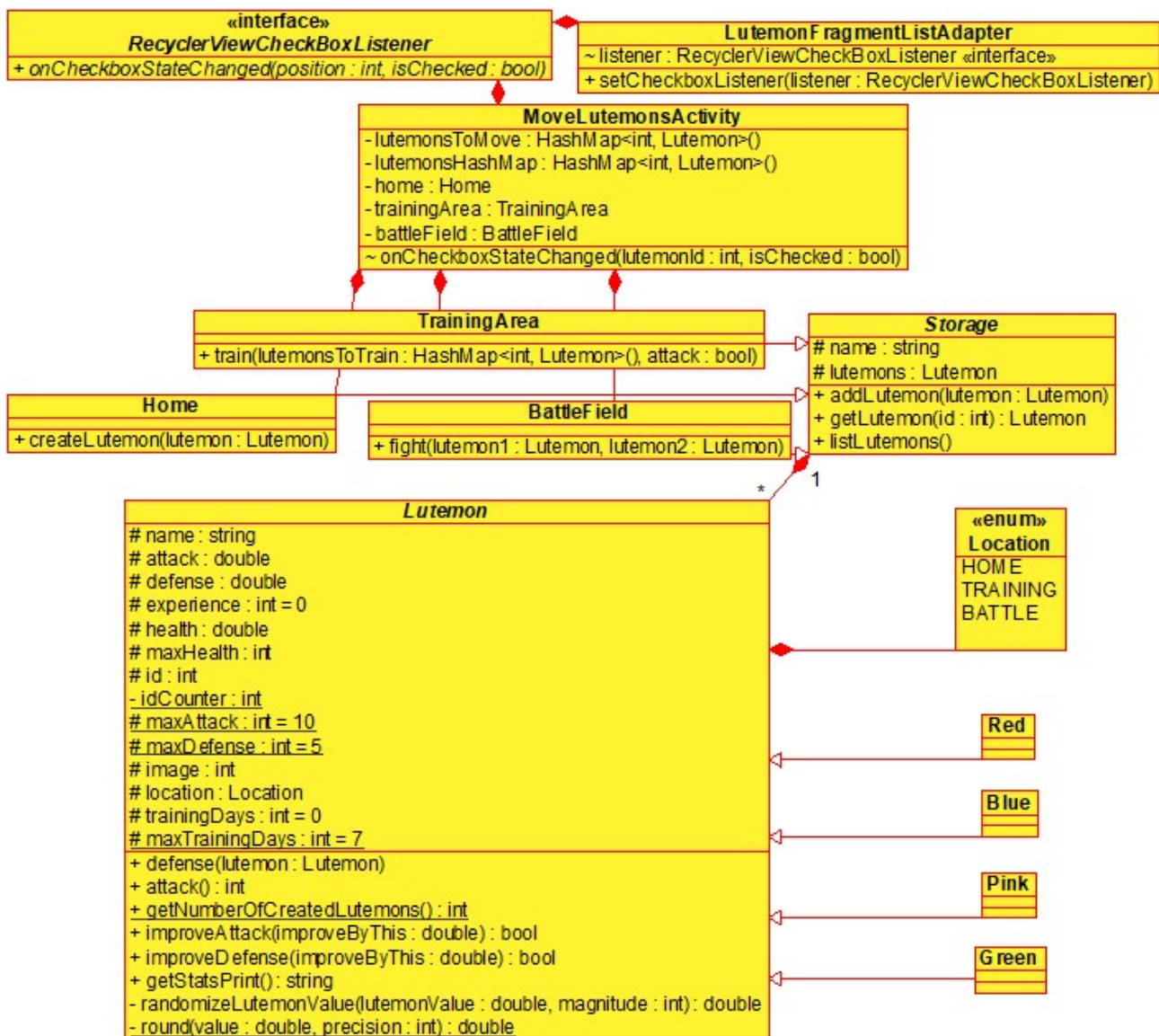


# HT-Dokumentointi

Samueli Aho 0565357 [samueli.aho@student.lut.fi](mailto:samueli.aho@student.lut.fi)

## 1 Luokkakaavio

Luokkakaavioon on otettu mukaan tärkeimmät luokat, ja käyttöliittymäluokat on jätetty poikkeuksia lukuunottamatta pois. Mukanaolevista luokista on kirjoitettu esiin ainoastaan tärkeimmät toiminnallisuudet, ja esim getterit ja setterit on jätetty pois.



## 2 Kuvaus työstä

### 2.1 Työnjako

Työ tehtiin itsenäisesti. Palautus on ryhmän nimeltä ”:(” palautuslaatikosta, koska työ aloitettiin tässä ryhmässä ja työsuunnitelma tehtiin tämän ryhmän kanssa. Varsinainen työ suunnitelmaa lukuun ottamatta kuitenkin tehtiin kokonaan itsenäisesti.

### 2.2 Implementoidut ominaisuudet

Alla työohjeesta kopioitu taulukko, johon on otettu mukaan vain ne rivit, jotka on työssä mukana. Kuvaus on joko työohjeen kuvaus tai itse täydennetty kuvaus implementoidusta ominaisuudesta.

Vaatus	Kuvaus	Pisteet
Oliokoodia	Ohjelma on koodattu olioparadigman mukaisesti	Pakollinen
Koodi on englanniksi	Ohjelman kaikki koodi ja kommentit ovat englanniksi. Dokumentaatio ja käyttöliittymäelementtien tekstit voivat olla suomeksi.	Pakollinen
Ohjelma toimii Androidilla	Ohjelman tulee toimia erilaisilla Android-puhelimilla ja se on koodattu Javalla Android-studiolla	Pakollinen
Ohjelma sisältää perustoiminnallisuuden	Ohjelma kattaa kaikki ne ominaisuudet, jotka ylempänä on määritetty, joten sillä voidaan pelaila Lutemonien kanssa	Pakollinen
Dokumentaatio	HT:n kanssa palautetaan dokumentaatio (PDF/Githubin .md-tiedosto), josta löytyy vähintään seuraavat asiat: luokkakaavio (käyttöliittymäluokkia [esim. Aktiviteetit] ei tarvitse sisällyttää luokkakaavioon), ryhmän jäsenten työnjako (kuka teki ja mitä), implementoidut ominaisuudet, yleinen kuvaus työstä.	
RecyclerView	Ohjelmassa käytetään RecyclerView-komponenttia listattaessa Lutemoneja ja niiden tietoja	3
Lutemoneilla on kuva	Erilaiset lutemonit on visualisoitu erilaisin kuvin. <ul style="list-style-type: none"><li>- Tehty. Lutemonin tyyppi ja kuva kulkevat käsi kädessä: niitä ei valita erikseen vaan esim. vihreällä lutemonilla on aina sama ennalta määritetty kuva.</li></ul>	2

Tilastot	<p>Pidä ohjelmassa kirjaa kuinka kukin taistelija on pärjännyt, eli kerro kuinka monta taistelua ja voittoa on tullut. Entäpä treenipäiviä?</p> <ul style="list-style-type: none"> <li>- Voitetut ja hävityt taistelut sekä treenipäivät nähtävillä kaikkien Lutemonien yhteisessä listauksessa.</li> </ul>	2
Kuolema pois	<p>Entäpä jos hävinnyt lutemoni ei kuolisikaan vaan palautuisi alkutilaan? Tällöin tilastoihin voisi tulla myös hävityt taistelut.</p> <ul style="list-style-type: none"> <li>- Lutemon kuolee vasta 3 hävityn taistelun jälkeen.</li> </ul>	1
Satunnaisuus	<p>Voisiko taisteluissa olla mukana myös himan satunnaisuutta (attack +Math.random()*3)?</p> <ul style="list-style-type: none"> <li>- Math.random() lisää tai vähentää Lutemonin attack ja defense -taitoja rajallisen määrän Lutemonin luonnin yhteydessä.</li> <li>- Taistelun aloittaja valitaan Math.random():in avulla (onko luku yli vai alle 0,5).</li> <li>- Lutemonia treenatessa ominaisuuden parannuksessa on hyödynnetty Math.random():ia.</li> </ul>	1
Fragementit	<p>Fragmentteja on käytetty työssä mielekkäästi</p> <ul style="list-style-type: none"> <li>- Lutemonien siirto-aktiviteetti on tehty fragmenttien avulla välilehti-designiä hyödyntäen.</li> </ul>	3
Oma ominaisuus: Treenaamiselle on rajat	<ul style="list-style-type: none"> <li>- Lutemonien treenaamista on rajoitettu. Koska treenaaminen kehittää valittua ominaisuutta, treenattaville ominaisuuksille on asetettu ylärajat, ja jotta nämä eivät olisi aina saavutettavissa, treenauskerroille eli ”treenipäiville” on Lutemon-kohtainen yläraja, 7. Tämän voi jakaa hyökkäys- ja puolustustreenaukseen miten haluaa.</li> <li>- Ominaisuuden parantuminen riippuu myös Math.random():sta: sitä ei voi tarkasti tietää etukäteen. Suuruusluokka 0-1,5 per treenipäivä.</li> </ul>	2 ?

### 2.3 Ohjelman käyttöliittymän rakenne:

#### MainActivity

- AddNewLutemonActivity
  - o Valitaan Lutemonin tyyppi, annetaan nimi ja luodaan Lutemon
    - Tyyppi määrittää lähtöarvot Lutemonin hyökkäys- puolustus ja maksimi elämäarvoille, joista hyökkäystä ja puolustusta muutetaan randomisoinnilla

- Lutemon menee automaattisesti luomisen yhteydessä sijaintiin Koti
- ListLutemonsActivity
  - Kaikki Lutemonin listataan (todennäköisesti aina) luomis- eli ID-järjestyksessä
  - Näkyvät tiedot: ID, kuva, nimi, tyyppi, sijainti, hyökkäys, puolustus, elämäarvo/maksimi elämäarvo, kokemuspisteet, voitot, tappiot ja treenipäivät
- MoveLutemonsActivity
  - Yläosa näkymästä on välilehtinäköymä, jossa on välilehdet Kotona/Treeni/Taistelu ja jokaisella välilehdellä on RecyclerView kyseisessä paikassa olevista Lutemoneista, jossa jokaisen Lutemonin kohdalla on karsitusti ainoastaan Checkbox ja sen perässä Lutemonin ID, nimi ja tyyppi
  - Alaosassa näkymää on pysyvästi RadioGroup siirron määränpääsijainnista ja nappi, jolla siirto suoritetaan
  - Valitaan siirrettävät Lutemonit Checkboxeilla, niiden määränpää ja siirretään.
- TrainLutemonsActivity
  - MoveLutemonsActivityn fragmenttien kanssa yhteisellä adapterilla tuotettu RecyclerView Lutemoneista, jotka ovat sillä hetkellä TrainingField-sijainnissa
  - Valitaan treenattavat Lutemonit Checkboxeilla sekä ominaisuus (hyökkäys tai puolustus) ja treenataan
  - Alhaalle tulee tekstimuotoinen palaute, jossa on treenattujen Lutemonien seuraavat tilastot: hyökkäys, puolustus, treenipäivät; sekä tarvittaessa huomautus siitä, että kaikki treenipäivä (kaikilla Lutemoneilla alussa 7) on jo käytetty
- BattleFieldActivity
  - Myös tässä on MoveLutemonsActivityn fragmenttien kanssa yhteisellä adapterilla tuotettu RecyclerView Lutemoneista, jotka ovat sillä hetkellä Battlefield-sijainnissa
  - Valitaan Lutemonit taisteluun Checkboxeilla (tasan 2) ja aloitetaan taistelu napista
  - Alhaalle tulee tekstimuotoinen palaute taistelun vaiheista. Taistelu koostuu kierroksista, joissa jompikumpi Lutemoneista hyökkää toisen kimppuun. Mikäli puolustavalla on vielä elämäpisteitä jäljellä tämän jälkeen, vuorot vaihtuvat ja tulee

uusi kierros. Ensimmäisen kierroksen aloittaja arvotaan `Math.random()`-luvulla. Jokaisen kierroksen alussa tulostetaan molempien perustiedot, sekä kierroksen lopuksi, että vieläkö puolustava Lutemon pystyy jatkamaan taistelua.

- Esimerkkituloste:

- 2 ensimmäistä kierrosta:

ID 5: Pinky (Pink) A: 7.8; D: 2.0; E: 0; H: 18.0/18  
ID 4: Vihru (Green) A: 5.2; D: 2.1; E: 0; H: 19.0/19  
5: Pinky(Pink) attacks 4: Vihru(Green)  
Vihru(Green) manages to escape defeat.

ID 4: Vihru (Green) A: 5.2; D: 2.1; E: 0; H: 13.3/19  
ID 5: Pinky (Pink) A: 7.8; D: 2.0; E: 0; H: 18.0/18  
4: Vihru(Green) attacks 5: Pinky(Pink)  
Pinky(Pink) manages to escape defeat.

- Viimeinen kierros:

ID 5: Pinky (Pink) A: 7.8; D: 2.0; E: 0; H: 8.4/18  
ID 4: Vihru (Green) A: 5.2; D: 2.1; E: 0; H: 1.9/19  
5: Pinky(Pink) attacks 4: Vihru(Green)  
Vihru(Green) is defeated in a fight.  
Vihru(Green) continues to live with 1 defeats.

- Jos Lutemon kokee 3. tappionsa elinaikanaan, se kuolee ja poistetaan lopullisesti.

## 2.4 Yleinen kuvaus työstä

Sovellus täyttää vaaditut kriteerit läpipääsyyn, mutta siinä on puutteita käytännöllisyyteen ja käyttäjäkokemukseen liittyen. Isoin puute on, että kaikissa kohteissa, joissa hyödynnetään `RecyclerView`iä saatavilla olevien Lutemonien tuomiseen esille valittavaksi erilaisten toimenpiteiden kohteeksi. Tällöin kyseiset listaukset tulisi ohjelman hyvän käyttökokemuksen tulisi olla dynaamisia ja päivittyä reaaliajassa kun Lutemoneja siirtyy paikasta toiseen, mutta lukuisista yrityksistä huolimatta tätä ei onnistuttu tässä harjoitustyössä toteuttamaan. Kyseiset listat päivittyvät ruudulle vasta, kun aktiviteetti käynnistetään uudelleen. Tämä olisi voinut olla helpompaa toteuttaa, jos näitä listoja ei olisi toteutettu `RecyclerView`illä, mikä olisi ollut pisteytettävien toiminnallisuuksien

puolesta myös hyväksyttyä, koska RecyclerViewä olisi joka tapauksessa käytetty kaikkien Lutemonien yhteisessä listauksessa. Listauksen olisi voinut esimerkiksi toteuttaa makeCheckboxes()-tyylisellä suunnittelulla siten, kyseisessä fragmentissa on ainoastaan tyhjä tila johon checkboxeja luodaan käskystä saatavilla olevan tietorakenteen mukaisesti oikea määrä, ja metodin olisi voinut suorittaa haluamassani kohdassa.

Ohjelmassa voi luoda Lutemoneja, treenauttaa niitä, tarkastella niiden tilastoja, siirtää kotiin lepäämään ja palauttamaan elämäpisteet maksimiin, siirtää eri sijainteihin, taistella Lutemoneilla toisiaan vastaan sekä seurata niiden taistelutilastoja ja jännittää, pysyykö Lutemon hengissä pidemmän aikaa vai tuleeeko se tapetuksi vahvempien toimesta.