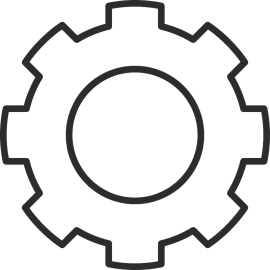
**Get it in, bro!**

**Ontwikkeld door Kodi Coding**

**C:\Users\Dion\Documents\Visual Studio 2010\Projects\Getitinbro\Press Kit\logo.jpg**

**Technisch ontwerp**

Inhoud

[Introductie 3](#_Toc369257955)

[Projectopdracht 4](#_Toc369257956)

[Doel van de mobile game 4](#_Toc369257957)

[Obstakels 5](#_Toc369257958)

# **Introductie**

Kodi Coding is een onofficieel bedrijf opgericht door Koen Klever en Dion van ’t Veer. Dit team, bestaande uit twee studenten Technische Informatica op de Hogeschool Rotterdam, is een team van twee softwareontwikkelaars met ervaring in C#, Java, Pascal, Visual Basic, HTML, JavaScript, PHP, ASP.NET, C, en C++.

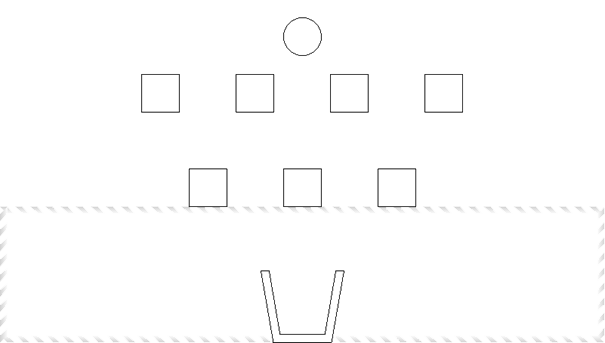
# **Projectopdracht**

In het eerste kwartaal van de minor Game Design en Development is ons een opdracht toegewezen. In deze opdracht werd ons gevraagd binnen 6 weken een retrogame een nieuwe draai te geven en te publiceren op een mobile device.

Wij hebben deze opdracht aangenomen en na overleg besloten een remake te maken van het onder de meesten bekende spel ‘Line Rider’. Aangezien ons team bestaat uit twee softwareontwikkelaars en niet uit grafische ontwerpers, hebben we ervoor gekozen het grafische ontwerp simpel te houden. We hebben ons als opdracht opgelegd om een werkende ‘physics engine’ te maken. Dit houd in dat we zelf een wereld creëren en hierin zwaartekracht, wrijving en andere natuurkundige wetten verwerken.

# Doel van de mobile game

Zodra de spelomgeving wordt gestart, krijgt de gebruiker een ‘wereld’ te zien met hierin een bal, een emmer en obstakels. Het doel van het spel is om de gebruiker zelf een weg te laten tekenen, om zo de bal in de emmer te krijgen. In figuur 1 is een voorbeeld van een (simpel lijkend) level.



Figuur 1: Een simpel lijkend level.

# Obstakels

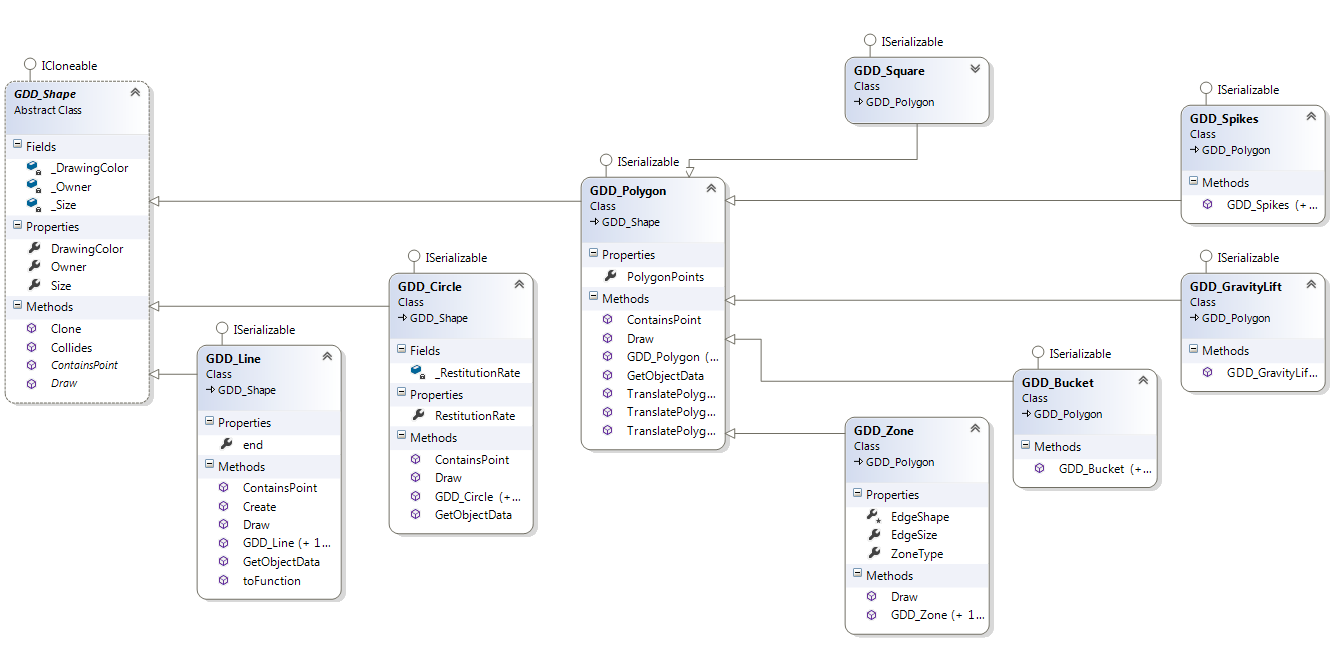
De obstakels in deze mobile game zijn simpel weergegeven. Onderstaand staat een lijst met de te implementeren obstakels.

* ‘Muren’ in de vorm van vierhoeken welke niet gepasseerd kunnen worden.
* ‘Spikes’ welke bij aanraking de bal lek prikt.
* ‘Gravity Lift’ welke de bal in een bepaalde richting schiet.
* ‘No-Draw zone’ waar niet in getekend kan worden.

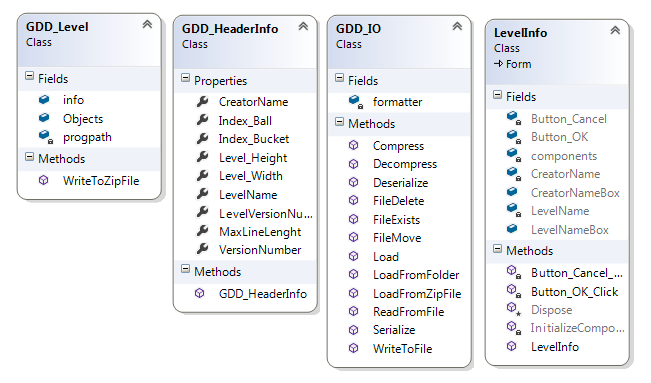
In Figuur 1 zijn de ‘muren’ en ‘No-Draw zone te zien’.

# Ontwikkelmethode

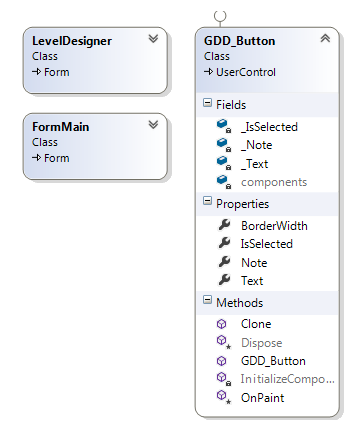
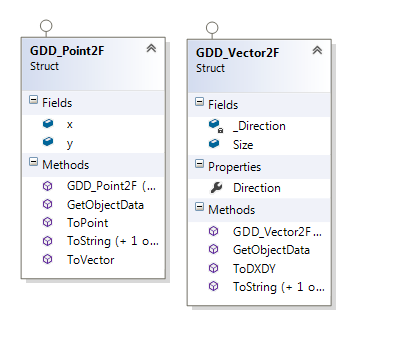
In dit project is gekozen om te ontwikkelen in C#, met als ontwikkelomgeving Visual Studio 2012. Deze keuze geeft de mogelijkheid de mobile game te laten draaien op elk apparaat dat Windows XP of hoger draait, inclusief tablets. In Figuur 2.1 tot en met Figuur 2.7 zijn klasse-diagrammen te zien, welke de structuur van de broncode bevat.



Figuur 2.1: De klasse-diagram van Shapes en zijn afgeleiden

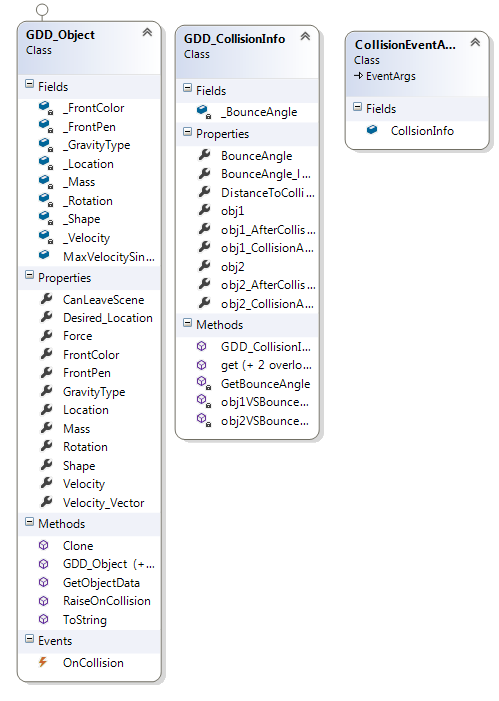
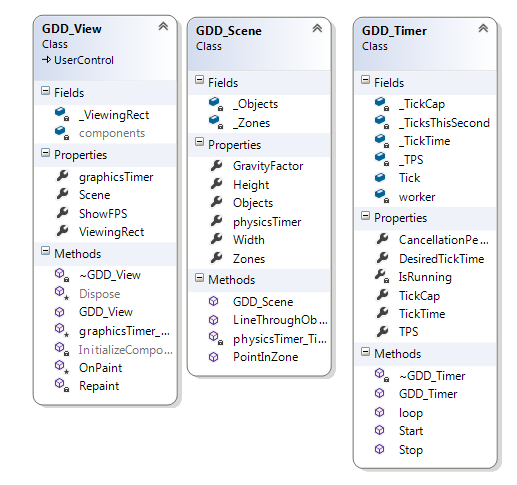


Figuur 2.2: De klasse diagrammen van de Level-klassen



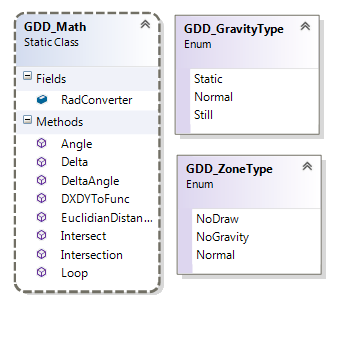
Figuur 2.3: De Types

Figuur 2.4: De zelfgemaakte User Controls



Figuur 2.6: De klassen waarin wordt getekend

Figuur 2.5: De klasse Object en Collision



Figuur 2.7: Overige klassen

# Opslagstructuur

De mobile game bevat een bepaalde opslagstructuur. Levels die door de gebruiker worden gemaakt in de daarvoor geleverde Level Designer worden automatisch opgeslagen in de map:

**(Game directory)/Saved levels/Custom**

Als (levelnaam).zip bestand. Van tevoren meegeleverde levels zijn te vinden in:

**(Game directory)/Levels/Chapter(chapternummer)/ch(chapternummer)lev(levelnummer).zip**

Een voorbeeld hiervan is, uitgaande van Chapter 1 en Level 1:

**(Game directory)/Levels/Chapter1/ch1lev1.zip**

Custom games worden geladen vanuit de bovenstaande map.

# Physics engine

In de game “Get it in, bro!” is een eigen physics engine gemaakt. Deze physics engine is een stuk door Kodi Coding zelf geschreven code die de natuurkundige wetten zo realistisch mogelijk beschrijft.