



# Uitbreiding OVPong

Alexander Drechsel  
Sander Vlutters  
Stanley Huang  
Henley Ding  
Jeffrey Steen

[September 2013 - Januari 2014]



# 1 Inhoud

2	Inleiding .....	5
2.1	De aanleiding .....	5
2.2	De opdracht .....	5
2.3	Het doel .....	5
2.4	De uitwerking .....	5
3	Projectorganisatie.....	6
3.1	Taakverdeling .....	6
3.2	Begeleiding .....	6
3.3	Ontwikkelmethodiek .....	6
3.4	Documentversie .....	7
3.5	Bibliografie.....	7
4	Analyse .....	8
4.1	Domeinanalyse .....	8
4.1.1	Introductie .....	8
4.1.2	Terminologie.....	8
4.1.3	Algemene kennis domein .....	8
4.1.4	Klanten en gebruikers van het systeem .....	8
4.1.5	Taakomschrijving oude systeem .....	9
4.2	Requirements Analyse.....	10
4.2.1	Inleiding .....	10
4.2.2	Achtergrond.....	10
4.2.3	Omgeving en systeem modellen .....	10
4.2.4	Functionele eisen.....	10
4.2.5	Niet-functionele eisen .....	11
4.2.6	Wensen.....	11
4.3	Use-case analyse.....	12
4.3.1	Inleiding .....	12
4.3.2	Uitwerking .....	12
5	Ontwerp.....	13
5.1	Architectural Design .....	13
5.1.1	Client-server verbinding .....	13
5.1.2	Besturingssysteem.....	13
5.1.3	Interface .....	13
5.1.4	Power-ups/downs .....	17

5.1.5	Highscores .....	18
5.1.6	High-level klassendiagram .....	19
5.1.7	High-level klassendiagram server .....	20
5.2	Detailed Design.....	22
5.2.1	Detailed Activity-level.....	22
5.2.2	Detailed Network.....	25
5.2.3	Detailed Server .....	26
5.2.4	Detailed Model .....	28
5.2.5	Detailed Obstacles.....	30
5.2.6	Detailed PowerUp.....	31
5.2.7	Detailed Shared .....	33
5.2.8	Detailed View.....	35
5.3	Uitleg Protocol.....	37
6	Testen .....	38
7	Evaluatie .....	39
7.1	Het begin .....	39
7.2	Samenwerking .....	39
7.3	Communicatie met begeleiders .....	39
7.4	Watervalmodel.....	39
7.5	Opslag en versiebeheer .....	40
7.6	Aanbevelingen .....	40
7.6.1	Verbinding verbeteren .....	40
7.6.2	Automatisch inloggen.....	40
7.6.3	Encryption.....	40
7.6.4	Achievements .....	40
7.6.5	Globale highscorelijst .....	40
7.6.6	Chat.....	40
7.6.7	Meer power-ups en obstakels.....	40
7.6.8	DNS .....	41
8	Appendix A – Use Cases.....	42
8.1.1	Traceability Matrix: Requirements Use-Cases.....	58
8.1.2	Traceability Matrix: Activity’s Use-Cases.....	59
8.1.3	Use case Diagram .....	60
9	Appendix B – Protocol .....	61
9.1	Activity Events .....	61

9.1.1	LoginActivity .....	61
9.1.2	MainMenuActivity .....	61
9.1.3	LobbyActivity .....	61
9.1.4	RoomActivity .....	62
9.1.5	Meerdere Activity's .....	62
9.2	Game Events .....	63
9.2.1	Intern .....	63
9.2.2	Extern.....	63
10	Appendix C - Testcases .....	64
10.1.1	Traceability Matrix: Requirements Testcases.....	78
10.1.2	Traceability Matrix: Usecases Testcases.....	79
11	Appendix D - User Manual.....	81

## 2 Inleiding

### 2.1 De aanleiding

Dit project is onderdeel van het vak Ontwerpproject aan de Universiteit Twente. Voor dit vak moet een software product ontworpen en ontwikkeld worden. Er werden enkele opties voorgesteld. De meest aantrekkelijke optie was het tablet project van OVSoftware. Dit project bleek vorig jaar al uitgevoerd te zijn, maar er werd door OVSoftware een uitbreiding op dat project voorgesteld. Er is besloten om deze uitbreiding uit te voeren.

### 2.2 De opdracht

Het product dat door de voorgaande groep is gemaakt, is een applicatie voor Android om via een peer-to-peer structuur met andere Android-apparaten over Wi-Fi pong te spelen. Hierbij was het alleen mogelijk om het spel met meerdere mensen te spelen op hetzelfde Wi-Fi-netwerk. Ook was er niet de mogelijkheid om te kiezen met welke andere spelers er gespeeld kon worden. De opdracht bevat het vervangen van de peer-to-peer structuur met een client-server structuur. Deze server zal over het internet bereikt kunnen worden, zodat er overal verbinding met de server kan worden opgezet. Op deze server zal er informatie centraal worden opgeslagen en gedeeld, zoals behaalde highscores en accountgegevens. Alle communicatie tussen de clients zal via de server lopen. Clients moeten inloggen op een account, waarna ze de lobby kunnen binnengaan om een spel te vinden. In de lobby kunnen ze kiezen om zelf kamers aan te maken waar andere spelers kunnen binnengaan en ze andere spelers kunnen uitnodigen, of een al bestaande kamer binnen te gaan. Verder bevat de opdracht het uitbreiden van het spel met power-ups en power-downs. Een van deze power-downs is het verschijnen van obstakels, die op het speelveld moeten worden weergegeven als aan OVSoftware gerelateerde objecten. Verdere mogelijke power-ups zijn: versnelling of vertraging van de bal en het vergroten of verkleinen van de bat.

### 2.3 Het doel

Het doel van het project is om beter bekend te raken met de praktijk van het ontwikkelen van een product. Er wordt ervaring opgedaan met het opstellen van de eisen die de opdrachtgever voor het te ontwikkelen systeem heeft, het ontwerpen van dit systeem, bij het ontwerp een prototype te maken, dit prototype te testen, het produceren van een eindproduct en het documenteren van alle stappen. Verder wordt er ervaring opgedaan met het afstemmen van het ontwerp met de opdrachtgever en het samenwerken in een team. Dit alles zal zowel mondeling als schriftelijk gepresenteerd worden.

Dit product zal bij OVSoftware als demoproduct aan potentiële klanten worden laten zien.

### 2.4 De uitwerking

Bij het ontwikkelen van het systeem is er wekelijks een moment geweest waarop er bij OVSoftware aan het systeem gewerkt wordt, zodat er voortgang kon worden getoond en er om advies en commentaar kon worden gevraagd. Op andere dagen werd er op de UT gewerkt. Ook werd er wekelijks met de begeleider op de UT de voortgang besproken.

Het systeem werd ontwikkeld met behulp van het watervalmodel. Dit wordt verder besproken in paragraaf 3.2.

Het opzetten van het project duurde langer dan gepland, omdat de begeleider die eerst was voorgesteld geen tijd had, en een andere begeleider gevonden moest worden. Ook liep de communicatie met OVSoftware aan het begin moeizaam, waardoor het project vier weken later dan gepland pas van start kon gaan.

## 3 Projectorganisatie

### 3.1 Taakverdeling

Naam	Taak
Alexander Drechsel	Client code, contactpersoon, quality control
Sander Vlutters	Client code, testen
Henley Ding	Server code, testen
Jeffrey Steen	Server code, testen
Stanley Huang	Client code, documentatie

### 3.2 Begeleiding

Naam	Taak
Wojciech Mostowski	Begeleiding UT
Marieke Huisman	Begeleiding UT
Eric Barten	Begeleiding OVSoftware en opdrachtgever

Er is begeleiding geweest vanuit zowel de Universiteit Twente en OVSoftware. Marieke Huisman en Wojciech Mostowski hebben het project begeleid en feedback gegeven over de loop van het project en over het projectverslag. Elke week is er een moment geweest waarop er met de begeleiders werd besproken welke taken iedereen verricht heeft en is er een vooruitzicht gegeven voor de komende week.

Vanuit OVSoftware is Eric Barten zowel de opdrachtgever als de begeleider van het project. Met hem werd elke week gekeken of het project op schema was en of er technische problemen waren. Ook is er feedback geweest van Eric Barten op het projectverslag.

### 3.3 Ontwikkelmethodiek

In dit project is gebruik gemaakt van een vorm van het watervalmodel. Dit is de meest voor de hand liggende methode voor ons, omdat deze het bekendst is in de groep en eerder is gebruikt in Software Engineering Modellen. Ook komt het watervalmodel het meest overeen met de uiteindelijke deliverables. De structuur van het uiteindelijke verslag bestaat namelijk uit gedeelten over de requirements, het globale ontwerp, het detailontwerp en een testplan met testresultaten.

### 3.4 Documentversie

Versie	Datum	Beschrijving	Auteur
0.1	14-10-2013	Eerste opmaak	Alexander Drechsel
0.11	16-10-2013	Use cases etc. toegevoegd	Iedereen
0.12	15-11-2013	Power-ups/downs en Highscores van het Ontwerp toegevoegd	Sander Vlutters
0.13	22-11-2013	Voorlopig definitieve versie van het protocol toegevoegd, Schets van black box testcases	Alexander Drechsel
0.14	25-11-2013	Black box testcases toegevoegd, kleine update van events	Alexander Drechsel
0.15	29-11-2013	Detailed Activity's en high-level design toegevoegd	Stanley Huang
0.16	02-12-2013	Detailed Activity's geüpdatet	Stanley Huang
0.17	04-12-2013	Aanpassingen na Eric's review	Stanley Huang
0.18	06-12-2013	Aanpassingen na Eric's review	Sander Vlutters
0.19	06-12-2013	Toevoeging server, kleine aanpassingen	Stanley Huang
0.191	06-12-2013	Kleine toevoeging bij detailed design server	Henley Ding
0.2	13-12-2013	Toevoegingen bij detailed design	Stanley Huang
0.21	16-12-2013	Detailed Shared toegevoegd, plaatjes mooier gemaakt, game gerelateerde test cases toegevoegd	Stanley Huang
0.22	18-12-2013	Test cases geüpdatet	Stanley Huang
0.3	06-01-2014	Appendix Protocol en Highlevel design geüpdatet	Alexander Drechsel & Stanley Huang
0.31	06-01-2014	Diagrammen geüpdate, kleine verbeteringen	Stanley Huang
0.32	06-01-2014	Test cases geüpdatet	Henley Ding, Jeffrey Steen
0.33	10-01-2014	Test cases geüpdatet + comments toegepast	Henley Ding, Jeffrey Steen, Stanley Huang
0.34	13-01-2014	Use cases geüpdatet + nieuwe tracability matrixes	Alexander Drechsel
0.35	15-01-2014	High level client-server toegevoegd	Stanley Huang
0.36	19-01-2014	Evaluatie toegevoegd	Stanley Huang
0.37	25-01-2014	Testcases update	Henley Ding
0.38	29-01-2014	Testcases netter gemaakt, Testen toegevoegd	Stanley Huang, Henley Ding
0.39	30-01-2014	Evaluatie verbeterd, spellingscontrole	Stanley Huang, Henley Ding, Jeffrey Steen, Alexander Drechsel
1.0	31-01-2014	Uiteindelijke versie	Iedereen

### 3.5 Bibliografie

Bronnen
<a href="http://stackoverflow.com">stackoverflow.com</a>
<a href="http://developer.android.com">developer.android.com</a>

## 4 Analyse

### 4.1 Domeinanalyse

#### 4.1.1 Introductie

In dit hoofdstuk wordt de domeinanalyse uitgevoerd. Hieronder vallen terminologie, informatie over het domein, gebruikers van het systeem en omschrijving van het oude systeem. De reden om een domeinanalyse uit te voeren is voor een snellere ontwikkeling van het product, omdat het duidelijk is in welke omgeving het gebruikt gaat worden zodat er gericht ontwikkeld kan worden. Het domein kan omschreven worden als “Android games voor OVSoftware”.

#### 4.1.2 Terminologie

Term	Beschrijving
Account	Gegevens waaraan de server de speler herkent.
Bat	Het balkje dat in het spel wordt gebruikt om de bal af te stoten.
Host	De persoon die de room aangemaakt heeft en daarmee de beheerder van de room is.
Invite	Een uitnodiging voor een speler om een room toe te treden.
Kick	Een functionaliteit van de host om een speler uit de room te zetten.
Lives	Het aantal keren dat een speler de bal mag missen voordat hij af is.
Lobby	Een ruimte waar elke speler van OVPong heen kan waarin je rooms aan kan maken of in kan gaan.
Obstakel	Een rechthoekige hindernis in het speelveld die een éénmalig effect heeft op de bal.
OVPong	Het Pong-spel dat in 2013 ontwikkeld is en aangepast is aan de eisen van OVSoftware.
OVSoftware	Het bedrijf waar het project wordt uitgevoerd.
Power-up	Een object dat als het opgepakt wordt een positief effect heeft voor die speler.
Power-down	Een object dat als het opgepakt wordt een negatief effect heeft voor een andere speler.
Quick Play	Een functie in het spel die ervoor zorgt dat een individuele speler automatisch wordt gekoppeld aan een andere speler voor een spelletje OVPong.
Room	Een kamer van waaruit spelers samen een spel kunnen starten.
Slot	Een ruimte in de room waar één speler in past. Deze ruimte kan gesloten worden, zodat er geen spelers in kunnen.

#### 4.1.3 Algemene kennis domein

Er bestaat al een oudere versie van OVPong die op Android werkt.

OVPong is gebaseerd op het spelletje Pong. Pong wordt gespeeld door 2 personen, die van elkaar moeten winnen door een bal heen en weer te kaatsen met behulp van een bat. Als de tegenstander de bal niet terug kan kaatsen, dan krijg je één punt.

#### 4.1.4 Klanten en gebruikers van het systeem

OVSoftware wil het nieuwe systeem graag intern gebruiken en het als demonstratie laten zien aan potentiële klanten.



#### 4.1.5 Taakomschrijving oude systeem

De oude versie van OVPong laat je een peer-to-peer spel opstarten met de eerste tegenspelers die gevonden worden. Daarbij is het ook mogelijk om meerdere spelers tegen elkaar te laten spelen en bestaat er de optie om meerdere ballen te gebruiken. Het is alleen mogelijk om lokaal spellen op te starten met behulp van een Wi-Fi-verbinding.

## 4.2 Requirements Analyse

### 4.2.1 Inleiding

Het doel van deze fase is de systeemgrenzen bepalen en op basis hiervan requirements opstellen die dit zullen volbrengen.

### 4.2.2 Achtergrond

De oude versie van OVPong ondersteunt spellen van meerdere spelers als deze op hetzelfde netwerk zitten en het netwerk UDP-broadcasting toestaat. OVPong start met de eerste spelers die het kan vinden, dus als bijvoorbeeld 2 verschillende groepen op hetzelfde netwerk een spel starten, kan het zijn dat spelers in het spel van de andere groep terechtkomen. Het spel zelf is een niet-gecompliceerde variant van Pong waarbij elke speler alleen zijn eigen speelveld en bat te zien krijgt.

### 4.2.3 Omgeving en systeem modellen

Aangezien er een bestaand programma wordt uitgebreid ligt het voor een groot gedeelte al vast waar het programma op draait. Het programma wordt alleen ontwikkeld voor Android (3.0 of hoger). De communicatie verloopt over Wi-Fi.

### 4.2.4 Functionele eisen

Het systeem zal:

- F1: een werkend OVPong spel zijn met meerdere power-ups/power-downs die tijdens het spel getoond worden.  
*Één van de doelen van het project is de bestaande versie van OVPong uitbreiden met power-ups.*
  - F1.1: aan OVSoftware gerelateerde obstakels bevatten.  
*Hier is specifiek om gevraagd door OVSoftware*
- F2: een duidelijke interface hebben.
  - F2.1: alleen het eigen deel van het speelveld weergeven.  
*In de huidige versie van OVPong wordt het eigen deel getoond. Aangezien het aantal spelers variabel is, blijft dit nog steeds een logische keuze.*
  - F2.2: een instructiescherm kunnen tonen met uitleg over de loop van het spel en de power-ups.  
*Omdat vanuit F2.1 alleen het eigen deel van het speelveld wordt weergegeven en een deel van de power-ups tegenstanders beïnvloedt, is het handig om een scherm te hebben waar de effecten van power-ups worden uitgelegd.*
  - F2.3: touch-besturing bevatten.  
*De huidige versie van OVPong wordt door touch-besturing bestuurd. Omdat er voor tablets en smartphones wordt ontwikkeld is touch-besturing nog steeds een goede keuze.*
- F3: highscores bijhouden.  
*Één van de doelen van het project is de bestaande versie van OVPong uitbreiden met highscores.*
  - F3.1: de behaalde highscores evenredig maken met het aantal spelers dat aan het spel heeft meegedaan.  
*Voor elk spel moet de te verdienen highscore gelijk zijn, evenredig aan de behaalde prestatie.*
- F4: spelers toestaan om met specifieke spelers te spelen.  
*Één van de doelen van het project is de bestaande versie van OVPong uitbreiden met de optie om specifieke spelers uit te dagen.*
- F5: een nieuw spel kunnen opstarten als het huidige spel is afgelopen zonder de app opnieuw te moeten opstarten.  
*Om het gebruiksgemak bij meerdere opeenvolgende spellen te vergroten, is het netjes afsluiten van het spel en terugkeren naar een hoofdmenu gewenst.*

#### 4.2.5 Niet-functionele eisen

Het systeem zal:

- N1: werkend op Android opgeleverd worden.  
*Net als het oorspronkelijke programma wordt ook de uitgebreide versie voor Android ontwikkeld.*
- N2: ontwikkeld worden als demoproduct voor OVSoftware.  
*Volgens domeinanalyse: beschrijving van de klanten en gebruikers, wordt het product ontwikkeld als demoproduct voor OVSoftware.*
  - N2.1 Voldoende netjes zijn afgewerkt  
*Volgens N2 wordt het als demoproduct ontwikkeld en moet het dus toonbaar zijn.*
  - N2.2 Branding van OVSoftware bevatten.  
*Volgens N2 wordt het als demoproduct ontwikkeld voor OVSoftware.*
  - N2.3 Geschikt zijn voor gebruik door zowel technische als niet-technische gebruikers.  
*Volgens N2 wordt het als demoproduct ontwikkeld. De klanten aan wie OVSoftware het product toont zijn mogelijk niet-technische gebruikers en het product moet dus ook voor niet-technische gebruikers bruikbaar zijn.*

#### 4.2.6 Wensen

Het systeem zal:

- W1: de bal realistischer laten ketsen.  
*In de huidige versie van het systeem kunnen ballen maar een gelimiteerd aantal richtingen opgaan.*
- W2: zijn communicatieberichten beveiligen.  
*Als berichten beveiligd zijn wordt het lastiger om een spel van buiten te verpesten of highscores te manipuleren.*
- W3: singleplayer ondersteunen.  
*Omdat een speler alleen zijn eigen scherm ziet is het goed mogelijk om een singleplayer variant te maken.*
- W4: achievements ondersteunen.  
*Het is een mooie toevoeging naast highscores.*
- W5: chatten ondersteunen.  
*Communicatie in de lobby is gewenst.*
- W6: globale highscorelijst bevatten.  
*Een oude functie die een wens is geworden.*

## 4.3 Use-case analyse

### 4.3.1 Inleiding

Een use case beschrijft de verschillende manieren waarop een gebruiker interactie heeft met het systeem en de respons van het systeem daarop.

### 4.3.2 Uitwerking

De use cases zijn te vinden in appendix A.

Er zijn 4 actoren aanwezig in de use cases, namelijk user, host, kicked player en targeted player. Een user is elke gebruiker en heeft alle functionaliteiten met uitzondering van een aantal functionaliteiten in een room die alleen ter beschikking zijn van een host. Een host is een user, maar is daarnaast ook de houder van een room. Deze host heeft naast de functionaliteiten die de user heeft ook nog de mogelijkheid om binnen de room de slots te beheren (d.w.z. de grootte van de room te veranderen door slots te openen/sluiten en players te kicken), instellingen te veranderen (de hoeveelheid lives en balls aanpassen), spelers te inviten en tenslotte nog de mogelijkheid om het spel te starten. Een kicked player is een user die gekickt is. Een targeted player is een user die een power-down heeft ontvangen.

## 5 Ontwerp

### 5.1 Architectural Design

In het volgende deel zal nader worden ingegaan op het ontwerp en de keuzes die daarbij op basis van de requirements gemaakt zijn. Deze betreffen de client-server verbinding, het besturingssysteem, de interface, de power-ups/downs en de highscores.

#### 5.1.1 Client-server verbinding

In het systeem wordt gebruik gemaakt van een client-server verbinding. De server houdt informatie van de gebruikers bij, zoals gebruikersnamen, wachtwoorden en highscores. Hierdoor kan de client inloggen op de server voordat er een spel gespeeld kan worden. Aan de client kant zijn er verschillende schermen aanwezig met verschillende functies per scherm. Dit ondersteunt de server door per client bij te houden in welk scherm deze zich bevindt, mocht het relevant zijn voor de server. Dat kunnen bijvoorbeeld het kamerscherm of het spelscherm zijn.

#### 5.1.2 Besturingssysteem

Het programma OVPong is ontwikkeld om te werken op Android-apparaten. Net als alle andere Android-applicaties draait het programma op Java. Dit komt omdat Android standaard ook op Java draait.

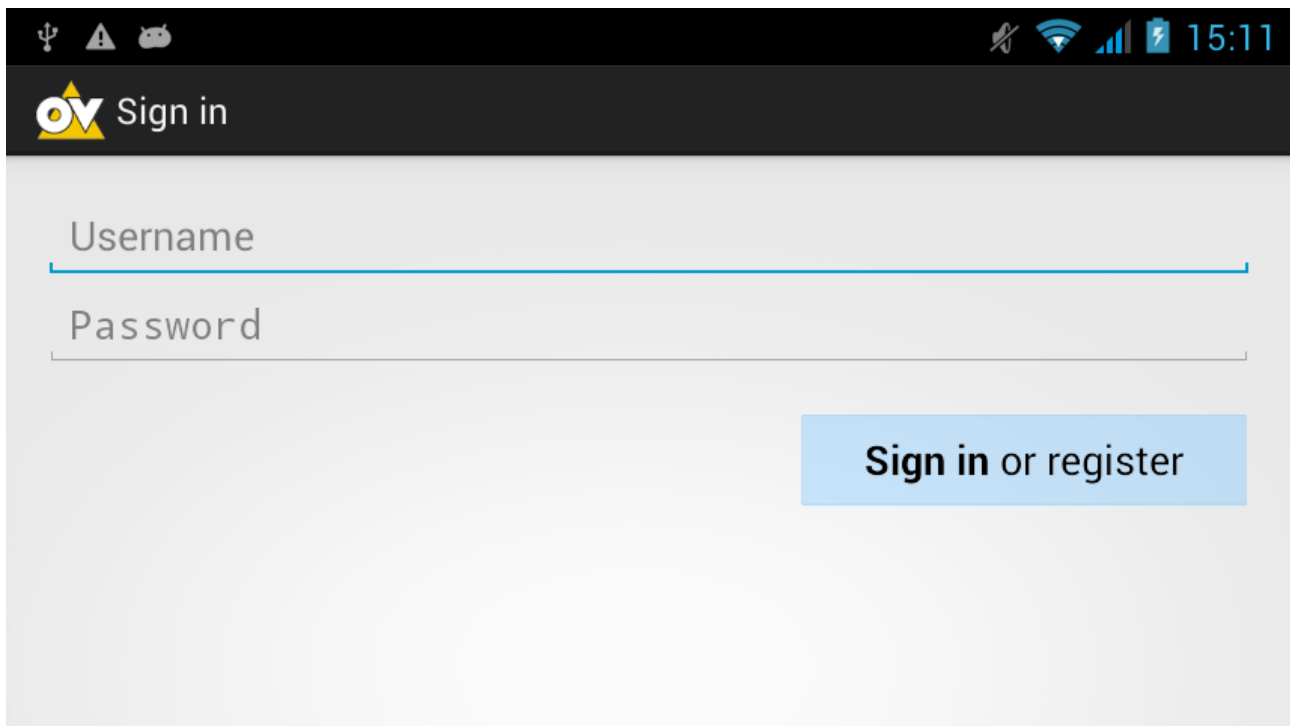
De server is ook een Java-programma, maar aangezien de server geen Android-specifieke functionaliteit heeft, kan de server op verschillende besturingssystemen opgezet worden.

#### 5.1.3 Interface

De interface heeft in het huidige design veel meer knoppen en opties dan de vorige versie van OVPong. Dat komt doordat er nieuwe functionaliteiten aan toegevoegd zijn, zoals een lobby-systeem met kamers. Er is eerst gekeken naar de hoeveelheid schermruimte die beschikbaar is en aan de hand daarvan is besloten dat er meerdere schermen nodig zijn. Als er te veel interfacefuncties, zoals knoppen en tekst, op één scherm staan, zal het lastig worden voor de smartphonegebruiker om de app te gebruiken. Ook wordt het onoverzichtelijk door een overvloed aan knoppen en opties.

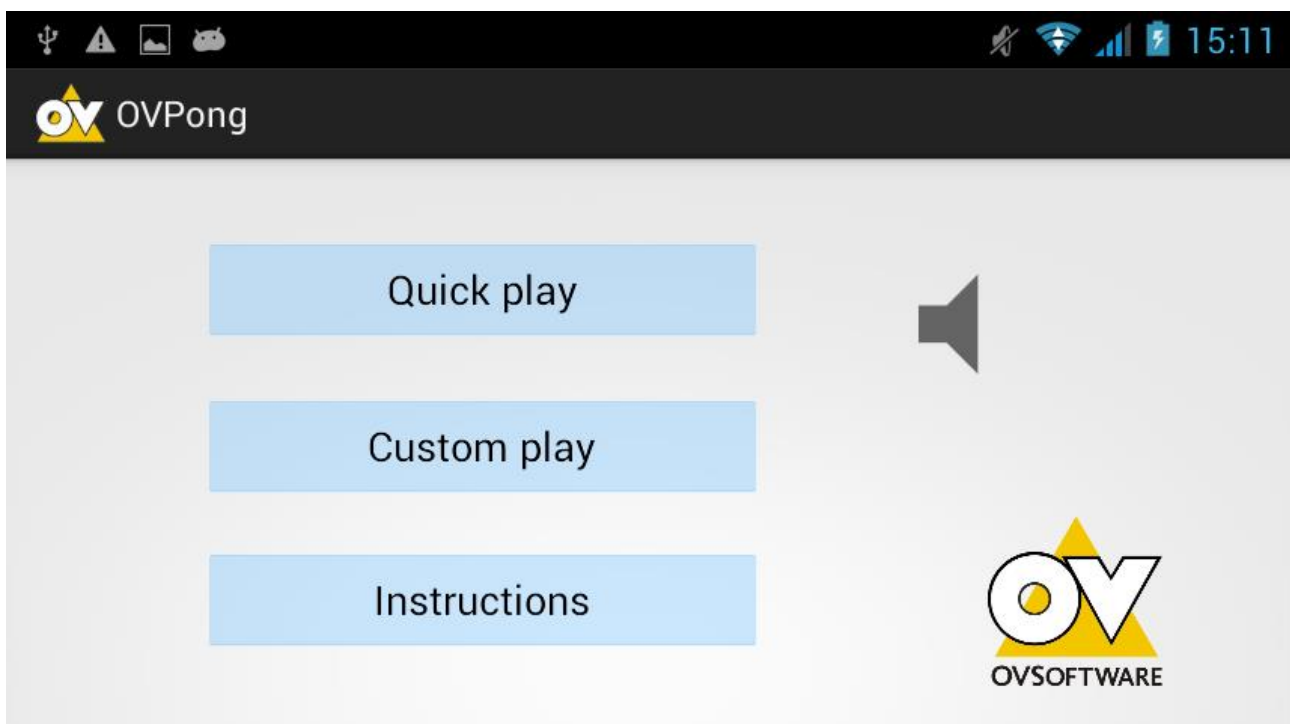
Het huidige design is opgedeeld in 7 schermen, namelijk splashscreen, loginscherm, menuscherm, lobby-scherm, kamerscherm, instructiescherm en een gamescherm.

Als de app wordt gestart, zal het splashscreen worden weergegeven. Dit scherm is overgenomen van het vorige systeem, en vervult eis N2.2. Daarna komt de gebruiker direct in het loginscherm, waar met een account (F3, F4) ingelogd kan worden. Een account is nodig om de highscorefunctie en de custom game functie te ondersteunen, omdat elke speler uniek identificeerbaar en herkenbaar moet zijn. Het account bestaat uit een naam en een wachtwoord, waarvan de naam uniek moet zijn op de server. Als er op de "OK"-knop wordt gedrukt, kunnen er 3 dingen gebeuren. Als het account al een keer eerder aan is gemaakt, wordt er gecontroleerd of het wachtwoord klopt. Als dat klopt, dan wordt de user ingelogd in het systeem met dit account. Indien dit niet klopt, dan zal de user een pop-up te zien krijgen waarin staat dat het inloggen niet gelukt is. Als de naam nog niet in het systeem geregistreerd is, is er nog geen account met die specifieke naam aangemaakt, dus zal er één worden geregistreerd in het systeem met het bijbehorende wachtwoord, waarna de user in wordt gelogd.



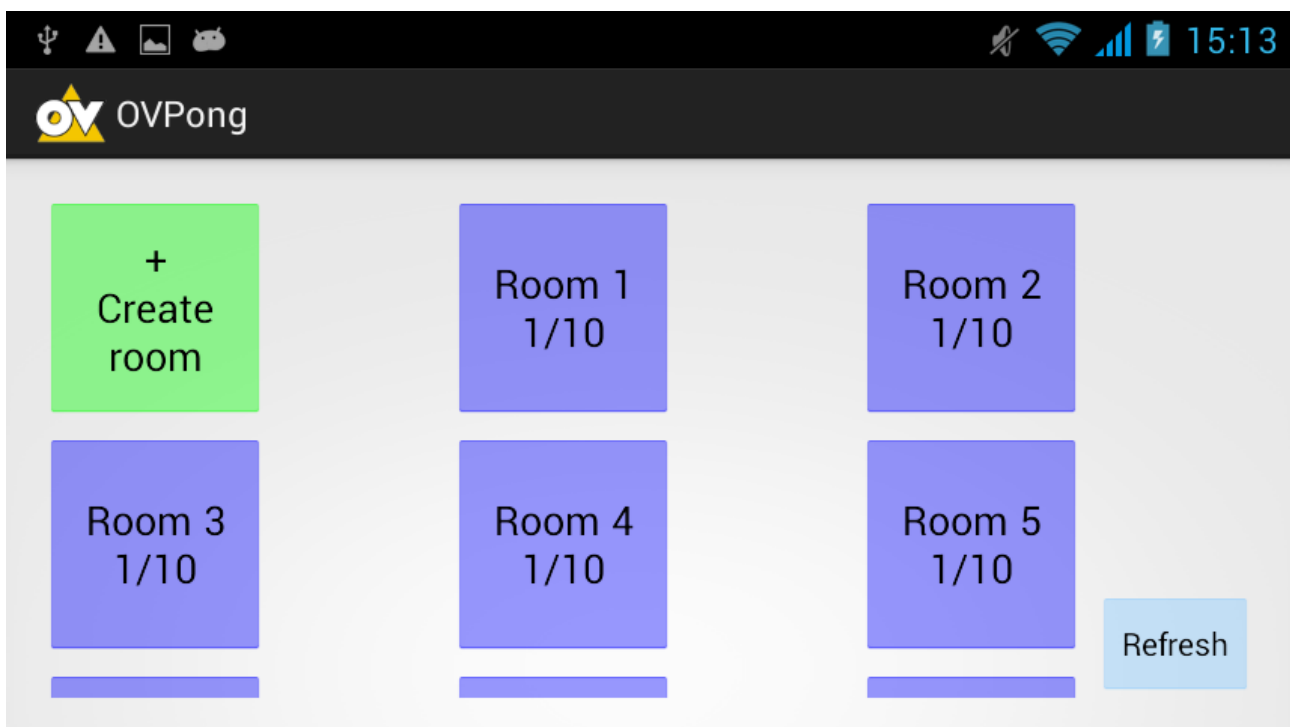
*Figuur 1 - Het loginscherm dat na de splashscreen komt*

Nadat de user succesvol ingelogd is, zal hij terecht komen in het menuscherm. Van hieruit kan snel een spel gestart worden, de lobby (F4) ingegaan worden, kunnen de spelinstructies (F2.2) gelezen worden, en kan het geluid aan- of uitgezet worden. Dit scherm is nodig om de navigatiestructuur in het programma overzichtelijk te houden.



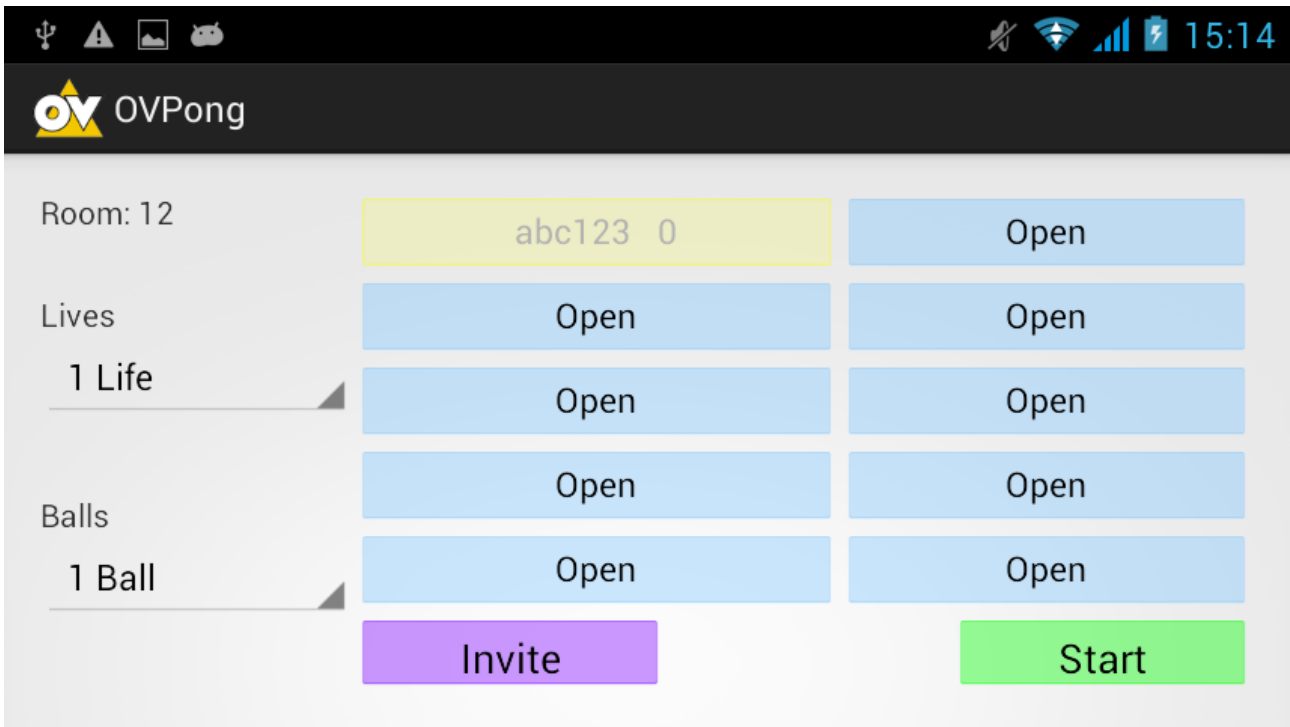
*Figuur 2 - Het menuscherm*

Als op de knop 'Custom play' gedrukt wordt, zal de user in het lobby scherm komen. Om efficiënt gebruik te maken van de ruimte op het scherm, is ervoor gekozen om alleen de rooms te laten zien. Om een nieuwe room te maken, moet op de eerste knop met het plusje worden gedrukt. Deze knop is te zien in figuur 3. De user kan de rooms binnen gaan door erop te drukken. Alle rooms worden bij het aanmaken genummerd om de room te identificeren. Er is nu de mogelijkheid om zelf te kiezen met wie je een spel wilt spelen (F4). De rooms zijn te zien in een grid en er bestaat geen maximum op het aantal rooms dat gelijktijdig kan bestaan. Als er meer rooms komen dan op het scherm passen, zal het mogelijk worden om naar beneden te scrollen. In figuur 3 is er een room aangemaakt voor 10 spelers en is 1 speler aanwezig. Dat wordt weergegeven door de tekst "(1/10)".



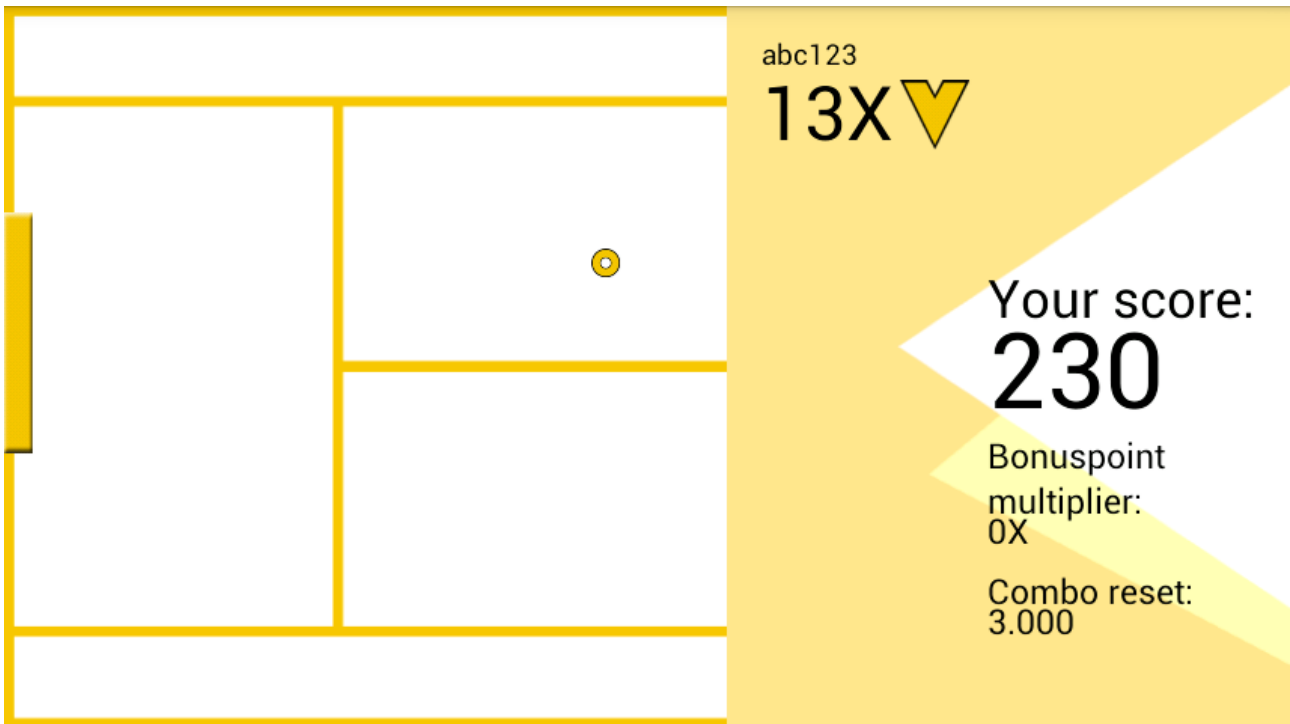
Figuur 3 - Lobby met kamers

Als er op de knop met het plusje of op een al bestaande niet-volle room wordt gedrukt, dan komt de user in het scherm van de room terecht. De spelers in de room zijn zichtbaar in een grid en het nummer van de room is linksboven in het scherm te zien. De speler die de room aangemaakt heeft is de host. Wanneer de host slots opent/sluit, verandert de grootte van de room en daarmee het aantal spelers die in deze room kunnen. De maximum grootte van de room blijft echter 10. Om het spel te starten moet de host op de "Start"-knop drukken. Dit zal een spel starten met de spelers die in de room aanwezig zijn.



Figuur 4 - Een kamer met 1 speler

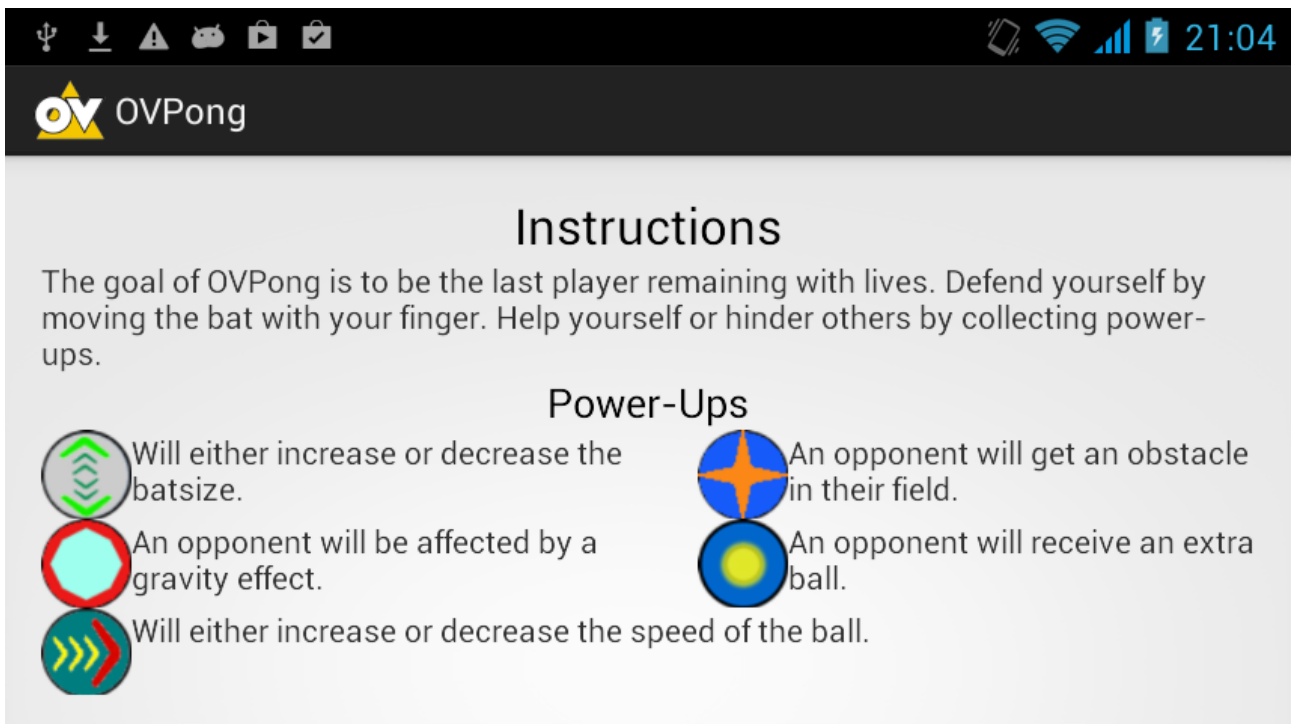
Als het spel is gestart, zullen de namen van de spelers, hun levens en hun score zichtbaar zijn op het scherm, maar de speelvelden van de andere spelers zullen niet zichtbaar zijn (F2.1). Tijdens het spel zullen af en toe verschillende power-ups en power-downs (F1) verschijnen. Na het spel wordt de speler terug gestuurd naar de lobby (zie figuur 3) en is er nogmaals de mogelijkheid om een nieuw spel te starten (F5).



Figuur 5 - Screenshot van het gamescherm van OVPong.



Tenslotte is er nog een apart scherm aangemaakt (zie figuur 6) waarin de verschillende power-ups/downs en hun effecten uitgelegd worden zodat deze duidelijk zijn voor de user (F2.2). Dit scherm is toegankelijk vanuit het menuscherm (figuur 2).



Figuur 6 - Instructiescherm

#### 5.1.4 Power-ups/downs

Power-ups/downs (F1) verschijnen gemiddeld elke 15 seconden met een variatie van 10 seconden in de vorm van een icoon. Bij elk verschillend icoon hoort een unieke power-up/down. Het icoon zal verschijnen op een willekeurige plaats bij het vlak tegenover het vlak waarin zich de bat bevindt en zal met een versnelling en in een loodrechte lijn richting het vlak van de bat bewegen. Wanneer het icoon opgevangen wordt door de bat, zal de power-up/down direct in werking gesteld worden en verdwijnt het icoon. Indien het icoon niet opgevangen wordt door de bat zal het icoon ook verdwijnen zonder dat de power-up/down in werking wordt gesteld. Power-ups hebben een effect op je eigen speelveld, terwijl power-downs een negatief effect hebben op het speelveld van één van je tegenstanders. Verder kunnen power-ups/downs niet botsen met ballen. Het spel kent de volgende power-ups/downs:

- Vergroot/verkleint de grootte van je eigen bat.
- Versnelt/vertraagt de bal.
- Alle ballen op het scherm van één van je tegenstanders krijgen voor een bepaalde tijd een constante versnelling richting de boven of onderkant van het scherm. Zwaartekracht wordt hiermee nagebootst.
- Serveert een nieuwe bal op het veld van één van je tegenstanders, waardoor er meer ballen in het spel komen.
- Zet een obstakel neer op het speelveld van één van je tegenstanders. Een obstakel heeft een bepaalde werking op de bal, zoals de bal een willekeurige richting in sturen, of de bal een versnelling geven. Nadat een bal het obstakel aangeraakt heeft, zal het obstakel verdwijnen. Het obstakel wordt weergegeven als een icoon dat gerelateerd is aan OVSoftware (F1.1).

Daarnaast bestaan er voor de power-ups/downs de volgende restricties:

- Een bat heeft 3 verschillende groottes, namelijk de verkleinde versie, de standaard grootte en de vergrote versie. Het opvangen van een power-up die de grootte van de bat verandert, zal de bat met één stap vergroten of verkleinen.
- Een bal kan maar naar één richting een versnelling krijgen (óf de bovenkant óf de onderkant, maar nooit naar beide kanten). Wanneer er een power-down opgepakt wordt die de versnelling verandert naar bijvoorbeeld de bovenkant wanneer de bal al reeds een versnelling heeft naar de onderkant, dan wordt de versnelling naar de bovenkant in werking gesteld en is de versnelling naar de onderkant niet meer in effect.

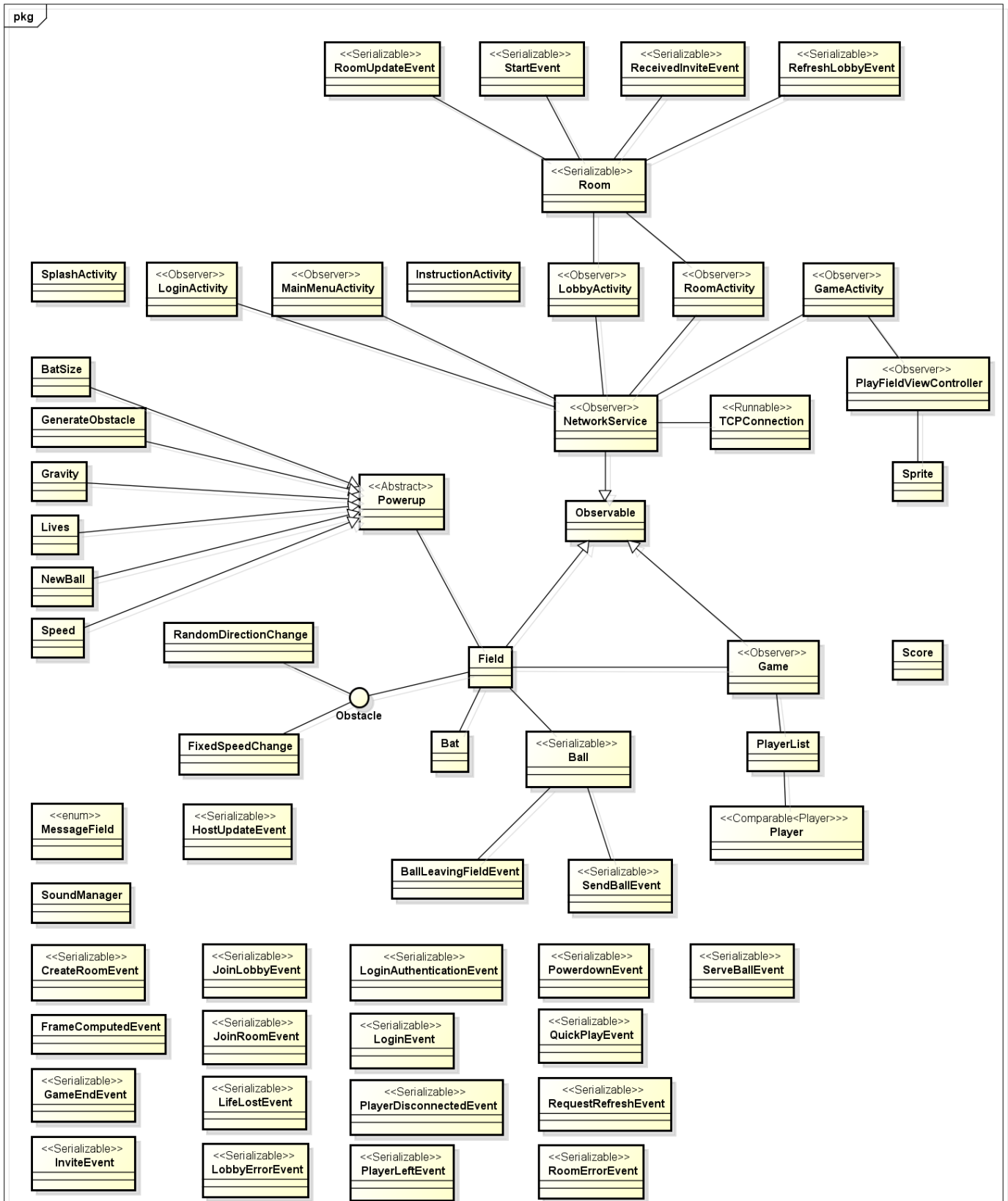
### 5.1.5 Highscores

Highscores worden per speler bijgehouden (F3) en zijn, wanneer een speler in een Room zit, voor iedereen in deze room zichtbaar. Er zijn een aantal variabelen waarop een highscore gebaseerd kon worden, namelijk de totale tijd uitgehouden in het spel, de uiteindelijke positie ten opzichte van andere spelers, de hoeveelheid ballen die de speler in het spel terug geslagen heeft, en de hoeveelheid power-ups die door deze speler opgepakt zijn. Daarnaast moet de score schalen op de hoeveelheid spelers, omdat bijvoorbeeld een 2<sup>e</sup> plaats met 10 spelers natuurlijk een betere score krijgt dan een 2<sup>e</sup> plaats met maar 2 spelers. Ook moet de score niet af gaan hangen van de hoeveelheid spelers. Dit wil zeggen dat het behalen van een 1<sup>e</sup> plaats met 2 spelers gelijk moet zijn aan het behalen van een 1<sup>e</sup> plaats met bijvoorbeeld 10 spelers. Hier is voor gekozen, omdat het anders mogelijk zou zijn om een hogere score te halen naarmate er meer spelers zijn, wat het spelen met weinig spelers wellicht zou kunnen ontmoedigen (F3.1). Ook moet de score afhankelijk zijn van de hoeveelheid lives waarmee gespeeld wordt. Het is bijvoorbeeld niet de bedoeling dat het makkelijker is om een hogere score te behalen naarmate de hoeveelheid lives meer wordt. Daarom wordt de uiteindelijke score gedeeld door het aantal lives. Uiteindelijk is op de volgende formule uitgekomen om de score van een speler te berekenen:

$$score = \frac{10 * (players + 1 - pos)}{players} * \frac{inGameScore}{lives}$$

Hierbij is *score* de uiteindelijke score van de betreffende speler, *players* de totale hoeveelheid spelers waarmee een spel gespeeld werd, *pos* de uiteindelijke plaats die behaald is aan het einde van het spel, *lives* de hoeveelheid levens waarmee het spel gespeeld werd, en *inGameScore* de score die behaald is binnen het spel zelf. Deze *inGameScore* is gebaseerd op de hoeveelheid ballen die terug geslagen zijn en de hoeveelheid power-ups die opgepakt zijn gedurende het spel. Hierbij is het terug slaan van de bal 100 punten waard en het oppakken van een power-up/down 150 punten. Daarnaast zijn er bonuspunten te verdienen. Wanneer een bal geraakt wordt, zal er een tellertje verhoogd worden met 1. Dit tellertje wordt elke 3 seconden gereset naar 0, of wanneer een bal gemist wordt. Echter, wanneer binnen deze 3 seconden opnieuw een bal geraakt wordt, dan wordt het tellertje verhoogd en begint het 3 seconden aftellen weer opnieuw. Indien dit gebeurt, krijgt de speler 50 punten vermenigvuldigd met de hoogte van het tellertje. Op deze manier wordt de speler beloond door met meer ballen te spelen. Wanneer na een spel de totale score hoger blijkt te zijn dan de huidige highscore van de betreffende speler, dan wordt de highscore vervangen door deze nieuwe score.

## 5.1.6 High-level klassendiagram



Figuur 7 - High-level klassendiagram van OVPong

De structuur van het systeem is grotendeels identiek aan de structuur van de originele versie van OVPong. De Activity's, de verschillende schermen waar een Android-app zich in kan bevinden, zijn nog steeds verbonden met de NetworkService, omdat deze de communicatie regelen over het netwerk. Er zijn wel

enkele nieuwe aanpassingen aan de Activity-kant, zoals pop-ups en een variabel aantal knoppen in bijvoorbeeld LobbyActivity. Deze worden respectievelijk weergegeven als DialogFragments en ButtonAdapters. De ButtonAdapter is daarom verbonden met Room; het hangt af van hoeveel Rooms er bestaan.

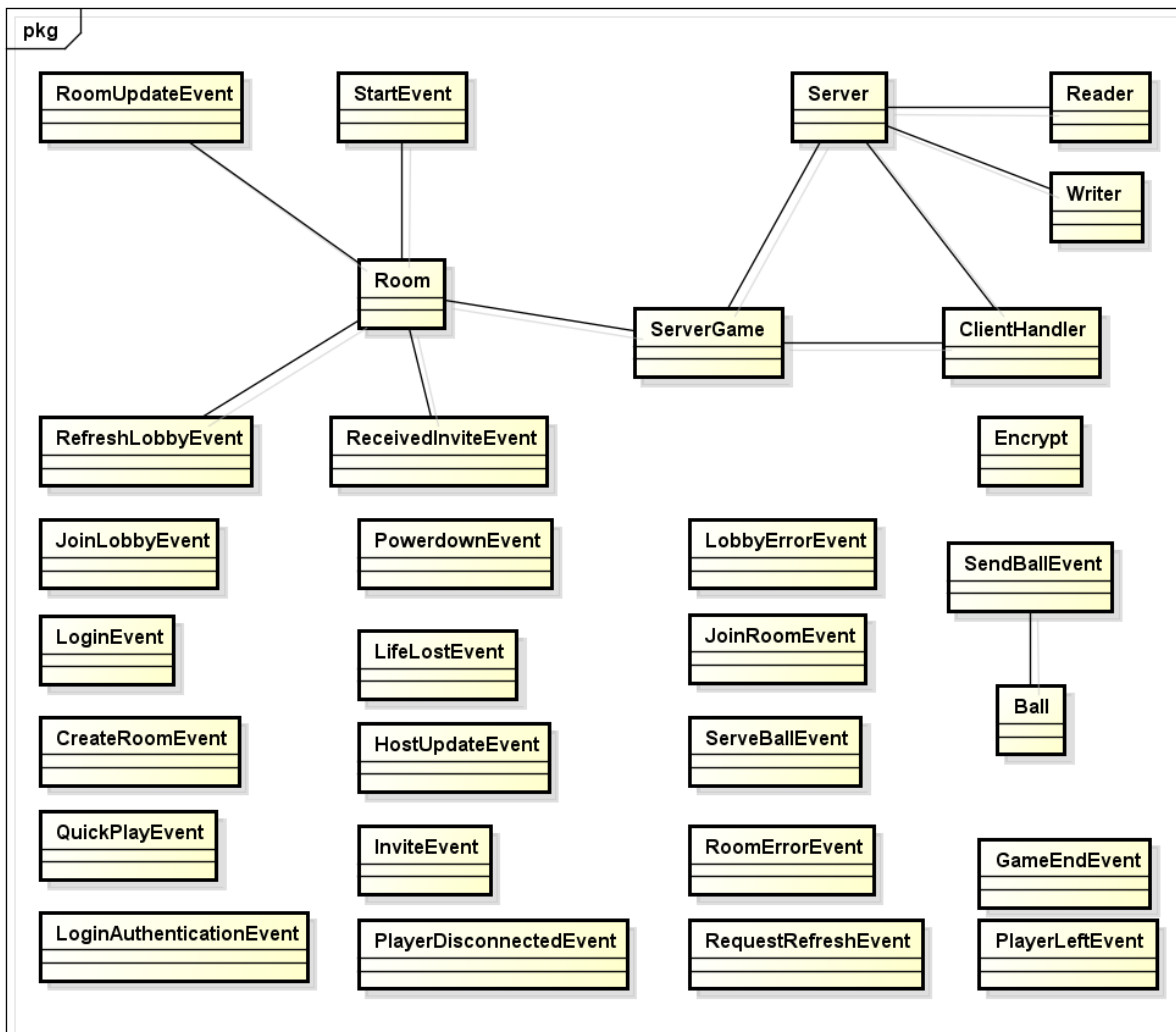
In de oude versie van OVPong gebruikte het netwerk nog een UDP-verbinding, maar dat is nu niet meer het geval, aangezien er nu gebruikt wordt gemaakt van een verbinding met een server. Er staat daarom ook maar één TCPConnection verbonden met NetworkService.

De events zijn Serializable, zodat ze over het netwerk te versturen zijn. Zo wordt communicatie geregeld met de server. Er staan 24 events in het diagram, waarvan er 6 verbonden zijn met een andere klasse in het systeem. Dat betekent dat deze 6 events de klassen doorsturen die er mee verbonden zijn.

### 5.1.6.1 Observer-patroon

Voor interne 'communicatie' in de client is ervoor gekozen om een Observer-patroon te gebruiken. Zo wordt dit patroon gebruikt tussen Activity's en NetworkService, Field en Game, en Game en NetworkService.

### 5.1.7 High-level klassendiagram server



Figuur 8 - High-level klassendiagram van Server

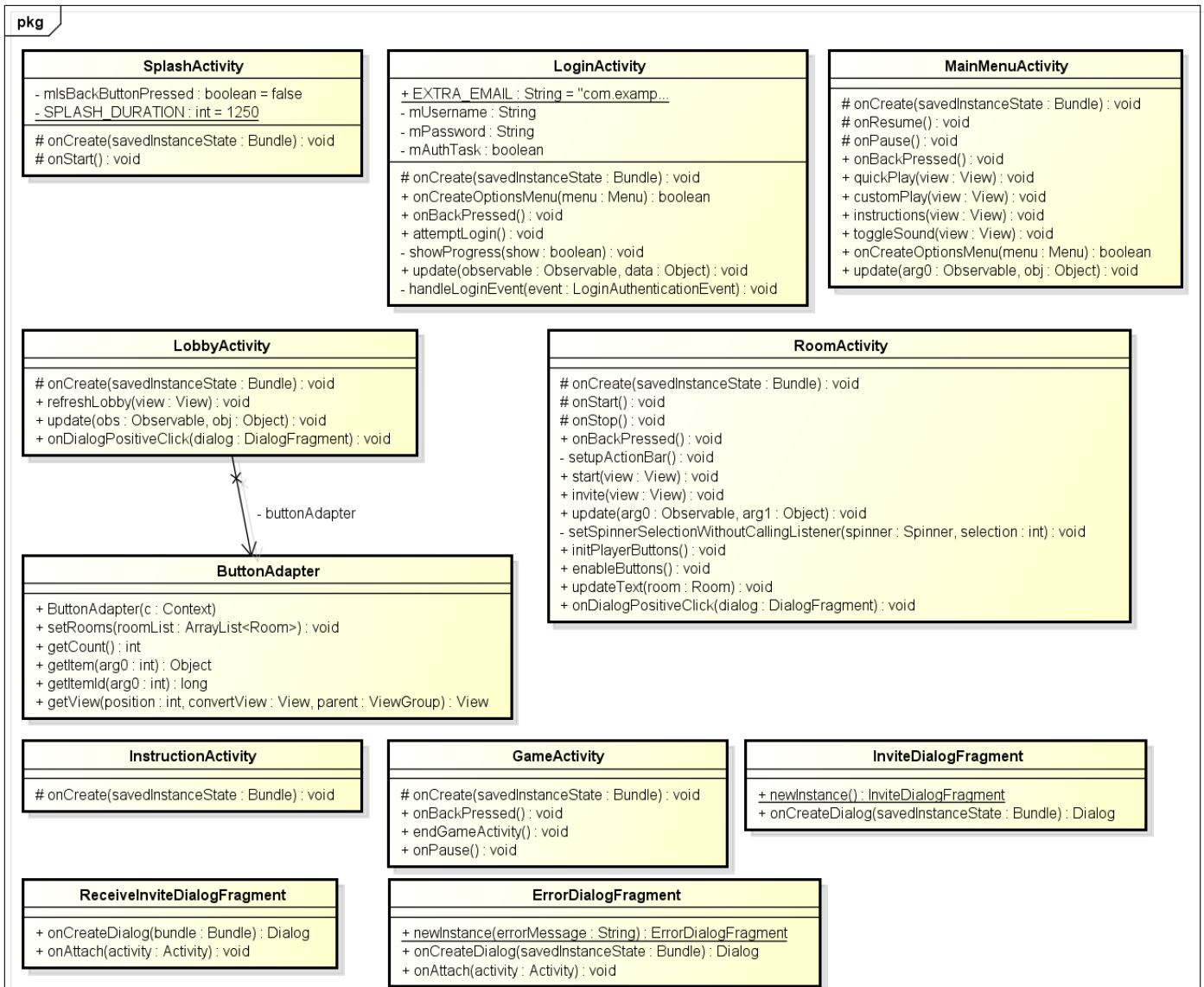
Het systeem heeft een centrale server die draait op Java. De server handelt alle communicatie af en houdt gegevens bij van gebruikers en spellen. De communicatie verloopt door middel van het sturen en ontvangen van event-klassen die elk wat anders betekenen.

Het servergedeelte heeft een klasse Server. Deze klasse is een thread die connecties accepteert, ClientHandlers bijhoudt, Games bijhoudt en alle gegevens opslaat en inleest. Reader leest opgeslagen bestanden en Writer schrijft naar opgeslagen bestanden. Encrypt versleutelt de wachtwoorden om die daarna op te slaan of om ze te vergelijken met de opgeslagen versleutelde wachtwoorden. ClientHandler handelt de connectie en communicatie tussen de Server en de Client af. Game handelt een kamer en het daarop volgende spel af. Room is gedeeld met het clientgedeelte, omdat deze meegestuurd wordt in events. Verder zijn er nog veel events die over het netwerk verstuurd worden en daarom Serializable zijn. Ook zijn er 5 events die verbonden zijn met Room, omdat ze een Room doorsturen over het netwerk.

## 5.2 Detailed Design

In het detailed design wordt in detail uitgelegd wat de individuele klassen doen in het systeem.

### 5.2.1 Detailed Activity-level



Figuur 9 - Detailed class diagram van de Activity's en de bijbehorende hulpklassen

De package *OVPong* bevat alle Activity's in het systeem en de bijbehorende hulpklassen. Activity's zorgen in Android voor de weergave van het programma. Elke Activity is hierbij een scherm.

#### 5.2.1.1 SplashActivity

Dit is de eerste Activity die start zodra de app begint. De enige functie hiervan is om een splashscreen te laten zien voor een bepaalde tijd, aangegeven door *SPLASH\_DURATION*. Direct na het verdwijnen van het splashscreen, zal *LoginActivity* worden gestart. Het scherm kan niet gedraaid worden in *SplashActivity*.

### **5.2.1.2 LoginActivity**

Deze Activity zorgt voor het inlogscherf van het systeem. Het is daarom ook verbonden met NetworkService door middel van een Listener-structuur. Het inlogscherf vraagt om een gebruikersnaam en wachtwoord, waarvoor voorwaarden aan verbonden zijn. Het wachtwoord mag niet minder dan 4 karakters bevatten en de gebruikersnaam mag maximaal 10 karakters lang zijn. Er zijn 2 namen die niet gekozen mogen worden, namelijk "Open" en "Closed". Deze namen worden in het loginscherf al afgevangen, zodat het geen problemen geeft in RoomActivity. Deze restricties worden allemaal intern in LoginActivity afgehandeld. Als de invoer geen fouten bevat en er op de "Sign in or register"-knop wordt gedrukt, zal er informatie verstuurd worden via NetworkService en zal er een laadscherf verschijnen. Het event dat verstuurd wordt over het netwerk is LoginEvent, met de ingevoerde gegevens. Als de server het goedkeurt met de juiste informatie in LoginAuthenticationEvent, zal LoginActivity naar MainMenuActivity gaan. Het scherf kan niet gedraaid worden in LoginActivity.

### **5.2.1.3 MainMenuActivity**

Dit is de Activity die het menuscherf laat zien. Er zijn netwerkfuncties nodig, dus is deze klasse verbonden met NetworkService met een Listener-structuur. Deze Activity zorgt ervoor dat het geluid aan- en uitgezet kan worden en dat de juiste acties worden uitgevoerd als er op een knop wordt gedrukt. De "Quick play"-knop zorgt ervoor dat een QuickPlayEvent verstuurd wordt, waarna de knop 'uitgezet' wordt. Hierna wordt gewacht tot er een bericht van de server komt, dat er een speler aanwezig is. Het spel zal daarna direct gestart worden met de gevonden speler. De "Custom play" knop zorgt ervoor dat er een JoinLobbyEvent verstuurd wordt en dat LobbyActivity wordt gestart. "Instructions" stuurt je naar InstructionActivity, dat het instructiescherf laat zien. Hierbij worden er geen Events over het netwerk gestuurd. Het scherf kan niet gedraaid worden in MainMenuActivity.

### **5.2.1.4 InstructionActivity**

Deze Activity laat het instructiescherf zien en is niet verbonden met NetworkService. Het scherf kan niet gedraaid worden in InstructionActivity.

### **5.2.1.5 LobbyActivity**

LobbyActivity laat het overzichtsscherf van kamers zien, waarvan de lijst met kamers in principe oneindig is. Voor deze functie wordt gebruik gemaakt van een hulpklasse, ButtonAdapter. In LobbyActivity is het mogelijk om een kamer te maken, een kamer in te gaan en om uitgenodigd te worden door een andere speler. Als er een kamer gemaakt wordt, zal LobbyActivity een CreateRoomEvent sturen naar NetworkService. Als een speler een al bestaande kamer wil ingaan, wordt er een JoinRoomEvent verstuurd. De laatste optie is het ontvangen van een ReceivedInviteEvent, wat zorgt voor het laten zien van een pop-up scherf, die ondersteund wordt door het uitbreiden van FragmentActivity. Dit is in andere Activity's gedaan door het uitbreiden van Activity. Er is ook een "Refresh"-knop aanwezig, dat voor een up-to-date lijst van kamers laat zien. Dat is mogelijk, omdat er een RequestRefreshEvent verstuurd wordt naar NetworkService. Uiteindelijk zal LobbyActivity een RefreshLobbyEvent ontvangen, met de bijbehorende set van kamers. Het scherf kan niet gedraaid worden in LobbyActivity.

### **5.2.1.6 ReceiveInviteDialogFragment**

Deze klasse breidt een DialogFragment uit. Dat maakt het mogelijk om een pop-up-dialoog te laten zien met 2 opties. In dit geval zijn dat "Accept" en "Decline" als antwoord op de vraag of de gebruiker op de uitnodiging in gaat.

### **5.2.1.7 RoomActivity**

RoomActivity laat een scherm zien met een aantal spelers en opties om het aantal ballen en het aantal levens aan te passen. Er passen maximaal 10 spelers in de kamer, waarvan de speler linksboven, die te herkennen is aan de gele knop, de beheerder van de kamer is. De beheerder kan spelers verwijderen en opties aanpassen.

In deze Activity is het mogelijk om een speler uit te nodigen. Hiervoor wordt de hulpklasse InviteDialogFragment gebruikt, die ervoor zorgt dat er een pop-up verschijnt. Hiervoor moet RoomActivity de klasse FragmentActivity uitbreiden. Spelers kunnen uit de kamer verwijderd worden door de knop met de spelernaam in te drukken. Het scherm kan niet gedraaid worden in RoomActivity.

### **5.2.1.8 InviteDialogFragment**

Deze klasse breidt een DialogFragment uit, wat ervoor zorgt dat het gebruikt kan worden als een pop-up dialoog. Er is een tekstveld aanwezig met 2 knoppen "Cancel" en "Send invite", die ervoor zorgen dat de invite werkt, of gestopt wordt.

### **5.2.1.9 ButtonAdapter**

ButtonAdapter wordt gebruikt in de klasse LobbyActivity voor het mogelijk maken van meerdere knoppen. Ook zal het gebruik van ButtonAdapter ervoor zorgen dat de lijst met knoppen in principe oneindig groot kan worden. Er zal dan ook een scrollbar verschijnen.

### **5.2.1.10 GameActivity**

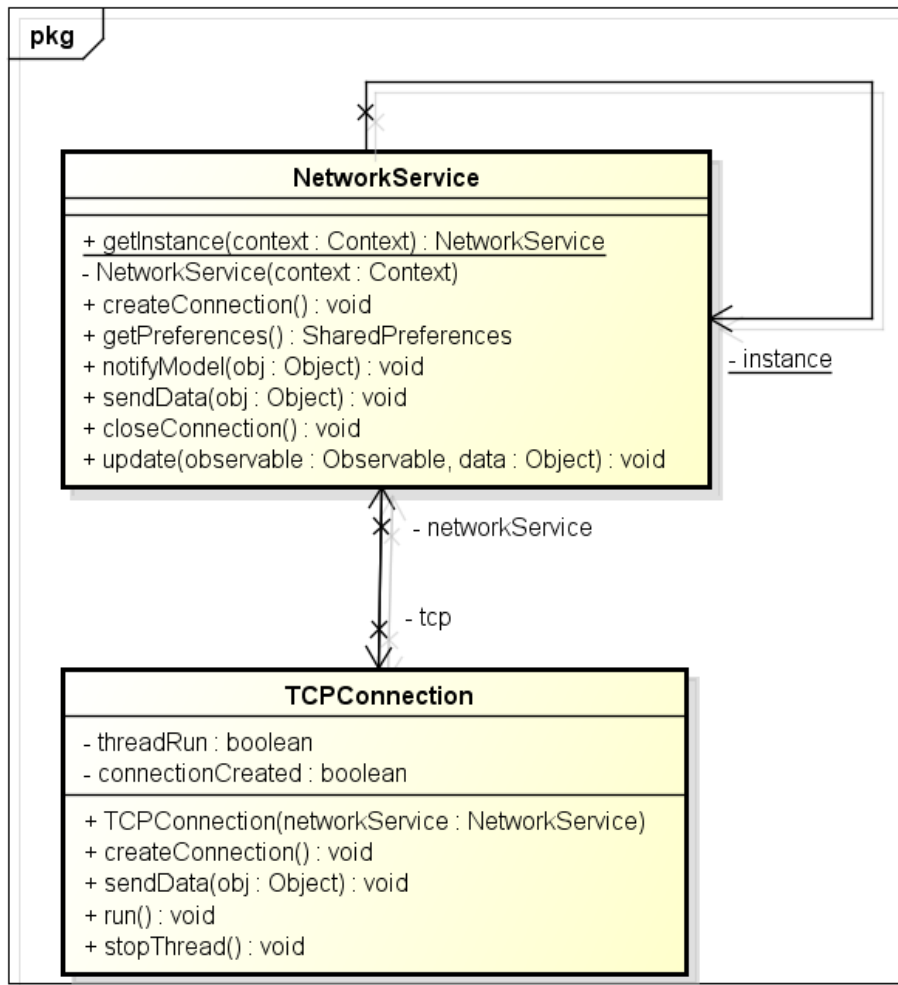
GameActivity laat het spelscherm zien met aan de rechterkant een spelerlijst en aan de linkerkant een speelveld. Als er op de back-knop wordt gedrukt, zal GameActivity dat afhandelen door een GameEndEvent te versturen naar de server. Het scherm kan niet gedraaid worden in GameActivity.

### **5.2.1.11 ErrorDialogFragment**

ErrorDialogFragment is een DialogFragment, dus het is een pop-up scherm. In het scherm staat alleen een tekst en een "Ok"-knop.



## 5.2.2 Detailed Network



Figuur 10 - Detailed class diagram van het netwerk

De package *network* bevat alle klassen die te maken hebben met de verbinding tussen de client en de server.

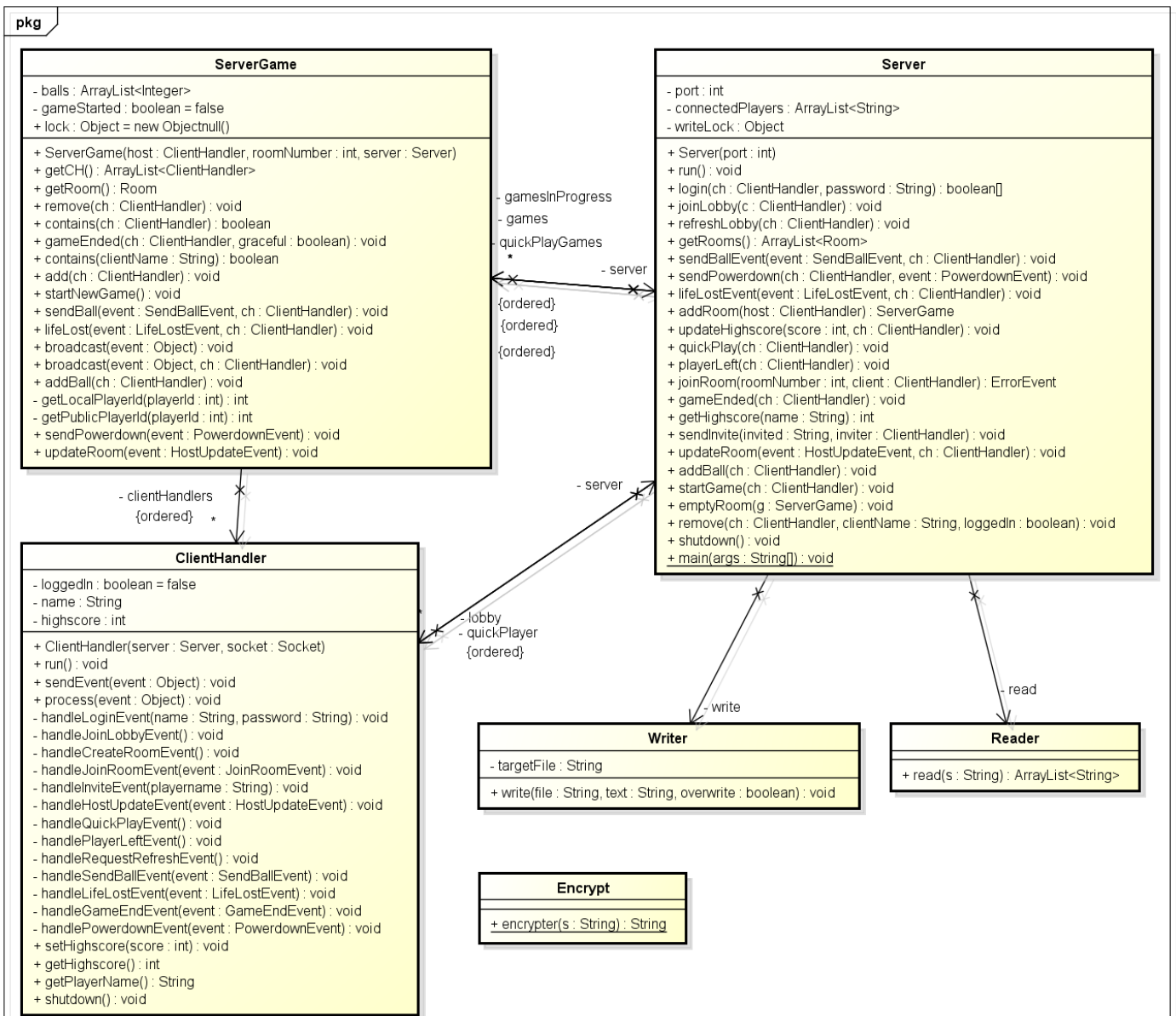
### 5.2.2.1 NetworkService

**NetworkService** wordt in het systeem gezien als een soort brug van communicatie tussen de modelkant en de netwerkkant van de applicatie. Zo breidt **NetworkService** **Observable** uit, zodat het model informatie doorgestuurd kan krijgen zodra het beschikbaar is van het netwerk. Daarnaast implementeert deze klasse de interface **Observer**. Dit maakt het mogelijk om informatie van het model te ontvangen en vervolgens te versturen over het netwerk, via de klasse **TCPConnection**. In het huidige systeem is het nodig om maar één instantie van het netwerk te hebben, aangezien er maar één verbinding wordt gemaakt met de server. Dit wordt gerealiseerd door van de **NetworkService** een singleton-klasse te maken. De reden dat voor deze 'brug' gekozen is, komt omdat Android het niet toestaat om een netwerkklasse te laten draaien op de 'main' Thread. Om toch het netwerk mogelijk te maken is een aparte klasse **TCPConnection** gemaakt. Deze wordt in **NetworkService** aangeroepen en direct in een Thread gestopt, om zo parallel te draaien.

### 5.2.2.2 TCPConnection

Dit is de klasse die daadwerkelijk de TCP-verbinding maakt met de server. Het is nodig dat dit parallel gaat draaien aan de rest van het programma, vanwege de hierboven beschreven reden, dus implementeert het Runnable. TCPConnection houdt netwerkgerelateerde variabelen bij, zoals op welke Socket er wordt geluisterd en de bijbehorende ObjectInput- en ObjectOutputStreams. Ook is het nodig om een instantie van NetworkService bij te houden, omdat dat het mogelijk maakt om informatie te versturen en te ontvangen van NetworkService. Vanaf het moment dat TCPConnection wordt geïnstantieerd, wordt het IP-adres en het poortnummer van de server direct uit de code gehaald. Het IP-adres en het poortnummer zijn hardcoded, omdat user-input ongewenst is en een config file lastig is in Android. Deze informatie wordt gebruikt om een Socket te maken en het netwerk te initiëren. Er bestaan ook methoden om de data te versturen en de huidige verbinding (en dus de Thread) te stoppen.

### 5.2.3 Detailed Server



Figuur 11- Detailed class diagram van de server

De klassen in het diagram zijn verspreid over meerdere packages, maar zijn samengevoegd voor overzichtelijkheid. De package *server* bevat klassen die de communicatie met de client afhandelt en de package *account* bevat de klassen die gegevens kunnen opslaan en schrijven.

#### **5.2.3.1 Server**

Dit is de centrale klasse. Deze klasse moet bijhouden welke clients in de lobby zitten en welke in een spel. De Server is ook verantwoordelijk voor het verplaatsen van clients naar andere locaties in het systeem, bijvoorbeeld de lobby. De Server is een Thread die binnenkomende connecties accepteert en dan voor elke clientconnectie een ClientHandler aanmaakt die deze connectie afhandelt. Ook houdt de Server alle ServerGame's bij. Daarnaast werkt de Server de opgeslagen bestanden bij en leest deze gegevens in met behulp van de Reader en Writer.

#### **5.2.3.2 Reader**

Reader leest de gegevens uit een tekstfile en geeft deze terug.

#### **5.2.3.3 Writer**

Writer schrijft gegevens naar een tekstfile. Het kan de oude gegevens overschrijven of de gegevens achter de oude gegevens schrijven.

#### **5.2.3.4 Encrypt**

Encrypt versleutelt een String en geeft deze terug. Dit wordt gebruikt bij het opslaan en vergelijken van wachtwoorden.

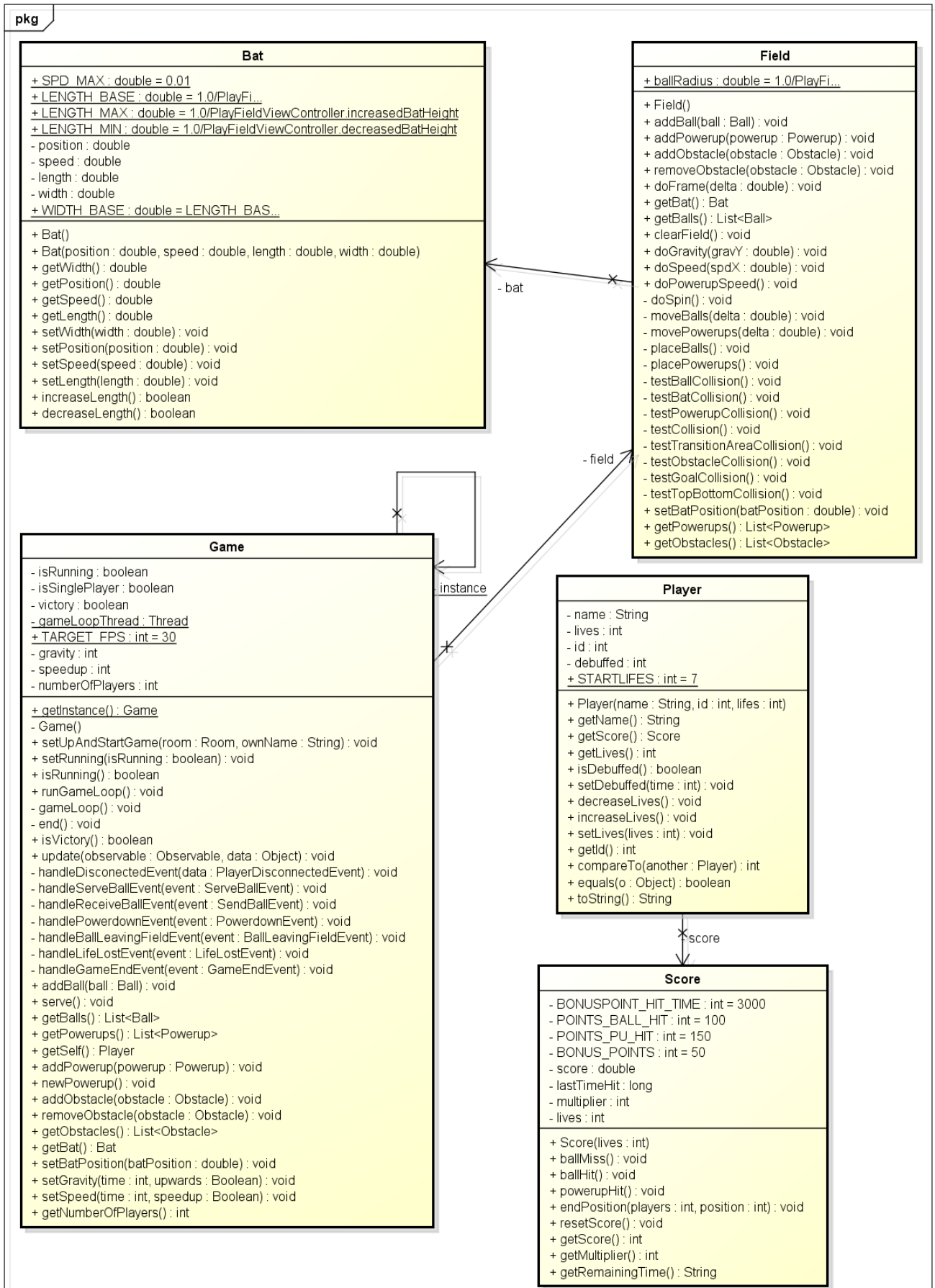
#### **5.2.3.5 ClientHandler**

Een ClientHandler wordt aangemaakt door de Server zodra een client connect. Deze klasse handelt de communicatie met de client af. Het is namelijk een Thread die constant Objecten leest van de connectie met de client. Het ziet welke events zijn ontvangen en stuurt relevante informatie door naar de Server. Server verwerkt deze verder. Verder meldt de ClientHandler aan de Server wanneer de client de connectie verbreekt.

#### **5.2.3.6 ServerGame**

ServerGame wordt aangemaakt zodra een client een kamer creëert in de lobby. Deze ServerGame houdt een Room bij. De host kan de instellingen van deze Room aanpassen, slots openen, slots sluiten, spelers kicken en spelers uitnodigen. Zodra de host het spel start door middel van een HostUpdateEvent, stuurt ServerGame ServeBallEvent's naar clients gelijk aan het aantal ballen die door de host ingesteld waren. Tijdens het spel stuurt ServerGame SendballEvent's door tussen clients. Verder stuurt ServerGame naar elke andere client in het spel een PlayerDisconnectedEvent als een client de verbinding verbreekt, daarna worden alle ballen die deze client had verdeeld over de overgebleven clients. Als een client een leven verliest dan wordt dit naar alle clients doorgestuurd. ServerGame wordt ook aangemaakt als er een quickplay spel wordt opgestart. In dit geval wordt het spel onmiddellijk gestart, en kunnen er dus geen aanpassingen worden gemaakt aan Room.

## 5.2.4 Detailed Model



Figuur 12 - Detailed class diagram van het model

De package *model* wordt gebruikt voor klassen die de gegevens van het model van het systeem bijhouden. Dit zijn onder andere klassen zoals Field, Bat en Game.

#### **5.2.4.1 Game**

Deze klasse bevat alle methoden en variabelen die betrekking hebben op het spel zelf. De klasse bevat onder andere een PlayerList die de tegenstanders en de eigen Player bijhoudt, een Field, een variabele die bijhoudt of het spel een singleplayer-spel is. Dit is relevant, omdat de ballen en de obstakels dan naar jezelf verstuurd moeten worden in plaats van naar andere spelers. Ook worden de hoeveelheid frames per seconde bijgehouden. Dit geeft aan hoe vaak het Canvas opnieuw getekend wordt per seconde en hoe vaak er gekeken wordt of er botsingen plaatsvinden tussen ballen en muren, obstakels, bat of andere ballen.

#### **5.2.4.2 Field**

Deze klasse bevat alle methoden en variabelen die betrekking hebben op het speelveld binnen het spel. Binnen deze klasse wordt een lijst van Balls, een lijst van Power-ups, een lijst van Obstacles en de Bat bijgehouden. Daarnaast wordt voor elk frame getest of er een botsing plaatsvindt tussen ballen, tussen een bal en de muur, tussen een bal of een power-up/down en de bat, tussen een bal en een obstakel, wanneer een bal het veld verlaat of wanneer een bal of power-up/down de doellijn overschrijdt.

#### **5.2.4.3 Bat**

Deze klasse bevat alle methoden en variabelen die betrekking hebben op de bat. Bijgehouden wordt de breedte van de bat, de standaard lengte van de bat (de lengte waarmee het spel begonnen wordt), de maximale en minimale lengtes van de bat (de lengtes door power-ups/downs) en tenslotte de maximale snelheid waarmee de bat bewogen kan worden.

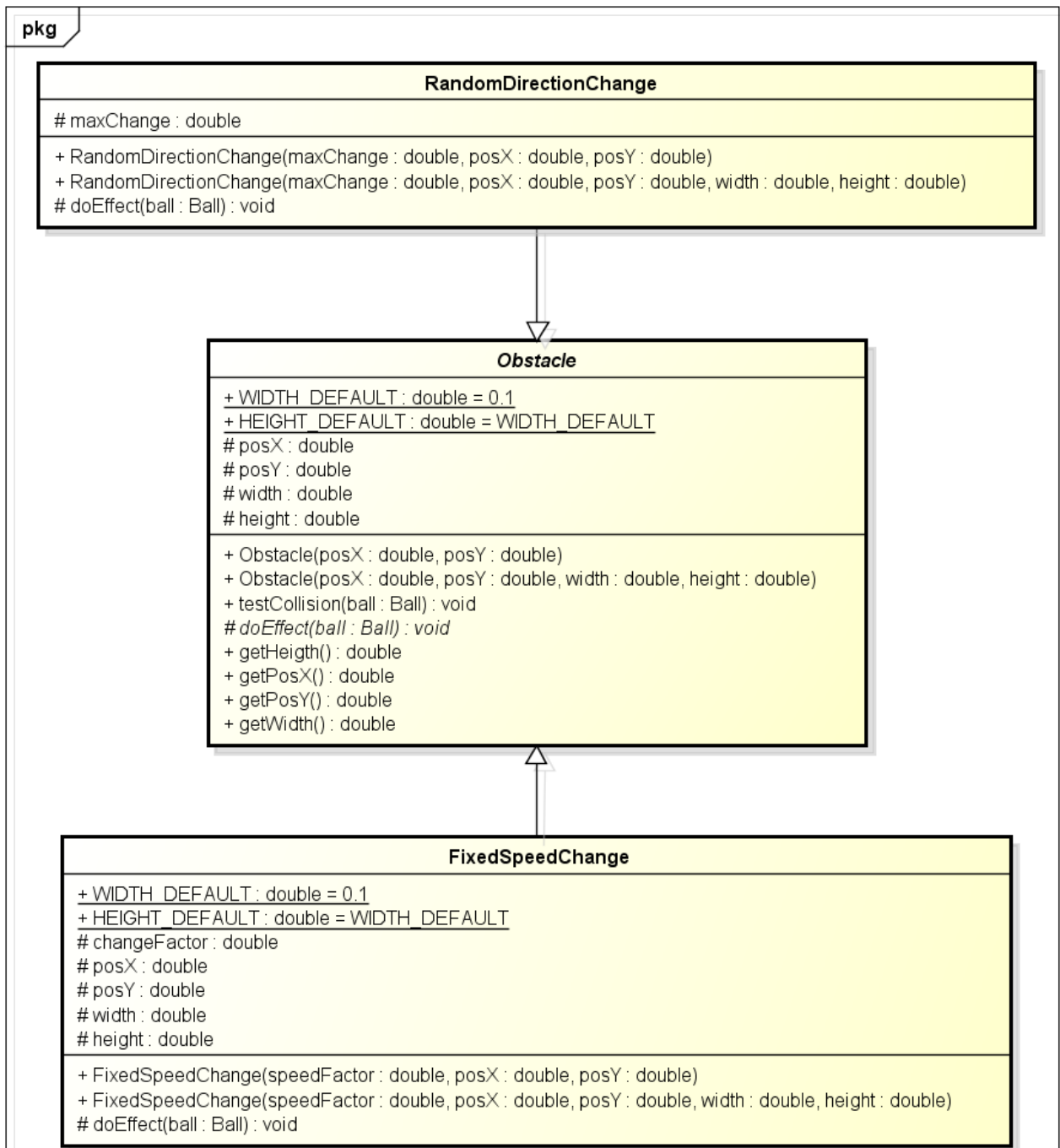
#### **5.2.4.4 Player**

Deze klasse bevat alle methoden en variabelen die betrekking hebben op de speler. Bijgehouden wordt de naam van de betreffende speler, de hoeveelheid resterende lives, de identieke id die een waarde heeft tussen de 0 en 9 om onderscheid te maken tussen de spelers binnen een spel, de Score, en tenslotte een variabele om bij te houden of de speler een power-down heeft.

#### **5.2.4.5 Score**

Deze klasse houdt de score bij en alle methoden en variabelen die hieraan gerelateerd zijn. Bijgehouden wordt onder andere de hoeveelheid punten het raken van de bal waard is, het oppakken van een power-up waard is, hoeveel bonuspunten er verkregen worden wanneer er meer ballen geraakt worden binnen de bijgehouden tijdsframe en de hoeveelheid ballen die geraakt zijn voordat het tijdsframe verlopen is.

## 5.2.5 Detailed Obstacles



Figuur 13 - Detailed class diagram van obstacles

De package *obstacles* wordt gebruikt voor alle klassen die obstakels bieden voor het spel.

### 5.2.5.1 Obstacle

Obstacle is een abstracte klasse die definieert dat Obstacles hun dimensies en positie bij kunnen houden. Het is een abstracte klasse, omdat bepaalde methoden in elke Obstacle hetzelfde moet zijn. Obstacle heeft methoden om te kijken of een Ball de Obstacle 'aanraakt' en wat voor effect het zal geven op de Ball. Dat wil zeggen, als een Ball binnen het bereik van deze Obstacle komt, wordt het effect geactiveerd en de Obstacle verwijderd van het speelveld.

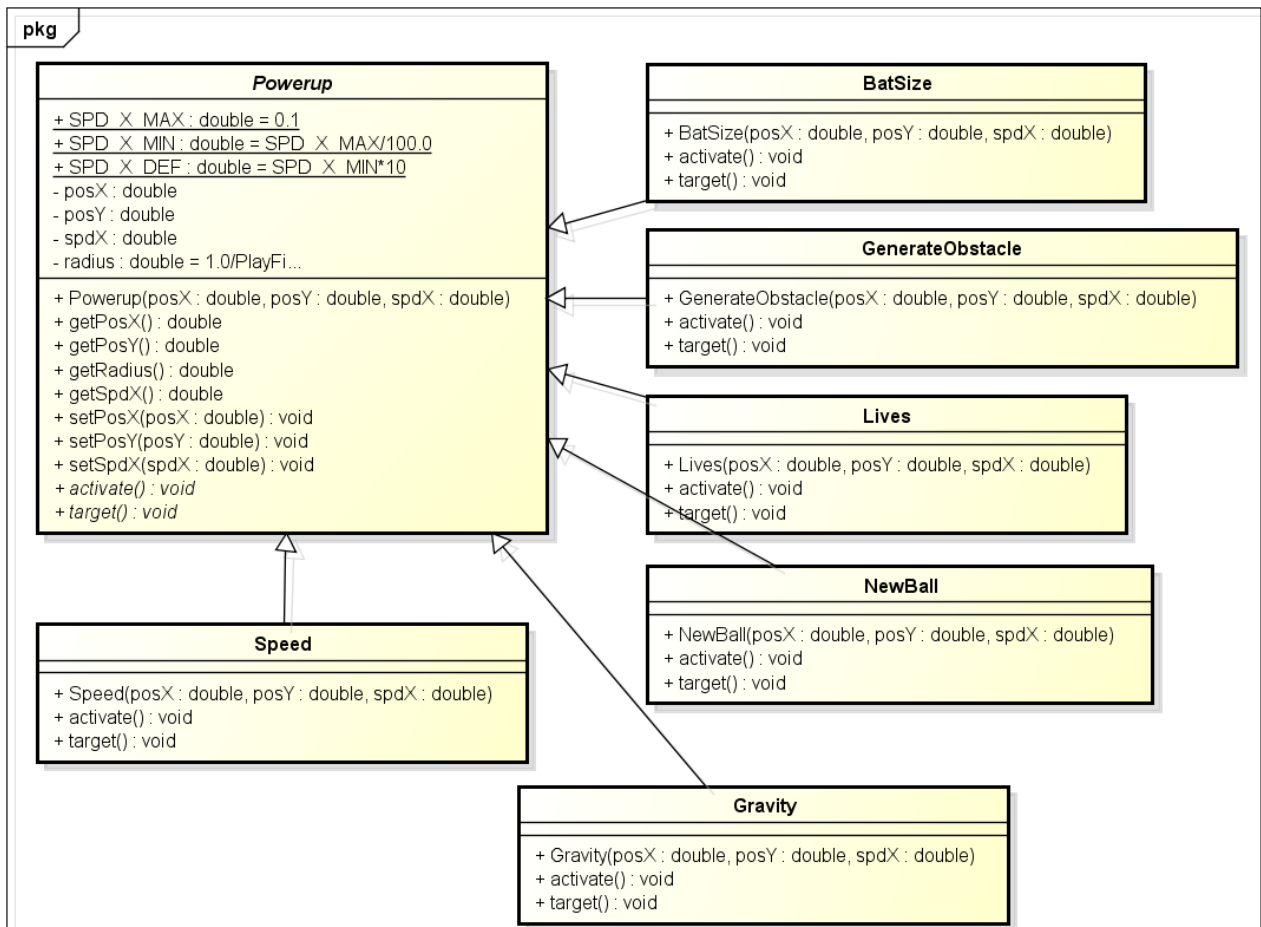
### 5.2.5.2 FixedSpeedChange

FixedSpeedChange breidt de abstracte klasse Obstacle uit met een effect. Dit is gedaan door de methode *doEffect* uit te breiden. Het effect zorgt ervoor dat de Ball die deze Obstacle aanraakt, versneld of vertraagd zal worden.

### 5.2.5.3 RandomDirectionChange

RandomDirectionChange breidt de abstracte klasse Obstacle uit met een effect. De methode *doEffect* zorgt ervoor dat de Ball die deze Obstacle aanraakt van richting veranderd wordt. Dit gebeurt willekeurig, maar er is een maximale hoek die aangepast mag worden.

## 5.2.6 Detailed PowerUp



Figuur 14 - Detailed class diagram van de power-ups

De package *powerUps* wordt gebruikt voor alle power-ups in het spel.

### 5.2.6.1 Powerup

Powerup is een abstracte klasse die definieert dat Powerups hun positie en snelheid in de X-richting bij kunnen houden. Ook heeft het een abstracte methode *activate()*, die door subclasses uitgebreid kunnen worden.

### 5.2.6.2 BatSize

BatSize breidt Powerup uit met een methode *activate()*. Deze zorgt voor het vergroten of verkleinen van de Bat. Dit gebeurt willekeurig met een kans op vergroting en een kans op verkleining.

### 5.2.6.3 Gravity

Gravity breidt Powerup uit met een methode *activate()*. Deze zorgt voor een versnelling op de Ball. Er is een kans op een versnelling in de Y-richting en een kans op een versnelling in de negatieve Y-richting. Dit effect duurt 10 seconden.

### 5.2.6.4 Speed

Speed breidt Powerup uit met een methode *activate()*. Deze zorgt voor een vaste versnelling of vertraging op de Ball. Er is een kans op een versnelling en een kans op een vertraging. Dit effect duurt 10 seconden.

### 5.2.6.5 NewBall

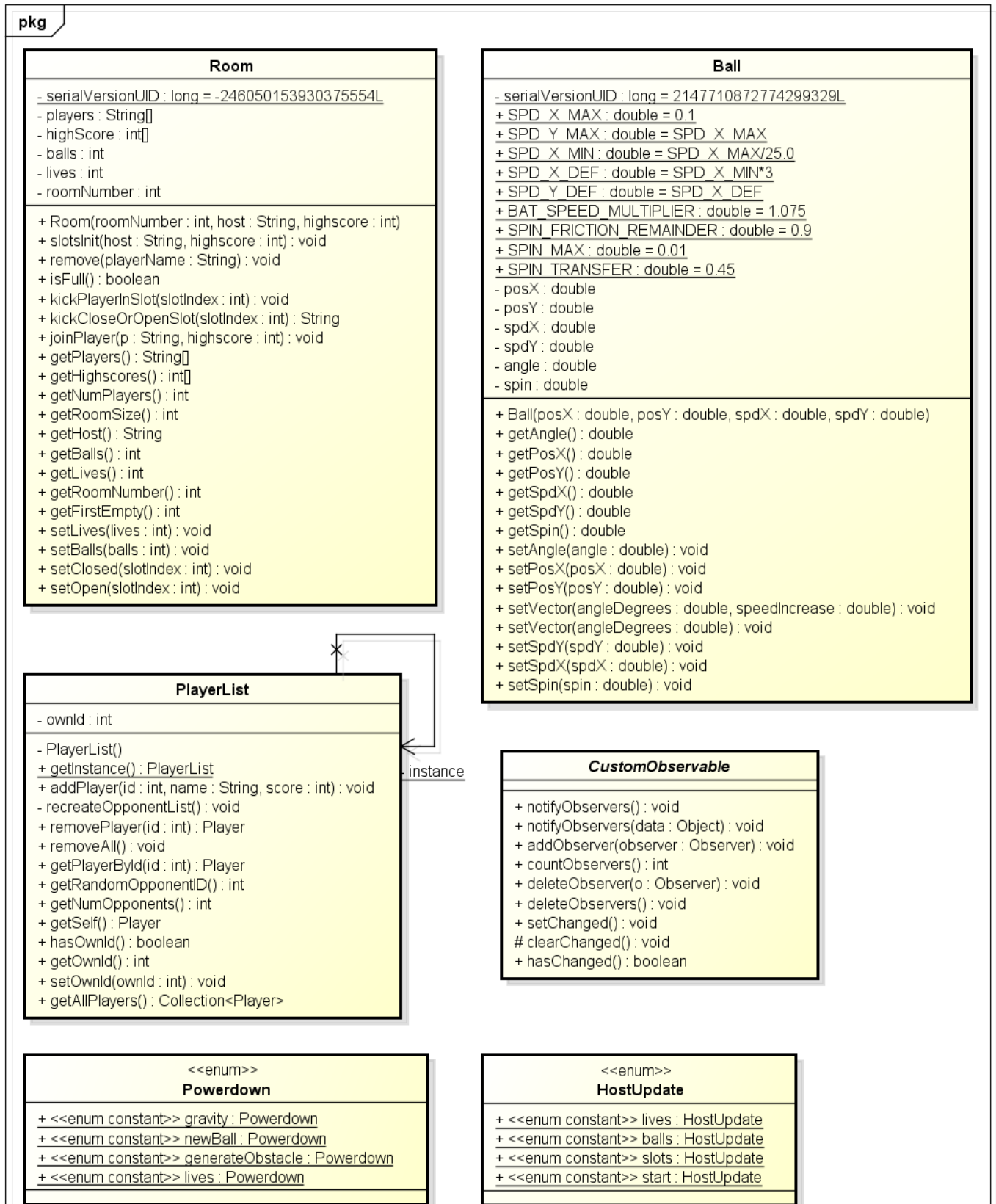
NewBall breidt Powerup uit met een methode *activate()*. Deze zorgt voor een nieuwe Ball in het veld met een willekeurige startpositie en een willekeurige hoek van beweging.

### 5.2.6.6 GenerateObstacle

GenerateObstacle breidt Powerup uit met een methode *activate()*. Deze zorgt voor een obstakel in het veld met een kans op een RandomDirectionChange en een kans op een FixedSpeedChange.



## 5.2.7 Detailed Shared



Figuur 15 - Detailed class diagram van shared

In de package *shared* staan een aantal klassen die gebruikt worden door meerdere andere klassen. Hier staan ook klassen die gebruikt worden tussen de client en de server. Dit is nodig, omdat de server een klein gedeelte van de implementatie van een aantal klassen nodig heeft.

### **5.2.7.1 Ball**

De klasse Ball houdt eigenschappen van de bal bij, zoals positie en snelheid en biedt methoden om die aan te passen. Ball wordt over het netwerk gestuurd, dus implementeert het Serializable.

### **5.2.7.2 CustomObservable**

CustomObservable is een Observable met nagenoeg dezelfde implementaties van methoden. Een klein verschil met het origineel zijn de synchronized methoden. Dit is nodig in het huidige systeem, omdat er meerdere aanroepen op dezelfde instantie kunnen zijn.

### **5.2.7.3 HostUpdate**

HostUpdate is een enumerator klasse om een type HostUpdate bij te houden. Dit wordt in het systeem gebruikt om netjes een HostUpdate aan te geven als er een HostUpdateEvent is ontvangen.

### **5.2.7.4 PlayerList**

PlayerList houdt een lijst met alle spelers in het spel en een lijst met alle tegenstanders bij. Voor elke speler wordt ook bijgehouden hoeveel levens de speler heeft. Daarnaast is het mogelijk om een willekeurige tegenstander te selecteren.

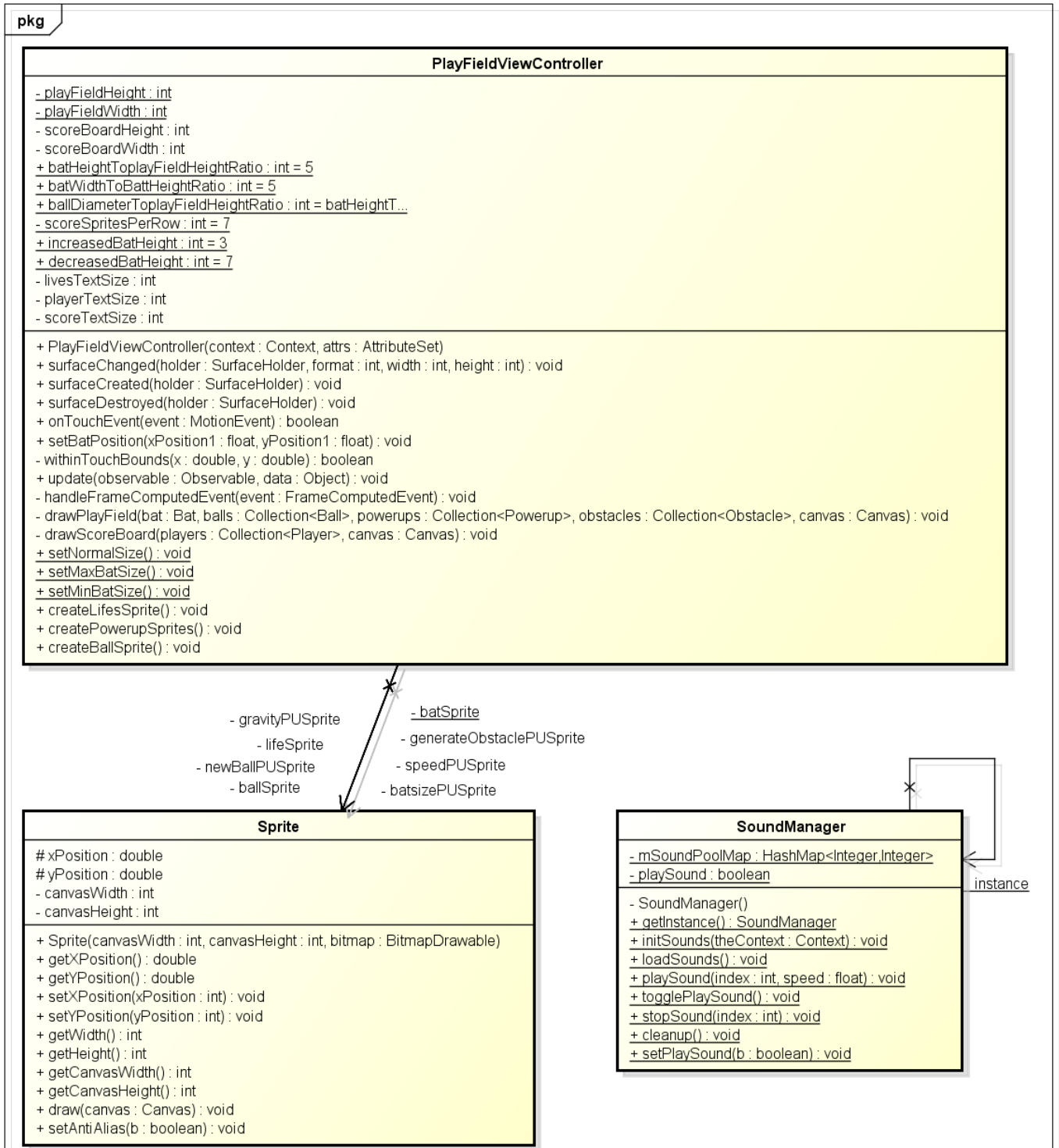
### **5.2.7.5 Room**

Room implementeert Serializable, omdat deze over het netwerk verstuurd wordt. Room houdt de spelers, hun highscore en de kamerinstellingen bij, zoals het aantal ballen en het aantal levens. Het is ook mogelijk om slots te openen, te sluiten en spelers te verwijderen uit de Room.

### **5.2.7.6 Powerdown**

Powerdown is een enumerator klasse om een type Powerdown bij te houden. Dit wordt in het systeem gebruikt om netjes een Powerdown aan te geven als er een PowerdownEvent is ontvangen.

## 5.2.8 Detailed View



Figuur 16 - Detailed class diagram van view

De package *view* wordt gebruikt voor klassen die beeld en geluid afhandelen in het systeem. Hierbij horen ook de continue aanpassingen van het speelveld, zoals het weergeven van de bal, power-ups en het tonen van de levens.

### **5.2.8.1 *PlayFieldViewController***

Deze klasse wordt gebruikt om het eigenlijke spel en het scorebord weer te geven met hierop de namen van de spelers, de hoeveelheid lives van de spelers en de eigen behaalde punten. De klasse houdt hiervoor een SurfaceHolder en een aantal Bitmaps bij. De SurfaceHolder bevat het Canvas waarop het speelveld, het scorebord en alle andere spelelementen getekend worden. De Bitmaps bevatten de afbeeldingen voor de bat, Ball, power-ups en het scoreboard op de juiste schaal. De klasse komt uit de oude versie van OVPong en is aangepast om de toegevoegde power-ups weer te kunnen geven, de grootte van de bat groter/kleiner te kunnen maken, en de score bij te kunnen houden.

### **5.2.8.2 *SoundManager***

SoundManager wordt in Android gebruikt om geluiden af te spelen. De belangrijkste functies zijn het spelen van geluiden, het stoppen daarvan en het aan en uit zetten van geluid. Deze klasse komt uit de oude versie van OVPong en is niet aangepast.

### **5.2.8.3 *Sprite***

De klasse Sprite wordt in Android gebruikt om plaatjes te weergeven. Sprite houdt alleen de dimensies van het plaatje bij en het biedt een methode om een plaatje te laten zien op het scherm. Deze klasse komt uit de oude versie van OVPong en is niet aangepast.

### 5.3 Uitleg Protocol

Communicatie tussen verschillende onderdelen van de applicatie gebeurt met behulp van events. Deze events zijn kleine Java-klassen die de te versturen informatie bevatten in de vorm van hun klassenaam en eventuele public variabelen. De events zijn terug te vinden in Appendix B.

Clientside worden events via een Observer patroon verstuurd. Binnen het spel gebeurt dit op een nette manier met behulp van een CustomObservable. Activity's communiceren met NetworkService via een Mutual Observer patroon. Activity's kunnen niet Observable zijn, omdat ze al Activity extenden en Java-klassen maar één superklasse mogen hebben. Hierdoor kon niet het standaard Observer patroon gebruikt worden voor de communicatie van Activity's naar NetworkService. In plaats daarvan is de functionaliteit van Observable nagebootst door het direct aanroepen van *NetworkService.update*. Dit is mogelijk, omdat Activity's alleen met NetworkService communiceren en NetworkService een singleton-klasse is.

Het versturen van events tussen client en server gebeurt via een TCP-connectie, die bij het opstarten van de applicatie wordt opgezet. De server wordt gevonden via een hardcoded IP-adres en poortnummer, waarna een verbinding wordt opgezet via Sockets. Als verschillende clients met elkaar communiceren zal dit met de server als tussenpersoon gebeuren. Alle events die over het netwerk worden verstuurd zijn Serializable en kunnen hierdoor overal opnieuw gecreëerd worden.

## 6 Testen

In dit gedeelte wordt het testen van het programma besproken. Er zijn blackbox-tests aanwezig in appendix C. Whitebox-testing is niet gebruikt, omdat dat in dit geval te moeilijk te testen is in verband met Android.

Alle testcases zijn gehaald, behalve testcase 27. Deze testcase gaat over het onverwachts afsluiten van een spel. Daar bedoelen we mee dat een speler het spel verlaat voordat deze speler geen levens meer heeft door bijvoorbeeld de applicatie af te sluiten tijdens het spel. Mocht dit gebeuren, dan zou de Server op de hoogte gesteld moeten worden en moet de Server dat afhandelen. Dit werkt niet als gewenst.

Als de applicatie afgesloten wordt, dan zou de ClientHandler een IOException moeten krijgen. Deze Exception vangt de ClientHandler op en meldt het bij de Server, waarna Server deze client uitlogt. Er hebben zich helaas momenten voorgedaan waar de ClientHandler geen IOException krijgt wanneer dit wel had moeten gebeuren.

Na de demo op 15 januari is ontdekt dat bij meerdere spelers de ballen niet evenwichtig naar alle spelers werd verstuurd. Deze fout is verbeterd en de oplossing is getest met behulp van een whitebox-test voor de specifieke methode. Deze test is meegeleverd als een apart project genaamd testOpponentID.

## 7 Evaluatie

In de evaluatie wordt nader ingegaan op de organisatie van het project, hoe het verliep en de uitbreidingen waar in dit project nog geen aandacht aan besteed was.

### 7.1 Het begin

Het begin van het project is erg traag begonnen, omdat in eerste instantie de communicatie met OVSoftware niet goed verliep. Hierdoor begon de opdracht pas 4 weken na het begin van het vak. Ook is het vinden van begeleiders moeizaam verlopen. Deze hebben we pas 2 weken later kunnen regelen. Ondanks een trage start is er elke week voldoende gewerkt aan het project. Ook is er elke week contact geweest met zowel de begeleider als de opdrachtgever en werd het tekort aan het begin wekelijks steeds verder ingehaald. Voor de volgende keer is het beter om niet te wachten tot de opdracht duidelijk is voordat er een begeleider gevonden gaat worden.

### 7.2 Samenwerking

Er is met vijf personen gewerkt aan dit project met allemaal verschillende taken. Een deel van de groep heeft zich bezig gehouden met de server en een ander deel met de client. Ondanks deze verdeling is toch overal door iedereen wat aan gewerkt. Zo is het overzicht op het project goed gehouden en zijn geschreven stukken tekst en code voldoende onderling gecontroleerd. De communicatie in de groep was goed. Elke week is er op gezette tijden samengewerkt aan het project. De rest van de communicatie ging via e-mails en sms als er bijvoorbeeld een afspraak afweek of als iemand wat later kwam. De samenwerking verliep goed.

### 7.3 Communicatie met begeleiders

Er is elke week een moment geweest waar we samen zijn gekomen met de begeleiders vanuit de Universiteit Twente en de begeleider van OVSoftware. Bij deze bijeenkomsten zijn voortgang en resultaten besproken. Het contact met de begeleiders verliep goed, er was veel contact, zowel fysiek als per e-mail en de responstijd van de begeleiders was snel. Er zijn 3 begeleiders geweest die feedback konden geven op het projectverslag. Zo is er veel verbetering in het verslag geweest.

### 7.4 Watervalmodel

Aan het begin van het project is er door de opdrachtgevers afgeraden om het standaard watervalmodel te gebruiken. Volgens hen was het watervalmodel in het praktijk niet handig en wordt het dus ook nooit gebruikt. Er is uiteindelijk toch gekozen om het watervalmodel te hanteren, aangezien deze het bekendst is in de groep en het beste past bij de gewenste verslagstructuur. Als er een verandering in een later stadium van het watervalmodel was, moesten de voorgaande stadia worden aangepast. Dit kostte niet veel tijd qua documenteren, maar het aanpassen van het verslag werd af en toe vergeten. Uiteindelijk zijn alle veranderingen geüpdatet en is het watervalmodel in dit project toch handig gebleken. De opdrachtgever veranderde namelijk niet constant van gedachten, dus hoefde er niet te veel aangepast te worden. Toch is het een goed idee om de volgende keer naar andere ontwikkelmethoden te kijken, omdat het watervalmodel niet in elke situatie handig is.

## 7.5 Opslag en versiebeheer

Voor het opslaan van de code, het verslag en alle andere documenten is Dropbox gebruikt. Hier is voor gekozen, omdat Dropbox in andere soorten projecten prima werkte. Achteraf gezien is het gebruik van Dropbox in dit project een slecht idee gebleken. Er werd namelijk door verschillende personen tegelijkertijd aan verschillende delen van de code gewerkt. Achteraf moest dat weer samengevoegd worden, en dat kostte tijd. Vaak werd het samenvoegen ook verkeerd gedaan of werd er iets vergeten. Zo raakten veranderingen verloren in de nieuwste versies van de code. Dropbox is toch nog tot het eind van het project gebruikt en alle veranderingen zijn uiteindelijk wel doorgekomen. Het is een goed idee om de volgende keer een ander programma te gebruiken voor het opslaan en delen van de code. Een alternatief is Github.

## 7.6 Aanbevelingen

Er zijn in dit project al heel wat uitbreidingen gerealiseerd voor OVPong, maar er waren nog een aantal andere ideeën, waar niet genoeg tijd voor was.

### 7.6.1 Verbinding verbeteren

De client-server-verbinding was niet ideaal, bleek uit de demonstratie die bij OVSoftware was uitgevoerd. Als een client de verbinding verliest, hoort de server dat te weten en af te vangen. Dit gebeurt momenteel alleen nog als de client weet dat deze de verbinding verliest. Als er een probleem met het netwerk is of met het apparaat dat de client draait, kan de verbinding onverwachts verbreken. In dat geval is er nog geen ingebouwd systeem om dat af te vangen. Een voorstel is om dit te doen met “keep-alive”-berichten, die tussen de server en de client gestuurd kunnen worden.

### 7.6.2 Automatisch inloggen

Het inlogproces moet momenteel elke keer handmatig verricht worden door de gebruiker. De mogelijkheid bestaat om de applicatie de inloggegevens te laten onthouden en automatisch in te laten loggen.

### 7.6.3 Encryption

In het verslag van de originele makers van OVPong is al aanbevolen om berichten die over het netwerk gaan versleuteld te maken. Dit is in deze iteratie van OVPong niet opgepikt en is nog steeds een mogelijke uitbreiding.

### 7.6.4 Achievements

Initieel was het plan om ook achievements toe te voegen aan de applicatie, maar deze functie is in een vroeg stadium van het project verworpen. Het is echter nog steeds een leuk idee om te implementeren, aangezien er al accounts en bijbehorende highscores bestaan.

### 7.6.5 Globale highscorelijst

Het spel ondersteunt momenteel al highscores en een toevoeging kan een globale highscorelijst zijn, waar de spelers met het meest aantal punten bovenaan staan.

### 7.6.6 Chat

In de eerste versies van de user interface is er rekening gehouden met de mogelijkheid om te chatten. Door een tekort aan schermruimte is dit idee intern afgewezen. Het is echter nog steeds een goed idee om een vorm van communicatie tussen gebruikers te hebben.

### 7.6.7 Meer power-ups en obstakels

Het idee van power-ups en obstakels is in dit project gerealiseerd, maar er kunnen meerdere soorten power-ups en obstakels zijn. Er moet dan vooral gekeken worden naar OVSoftware-gerelateerde obstakels.



### 7.6.8 DNS

Een suggestie die al gedaan is door de opdrachtgevers bij OVSoftware, is het gebruiken van een DNS-naam, in plaats van een hardcoded IP-adres en poortnummer. Dit is uiteindelijk niet gerealiseerd, maar het kan een leuke implementatie zijn, die de puntjes op de “i” zet voor de applicatie.

## 8 Appendix A – Use Cases

UC Nr:	1		
Authors:	Jeffrey, Henley, Sander, Stanley, Alexander		
Version:	4	Date:	13-01-14
UC Name:	Login		
Summary:	User logs into the system.		
Actors:	User		
Goals	Log in.		
Preconditions:	User is not logged in.		
Trigger event	Program starts up.		
Related UC's:			
Description normal flow:	User  Types username and password.  Confirms username and password.	System  Asks for username and password.    Shows the menu.	
Description alternative flow  New account	User  Types username and password.  Confirms username and password.	System  Asks for username and password.   Adds the username, password combination to the database of accounts.  Shows the Menu	
Description alternative flow  Short password	User  Types username and a short password.  Confirms username and	System  Asks for username and password.	

	password.	Shows an error for password.
Description alternative flow  incorrect password	User  Types username and a password.  Confirms username and password.	System  Asks for username and password.    Shows an error for password.
Description alternative flow  Username too long	User  Types a long username and a short password.  Confirms username and password.	System  Asks for username and password.    Shows an error for username.
Description alternative flow  Username reserved	User  Types username and a short password.  Confirms username and password.	System  Asks for username and password.    Shows an error for username.
Description alternative flow  username already logged-in	User  Types username and a short password.  Confirms username and password.	System  Asks for username and password.    Shows an error for username.
Postconditions	User is logged in and in the main menu.	

UC Nr:	2		
Authors:	Jeffrey, Henley, Sander, Stanley,Alexander		
Version:	4	Date:	13-01-14
UC Name:	Enter lobby		
Summary:	User enters the lobby.		
Actors:	User		
Goals	Enter the lobby.		
Preconditions:	User is in the menu		
Trigger event	User presses the "Custom play" button.		
Related UC's:	UC:1		
Description normal flow:	User Presses the "Custom play" button.	System	Shows the lobby.
Description alternative flow			
Postconditions	User is in the lobby.		

UC Nr:	3		
Authors:	Jeffrey, Henley, Sander, Stanley, Alexander		
Version:	4	Date:	13-01-14
UC Name:	Quick play		
Summary:	User starts a game via quick play.		
Actors:	User		
Goals	Play a quick game.		
Preconditions:	User is in the menu.		
Trigger event	User presses the "quick play" button.		
Related UC's:	UC:1, UC:6		
Description normal flow:	User Presses the "quick play" button.	System  Opens a game screen and starts the game.	
Description alternative flow	User Presses the "quick play" button.	System  Shows it's waiting for other players.  Opens a game screen and starts the game.	
Postconditions	Game has started.		

UC Nr:	4		
Authors:	Jeffrey, Henley, Sander, Stanley, Alexander		
Version:	4	Date:	13-01-14
UC Name:	Make new room.		
Summary:	User starts a new room.		
Actors:	User		
Goals	Start a new room.		
Preconditions:	User is in the lobby.		
Trigger event	User presses the "+" button.		
Related UC's:	UC:2		
Description normal flow:	User Presses the "+" button.	System  Opens a room screen.	
Description alternative flow			
Postconditions	New room is created and the user is the host.		

UC Nr:	5		
Authors:	Jeffrey, Henley, Sander, Stanley, Alexander		
Version:	3	Date:	13-01-14
UC Name:	Join room.		
Summary:	User joins a room.		
Actors:	User		
Goals	Join a room.		
Preconditions:	User is in the lobby.		
Trigger event	User presses a room.		
Related UC's:	UC:2, UC:4		
Description normal flow:	User Presses a room.	System  Opens a room screen.	
Description alternative flow Room Full	User Presses a room.	System  Show a room full error	
Description alternative flow Room non-existent	User Presses a room.	System  Show a room non-existent error.	
Postconditions	User has entered the room.		

UC Nr:	6		
Authors:	Jeffrey, Henley, Sander, Stanley, Alexander		
Version:	3	Date:	13-01-14
UC Name:	Start game.		
Summary:	Host starts the game.		
Actors:	Host		
Goals	Start a game.		
Preconditions:	Player is host of a room.		
Trigger event	Host presses the "start game" button.		
Related UC's:	UC:2, UC:3, UC:4, UC:5		
Description normal flow:	Host Presses the "start game" button.	System	Starts a game. Opens the game screen.
Description alternative flow			
Postconditions	Game has started.		



UC Nr:	7		
Authors:	Jeffrey, Henley, Sander, Stanley, Alexander		
Version:	3	Date:	13-01-14
UC Name:	Invite		
Summary:	Host invites a player.		
Actors:	Host		
Goals	Invite a player.		
Preconditions:	Player is host of a room.		
Trigger event	Host presses the "invite" button.		
Related UC's:	UC:4, UC:5, UC:8		
Description normal flow:	User Presses the "invite" button.  Enters a player name and confirms.	System   Sends an invite to player matching the player name.	
Description alternative flow  No player with matching name logged in	User Presses the "invite" button.  Enters a player name and confirms.	System	
Postconditions	Player is invited.		

UC Nr:	8		
Authors:	Jeffrey, Henley, Sander, Stanley, Alexander		
Version:	3	Date:	13-01-14
UC Name:	Accept invitation		
Summary:	User accepts an invitation.		
Actors:	User		
Goals	Accept an invitation.		
Preconditions:	User is in the lobby.		
Trigger event	User is being invited.		
Related UC's:	UC:2, UC:4, UC:5, UC:7		
Description normal flow:	User  Accepts the invitation.	System  Shows the invitation.  Opens the room screen.	
Description alternative flow Decline invitation	User  Declines the invitation.	System  Shows the invitation.  Closes the invitation.	
Postconditions	User is in the room.		

UC Nr:	9		
Authors:	Jeffrey, Henley, Sander, Stanley, Alexander		
Version:	3	Date:	13-01-14
UC Name:	Kick		
Summary:	Host kicks a player.		
Actors:	Host, kicked Player		
Goals	Kick a player.		
Preconditions:	Host is in a room with at least one other player.		
Trigger event	Host presses the "kick" button.		
Related UC's:	UC:2, UC:4, UC:5, UC:10, UC:11		
Description normal flow:	Host Presses the "kick" button behind the player's name.	System  Removes the player from the room.	
Description simultaneous flow Kicked player	Kicked Player  Acknowledges the message	System  Shows the kick message.  Shows the lobby	
Description alternative flow			
Postconditions	Player is removed from the room and is in the lobby.		

UC Nr:	10		
Authors:	Jeffrey, Henley, Sander, Stanley, Alexander		
Version:	4	Date:	13-01-14
UC Name:	Close slot		
Summary:	Host closes a slot.		
Actors:	Host		
Goals	Close a slot.		
Preconditions:	Not all slots are filled.		
Trigger event	Host presses the "close" button.		
Related UC's:	UC:4, UC:5, UC:9, UC:11		
Description normal flow:	Host  Presses the "close" button at the end of an open slot.	System  Closes the slot.	
Description alternative flow			
Postconditions	Slot is closed.		

UC Nr:	11		
Authors:	Jeffrey, Henley, Sander, Stanley, Alexander		
Version:	3	Date:	13-01-14
UC Name:	Open slot		
Summary:	Host opens a slot.		
Actors:	Host		
Goals	Open a slot.		
Preconditions:	A slot is closed.		
Trigger event	Host presses the "open" button.		
Related UC's:	UC:4, UC:5, UC:9, UC:10		
Description normal flow:	Host  Presses the "open" button at the end of a closed slot.	System	  Opens the slot.
Description alternative flow			
Postconditions	Slot is open.		

UC Nr:	12		
Authors:	Jeffrey, Henley, Sander, Stanley, Alexander		
Version:	3	Date:	13-01-14
UC Name:	Power-up/power-down		
Summary:	User gets power-up/power-down and the user is affected by the power-up or an opponent is affected by the power-down.		
Actors:	User, Targeted Player		
Goals	Activate power-up/power-down		
Preconditions:	User is playing a game.		
Trigger event	User collects power-up/power-down		
Related UC's:	UC:15		
Description normal flow:	User  Collects power-up	System  Shows power-up  Shows power-up effects	
Description alternative flow Power-down	User  Collects power-down	System  Shows power-down  Shows name of targeted player in red.	
Description simultaneous flow Targeted player	Targeted Player	System  Shows power-down effects	
Postconditions	User has power-up/opponent has power-down.		

UC Nr:	13		
Authors:	Jeffrey, Henley, Sander, Stanley, Alexander		
Version:	3	Date:	13-01-14
UC Name:	End game.		
Summary:	User goes back to the lobby or main menu after the game ends.		
Actors:	User		
Goals	Going to the lobby or main menu screen.		
Preconditions:	User is in the game.		
Trigger event	The game ends.		
Related UC's:	UC:2, UC3, UC:6		
Description normal flow:	User Plays a custom game. Ends game.	System  Shows lobby screen.	
Description alternative flow	User Plays a quick game. Ends game.	System  Shows menu screen.	
Postconditions	User is in the screen he came from.		

UC Nr:	14		
Authors:	Alexander		
Version:	2	Date	13-01-14
UC Name:	Instructions		
Summary:	User looks up Instructions		
Actors:	User		
Goals	Look up Instructions		
Preconditions:	User is in the menu.		
Trigger event	User presses the "Instruction" button		
Related UC's:	UC:1		
Description normal flow:	User Presses the "Instruction" button	System  Shows Instruction screen.	
Description alternative flow			
Postconditions	Instructions are visible including all power-ups.		



UC Nr:	15		
Authors:	Jeffrey, Henley, Stanley, Alexander		
Version:	3	Date	13-01-14
UC Name:	Obstacles		
Summary:	User gets an obstacle power-down which creates an obstacle in the field of an opponent.		
Actors:	User, Targeted Player		
Goals	Get a power-down		
Preconditions:	User is in the game		
Trigger event	User collects an obstacle power-down		
Related UC's:	UC:3, UC:6, UC:12		
Description normal flow:	User  Collects obstacle power-down.	System  Shows obstacle power-down.  Shows name of targeted player in red.	
Description simultaneous flow  Targeted player	Targeted Player	System  Shows the obstacle	
Description alternative flow			
Postconditions	There is an obstacle in the field of the targeted player.		

### 8.1.1 Traceability Matrix: Requirements Use-Cases

	F1	F1.1	F2	F2.1	F2.2	F2.3	F3	F4	F5
UC1						X	X	X	
UC2						X		X	
UC3	X			X		X			
UC4						X		X	
UC5						X		X	
UC6	X			X		X		X	
UC7						X		X	
UC8						X		X	
UC9						X		X	
UC10						X		X	
UC11						X		X	
UC12	X			X		X			
UC13									X
UC14					X	X			
UC15		X		X		X			

### 8.1.2 Traceability Matrix: Activity's Use-Cases

	SplashActivity	LoginActivity	MainMenuActivity	LobbyActivity	RoomActivity	GameActivity	InstructionActivity
UC1		X	X				
UC2			X	X			
UC3			X			X	
UC4				X	X		
UC5				X	X		
UC6					X	X	
UC7				X	X		
UC8				X	X		
UC9				X	X		
UC10					X		
UC11					X		
UC12						X	
UC13			X	X		X	
UC14			X				X
UC15						X	

### 8.1.3 Use case Diagram



Figuur 17 - Use case diagram van het systeem

## 9 Appendix B – Protocol

### 9.1 Activity Events

#### 9.1.1 LoginActivity

LoginEvent: dit event wordt gebruikt door de client om in te loggen. Het bevat een naam (String) en een wachtwoord (String). LoginEvent komt vanuit LoginActivity en gaat naar de ClientHandler van de server die bij de client hoort.

LoginAuthenticationEvent: dit event wordt gebruikt door de server om de resultaten van LoginEvent door te geven aan de client. LoginAuthenticationEvent bevat 2 booleans, de eerste geeft aan of het inloggen gelukt is en de tweede voor als het account al is ingelogd. Dit event komt vanuit de ClientHandler van de server die bij de client hoort en gaat naar de LoginActivity van de client.

#### 9.1.2 MainMenuActivity

QuickPlayEvent: dit event wordt gebruikt door de client om aan te geven dat de client geïnteresseerd is in een quickplay-spel. QuickPlayEvent heeft geen variabelen. Zodra de client naar een ander scherm gaat, stuurt hij een tweede QuickPlayEvent als deze een QuickPlayEvent gestuurd had. Als de server een tweede QuickPlayEvent ontvangt, wordt de client weer uit de wachtlijst voor een quickplay-spel verwijderd. Dit event komt vanuit MainMenuActivity en gaat naar ClientHandler van de server die bij de client hoort.

JoinLobbyEvent: Dit event wordt gebruikt om de server te notificeren dat een speler de lobby is ingegaan. JoinLobbyEvent heeft geen variabelen. Het komt vanuit MainMenuActivity en gaat naar ClientHandler van de server die bij de client hoort.

#### 9.1.3 LobbyActivity

LobbyErrorEvent: dit event wordt door de server naar de client gestuurd als er een fout is opgetreden en de client niet een RoomActivity mag starten. Het wordt alleen verstuurd, omdat de poging van de client om een kamer binnen te gaan is mislukt. LobbyErrorEvent komt vanuit de server en gaat naar de LobbyActivity van de client.

RequestRefreshEvent: dit event wordt gebruikt door de client om aan de server een geüpdatete lijst van rooms op te vragen. RequestRefreshEvent heeft geen variabelen. Dit event komt vanuit LobbyActivity en gaat naar ClientHandler van de server die bij de client hoort.

RefreshLobbyEvent: dit event wordt gebruikt door de server om een geüpdatete lijst met rooms naar de client te sturen. Het heeft een lijst van Rooms (ArrayList). RefreshLobbyEvent komt vanuit de ClientHandler van de server die bij de client hoort en gaat naar de LobbyActivity van de client.

ReceivedInviteEvent: dit event wordt gebruikt door de server om een client een invite te sturen. Het heeft een Room waaruit de naam van de persoon inviter (String) en het kamernummer(Int) kunnen worden gehaald. ReceivedInviteEvent komt vanuit de ClientHandler van de server die bij de client hoort en gaat naar de LobbyActivity van de client.

CreateRoomEvent: dit event wordt gebruikt door client als deze een room wil creëren als host. CreateRoomEvent heeft geen variabelen. CreateRoomEvent komt vanuit LobbyActivity en gaat naar

ClientHandler van de server die bij de client hoort. Als reactie hierop kan de server een RoomUpdateEvent versturen.

JoinRoomEvent: dit event wordt gebruikt door client als deze een bestaande room wil binnengaan. Het heeft een kamernummer (Int) die wordt gebruikt om de kamer te identificeren. JoinRoomEvent komt vanuit LobbyActivity en gaat naar ClientHandler van de server die bij de client hoort. Als reactie op dit event kan de server een RoomUpdateEvent versturen.

#### 9.1.4 RoomActivity

RoomErrorEvent: Dit event wordt door de server naar de client gestuurd als er een error is gebeurd of de speler gekicked is, en de client moet terugkeren naar LobbyActivity. RoomErrorEvent wordt alleen verstuurd als de client al in de room zat, en deze moet verlaten. Dit event wordt door Game van de server gegenereerd als de client gekicked wordt of Server als er een andere error is gebeurd. het event gaat naar RoomActivity van de clients die de room moeten verlaten.

HostUpdateEvent: Dit event wordt door de host van een room gebruikt om veranderingen van kamerinstellingen aan de server door te geven en om een spel te starten. HostUpdateEvent heeft een type kamer verandering(HostUpdate) en een integer variabele. Dit event komt vanuit RoomActivity en gaat naar ClientHandler van de server die bij de client hoort. Als reactie op dit event kan de server een StartEvent, RoomUpdateEvent of RoomErrorEvent versturen.

InviteEvent: Dit event wordt gebruikt door de host van een room als deze een specifieke speler wil uitnodigen. InviteEvent bevat een naam(String). InviteEvent komt vanuit RoomActivity en gaat naar ClientHandler van de server die bij de client hoort. Als reactie op dit event kan de server een ReceivedInviteEvent versturen.

PlayerLeftEvent: Dit event wordt gebruikt als een speler een kamer verlaat en terugkeert naar de lobby. PlayerLeftEvent heeft geen variabelen. Dit event komt vanuit RoomActivity en gaat naar ClientHandler van de server die bij de client hoort. Als reactie op dit event kan de server een RoomUpdateEvent of RoomErrorEvent versturen.

#### 9.1.5 Meerdere Activity's

StartEvent: Dit event geeft aan dat er naar GameActivity moet worden gegaan en een spel wordt gestart. StartEvent heeft een Room waarin de definitieve informatie over het spel staat (levens, ballen en spelers). Dit event wordt verstuurd door de server en ontvangen door MainMenuActivity en RoomActivity.

RoomUpdateEvent: Dit event wordt door de server gebruikt om alle clients binnen een room te notificeren van veranderingen in die room en om clients rooms binnen te laten gaan.

RoomUpdateEvent heeft een Room. Dit event wordt verstuurd door de server en ontvangen door LobbyActivity en RoomActivity.

## 9.2 Game Events

Game events worden gebruikt voor spel-gerelateerde communicatie, dit gebeurt zowel intern binnen een client als over het netwerk.

### 9.2.1 Intern

**FrameComputedEvent:** Dit event wordt gebruikt om `PlayFieldViewController` te notificeren dat een nieuw frame is berekend en dus kan updaten. `FrameComputedEvent` heeft geen variabelen. Dit event komt vanuit `Game` en gaat naar `PlayFieldViewController`.

**BallLeavingFieldEvent:** Dit event wordt gebruikt om `Game` te notificeren dat een `Ball` het veld heeft verlaten. Dit event heeft een `Ball`. `BallLeavingFieldEvent` komt vanuit `Field` en gaat naar `Game`.

### 9.2.2 Extern

Al deze Events zijn `Serializable`.

**ServeBallEvent:** Dit event wordt door de server gestuurd om nieuwe ballen te creëren. Dit Event wordt ook aan het begin van een spel gebruikt voor de initiële ballen. `ServeBallEvent` heeft geen variabelen. Dit event komt vanuit `ServerGame` er wordt naar `Game` van verschillende clients gestuurd.

**SendBallEvent:** Dit event wordt gebruikt om een `Ball` van de ene naar de andere client te sturen. `SendBallEvent` heeft een `Ball` die verstuurd wordt en een `playerid(int)` van de speler naar wie de bal verstuurd wordt. Dit event wordt verstuurd vanuit `Game` via de server naar een andere client.

**LifeLostEvent:** Dit event wordt gebruikt om iedereen te notificeren als een client een leven verliest. Dit Event heeft een `scoreAt(int)`. `LifeLostEvent` wordt verstuurd vanuit `Field`, van de client die een leven verloren heeft, naar zijn `Game` en vervolgens via server naar de `Game` van alle andere clients die in het spel betrokken zijn.

**GameEndEvent:** Dit event wordt gebruikt als een speler het spel verlaat, verliest of wint. `GameEndEvent` heeft een `graceful(boolean)` die aangeeft of het verlaten door het spel komt doordat de speler gewonnen of verloren heeft of niet en een `score(int)` die de score van de speler aan de server verstuurt. Dit event komt vanuit `Game` en gaat naar `ClientHandler` van de server die bij de client hoort. Vervolgens wordt `GameEndEvent` doorgestuurd naar alle andere clients die in het spel betrokken zijn.

**PlayerDisconnectedEvent:** Dit event wordt door de server gebruikt als een client een spel verlaat zonder dat deze een `GameEndEvent` met de boolean `graceful` op `true` heeft gestuurd.

`PlayerDisconnectedEvent` heeft een `playerid(int)`. Dit event komt vanuit `ServerGame` en wordt door `Game` van alle clients, die bij het spel horen waar de client van disconnect, ontvangen.

**PowerdownEvent:** Dit event wordt gebruikt om een powerdown van de ene naar de andere client te sturen. `PowerdownEvent` heeft een `type powerdown(Powerdown)` en een `playerid(int)`. Dit event wordt verstuurd door `Game` van de client die de powerdown verzamelde en de `Game` van de client die door de powerdown wordt beïnvloed.

## 10 Appendix C - Testcases

Test ID	1
Beschrijving	Test het opstarten van de applicatie
Requirements	Er draait een bereikbare server
Stappen	1. Start de applicatie
Verwacht resultaat	De applicatie start op met het splashscreen en zal na een paar momenten naar het loginscherm gaan. De client maakt connectie met de server.
Werkelijke resultaat	De applicatie start op met het splashscreen en gaat na een paar momenten naar het loginscherm. De client maakt connectie met de server.
Opruimen	Sluit de applicatie af, hierbij wordt de client ook bij de server gedisconnect.



Test ID	2
Beschrijving	Test het inloggen op de server
Requirements	Er is een opgestarte applicatie zoals aan de hand van test case 1
Stappen	<ol style="list-style-type: none"> <li>1. Voer een naam en wachtwoord in</li> <li>2. Druk op de sign-in knop</li> </ol>
Verwacht resultaat	De applicatie zal naar het hoofdmenu gaan na een kort voortgangsscherm gezien te hebben.
Alternative Flow 1 – incorrect wachtwoord	Als de naam al bestaat binnen de server maar de wachtwoorden komen niet overeen dan zal na het voortgangsscherm het loginscherm weer getoond worden met een error bij het wachtwoord.
Alternative Flow 2 – wachtwoord te kort	Als het wachtwoord minder dan 4 tekens is dan zal een error hierover verschijnen en er wordt geen poging gedaan om in te loggen.
Alternative Flow 3 – gebruikersnaam te lang	Als de gebruikersnaam meer dan 10 tekens is dan zal een error hierover verschijnen en er wordt geen poging gedaan om in te loggen.
Alternative Flow 4 – gebruikersnaam is gereserveerd	Als de gebruikersnaam gereserveerd is dan zal een error hierover verschijnen en er wordt geen poging gedaan om in te loggen.
Alternative Flow 5 – gebruiker is al ingelogd	Als de gebruiker op een account inlogt die op dat moment al is ingelogd dan zal na het voortgangsscherm het loginscherm weer getoond worden met een error bij de gebruikersnaam
Werkelijke resultaat	De applicatie zal naar het hoofdmenu gaan na een kort voortgangsscherm gezien te hebben.
Alternative Flow 1 – incorrect wachtwoord	Als de naam al bestaat binnen de server maar de wachtwoorden komen niet overeen dan zal na het voortgangsscherm het loginscherm weer getoond worden met een error bij het wachtwoord.
Alternative Flow 2 – wachtwoord te kort	Als het wachtwoord minder dan 4 tekens is dan zal een error hierover verschijnen en er wordt geen poging gedaan om in te loggen.
Alternative Flow 3 – gebruikersnaam te lang	Als de gebruikersnaam meer dan 10 tekens is dan zal een error hierover verschijnen en er wordt geen poging gedaan om in te loggen.
Alternative Flow 4 – gebruikersnaam is gereserveerd	Als de gebruikersnaam gereserveerd is dan zal een error hierover verschijnen en er wordt geen poging gedaan om in te loggen.
Alternative Flow 5 – gebruiker is al ingelogd	Als de gebruiker op een account inlogt die op dat moment al is ingelogd dan zal na het voortgangsscherm het loginscherm weer getoond worden met een error bij de gebruikersnaam
Opruimen	Sluit de applicatie af

Test ID	3
Beschrijving	Een test om te kijken op Quick Play werkt
Requirements	Er is een ingelogde apparaat zoals aan de hand van test case 2
Stappen	1. Druk op de Quick Play knop
Verwacht resultaat	De tekst van de quickplay knop verandert naar een wachttekst en de client vraagt de server naar een quickplay.
Werkelijke resultaat	De tekst van de quickplay knop verandert naar een wachttekst en de client vraagt de server naar een quickplay.
Opruimen	Sluit de applicatie af

Test ID	4
Beschrijving	Een test om te kijken op Quick Play werkt
Requirements	Er zijn 2 ingelogde apparaten zoals aan de hand van test case 2
Stappen	1. Druk op de Quick Play knop voor beide apparaten
Verwacht resultaat	Nadat beide apparaten op de Quick Play knop hebben gedrukt, start een kort spel met beide spelers met 1 bal en 3 levens.
Werkelijke resultaat	Nadat beide apparaten op de Quick Play knop hebben gedrukt, start een kort spel met beide spelers met 1 bal en 3 levens.
Opruimen	Sluit de applicaties af

Test ID	5
Beschrijving	Een test of de geluidknop werkt
Requirements	Er is een ander apparaat die op een quick play zit te wachten, zoals test case 3.
Stappen	1. Voer Test Case 2 uit 2. Zet in het hoofdmenu de toggleknop van het geluid op aan. 3. Druk op de Quick Play Knop
Verwacht resultaat	Er wordt een kort spel gestart waarbij het test apparaat geluid heeft.
Alternative Flow 1 – geluid uit	Bij stap 2 wordt de toggleknop van het geluid op uit gezet. Bij het korte spel zou het test apparaat geen geluid moeten hebben.
Werkelijke resultaat	Er wordt een kort spel gestart waarbij het test apparaat geluid heeft.
Alternative Flow 1 – geluid uit	Er wordt een kort spel gestart waarbij het test apparaat geen geluid heeft.
Opruimen	Sluit de applicaties af

Test ID	6
Beschrijving	Een test of er naar het instructie scherm kan worden gegaan.
Requirements	Er is een ingelogde applicatie, zoals Test Case 2.
Stappen	1. Druk op de instructie knop
Verwacht resultaat	De applicatie zal naar het instructie scherm gaan
Werkelijke resultaat	De applicatie gaat naar het instructie scherm
Opruimen	Sluit de applicaties af

Test ID	7
Beschrijving	Een test of er naar het lobby scherm kan worden gegaan
Requirements	Er is een ingelogde applicatie, zoals Test Case 2.
Stappen	1. Druk op de Custom Play knop
Verwacht resultaat	De applicatie zal naar het lobby scherm gaan, die na enkele momenten met de lijst van actieve kamers wordt geüpdatet.
Werkelijke resultaat	De applicatie gaat naar het lobby scherm, die na enkele momenten met de lijst van actieve kamers wordt geüpdatet.
Opruimen	Sluit de applicaties af

Test ID	8
Beschrijving	Een test of clients nieuwe rooms kunnen creëren
Requirements	Er is een applicatie in het lobby scherm, zoals Test Case 7
Stappen	1. Druk op de new room knop
Verwacht resultaat	De client komt in het room scherm van een nieuwe room met zichzelf als host
Werkelijke resultaat	De client komt in het room scherm van een nieuwe room met zichzelf als host
Opruimen	Sluit de applicaties af

Test ID	9
Beschrijving	Een test of de lobby kan worden gerefresht
Requirements	Er is een applicatie in het lobby scherm, zoals Test Case 7. De actieve rooms zijn hierna veranderd.
Stappen	1. Voer Test Case 8 uit op een ander device 2. Druk op de refresh knop
Verwacht resultaat	De applicatie zal na enkele moment met de lijst van actieve kamers updaten.
Werkelijke resultaat	De applicatie wordt na enkele momenten met de lijst van actieve kamers geüpdatet.
Opruimen	Sluit de applicaties af

Test ID	10
Beschrijving	Een test of clients rooms kunnen joinen
Requirements	Er is een applicatie in het lobby scherm, zoals Test Case 7. Er is op zijn minst 1 actieve room.
Stappen	1. Druk op een room
Verwacht resultaat	De client komt in het room scherm van de selecteerde room waar alle buttons niet drukbaar zijn.
Alternative Flow 1 – Room Full	Als de geselecteerde room vol is dan zal er een error verschijnen en zal de client in het lobby scherm blijven.
Alternative Flow 2 – Room non-existent	Als de geselecteerde room niet meer bestaat dan zal er een error verschijnen en zal de client in het lobby scherm blijven.
Werkelijke resultaat	De client komt in het room scherm van de selecteerde room waar alle buttons niet drukbaar zijn.
Alternative Flow 1 – Room Full	Als de geselecteerde room vol is dan zal er een error verschijnen en zal de client in het lobby scherm blijven.
Alternative Flow 2 – Room non-existent	Als de geselecteerde room niet meer bestaat dan zal er een error verschijnen en zal de client in het lobby scherm blijven.
Opruimen	Sluit de applicaties af

Test ID	11
Beschrijving	Een test of de host het ingestelde aantal levens kan veranderen
Requirements	Er is een room met een aantal spelers
Stappen	1. De host selecteert een nieuw aantal levens
Verwacht resultaat	Het aantal levens verandert naar het nieuwe aantal voor alle spelers
Werkelijke resultaat	Het aantal levens verandert naar het nieuwe aantal voor alle spelers
Opruimen	Sluit de applicaties af

Test ID	12
Beschrijving	Een test of de host het ingestelde aantal ballen kan veranderen
Requirements	Er is een room met een aantal spelers
Stappen	1. De host selecteert een nieuw aantal ballen
Verwacht resultaat	Het aantal ballen verandert naar het nieuwe aantal voor alle spelers
Werkelijke resultaat	Het aantal ballen verandert naar het nieuwe aantal voor alle spelers
Opruimen	Sluit de applicaties af

Test ID	13
Beschrijving	Een test of de host de speler slots kan veranderen
Requirements	Er is een room met een aantal spelers
Stappen	1. De host drukt op de button bij een geopend slot
Verwacht resultaat	Het geopende slot wordt bij alle spelers veranderd naar een gesloten slot.
Alternative Flow 1 – Gesloten Slot	De host drukt op de button bij een gesloten slot. Deze gesloten slot wordt dan bij alle spelers veranderd naar een open slot
Alternative Flow 2 - Kick	De host drukt op de button bij een slot waar een speler zit. Deze slot wordt voor alle spelers behalve de geselecteerde naar open veranderd. De gekickte speler krijgt een bericht dat hij gekickt is.
Werkelijke resultaat	Het geopende slot wordt bij alle spelers veranderd naar een gesloten slot.
Alternative Flow 1 – Gesloten Slot	De host drukt op de button bij een gesloten slot. Deze gesloten slot wordt dan bij alle spelers veranderd naar een open slot
Alternative Flow 2 – Kick	De host drukt op de button bij een slot waar een speler zit. Deze slot wordt voor alle spelers behalve de geselecteerde naar open veranderd. De gekickte speler krijgt een bericht dat hij gekickt is.
Opruimen	Sluit de applicaties af

Test ID	14
Beschrijving	Een test of de host specifieke spelers kan inviten.
Requirements	Er is een room met een aantal spelers en er is een speler in de lobby
Stappen	<ol style="list-style-type: none"> <li>1. De host drukt op de invite button</li> <li>2. Bij de verschenen pop-up vult de host de naam van de speler in de lobby in.</li> <li>3. De host drukt op OK</li> </ol>
Verwacht resultaat	De speler in de lobby krijgt een invite.
Werkelijke resultaat	De speler in de lobby krijgt een invite.
Opruimen	Sluit de applicaties af

Test ID	15
Beschrijving	Een test of invites werken
Requirements	Er is een room met een aantal spelers en er is een speler in de lobby. De speler in de lobby wordt uitgenodigd zoals in Test Case 14
Stappen	1. De geïnviteerde speler drukt op Accept
Verwacht resultaat	De geïnviteerde speler komt in de room op de eerste volgende lege plek en dit wordt voor alle spelers geüpdatet.
Werkelijke resultaat	De geïnviteerde speler komt in de room op de eerste volgende lege plek en dit wordt voor alle spelers geüpdatet.
Opruimen	Sluit de applicaties af

Test ID	16
Beschrijving	Een test voor het afsluiten van de host van de kamer
Requirements	Er is een room met een aantal spelers
Stappen	1. De Host sluit zijn applicatie af
Verwacht resultaat	Alle overige spelers krijgen een pop-up en keren terug naar de lobby.
Alternative Flow 1 – De host keert terug naar de lobby	In plaats van de applicatie af te sluiten drukt de host op back. Alle overige spelers krijgen nog steeds een error en keren terug naar de lobby.
Werkelijke resultaat	Alle overige spelers krijgen een pop-up en keren terug naar de lobby.
Alternative Flow 1 – De host keert terug naar de lobby	Alle overige spelers krijgen nog steeds een error en keren terug naar de lobby.
Opruimen	Sluit de applicaties af

Test ID	17
Beschrijving	Een test voor het afsluiten van een speler uit de kamer
Requirements	Er is een room met een aantal spelers
Stappen	1. Een speler die niet de host is sluit zijn applicatie af
Verwacht resultaat	De slot waar de speler in zat wordt open voor alle andere spelers.
Alternative Flow 1 – De host keert terug naar de lobby	In plaats van de applicatie af te sluiten drukt de speler op back. Voor alle overige spelers gaat de slot nog steeds open.
Werkelijke resultaat	De slot waar de speler in zat wordt open voor alle andere spelers.
Alternative Flow 1 – De host keert terug naar de lobby	Voor alle overige spelers gaat de slot nog steeds open.
Opruimen	Sluit de applicaties af

Test ID	18
Beschrijving	Een test of een spel kan worden gestart
Requirements	Er is een room met een aantal spelers
Stappen	1. De Host drukt op de start knop
Verwacht resultaat	Alle spelers gaan naar het spel scherm, er wordt een spel gestart en de instellingen van de room komen overeen met het spel.
Werkelijke resultaat	Alle spelers gaan naar het spel scherm, er wordt een spel gestart en de instellingen van de room komen overeen met het spel.
Opruimen	Sluit de applicaties af

Test ID	19
Beschrijving	Een test of de highscores goed geüpdatet worden
Requirements	Highscores is >0
Stappen	<ol style="list-style-type: none"> <li>1. Speel een spel en haal een hogere highscore</li> <li>2. Creëer een room</li> </ol>
Verwacht resultaat	Het aantal levens verandert naar het nieuwe aantal voor alle spelers
Alternative Flow 1 – behalen van lagere highscore	<ol style="list-style-type: none"> <li>1. Speel een spel en haal een lagere highscore</li> <li>2. Creëer een room</li> </ol>
Werkelijke resultaat	Het aantal levens verandert naar het nieuwe aantal voor alle spelers
Alternative Flow 1 – behalen van lagere highscore	<ol style="list-style-type: none"> <li>1. Speel een spel en haal een lagere highscore</li> <li>2. Creëer een room</li> </ol>
Opruimen	Sluit de applicaties af

## Old in-game tests

Om de oude functionaliteit te testen zijn er oude test cases met kleine veranderingen hergebruikt.

Test ID	20
Beschrijving	De naam-invoer testen
Stappen	<ol style="list-style-type: none"><li>1. Log in met een naam</li><li>2. Start een nieuw spel.</li></ol>
Verwacht resultaat	De naam staat op het scorebord, met het aantal levens eronder.
Werkelijke resultaat	De naam staat op het scorebord, met het aantal levens eronder.
Opruimen	Sluit de applicatie af

Test ID	21
Beschrijving	Een test of de bal goed wordt afgeweerd
Stappen	<ol style="list-style-type: none"><li>1. Start een nieuw spel</li><li>2. Blokkeer op een van de devices de bal met het batje.</li></ol>
Verwacht resultaat	De bal stuitert van het batje af.
Werkelijke resultaat	De bal stuitert van het batje af.
Opruimen	Sluit de applicatie af

Test ID	22
Beschrijving	Een test of het scoren goed wordt afgehandeld.
Stappen	<ol style="list-style-type: none"><li>1. Start een nieuw spel.</li><li>2. Laat op een van de devices een bal de linkerkant van het scherm verlaten.</li></ol>
Verwacht resultaat	De bal verdwijnt, de score onder je naam wordt met 1 verlaagd op zowel jouw device als de anderen en er wordt een nieuwe bal gereserveerd.
Werkelijke resultaat	De bal verdwijnt, de score onder je naam wordt met 1 verlaagd op zowel jouw device als de anderen en er wordt een nieuwe bal gereserveerd.
Opruimen	Sluit de applicatie af

Test ID	23
Beschrijving	Een test of een bal goed wordt doorgestuurd.
Stappen	<ol style="list-style-type: none"><li>1. Start een nieuw spel</li><li>2. Laat op een van de devices een bal de rechterkant van het scherm bereiken</li></ol>
Verwacht resultaat	De bal verdwijnt, en verschijnt bij de andere speler
Werkelijke resultaat	De bal verdwijnt, en verschijnt bij de andere speler
Opruimen	Sluit de applicatie af



Test ID	24
Beschrijving	Een test of een spel goed verloren kan worden.
Stappen	<ol style="list-style-type: none"> <li>1. Start een nieuw spel.</li> <li>2. Verlies het spel door al je levens te verliezen.</li> </ol>
Verwacht resultaat	Je krijgt het "Game Over" graphic te zien en komt naderhand in het hoofdmenu terecht.
Werkelijke resultaat	Je krijgt het "Game Over" graphic te zien en komt naderhand in het hoofdmenu terecht.
Opruimen	Sluit de applicatie af

Test ID	25
Beschrijving	Een test of een spel goed gewonnen kan worden.
Stappen	<ol style="list-style-type: none"> <li>1. Start een nieuw spel.</li> <li>2. Win het spel door als laatste levens over te hebben.</li> </ol>
Verwacht resultaat	Je krijgt het "You Win" graphic te zien en komt naderhand in het hoofdmenu terecht.
Werkelijke resultaat	Je krijgt het "You Win" graphic te zien en komt naderhand in het hoofdmenu terecht.
Opruimen	Sluit de applicatie af

Test ID	26
Beschrijving	Een test van het onverwacht afsluiten van een spel.
Stappen	<ol style="list-style-type: none"> <li>1. Start een nieuw spel.</li> <li>2. Sluit met alle devices behalve 1 het spel af.</li> </ol>
Verwacht resultaat	Het spel op de overgebleven device geeft het "You Win" graphic aan en komt naderhand in het hoofdmenu terecht.
Werkelijke resultaat	Het spel op de overgebleven device geeft het "You Win" graphic aan en komt naderhand in het hoofdmenu terecht.
Opruimen	Sluit de applicatie af

Test ID	27
Beschrijving	Een test van het onverwacht afsluiten van een spel.
Stappen	<ol style="list-style-type: none"> <li>1. Start een nieuw spel.</li> <li>2. Sluit met 1 device het spel af.</li> </ol>
Verwacht resultaat	Het spel gaat gewoon tussen de overige devices door maar de afgesloten speler is weg. Er wordt een nieuwe bal in het spel gebracht op één van de andere overgebleven devices.
Werkelijke resultaat	Het spel gaat gewoon tussen de overige devices door maar de afgesloten speler is weg. Er wordt een nieuwe bal in het spel gebracht op één van de andere overgebleven devices. Maar soms gaat er iets mis (netwerk?)
Opruimen	Sluit de applicatie af

Test ID	28
Beschrijving	Spel spelen
Stappen	<ol style="list-style-type: none"> <li>1. Start een nieuw spel.</li> <li>2. Beweeg je vinger over het scherm op en neer, beginnen bij de hoogte van de bat</li> <li>3. Beweeg de bat naar de locatie van een inkomende bal</li> <li>4. Beweeg de bat naar een locatie waar de inkomende bal niet komt</li> <li>5. Ga door totdat nog een speler levens overheeft</li> </ol>
Verwacht resultaat (na stap 4)	<ol style="list-style-type: none"> <li>4. De bat volgt je vinger wat betreft de hoogte</li> <li>5. De bal weerkaatst</li> <li>6. Het aantal levens van de huidige speler gaat één omlaag bij beide devices. Na een korte tijd verschijnt er een nieuwe bal bij de huidige speler</li> <li>7. Bij de speler waar de levens op zijn, verschijnt het bericht 'game over' en bij de andere speler verschijnt het bericht 'you win'. Na een korte tijd gaan beide weer terug naar het menu.</li> </ol>
Opruimen	Sluit de applicatie af

## New in-game tests

Test ID	29
Beschrijving	Test of het juiste aantal levens weergeven
Stappen	<ol style="list-style-type: none"><li>1. Start een nieuw spel met X levens</li><li>2. Bekijk het spelscherm en inspecteer het aantal levens</li></ol>
Verwacht resultaat	Elke speler heeft aan het begin van het spel X levens
Werkelijke resultaat	Elke speler heeft aan het begin van het spel X levens
Opruimen	Sluit de applicatie af

Test ID	30
Beschrijving	Test of levens kunnen verminderen
Stappen	<ol style="list-style-type: none"><li>1. Start een nieuw spel met 10 levens</li><li>2. Vang de eerste bal niet op met de bat</li><li>3. Bekijk het aantal levens van jezelf</li></ol>
Verwacht resultaat	Jouw levens staan op 9
Werkelijke resultaat	Jouw levens staan op 9
Opruimen	Sluit de applicatie af

Test ID	31
Beschrijving	Test of BatSize werkt
Stappen	<ol style="list-style-type: none"><li>1. Start een nieuw spel</li><li>2. Speel tot er een BatSize verschijnt</li><li>3. Vang de BatSize op</li><li>4. Bekijk je bat</li></ol>
Verwacht resultaat	Je bat is groter of kleiner geworden
Werkelijke resultaat	Je bat is groter of kleiner geworden
Opruimen	Sluit de applicatie af

Test ID	32
Beschrijving	Test of GenerateObstacle werkt
Stappen	<ol style="list-style-type: none"><li>1. Start een nieuw spel met één ander persoon</li><li>2. Speel tot op jouw scherm er een GenerateObstacle verschijnt</li><li>3. Vang op jouw scherm de GenerateObstacle op</li><li>4. Bekijk het veld van de andere persoon</li></ol>
Verwacht resultaat	Er is een obstakel op het veld verschenen van de andere persoon
Werkelijke resultaat	Er is een obstakel op het veld verschenen van de andere persoon
Opruimen	Sluit de applicatie af van zowel jezelf als de andere persoon

Test ID	33
Beschrijving	Test of Gravity werkt
Stappen	<ol style="list-style-type: none"> <li>1. Start een nieuw spel met één ander persoon</li> <li>2. Speel tot op jouw scherm er een Gravity verschijnt</li> <li>3. Vang op jouw scherm de Gravity op</li> <li>4. Bekijk het veld van de andere persoon</li> </ol>
Verwacht resultaat	De bal heeft een parallelle versnelling ten opzichte van de bat op het veld van de andere persoon
Werkelijke resultaat	De bal heeft een parallelle versnelling ten opzichte van de bat op het veld van de andere persoon
Opruimen	Sluit de applicatie af van zowel jezelf als de andere persoon

Test ID	34
Beschrijving	Test of NewBall werkt
Stappen	<ol style="list-style-type: none"> <li>1. Start een nieuw spel met één ander persoon</li> <li>2. Speel tot op jouw scherm er een NewBall verschijnt</li> <li>3. Vang op jouw scherm de NewBall op</li> <li>4. Bekijk het veld van de andere persoon</li> </ol>
Verwacht resultaat	Het veld van de andere persoon heeft een extra bal
Werkelijke resultaat	Het veld van de andere persoon heeft een extra bal
Opruimen	Sluit de applicatie af van zowel jezelf als de andere persoon

Test ID	35
Beschrijving	Test of de Speed werkt
Stappen	<ol style="list-style-type: none"> <li>1. Start een nieuw spel</li> <li>2. Speel tot er een Speed verschijnt</li> <li>3. Vang de Speed op</li> <li>4. Bekijk de ballen op je veld</li> </ol>
Verwacht resultaat	De ballen op je veld zijn versneld of vertraagd
Werkelijke resultaat	De ballen op je veld zijn versneld of vertraagd
Opruimen	Sluit de applicatie af

Test ID	36
Beschrijving	Test of het RandomDirectionChange werkt
Stappen	<ol style="list-style-type: none"> <li>1. Start een nieuw spel met één ander persoon</li> <li>2. Speel tot op het scherm van de andere persoon een RandomDirectionChange wordt gegenereerd door op jouw scherm een GenerateObstacle op te vangen</li> <li>3. Speel tot de bal van de andere persoon het RandomDirectionChange raakt</li> <li>4. Bekijk de richting van de bal op het veld van de andere persoon</li> </ol>
Verwacht resultaat	De bal op het veld van de andere persoon verandert van richting
Werkelijke resultaat	De bal op het veld van de andere persoon verandert van richting
Opruimen	Sluit de applicatie af

Test ID	37
Beschrijving	Test of FixedSpeedChange werkt
Stappen	<ol style="list-style-type: none"> <li>1. Start een nieuw spel met één ander persoon</li> <li>2. Speel tot op het scherm van de andere persoon een FixedSpeedChange wordt gegenereerd door op jouw scherm een GenerateObstacle op te vangen</li> <li>3. Speel tot de bal van de andere persoon het FixedSpeedChange obstakel raakt</li> <li>4. Bekijk de snelheid van de bal op het veld van de andere persoon</li> </ol>
Verwacht resultaat	De bal op het veld van de andere persoon is versneld
Werkelijke resultaat	De bal op het veld van de andere persoon is versneld
Opruimen	Sluit de applicatie af

### 10.1.1 Traceability Matrix: Requirements Testcases

	F1	F1.1	F2	F2.1	F2.2	F2.3	F3	F3.1	F4	F5
<b>ACTIVITY</b>										
TC1							X		X	
TC2			X			X	X		X	
TC3			X			X				
TC4	X	X		X		X				
TC5	X		X			X				
TC6			X		X	X				
TC7			X			X			X	
TC8			X			X			X	
TC9			X			X			X	
TC10			X			X			X	
TC11			X			X				
TC12			X			X				
TC13						X			X	
TC14			X			X			X	
TC15			X			X			X	
TC16										
TC17										
TC18	X		X	X		X				
TC19							X			
<b>GAME</b>										
<b>OULD</b>										
TC20	X									
TC21	X					X				
TC22	X			X						
TC23	X			X						
TC24	X									X
TC25	X									X
TC26	X									X
TC27	X									
TC28	X					X				
<b>NIEUW</b>										
TC29	X		X							
TC30	X		X							
TC31	X									
TC32	X	X								
TC33	X									
TC34	X									
TC35	X									
TC36	X	X								
TC37	X	X								

### 10.1.2 Traceability Matrix: Usecases Testcases

	UC 1	UC 2	UC 3	UC 4	UC 5	UC 6	UC 7	UC 8	UC 9	UC 10	UC 11	UC 12	UC 13	UC 14	UC 15
<b>ACTIVITY</b>															
TC1															
TC2	X														
TC3			X												
TC4			X												
TC5															
TC6														X	
TC7		X													
TC8				X											
TC9															
TC10					X										
TC11															
TC12															
TC13									X	X	X				
TC14							X	X							
TC15								X							
TC16															
TC17															
TC18						X									
TC19															
<b>GAME</b>															
<b>OUD</b>															
TC20															
TC21															

TC22															
TC23															
TC24															
TC25															
TC26															
TC27															
TC28												X			
<b>NIEUW</b>															
TC29															
TC30															
TC31											X				
TC32											X				
TC33											X				
TC34											X				
TC35											X				
TC36											X				X
TC37											X				X



## **11 Appendix D - User Manual**

OVPong

Gebruikershandleiding

# Installatie

## Systeemvereisten

- Android 3.0 Honeycomb (API level 11)
- Internetverbinding

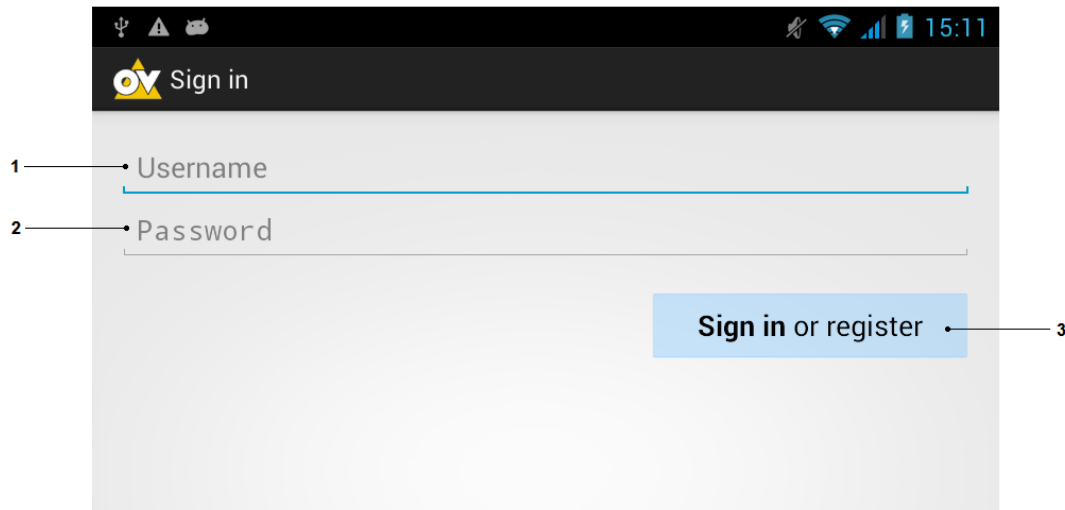
## Stappen

<b>1</b>	Download de OVPong.apk op het device waar het geïnstalleerd moet worden.
<b>2</b>	Ga naar "Downloads" in de applicatielijst.
<b>3</b>	Druk op de OVPong.apk.
<b>4</b>	Selecteer de "Package installer" en installeer OVPong.

# Spelen

## Inloggen

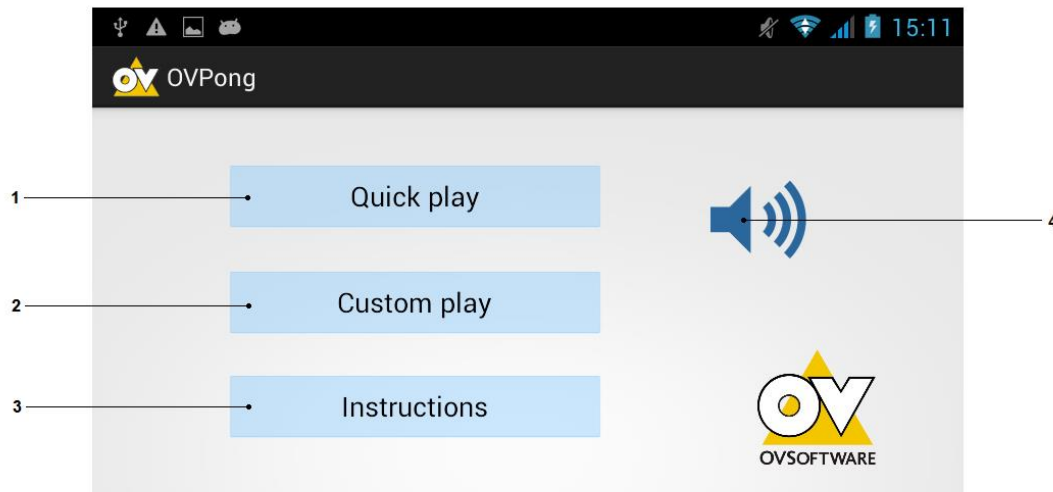
Als je OVPong opstart, dan wordt er een loginscherm getoond. Er is een account nodig om dit spel te kunnen spelen. Het account kan in dit scherm worden aangemaakt door het invoeren van een unieke gebruikersnaam in het vakje met "Username" en een wachtwoord in het vakje met "Password". De gebruikersnaam mag maximaal 10 tekens lang zijn en het wachtwoord moet minimaal 4 tekens lang zijn. Vervolgens moet er op "Sign in or register" gedrukt worden. Mocht dit goed verlopen, dan zal het ook meteen het account inloggen en doorgaan naar het Menu. Het inloggen verloopt hetzelfde als het account aanmaken, maar dan met een al bestaande account.



Num.	Uitleg
1	Het tekstvak waar de gebruikersnaam dient ingevoerd te worden.
2	Het tekstvak waar het wachtwoord dient ingevoerd te worden.
3	Dit is de knop om met de ingevulde gegevens in te loggen of om een nieuwe account te creëren.

# Menu

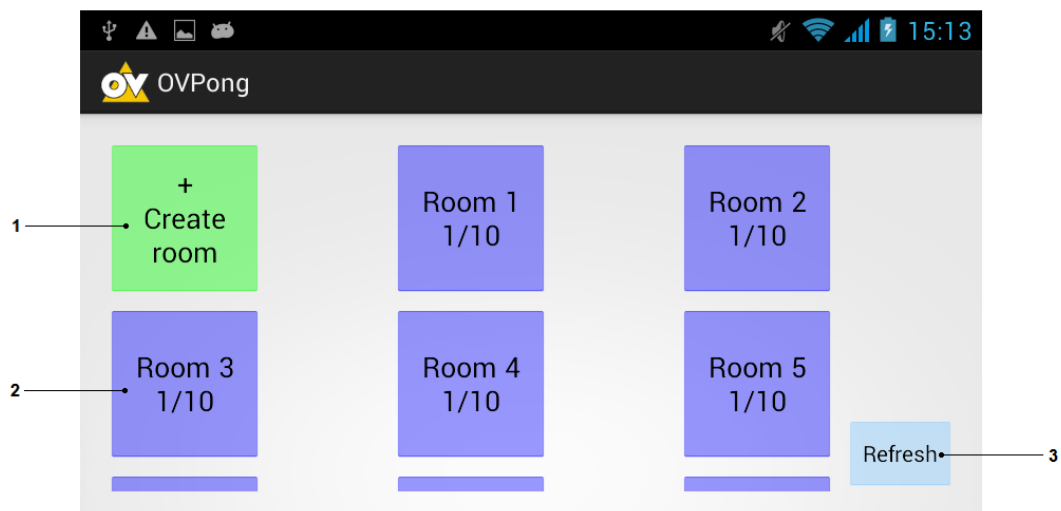
Na het inloggen verschijnt het menuscherm. Vanuit hier kan je een quick play game starten, naar de lobby gaan, het instructiescherm tevoorschijn halen en het geluid aan/uit doen.



Num.	Uitleg
1	Een quick play game is een spel waar je 1 tegen 1 speelt met 1 bal en 3 levens. Na het drukken op de “Quick play”-knop word je in de wachtrij gezet totdat er een tweede speler wordt gevonden. De wachtrij kan verlaten worden door op de back-knop, “Custom play”-knop of “Instructions”-knop te drukken.
2	“Custom play” zal je sturen naar de lobby waar je met meerdere andere spelers kan samenspelen (t/m 10 spelers in 1 spel).
3	“Instructions” zal de instructies van het spel tevoorschijn halen, daar vind je wat power-ups doen.
4	Een druk op dit geluidicoon zal het geluid aan/uit zetten.

# Lobby

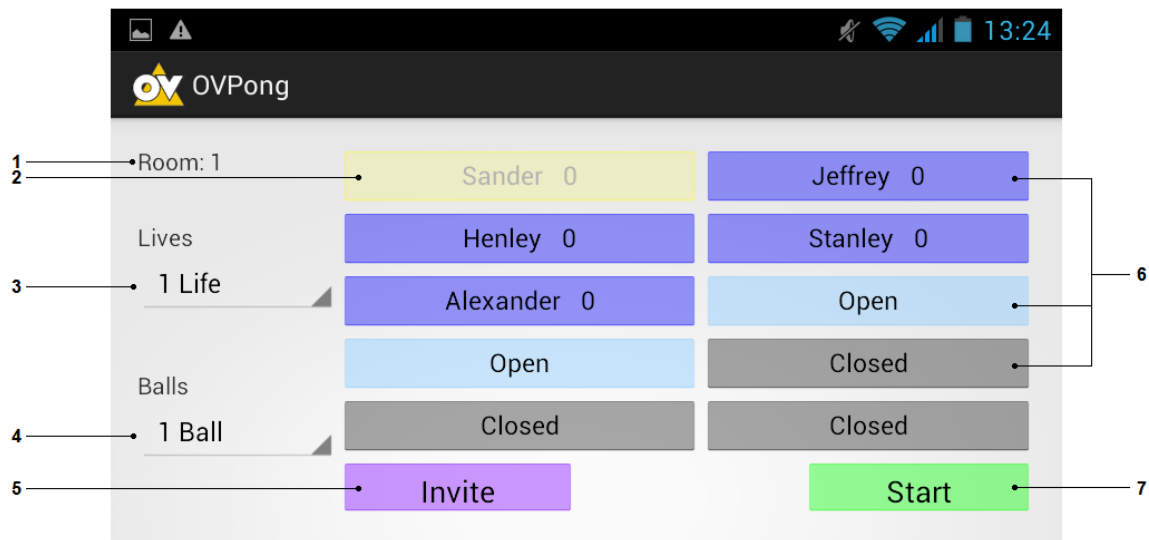
De lobby is een plaats waar spelers kamers kunnen maken en toetreden. Een kamer is een ruimte waar spelers verzamelen om samen een spel te spelen. In dit scherm zie je een kamerlijst, als er meer kamers zijn dan het scherm kan tonen, kan je over het scherm swipen om de andere kamers te zien.



Num.	Uitleg
1	Als er op deze knop gedrukt wordt, zal er een nieuwe kamer worden aangemaakt met jou als host.
2	Een kamer die toegetreden kan worden, als de kamer niet vol is.
3	Knop om de lijst met kamers te vernieuwen.

## Kamer

Een kamer heeft 10 kamerslots waar één host en 9 andere spelers in kunnen. De host, de speler die de kamer heeft gemaakt, heeft speciale rechten in deze kamer en is aangegeven met de gele kamerslot. Deze speler kan andere spelers uitnodigen, spelers verwijderen, kamerslots openen/sluiten, spelinstellingen aanpassen en het spel starten. Een kamerslot wordt weergegeven met “Open” als het open is, “Closed” als het dichtgemaakt is door de host of met de naam van een speler en de highscore van deze speler als er een speler aanwezig is.



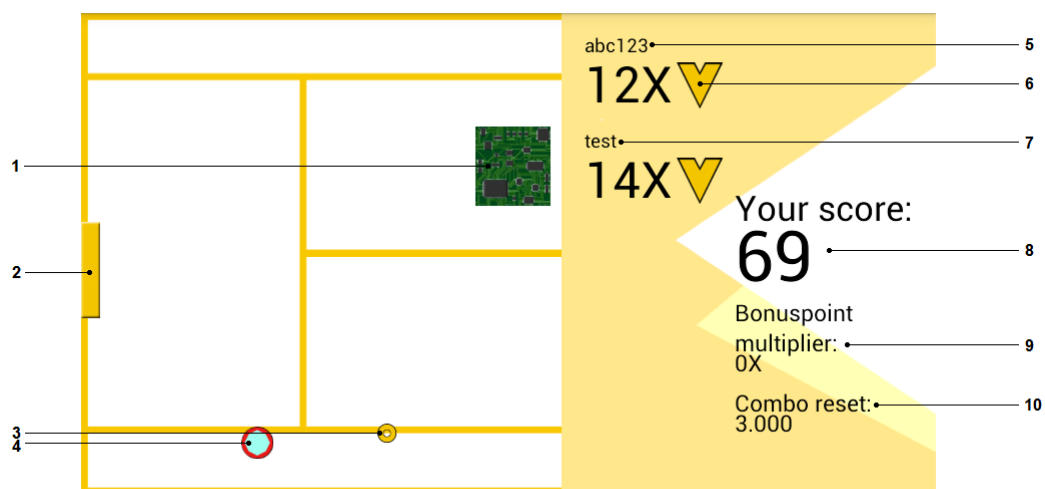
Num.	Uitleg
1	Geeft het kamernummer van deze kamer weer.
2	De host van de kamer.
3	Het aantal levens dat is ingesteld voor deze kamer, de host kan dit veranderen door erop te drukken en vervolgens het gewenste aantal levens te drukken.
4	Het aantal ballen dat is ingesteld voor deze kamer, de host kan dit veranderen door erop te drukken en vervolgens het gewenste aantal ballen te drukken.
5	De host kan met deze knop spelers uit de lobby uitnodigen. Als er op deze knop gedrukt wordt verschijnt er een pop-up waar een naam ingevoerd moet worden.

<b>6</b>	Een kamerslot waar een speler in kan als het open is. De host kan erop drukken om het te sluiten als het open is, het te openen als het gesloten is en om een speler te verwijderen als een speler het bezet.
<b>7</b>	Knop om het spel te starten met de huidige spelers in de kamer. Alleen te gebruiken door de host.



# Spel

Op het spelscherm is een speelveld te zien. Helemaal links op het speelveld heb je een batje die je kan besturen. Het is de bedoeling om ballen tegen te houden met dat batje. Naast ballen komen ook power-ups en obstakels op het speelveld. Power-ups verplaatsen zich richting naar links. Deze kunnen opgepakt worden door ze op te vangen met het batje en zullen een bepaald effect geven. Informatie over de effecten van power-ups zijn te vinden in het instructiescherm. Obstakels kunnen op het speelveld verschijnen als een tegenstander een obstakel gerelateerde power-up oppakt. Er zijn 2 soorten obstakels. Één soort verandert de snelheid van de bal, de andere soort verandert de richting van de bal. Rechts van het speelveld zie je de spelers en de bijbehorende levens. Ook zijn je score, bonuspoint multiplier en de combo reset te zien. Een speler is af als het geen levens meer heeft. Het spel is afgelopen als er 1 speler over blijft in het spel.



Num.	Uitleg
1	Een obstakel.
2	Het batje dat bestuurd kan worden. Het batje gaat met de vinger mee in de verticale richting als de vinger op het scherm gedrukt wordt gehouden.
3	De bal die dient tegengehouden te worden met het batje.

<b>4</b>	Één van de power-ups die opgepakt kan worden.
<b>5</b>	Hier wordt de naam van een speler weergegeven.
<b>6</b>	Het aantal levens dat hoort bij de spelernaam die erboven staat.
<b>7</b>	De namen en levens van eventuele meerdere spelers.
<b>8</b>	Je huidige score. Als je af bent dan komt er nog een multiplier over deze score heen dat afhangt van hoe goed je gespeeld hebt.
<b>9</b>	Hier wordt je eventuele multiplier bijgehouden. Als je meerdere ballen moet raken binnen 3 seconden, dan krijg je extra punten door deze multiplier. Deze multiplier hangt af van je combo en wordt gereset als er 3 seconden geen bal wordt geraakt.
<b>10</b>	De timer voor de multiplier. Als deze 0 is, wordt de combopoint multiplier gezet naar 0X.