenység) élvezhetőbb lesz. A dinamizmusok elemek hierarchiájának legfelsőbb szintjén találhatók. A legfontosabb dinamizmusok a következők:

1. Korlátok (korlátozások vagy kényszerített kompromisszumok)
2. Érzelmek (kíváncsiság, csalódottság, boldogság)
3. Elbeszélés (egy következetes, folyamatos történet)
4. Fejlődés (a játékos karakterének fejlesztése)
5. Kapcsolatok (társadalmi interakciók, bajtársiasság, altruizmus)

3.3.1.2 Játék mechanizmusok

A játék mechanizmusok szabály alapú rendszerek, melyek cselekvésre ösztönzik a felhasználót. A szakirodalom tíz fontos mechanizmust szokott kiemelni:

1. Kihívások (rejtvények és egyéb megoldásra váró feladatok)
2. Véletlen (elemi véletlenszerűség)
3. Vetélkedés (nyertesek és vesztesek)
4. Együttműködés (összedolgozás másokkal a cél elérése érdekében)
5. Visszacsatolás (információnyújtás a játékos tevékenységéről)
6. Erőforrásgyűjtés (hasznos és gyűjthető dolgok szerzése)
7. Jutalmak (juttatás valamilyen tevékenységért)
8. Tranzakciók (játékosok közötti kereskedelem)
9. Körökre osztás (játékosok cselekvésének korlátozása)

10. Győzelmi feltételek

Mindegyik mechanizmus valamilyen módon kötődik legalább az egyik dinamizmushoz az előbb felsorolt öt közül. Például egy figyelmeztetés nélkül felugró ablak (2. mechanizmus) kíváncsiságot (2. dinamizmus) válthat ki a felhasználóból. Ebből az is következik, hogy a mechanizmusok a dinamizmusok alatt helyezkednek el a hierarchián belül.

3.3.1.3 Játék komponensek

A hierarchia legalsó szintjén találhatók a játék komponensek. Ezek a játékelemek adnak példákat és módszereket arra, hogy valósíthatjuk meg azokat a dolgokat, amelyeket a dinamizmusok és mechanizmusok reprezentálnak. Tizenöt fontos komponens különböztetünk meg:

1. Eredmények (meghatározott célok)
2. Avatárok (játékos karakterének vizuális reprezentációja)
3. Jelvények (eredmények vizuális reprezentációja)
4. Főellenség harcok (kifejezetten nehéz kihívás az adott pálya zárásaként)
5. Gyűjtemények (gyűjthető jelvények illetve tárgyak összessége)
6. Harc (meghatározott csaták)
7. Feloldható tartalmak
8. Ajándékozás (erőforrás-megosztás lehetőségének biztosítása)
9. Ranglisták (a játékos fejlődésének és eredményeinek vizuális megjelenítése)
10. Szintek (meghatározott lépcsőfokok a játékos fejlődésében)
11. Pontok (játék fejlődés numerikus reprezentációja)
12. Küldetések (előredefiniált kivívások konkrét célokkal és jutalmakkal)
13. Szociális hálózatok (a játékosok játékon belüli kapcsolatainak reprezentálása)
14. Csapatok (meghatározott játékosok egy csoportja, akik egy közös cél elérése érdekében együtt dolgoznak)
15. Virtuális javak (névleges vagy valódi értékkel bíró játékbeli tőke)

A mechanizmusokhoz hasonlóan a komponensek mindegyike egy vagy több felsőbb rétegbeli elemhez szorosan kapcsolódik.

3.3.2 A „PBL hármas”

A PBL a Points (pontok), Badges (jelvények), Leaderboards (ranglisták) játékelemek rövidítése. Vegyük észre, hogy mindhárom felsorolt elem a komponensek közé tartozik. Azért érdemes a PLB-vel külön foglalkozni, mivel az esetek többségében egy jó kiindulást biztosít egy gamification-t alkalmazó saját rendszer kialakításához.

A PBL nagyon hatékony és praktikus komponens, ha az ember megfelelően használja. De figyelembe kell venni azt is, hogy nem minden esetben garantálja az elvárt eredményt, megvan a maga korlátai. Ebből kifolyólag, ha maximálisan ki szeretnénk használni a gamification által nyújtott előnyöket, akkor szükséges egyéb komponensek igénybe vétele.

Mindezek ellenére a gyakorlat azt mutatja, hogy érdemes ezzel a három komponens vizsgálatával kezdeni a gamification folyamatát. Ehhez pedig tisztában kell lenni az egyes komponensek előnyeivel és hátrányaival.

3.3.2.1 Pontok

A pontok az egyik legalapvetőbb elemek közé tartoznak az olyan emberek motivációjával kapcsolatban, akik szeretnek dolgokat gyűjteni. Emellett nagyon egyszerűen lehet vele az adott játékos teljesítményét is mérni, így versengő felek közötti mérőszámként is hatékonyan alkalmazható. Természetesen egyéb más módon használhatjuk még a pontokat. A szakirodalom hat különböző esetet határozott meg, hogy milyen funkcionalitást tölthetnek be a pontok a gamification során:

1. **A pontok eredményt fejezhetnek ki.** Ez az egyik leggyakrabban alkalmazott használati mód. Ebben az esetben a pontok informálják a játékosokat, hogy hogyan végzik a tevékenységüket, például egy magasabb pontszám jelentheti azt, hogy az adott játékos többet játszott, vagy ügyesebben végezte a dolgát, mint az, akinek kevesebb pontja van.
2. **A pontok meghatározhatják, hogy mikor győztünk (ha létezik győzelmi feltétel).** Néha egyes folyamatok teljesítéséhez szükséges feltételeket pontokban határozzák meg a fejlesztők.
3. **A pontok kapcsolatot teremthetnek a végzett folyamat és az elérhető jutalmak között.** Számos gamification-t alkalmazó rendszerben a felhasználók az általuk gyűjtött pontokat valós vagy virtuális jutalmakra válthatják be.

4. **A pontok visszajelzést biztosíthatnak a játékosoknak.** A pontok segítségével nagyon egyszerűen hatékonyan kivitelezhető egy olyan visszacsatolást megvalósító mechanizmus, mely az adott játékosokat tájékoztatja, hogy milyen jól haladnak az adott tevékenységben.
5. **A pontok azonos tevékenységet végzők egyéni eredményeiről tájékoztatást nyújthatnak.** Ezek az értékek az résztvevők képességeit reprezentálhatják.
6. **A pontok adatokkal szolgálhatnak a fejlesztők számára.** A játékosok által elért pontok egyszerűen megfigyelhetők és tárolhatóak. Később ezeket az adatokat a fejlesztők elemezhetik, a kapott eredményeket felhasználhatják a rendszer hatékonyságának növelésére.

Ha tisztában vagyunk, hogy mit szeretnénk elérni a gamification során, akkor könnyen meghatározhatjuk, hogy a pontokat milyen módon használhatjuk fel. Viszont ügyelni kell, hogy ne várjunk el többet a pontok funkcionalitásaitól, mint amire valójában képesek.

3.3.2.2 Jelvények

A jelvények már párszor szóba kerültek az elmúlt pár fejezet során, most következnek egy kicsit részletesebb bemutatás. A jelvények voltaképp a felhasználó által elért eredmények vizuális reprezentációja. A jelvények és a pontok gyakran szoros kapcsolatban állnak egymással, gyakran definiálnak a fejlesztők olyan jelvényt, melyek bizonyos mennyiségű pontok eléréséért járnak.

Kutatások szerint egy jól megtervezett jelvényrendszer öt motivációt növelő jellemzővel rendelkezik:

1. A jelvények egyfajta célt adnak a felhasználónak, akinek azáltal, hogy e célok elérésére törekszik, növekszik az adott tevékenységgel kapcsolatos elkötelezettsége.
2. A jelvények útmutatást nyújtanak a rendszer által nyújtott lehetőségekről a felhasználónak. Fontos, hogy ebben az esetben a felhasználó az általa még nem birtokolt jelvényt is meg tudja nézni, illetve hogy e jelvények mellé fel legyen tüntetve a megszerzésük feltétele.
3. A jelvények a felhasználó által kedvelt dolgokat jelzik, illetve a tényleges tevékenységekkel kapcsolatos teljesítményekért járnak.

4. A jelvények egyfajta státuszsimbólumként szolgálnak.
5. A jelvények elősegítik a hasonló érdeklődést mutató személyek csoportba rendeződését.

A jelvények legfőbb tulajdonsága a rugalmasságuk. A jelvények számosságának és típusainak csak a fejlesztők képzelete és a játékosított tevékenység szab határt. Ebből következőleg a jelvények használatával sokkal változatosabb felhasználói kört tudunk megszólítani, mintha egy egyszerű pontrendszert vezettünk volna be. Bár a kategóriáknál bemutatott példában ez nem látszott, a jelvényeket érdemes internal gamification során is alkalmazni.

3.3.2.3 Ranglisták

A PBL utolsó bemutatott tagja a ranglisták. A három bemutatott komponens közül ez a legproblémásabb, mivel gyakran „kétélű fegyver” válhat belőle. Egyrészt, vannak olyan játékosok, akik tudni szeretnék, hogy a többiekhez képest milyen teljesítményt sikerült elérniük. A ranglista erre a feladatra tökéletes. Nagyon sok játékost hajtja a versenyszellem, gyakran motiváló hatást fejthet ki az a tudat, hogy csak pár ponttal van lemaradva rangsorban előtte lévő embertől.

Másrészt, ha az adott játékos látja, hogy legjobbaktól milyen messze áll a rangsorban, az könnyedén elveheti a kedvét az adott tevékenység folytatásától. Éppen ezért nagyon fontos, hogy ne egyetlen statikus rangsort készítsünk, sőt a legjobb megoldás, ha néhány szempontot figyelembe véve összetett ranglistákat hozunk létre.

4 Összefoglalás

Zárásképpen következzen egy rövid összefoglaló az eddig elvégzett és a jövőbeli feladatokkal kapcsolatban.

4.1 Eddig elvégzett feladatok

Az már a munka elkezdésekor nyilvánvalóvá vált, hogy a projektet nem lehet egy félév alatt befejezni. A csoporton belüli feladat szétosztásakor a rám eső feladatok olyan jellegűek, amelyek tényleges implementálásához a mérőrendszer legalább kezdetleges verziójára mindenképpen szükség van. Ennek hiányában a féléves munkám teljes egészét az irodalomkutatás és a feladatokkal kapcsolatos problémák megoldásának tervezése tette ki. A félév során elvégzett feladatok felsorolása tételesen:

- létező erőforrás-szabályozó módszerek keresése és tanulmányozása a projektben való használhatóság szempontjából
- Android platformon elérhető erőforrások megismerése, csoportosítása és a hozzájuk tartozó kvóták meghatározása
- kvóta rendszer implementációjának megtervezése, esetleges felmerülő problémák végiggondolása lehetséges megoldások keresése (nem került dokumentálásra)
- gamification módszerének megismerése (6 hetes videó tanfolyamon és az ott ajánlott szakirodalmon keresztül)
- mérőrendszer gamification-nal kapcsolatos elemeinek kezdetleges tervezése (nem került dokumentálásra)

4.2 Második félévre maradt feladatok

A diplomatervezés második felére jutott a tervezett komponensek implementációja. Remélhetőleg minél hamarabb lesz egy (az egész csoport számára) elérhető kezdetleges változata a mérőrendszernek. Ha ez mégsem lenne biztosított, akkor első lépésként módosítani kell az eddigi terveket az erőforrás korlátozás megvalósításához. A gamification-nel kapcsolatban nehezebb a mérőrendszert nélkülözni, ebben az esetben elképzelhető, hogy az implementációig nem jut el a

megtervezett megoldás. Ha minden a menetrend szerint halad, akkor az elsődleges feladat a kvóta rendszer implementálása lesz, majd ez a tesztelés következik és végezetül az eredmények dokumentációja. Ez után következhet a gamification-nel kapcsolatos elemek integrálása.

Tételesen kifejezve a jövőre vonatkozó feladatokat:

- kvótarendszer implementálása az eddig tervezettek során
- kvótákhoz meghatározott minta értékek véglegesítése
- tesztelés
- eredmények dokumentációja
- gamification elemek tervezésének befejezése és rendszerbe integrálása

Worst case scenario (nincs használható mérőrendszer):

- alternatív, egyszerűbb rendszer megvalósítása
- kvótarendszer implementálása az új körülményeket figyelembe véve
- tesztelés
- eredmények dokumentációja
- tervezett gamification elemek dokumentálása

Irodalomjegyzék

- [1] ORBIT(Open-Access Research Testbed for Next-Generation Wireless Networks), <http://orbit-lab.org/> (2013. május)
- [2] PlanetLab, <http://www.planet-lab.org/> (2013. május)
- [3] SETI@home, <http://setiathome.berkeley.edu/> (2013. május)
- [4] setrlimit, <http://docs.oracle.com/cd/E19082-01/819-2241/setrlimit-2/> (2013. május)
- [5] cpulimit, <http://cpulimit.sourceforge.net/> (2013. május)
- [6] quota, <http://unixhelp.ed.ac.uk/CGI/man-cgi?quota+1> (2013. május)
- [7] grsecutrity, <http://grsecurity.net/index.php> (2013. május)
- [8] cgroups, <https://www.kernel.org/doc/Documentation/cgroups/cgroups.txt> (2013. május)
- [9] PAM (Pluggable Authentication Modules), <http://www.linux-pam.org/> (2013. május)
- [10] S. Deterding, D. Dixon, R. Khaled és L. Nacke, „From Game Design Elements to Gamefulness: Defining "Gamification"” 2011.
- [11] Joshua Williams, Ross Smith, Score One for Quality!: Using Games to Improve Product Quality 2008