

Ontwerp: Rolling Stone

Lars Bade (4051513|IC)
Gerhard Klassen (4121538|IC)
Moritz Neikes (4099095|IC)
Tobias Schröter (4122283|IC)

13. Mei 2012

Inhoudsopgave

1	Inleiding	2
1.1	Wat is “Rolling Stone”?	2
1.2	Voor wie is “Rolling Stone”?	2
1.3	Wat heb je nodig?	2
2	Requirements	3
2.1	Functionele eisen	3
2.1.1	Het menu	3
2.1.2	Het spel	3
2.1.3	Webdatabase voor highscores	4
2.2	Niet-functionele eisen	4
2.2.1	Veiligheid	4
2.3	Use-case model en use-cases	4
2.3.1	Hoofdmenu	4
2.3.2	Speel	5
2.3.3	Neigen	5
3	Ontwerp	7
3.1	Globaal ontwerp	7
3.1.1	Controller	7
3.1.2	Model	8
3.1.3	View	8
3.2	Gegevensontwerp	8
3.2.1	Blok	8
3.2.2	Layer	9
3.2.3	Level	9
3.2.4	Levelset	9
3.2.5	Bal	9
3.2.6	Game	9
3.2.7	EindVanLevelDialog	9
3.2.8	Opslaan	10
3.2.9	Highscores	10
3.3	Detail-Ontwerp	10
3.4	GUIs	10

3.5	Planning	10
3.5.1	Week 1	10
3.5.2	Week 2	11
3.5.3	Week 3	11
3.5.4	Week 4	11
3.5.5	Prototype	11

Hoofdstuk 1

Inleiding

1.1 Wat is “Rolling Stone”?

“Rolling Stone” is een app voor Android systemen. Het is een spel, waar je een bal door nigen van je telefoon door een doolhof moet navigeren. Het spel bestaat uit verschillende levels die speelbaar worden als je het vorige level hebt opgelost. Ieder level bestaat uit meerdere vlakten tussen welke je door portalen in de doolhof kunt springen. Bovendien zijn er verschillende speelelementen ingebouwd, zoals schakelaars die deuren openen en diverse vallen, zodat je handigheid moet bewijzen om een level op te lossen. Afhankelijk van de tijd die je nodig hebt om een level op te lossen heb je de mogelijkheid in de globale highscore-lijst te staan te komen. Voor de beste spelers zijn er verborgen bonussen, die jouw actuele tijd verminderen als je ze inzamelt.

1.2 Voor wie is “Rolling Stone”?

“Rolling Stone” is een app voor jong en oud. Iedereen die zin heeft in een klein spelletje tussendoor en een beetje geduld en handigheid heeft kan zich met dit spelletje bezig houden.

1.3 Wat heb je nodig?

- Android-Systeem met niggingsensor (Clinometer)
- Android 2.3 of hoger

Hoofdstuk 2

Requirements

2.1 Functionele eisen

2.1.1 Het menu

In het menu kan de speler tussen de verschillende vensters kiezen. Deze zijn:

- Het spel
 - overzicht met levels (waarvan sommige speelbaar zijn)
- About

2.1.2 Het spel

Besturing

Als men het telefoon in een bepaalde richting nijgt, versnelt de bal in deze richting.

Spelelementen

- Schakelaars (om deuren te openen)
- Deuren (kunnen door schakelaars geopend worden)
- Verschillende vallen (gaten, etc.)
- Portalen (om tussen de vlakten van een level te wisselen)
- Bonusstenen (om de tijd te verminderen en dus de score te verbeteren)
- Doel

Doel

Doel van het spel is alle level in een zo kort mogelijk tijd te absolveren.

Spelafloop

Ieder level heeft een bepaald startpunt en een doel. Om het level te absolveren moet men de bal naar het doel navigeren. Het is mogelijk dat er meer dan een weg naar het doel is. De speler bepaald zelf welke weg bij gebruikt. Sommige wegen zijn dankzij deuren niet meteen vrij, maar moeten eerst met schakelaars passerbar gemaakt worden. Daarbij bestaat ieder level mogelijk uit meerdere vlakten tussen welke de speler met behulp van portalen kan springen. Op weg naar het doel zijn verschillende vallen welke niet aangeraakt mogen worden. Als dit gebeurt moet de speler het level opnieuw beginnen.

Highscores

Tijdens het spelen van een level wordt de tijd gemeten die de speler nodig heeft om het level op te lossen. Deze tijd wordt aan het eind van een spel doorgegeven aan een webdatabase waar deze tijd opgeslagen kan worden als de speler dat wil. De highscores worden per level opgeslagen.

2.1.3 Webdatabase voor highscores

Er bestaat een webdatabase waar de highscores opgeslagen worden. Deze webdatabase draait op een server. De interactie tussen database en applicatie gebeurt via een php-script.

2.2 Niet-functionele eisen

2.2.1 Veiligheid

Het is van belang, dat de applicatie veilig is. Daarbij behoort dat ze geen toestemming voor functies van het telefoon nodig heeft, die niet noodzakelijk zijn om de functionaliteit van het spel mogelijk te maken. De highscores worden in een webdatabase opgeslagen en moeten dus ook beveiligd worden, zodat het niet mogelijk is om op de een of ander manier highscores te vervalsen.

2.3 Use-case model en use-cases

2.3.1 Hoofdmenu

De gebruiker heeft de app opgestart. Hij ziet het hoofdmenu met de volgende menupunten:

- New Game
- About

Als hij op een van de buttons drukt wordt de bijhorende use case uitgevoerd.

New Game

De gebruiker ziet een overzicht over een aantal levelpacks. Als hij de terug-knop op zijn mobieltje drukt dan komt hij weer in het hoofdmenu terecht. Dus wordt de Hoofdmenu-usecase uitgevoerd. Als hij een pack kiest, ziet hij vervolgens een overzicht over alle beschikbare levels erin. Hij ziet ook welke ervan al speelbaar zijn en welke hij nog moet vrijschakelen. Als hij op dit moment op de terug-knop drukt dan ziet hij weer de overzicht over de verschillende packs. Als hij op een van de levels drukt dan krijgt hij een dialoog te zien waarin verschillende informatie over het gekozen level staan. Hij ziet de eerste drie posities in de highscore-lijst op de server en zijn persoonlijke beste tijd. Er zijn twee buttons: Een terug-button waarmee de gebruiker naar de leveloverzicht terug kan gaan en een play-button waarmee de gebruiker het level kan opstarten. In het laatste geval wordt de Speel-usecase uitgevoerd.

About

De gebruiker krijgt een info-pagina te zien waarop de naam en versie van de app staat gevolgd van onze groepsnaam en onze namen. Als de gebruiker op de terug-knop drukt dan komt hij weer in het hoofdmenu terecht.

2.3.2 Speel

Het gekozen level wordt opgestart. Dit wordt binnen de Speel.Start usecase afgehandeld. Nu kan de gebruiker de bal bewegen zoals boven beschreven. Daarbij wordt de Neigen-usecase uitgevoerd.

Speel.Start

De bal wordt op de startpositie gezet en de timer, die de benodigde tijd bijhoudt, wordt opgestart. De gebruiker ziet het vlak van het level waarop de bal ligt.

Speel.Eind

Als het doel van een level is bereikt wordt de tijd gestopt en de gebruiker krijgt een dialoogvenster te zien. Op deze dialoog staat de tijd van de gebruiker die hij in deze level nodig had. Deze tijd kan hij opslaan in de highscorelijst op onze server. In dit geval wordt de beste tijd ook lokaal op het mobieltje opgeslagen. Als de gebruiker geen highscore naar de server stuurt, wordt de tijd ook niet lokaal opgeslagen.

Verder kan de gebruiker het level herhalen, naar het menu teruggaan of met het volgende level doorgaan.

2.3.3 Neigen

De Clinometer geeft de huidige neiging van het mobieltje aan. Die wordt als snelheid aan de bal doorgegeven. Vervolgens wordt de interactie met het level

uitgevoerd. Er bestaan de volgende mogelijkheden:

- De bal bereikt het doel
- De bal komt op een portaal
- De bal komt op een schakelaar
- De bal komt op een bonussteen
- De bal rolt van het speelbord

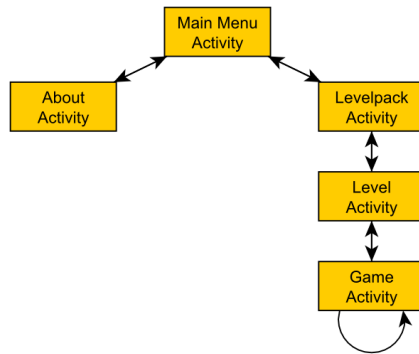
Als de bal op een van deze velden terecht komt wordt een bepaalde actie uitgevoerd. In het bijzonder wordt de `Speel.Eind` usecase uitgevoerd als de bal het eind bereikt.

Hoofdstuk 3

Ontwerp

3.1 Globaal ontwerp

De applicatie is onderverdeelt in vijf activities. In 3.1 is een overzicht te zien, hoe de gebruiker tussen de verschillende activiteiten kan navigeren. We gaan



Figuur 3.1: Overzicht over de activities

hier niet verder erop in, hoe precies de menus worden gerealiseerd.

De Game-activity vormt de kern van de applicatie. De architectuur van deze activity is strikt onderverdeelt naar het MVC-principe. 3.2 laat een overzicht zien, welke componenten er bestaan, bij welke groep ze behoren en hoe ze interageren.

3.1.1 Controller

De touch-controller reageert op alle touch-events die er optreden. Verder reageert de touch-controller ook als de gebruiker één van de buttons drukt. Als er

een event gebeurt, stuurt de touch-controller een bericht aan de game-component in het model.

De gravity-controller reageert op veranderingen van de neigingssensor in het mobieltje. Als zich de neiging verandert, stuurt de controller een bericht aan de bal in het model en verandert daardoor de snelheid van de bal in een bepaalde richting.

De ticker loopt in een eigen thread en roept regelmatig de bal aan, die dan een stukje probeert te rollen.

3.1.2 Model

De game-activity vormt de hoofd van het hele model en is dus voor een veeltal aan dingen verantwoordelijk. Het maakt het level en de bal aan en houdt met een timer de benodigde speeltijd bij. Het game biedt ook een mogelijkheid voor blokken om speciale acties uit te voeren. Als de bal het doel van het level bereikt, opent het game de einddialoog. Verder kan het game op een touch-event reageren. In zo een geval wordt het spel gepauzeerd of voortgezet en een pauze-dialoog wordt geopend.

De bal reageert op de gravity-controller en de ticker. Als zich de neiging verandert, wordt daardoor de snelheid van de bal verandert. Als de ticker de bal aanroept, probeert de bal met de gegeven snelheid in een bepaalde richting te rollen. Om te bepalen of de plek waar de bal naartoe wil rollen, vrij is, vraagt de bal het level, wat er voor blokken zijn. De bal stuurt een bericht aan de blok waarop hij op dit moment is. Daarop kan de blok reageren.

Het level biedt alle informatie over de omgeving, die de andere onderdelen nodig hebben. Het level weet waar de bal in het begin staat en geeft de bal aan of die op een bepaalde plek kan rollen of niet. Het level bestaat uit de verschillende layers.

Een layer bestaat uit verschillende blokken. Elk blok is van een bepaald type en weet zelf wat er gebeurt als de bal op deze blok rolt.

3.1.3 View

De Puzzleview laat de gebruiker datgene layer van het level zien, waarop de bal op dit moment is. De Puzzleview is een observer van de bal en wordt verversd als de bal zijn positie verandert.

De pausedialogue wordt aangeroepen als de gebruiker ergens op het scherm drukt of als de applicatie wordt afgesloten. De enddialogue wordt aangeroepen als het game een bericht krijgt dat de bal op een doelblok is.

3.2 Gegevensontwerp

3.2.1 Blok

Er bestaan verschillende blokken in ons spel. Ieder bloktype krijgt een eigen ID. Met behulp van deze IDs is het mogelijk de blokken te identificeren. Verder is

er een boolean variabele “solid” die aangeeft of de bal op deze blok mag rollen of niet. Speciale bloktypes hebben mogelijk nog andere gegevens. Bovendien krijgt de blok een variabele “state” die we vooral gebruiken om niet meerdere IDs voor een blok in verschillende toestanden te gebruiken. Zo heeft een deur altijd dezelfde ID als ze gesloten is of open staat.

3.2.2 Layer

Elk level bestaat uit verschillende layers. Een Layer wordt gerepresenteerd door een tweedimensionale array van blokken.

3.2.3 Level

Een level bestaat uit een rij van layers. Verder houdt het level nog andere gegevens bij. Deze zijn:

- start- en eindpunt
- persoonlijke highscore van dit level
- een boolean die aangeeft of de gebruiker dit level al een keer heeft opgelost

3.2.4 Levelset

Een levelset heeft een naam en bestaat verder uit een lijst van levels.

3.2.5 Bal

De “Rolling Stone” van onze app is een bal met deze attributen:

- positie (x,y) als double
- layerpositie op welke layer de bal is
- straal
- snelheid in x- en y-richting

3.2.6 Game

De klasse Game moet de gespeelde tijd bijhouden. Verder worden de objecten Level en Bal binnen deze klasse aangemaakt, dus zijn die ook onderdelen van Game.

3.2.7 EindVanLevelDialog

In deze klasse wordt alleen de naam van de speler onthouden.

3.2.8 Opslaan

De levelsets worden in vorm van een XML-bestand opgeslagen. Op die manier kan de gegevensarchitectuur van de levelpackages net zoals boven beschreven in een bestand worden opgeslagen en ook weer worden gelezen.

3.2.9 Highscores

De highscores worden in een online database opgeslagen. De communicatie van de app met deze database loopt via een klein php-scriptje, dat de gegevens uit de database haalt en in een van ons verwerkbaar vorm weergeeft op het scherm. Om de highscores te submitten wordt ook van een php-script gebruik gemaakt, dat met de parameters “naam” en “punten” wordt aangeroepen. Om misbruik zo veel mogelijk te vermijden is er nog een derde parameter “veiligheidscode” welke hardcoded in de app te vinden is en bij het opsturen van een highscore overeen moet komen met de veiligheidscode die in het php-script staat. Dit biedt helaas geen absolute veiligheid omdat de gegevens van een smartphone meestal via WiFi worden verstuurd en men in staat is om deze gegevens af te vangen, maar het biedt wel een beetje meer veiligheid dan de gegevens zonder beveiliging door te sturen. Verder zijn we aan het nadenken de gegevens versleuteld naar de server te sturen om nog meer veiligheid en correctheid van de gegevens te kunnen garanderen, maar hier weten we nog niet hoe ver dat mogelijk is.

3.3 Detail-Ontwerp

3.3 is een klassediagram dat een overzicht geeft hoe de onderdelen in elkaar zitten. In 3.4 is te zien hoe de interactie tussen de bal en het level eruitziet, als zich de neiging verandert of als de ticker een bericht naar de bal stuurt.

3.4 GUIs

3.5 t/m 3.10 laten zien hoe de gebruiker de verschillende onderdelen van de applicatie ziet.

3.5 Planning

3.5.1 Week 1

- Structurele opbouw op basis van het klassendiagram
- Besturing met behulp van de neiging
- Communicatie tussen de onderdelen binnen de Game-Activity

3.5.2 Week 2

- View
- Een volledig level met alle speelelementen

3.5.3 Week 3

- De menu's
- Dialogen
- Mogelijkheid om highscores uit de online-database te lezen en te schrijven

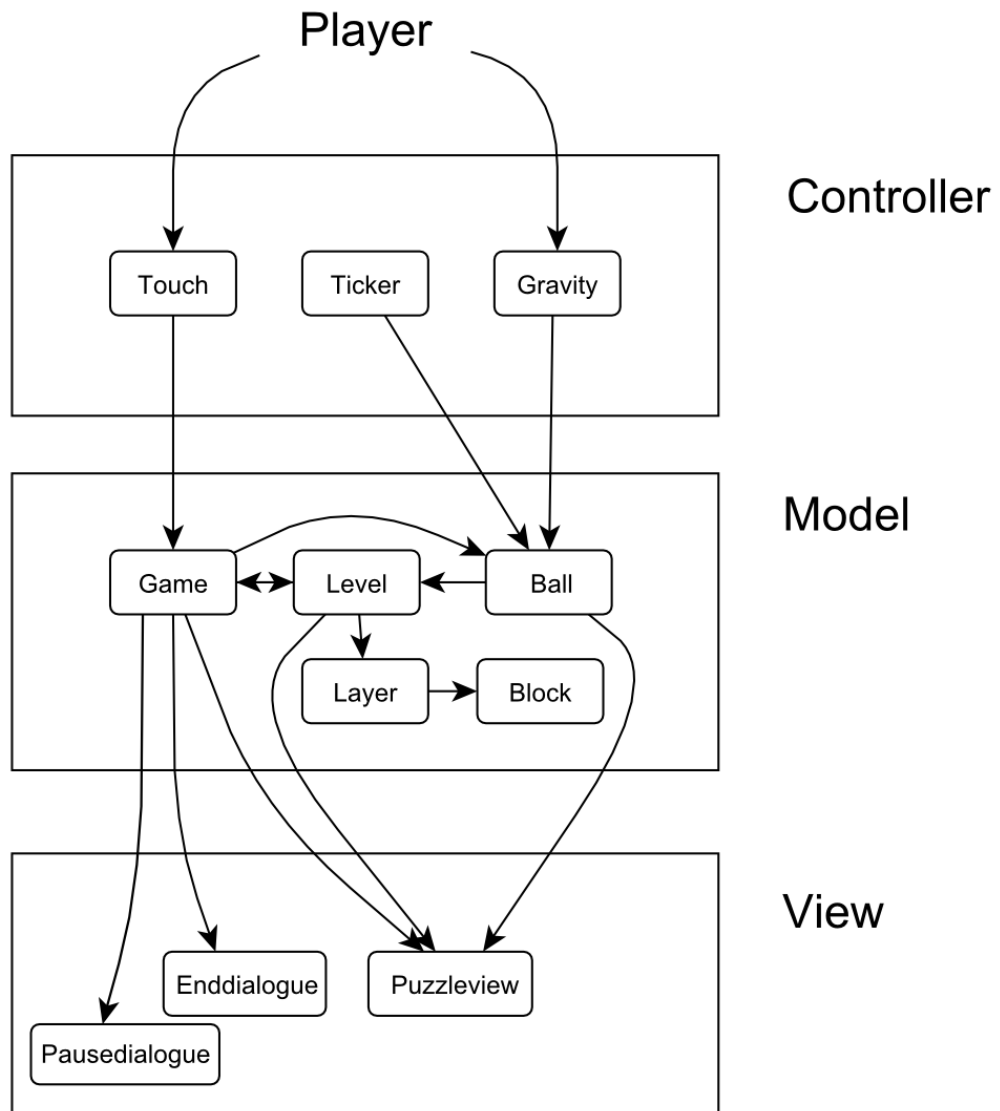
3.5.4 Week 4

- Bugfixing
- Meer levels

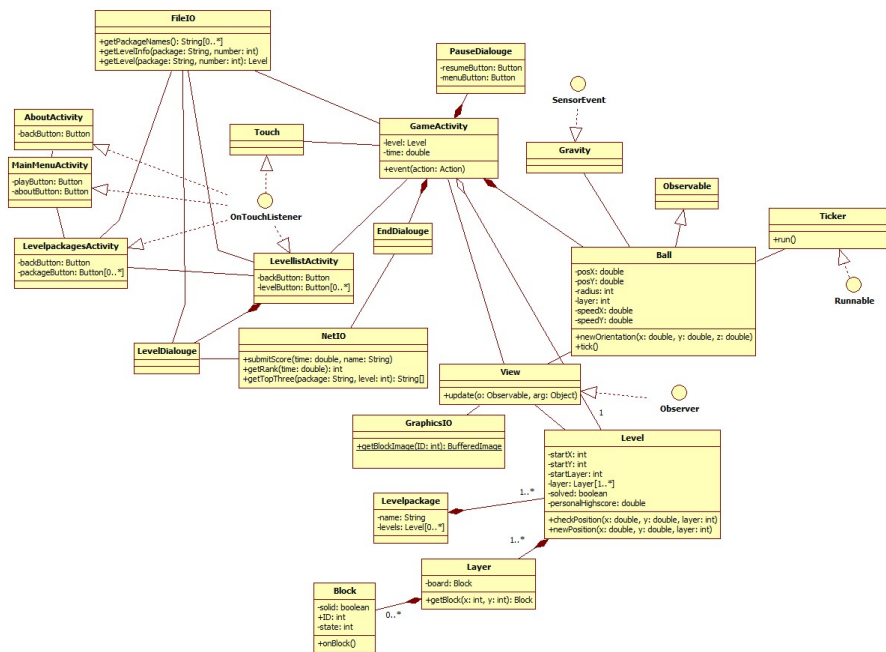
3.5.5 Prototype

Zoals in de werkplanning hierboven al te zien is, willen wij in de prototype een speelbaar level met meerdere layers en alle speelelementen hebben. Daarbij wordt ook de benodigde tijd al bijgehouden. De grafieken moeten op dit moment nog niet finaal zijn en we hebben mogelijk nog niet alle grafieke features geïmplementeerd.

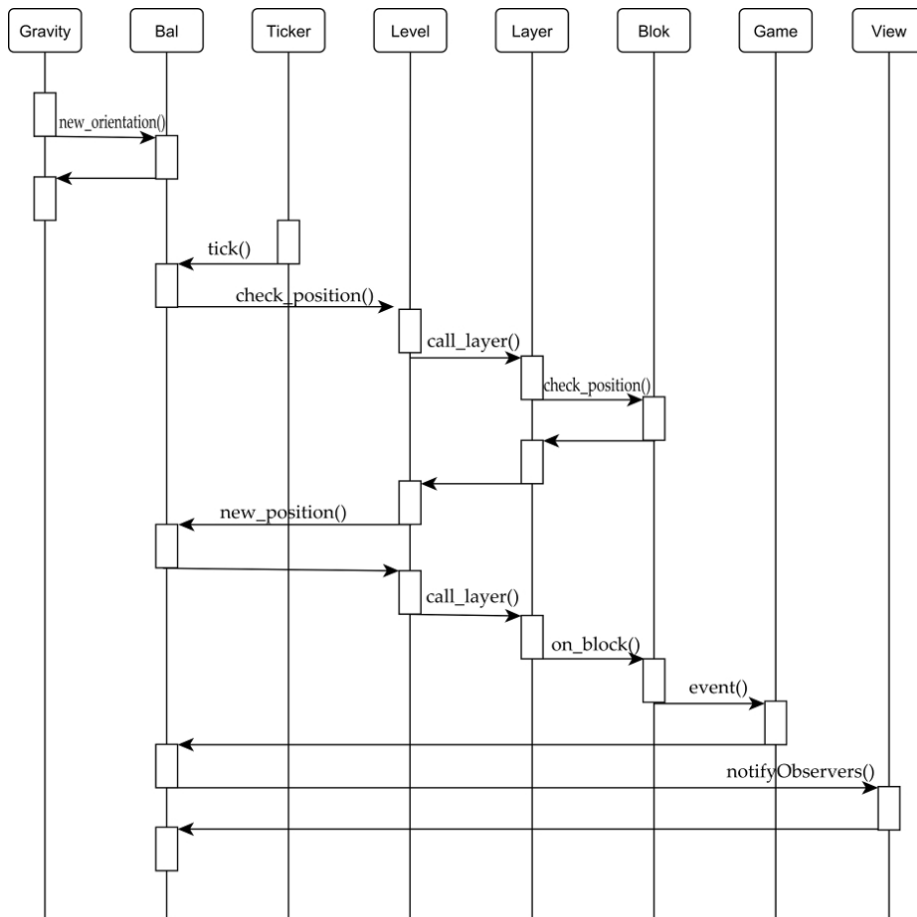
Wat er in de prototype dus zeker nog ontbreekt is een highscore-functie en de menu's.



Figuur 3.2: De architectuur van de game-activity



Figuur 3.3: Overzicht over klassen en relaties



0

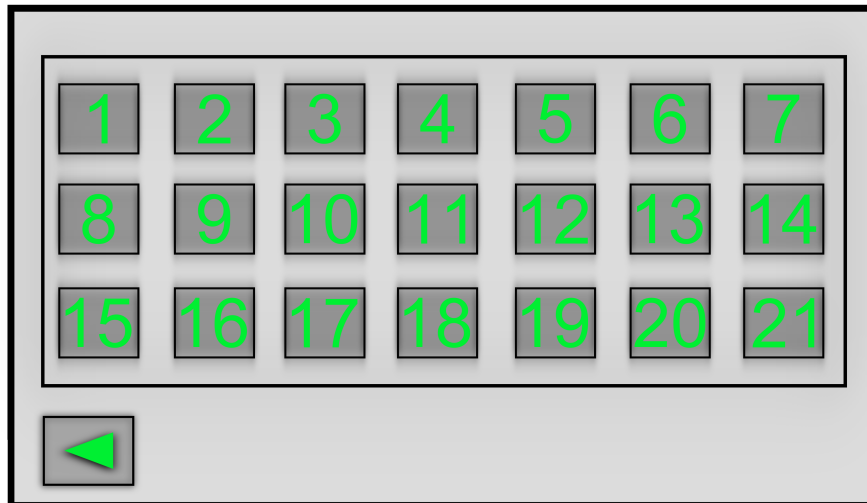
Figuur 3.4: Sequence diagram voor communicatie binnen de game-activity



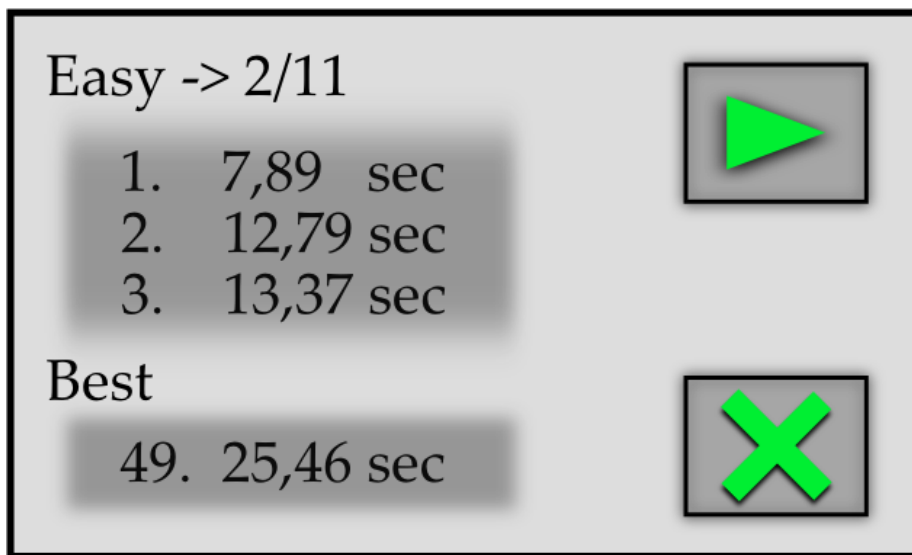
Figuur 3.5: Het hoofdmenu



Figuur 3.6: Het menu waar de gebruiker een levelpack kiest



Figuur 3.7: Het menu waar de gebruiker een level kiest



Figuur 3.8: De dialoog die naar het kiezen van een level te zien is



Figuur 3.9: Een voorbeeld hoe de gebruiker een level ziet



Figuur 3.10: De dialoog die de gebruiker aan het eind van een level ziet