

Ontwerpdocument Stuiterballen

Stan van Lier, Tom Evers & Sven Arissen

21 mei 2013

Inhoudsopgave

1	Inleiding	3
2	Verantwoording	3
2.1	Entertainment factor	3
2.2	Utility factor	3
2.3	Usability factor	3
2.4	Funtionality factor	4
3	Requirements	4
3.1	Use-case model	4
3.2	Use-cases/Functionele requirements	5
3.3	Niet-functionele requirements	7
4	Globaal Ontwerp	8
4.1	Bal	8
4.2	GameModel	8
4.3	GameView	8
4.4	GameActivity	9
4.5	StartActivity	9
4.6	HighscoreActivity	9
4.7	OptionsActivity	9
4.8	Powerup	9
4.9	Obstakel	9
4.10	Opmerkingen	10
5	Gegevens ontwerp	10
5.1	Speler	10
5.2	Bal	10
5.3	Handen	10
5.4	Omgeving	11
5.5	Apparaat	11
6	Detail-ontwerp	12

7	Gebruikersinterface	13
7.1	Menu scherm	13
7.2	Speel scherm	14
7.3	Highscores scherm	15
7.4	Opties scherm	16
8	Planning	16
8.1	Projectweek 5	17
8.2	Projectweek 6	17
8.3	Projectweek 7	17
8.4	Projectweek 8	17

1 Inleiding

Aan het begin van het project kwam Stan met het geniale idee om een app te maken die eigenlijk volkomen nutteloos was, maar gewoon leuk om als tijdverdrijf mee te klooiën. Het kwam neer op een stuiterbal: door te tablet of telefoon te kantelen verander je de zwaartekracht voor een bal, die dan begint te stuiteren. Op zich een leuk idee, natuurlijk, maar we wilden er iets meer van. Daarom besloten we er een spelletje van te maken. Als speler moet je de bal door een soort doolhof of grot heen navigeren, zonder af te gaan. Je zou dan af gaan als je er te lang over doet, of bepaalde obstakels raakt. Iedereen moet het kunnen spelen, dus we moeten het vrij simpel houden. Ook moet de grot eindeloos doorgaan, zodat je voor uren aan speelplezier hebt. Verslaafde gebruikers zijn altijd mooi meegenomen, natuurlijk.

2 Verantwoording

Uit ervaring weten we dat simpele apps en spelletjes vaak het meest worden gedownload/gekocht. Denk daarbij bijvoorbeeld aan Doodle Jump, Helicopter Game en Jetpack Joyride, spelletjes die in essentie lijken op wat wij van plan zijn. Net als bij ons hebben die spellen een eindeloze map en wordt het steeds moeilijker om verder te komen. In tegenstelling tot deze spellen hebben wij echter geen statische richting: bij onze app maak je echt gebruik van de zwaartekracht om in het spel een zwaartekracht te simuleren. Dit principe wordt niet vaak gebruikt, omdat dat heel specifiek voor handheld-apparaten bedoeld is. Het probleem van een spel als doel, is het komen met een goed idee. Gelukkig viel het bedenken van een leuk spel erg mee, zoals ook al uiteengezet in de inleiding. Dit spel verschilt wezenlijk van de grote en bekende namen op de markt, en zou wel eens een grote hit kunnen worden.

2.1 Entertainment factor

- Spelelement
- Uitdaging: jezelf en anderen verbeteren, highscores
- Veel bewegen met de tablet of telefoon

2.2 Utility factor

- Gebruiker vermaken

2.3 Usability factor

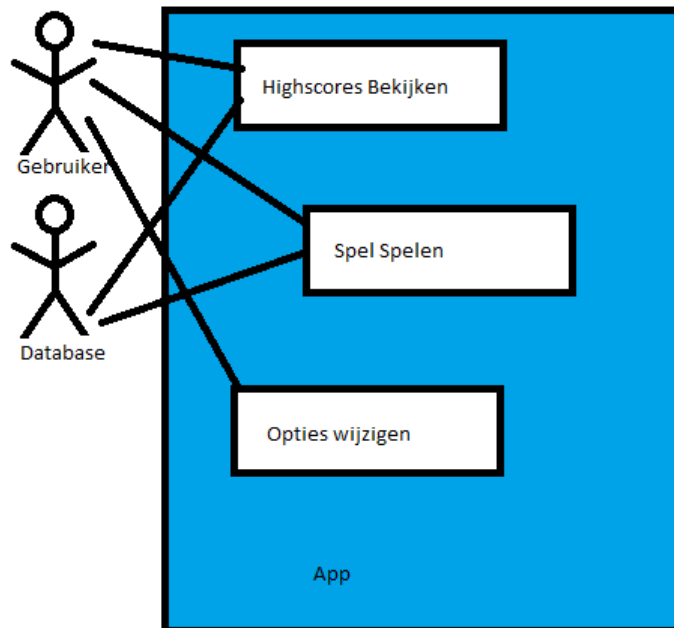
- Reageert direct op motion sensor
- Simpel spel: weinig bugs

2.4 Funtionality factor

- Motion sensor wordt actief gebruikt

3 Requirements

3.1 Use-case model



3.2 Use-cases/Functionele requirements

Naam: Spel spelen
Actors: Gebruiker, Database
Beschrijving: De gebruiker gaat het spel zelf spelen
Basic Course of Events: <ol style="list-style-type: none">1. De gebruiker geeft aan dat hij wil beginnen2. Het systeem toont het spel3. De gebruiker speelt het spel tot hij verliest4. Het systeem geeft de score aan5. Het systeem vraagt aan de database of deze score hoog genoeg is voor de highscores en waar hij dan moet staan6. De database geeft aan dat de score hoog genoeg is en op welke plek7. Het systeem voegt de score op die plek toe8. De gebruiker sluit de app
Alternative Path: De gebruiker sluit bij stap 8 het spel niet, maar gaat terug naar het startscherm Bij stap 6 kan de database ook aangeven dat de score niet hoog genoeg is, dan wordt stap 7 overgeslagen
Trigger: De gebruiker kiest Start op het menu scherm van de app
Preconditions: De smartphone staat aan met de app opgestart
Postconditions: De gebruiker heeft het spel 1 keer gespeeld

Naam: Highscores bekijken
Actors: Gebruiker, Database
Beschrijving: De gebruiker bekijkt de highscores
Basic Course of Events: <ul style="list-style-type: none"> 1. De gebruiker geeft aan dat hij de highscores wil bekijken 2. het systeem vraagt de highscores aan de database 3. De database geeft de highscores 4. Het systeem toont de highscores 5. De gebruiker sluit de app
Alternative Path: <p>De gebruiker sluit bij stap 5 het spel niet, maar gaat terug naar het startscherm</p>
Trigger: De gebruiker kiest Highscores op het menu scherm van de app
Preconditions: De smartphone staat aan met de app opgestart
Postconditions: De gebruiker heeft de highscores gezien

Naam: Opties wijzigen
Actors: Gebruiker
Beschrijving: De gebruiker gaat een optie wijzigen
Basic Course of Events: <ol style="list-style-type: none"> 1. De gebruiker geeft aan dat hij de opties wil zien 2. Het systeem toont de opties 3. De gebruiker selecteert de optie die hij wil veranderen 4. Het systeem toont de optie die verandert wordt 5. de gebruiker geeft aan hoe hij die optie wil veranderen 6. het syteem verandert de optie 7. De gebruiker sluit het spel
Alternative Path: De gebruiker sluit bij stap 7 het spel niet, maar gaat terug naar het startscherm
Trigger: De gebruiker kiest Opties op het menu scherm van de app
Preconditions: De smartphone staat aan met de app opgestart
Postconditions: De gebruiker heeft 1 optie verandert

3.3 Niet-functionele requirements

- De app moet meerdere schermen ondersteunen.
- De app moet zowel in Nederlands als Engels te bekijken zijn.
- Het spel moet vloeiend runnen.

4 Globaal Ontwerp

4.1 Bal

Responsibilities	Collaborations
Doen:	
Berekenen hoe ver de bal over de x/y as bewogen is	
De bal van een vlak af laten stuiteren	
Weten:	
De op dit moment actieve powerup	
De massa van de bal	
De zwaartekracht op de bal	

4.2 GameModel

Responsibilities	Collaborations
Doen:	
Berekenen hoe ver de bal bewogen is over de x/y as	Bal
Controleren of de bal niet moet stuiteren	
De bal laten stuiteren	Bal
Controleren of de bal een obstakel of een powerup raakt	
De powerup uitvoeren	Powerup
Het obstakel uitvoeren	Obstakel
Bepalen welk deel van het scherm door de nieuwe positie van de bal wegvalt en wat er nieuw bijkomt	
Een nieuw willekeurig gegenereerd stuk scherm berekenen	
Weten:	
De positie van de bal	
De positie van de obstakels die op dit moment op het scherm zijn	
De positie van de muren die op dit moment op het scherm zijn	
De positie van de powerups die op dit moment op het scherm zijn	

4.3 GameView

Responsibilities	Collaborations
Doen:	
Het scherm tekenen	GameModel
Weten:	
De grote van het scherm	
De positie van de bal	
De positie van de muren	
De positie van de powerups	
De positie van de obstakels	

4.4 GameActivity

Responsibilities	Collaborations
Doen:	
De GameActivity opstarten en de GameView eraan toevoegen	
Weten:	
De hoek waaronder de gebruiker de smartphone houdt	

4.5 StartActivity

Responsibilities	Collaborations
Doen:	
De StartActivity opstarten	
Na input de GameActivity of de HighscoreActivity of de OptionsAcitivty starten	
Weten:	

4.6 HighscoreActivity

Responsibilities	Collaborations
Doen:	
De HighscoreAcitivity opstarten en de highscore gegevens uit de sql database halen	
Weten:	
De sql-database	

4.7 OptionsActivity

Responsibilities	Collaborations
Doen:	
De OptionsActivity opstarten en de info op dit moment uit de sharedpreferences halen	
Wachten op userinput en aan de hand daarvan de sharedpreferences wijzigen	
Weten:	

4.8 Powerup

Responsibilities	Collaborations
Doen:	
De powerup uitvoeren	
Weten:	

4.9 Obstakel

Responsibilities	Collaborations
Doen:	
Het obstakel uitvoeren	
Weten:	

4.10 Opmerkingen

De `Obstakel` en `Powerup` klasse zijn uiteraard interfaces, omdat we nog niet precies weten welke opstakels we willen gebruiken en dat ook eerst zullen moeten testen voordat we daar definitieve beslissingen over nemen, hebben we de details daarvan nog niet hierin opgenomen. De implementaties van die powerups en obstakels zal ook verder niets aan de interface toevoegen behalve het effect van de powerups en obstakels.

5 Gegevens ontwerp

5.1 Speler

- **Naam speler**
Variabele van `Speler`. Deze wordt gebruikt bij het opslaan van de high-scores.
- **Score**
Variabele van `Speler`. Hierin wordt bijgehouden hoever de speler is in het spel. Als het spel is afgelopen wordt dit opgeslagen in de highscores.

5.2 Bal

- **Straal**
Variabele van `Bal`. Hier wordt de grote van de bal mee bepaald.
- **Stuiterkracht**
Variabele van `Bal`. Hier wordt mee bepaald hoe hard de bal stuiteret.
- **Snelheid**
Variabele van `Bal`. De snelheid die de bal op het moment heeft.
- **Richting**
Variabele hoek van `Bal`. De richting die de bal op het moment heeft.
- **Positie**
Variabelen x-positie en y-positie van `Bal`. De positie die de bal in de grot heeft op het moment.

5.3 Handen

- **Snelheid**
Variabele van `Handen`. De snelheid waarmee de handen zich bewegen.
- **Positie**
Variabelen x-positie en y-positie van `Handen`. De positie die de handen in de grot hebben op het moment.

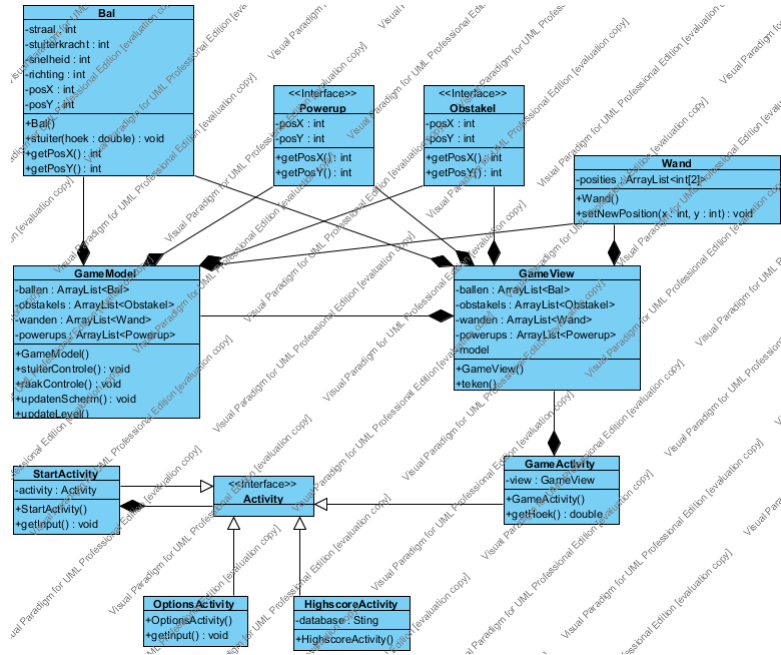
5.4 Omgeving

- Contouren van de grot
Bestaat uit twee objecten, Bovenkant en Onderkant.
Dit geldt voor ieder van deze objecten:
Rij van variabelen x-positie en y-positie, tussen de variabele oplopend in de rij wordt een lijn getrokken. Deze lijn wordt getoond zodat de contouren van de grot zichtbaar zijn. Als de bal hier tegenaan komt wordt zijn Snelheid en Richting veranderd ten opzichte van de hoek van deze lijn en de Snelheid, Richting en Stuiterkracht die de bal heeft.
- Obstakels (optioneel)
Verschillende objecten:
Ieder object heeft een variabele voor de x-positie en y-positie in de grot en een methode die uitgevoerd wordt als het obstakel geraakt wordt door een bal. Als het obstakel geraakt wordt door een bal zal ook hetzelfde gebeuren als wat er gebeurt als de bal de contouren van de grot raakt.
- Powerups (optioneel)
Verschillende objecten:
Ieder object heeft een variabele voor de x-positie en y-positie in de grot en een methode die uitgevoerd wordt als de powerup geraakt wordt door een bal.
- Zwaartekracht
Bestaat uit twee variabelen, zwaartekracht en zwaartekracht-positie. De bal zal met de zwaartekracht aangetrokken worden naar de zwaartekracht-positie.
- Beeld
Variabelen x-positie en y-positie van waar het beeld zich bevind, en variabelen breedte en hoogte van de afmetingen van het beeld. Alles wat binnen de breedte en hoogte vanaf de x-positie en y-positie valt zal te zien zijn.

5.5 Apparaat

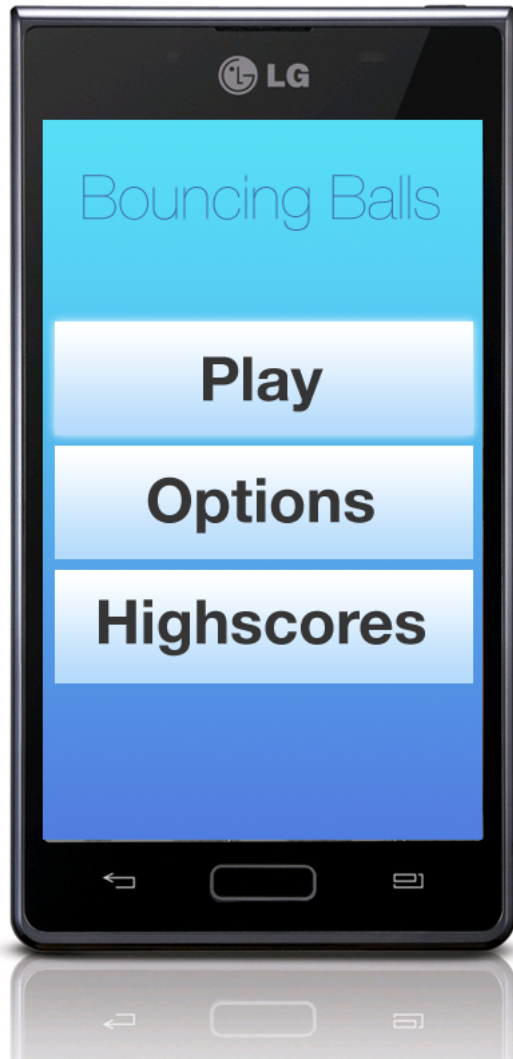
- Hoek
Variabele kantel-richting, de hoek waarmee de mobiel gekanteld is. Hiermee wordt de zwaartekracht-positie van de Omgeving en de richting van de Bal veranderd.

6 Detail-ontwerp

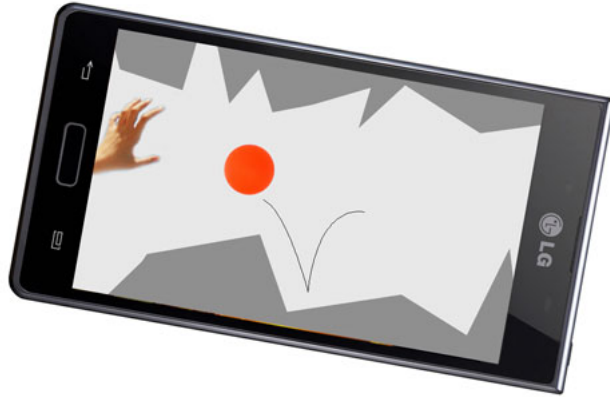


7 Gebruikersinterface

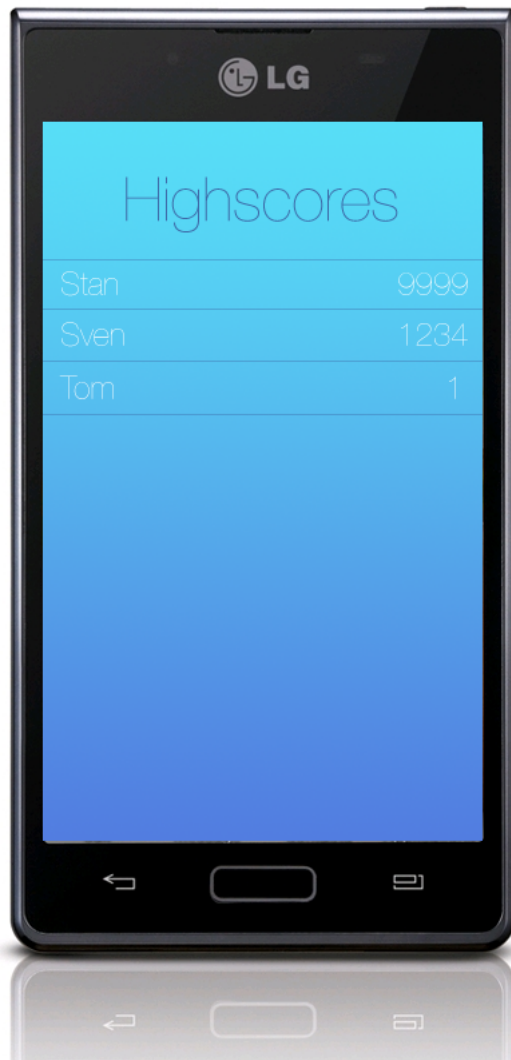
7.1 Menu scherm



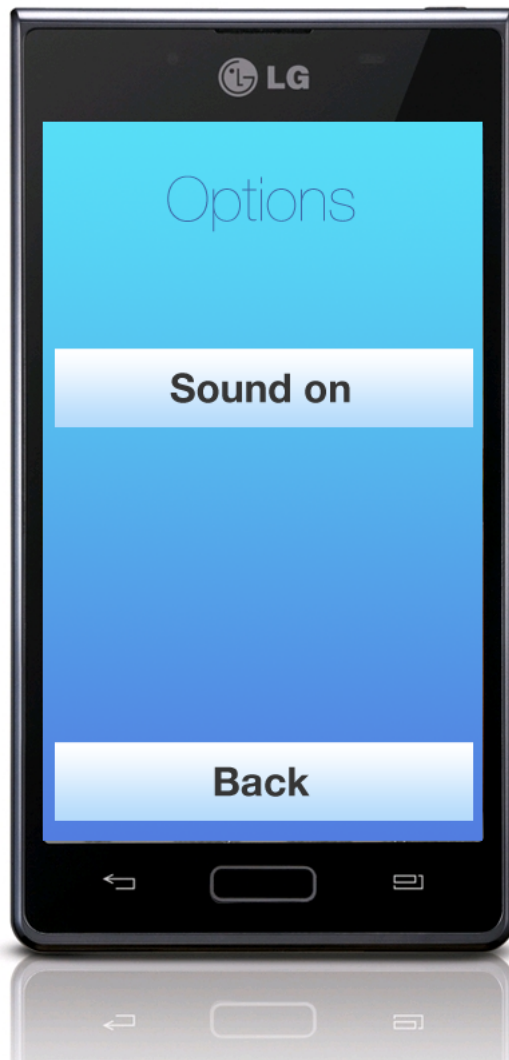
7.2 Speel scherm



7.3 Highscores scherm



7.4 Opties scherm



8 Planning

We hebben toch wel redelijk wat lastige onderdelen in dit project:

- De highscores bijhouden - database
- Hoe laten we de bal goed stuiten - physics en gebruikmaken van de motion sensor

- Hoe laten we het level eindeloos doorgaan - random level rendering
- De bal mag niet te ver terug stuiteren, want dan ben je af - grot vergeten voorbij een bepaalde afstand terug
- Powerups en vijanden/obstakels - hoe houden we balance

8.1 Projectweek 5

- Menu/gebruikersinterface
- Highscores

8.2 Projectweek 6

- Stuiteren: physics en motionsensor in sandbox

8.3 Projectweek 7

- Levelrendering: de grot
- Powerups/enemies

8.4 Projectweek 8

- Uitloop, afronding
- Inleveren