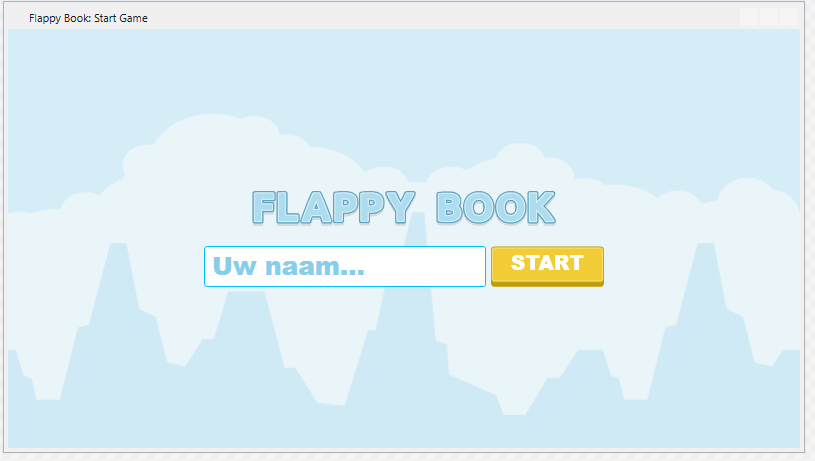
# Flappy Book

Deze periode zal er geen tussentijdse toets zijn, in plaats hiervan gaan jullie aan de gang met een project. Dit project maken jullie in 2-tallen. Let wel op, ieder teamlid moet alle code kunnen verklaren aan de docent indien hierom gevraagd wordt.

## Het project

In jullie tweetal gaan jullie een op Flappy Bird lijkend spel maken, er wordt vooraf een project aangeleverd waar de basis-layout al in uitgewerkt is. Verzin zelf een naam voor jullie versie van het spel.

### MainWindow:



Dit scherm geeft de gebruiker de mogelijkheid om zijn/haar naam in te geven en het spel te starten:

**Must have**

* Controleer of de naam minimaal 3 tekens lang is
* GameWindow moet geopend worden bij het klikken op “START”
* De ingevoegde naam moet meegegeven worden aan het GameWindow

**Should have**

* Een melding tonen als de naam niet juist ingevuld is
* De juiste spelnaam moet getoond worden

**Could have**

* Voorkom dat de gebruiker iets anders in kan typen dan letters en spaties

**Won’t have**

* Toon de huidige highscore

### XML level

De levels van het spel moeten worden opgeslagen in XML-formaat, maak in het project een map XML en plaats hierin 1 of meer XML bestanden welke ongeveer zo uit zien:

<?xml version="1.0" encoding="utf-8"?>

<GameLevel number="1">

<Settings>

<Name>Mania</Name>

<Dimensions>

<Width>1200</Width>

<Height>500</Height>

</Dimensions>

</Settings>

<Blocks>

<Block Position="100" GapHeight="100" />

<Block Position="250" GapHeight="20" />

<Block Position="320" GapHeight="400" />

<Block Position="540" GapHeight="300" />

<Block Position="870" GapHeight="50" />

</Blocks>

</GameLevel>

Het level ontwerp staat eigenlijk opgeslagen als Block elementen in de Blocks collectie van het GameLevel element. Ieder “Block” heeft een positie vanaf links en een hoogte van de rectangle aan de onderkant.

Op basis van de afstand tussen de Rectangle boven en onder moet de positie van de bovenste rectangle berekend worden. Het eerste “Block” uit het voorbeeld hierboven wordt dus getoond als volgt:

<Rectangle Margin="100,0,0,0" Width="75" Height="300"

Fill="LightBlue" Stroke="SkyBlue" StrokeThickness="5"

VerticalAlignment="Top" HorizontalAlignment="Left" />

<Rectangle Margin="100,0,0,0" Width="75" Height="100"

Fill="#FF97714A" Stroke="#FF96C23A" StrokeThickness="5"

VerticalAlignment="Bottom" HorizontalAlignment="Left" />

De hoogte van het onderste “Rectangle” element komt uit de XML en is 100px, de hoogte van het scherm is 500px zoals gedefinieerd in het XML-bestand en ik wil dat er 100px tussen de twee elementen zit waar het karakter tussendoor past.

Dus de hoogte van het bovenste “Rectangle” wordt dat al volgt berekend:

**hoogte\_rectangle\_boven =** *hoogte\_scherm - hoogte\_rectangle\_onder – gewenste\_vrije\_ruimte*

**hoogte\_rectangle\_boven** = 500 – 100 – 100 = 300

### GameWindow:



Dit scherm laad een level uit een XML-bestand, toont dit level en geeft de gebruiker de mogelijkheid om tussen de “Rectangle” elementen door te vliegen met een karakter.

**Must have**

* Toon de naam van de gebruiker linksboven in het scherm
* Lees het “Settings” element uit het level XML-bestand uit
* Verander de scherminformatie met de gegevens uit het “Settings” element.
* Lees de verzameling “Blocks” uit het level XML-bestand uit
* Toon voor ieder “Block” twee “Rectangle” elementen, één boven aan het scherm en één onderaan het scherm. Zoals ook omschreven staat op de vorige pagina.
* Toon het karakter op een positie aan de linkerkant van het scherm waar deze niet botst met eventuele “Rectangle” elementen
* 3 seconden na het tonen van het scherm moet het volgende gebeuren:
  + De afbeelding met “GET READY” wordt verborgen
  + Vanaf nu moet iedere 50 milliseconden:
    - Het karakter 2px naar onderen en 5 naar rechts verplaatsen
    - De score (rechtsboven in het scherm) verhoogd worden met 1
* Bij het drukken op de spatiebalk verplaatst het karakter met 15px omhoog

**Should have**

* Als het karakter een “Rectangle” raakt stopt het spel en wordt de “GAME OVER” afbeelding getoond.
* Als het karakter de rechterkant van het scherm bereikt wordt indien beschikbaar het volgende level geladen en anders herstart het eerste level.

**Could have**

* Toon de naam van de gebruiker linksboven in het scherm opgebouwd uit letter afbeeldingen
* Toon de score van de gebruiker linksboven in het scherm opgebouwd uit letter afbeeldingen

**Won’t have**

* Een level ontwerper