Projectdocument Git Visualisatie   
(Make SE great again)

Andrès Ooghe & Brend Lambert

# Plan van aanpak:

Initieel was de gevraagde opdracht het visualiseren van verschillende git projecten en het mogelijk maken om deze te spelen. Na verder overleg met de opdrachtgever, werd er beslist om de opdracht wat bij te sturen.

Het is momenteel de bedoeling om een project te ontwikkelen dat getoond kan worden op opendeurdagen of infomomenten. Hierbij kunnen bezoekers aan de hand van een User Interface, verschillende stukken code op de juiste plaats zetten. Daarna worden deze stukken code uitgevoerd op een spel (Flappy Bird) en kunnen ze visueel de veranderingen zien die ze aangebracht hebben (kleur veranderen, snelheid van vallen/vooruit gaan, richting van vallen…).

Met dit project is het de bedoeling dat we de toekomstige studenten al een lichte voorsmaak geven van wat de afstudeerrichting SE inhoudt. Hiermee willen we de studenten motiveren om voor SE te kiezen, wat uiteraard een zeer boeiende afstudeerrichting is.

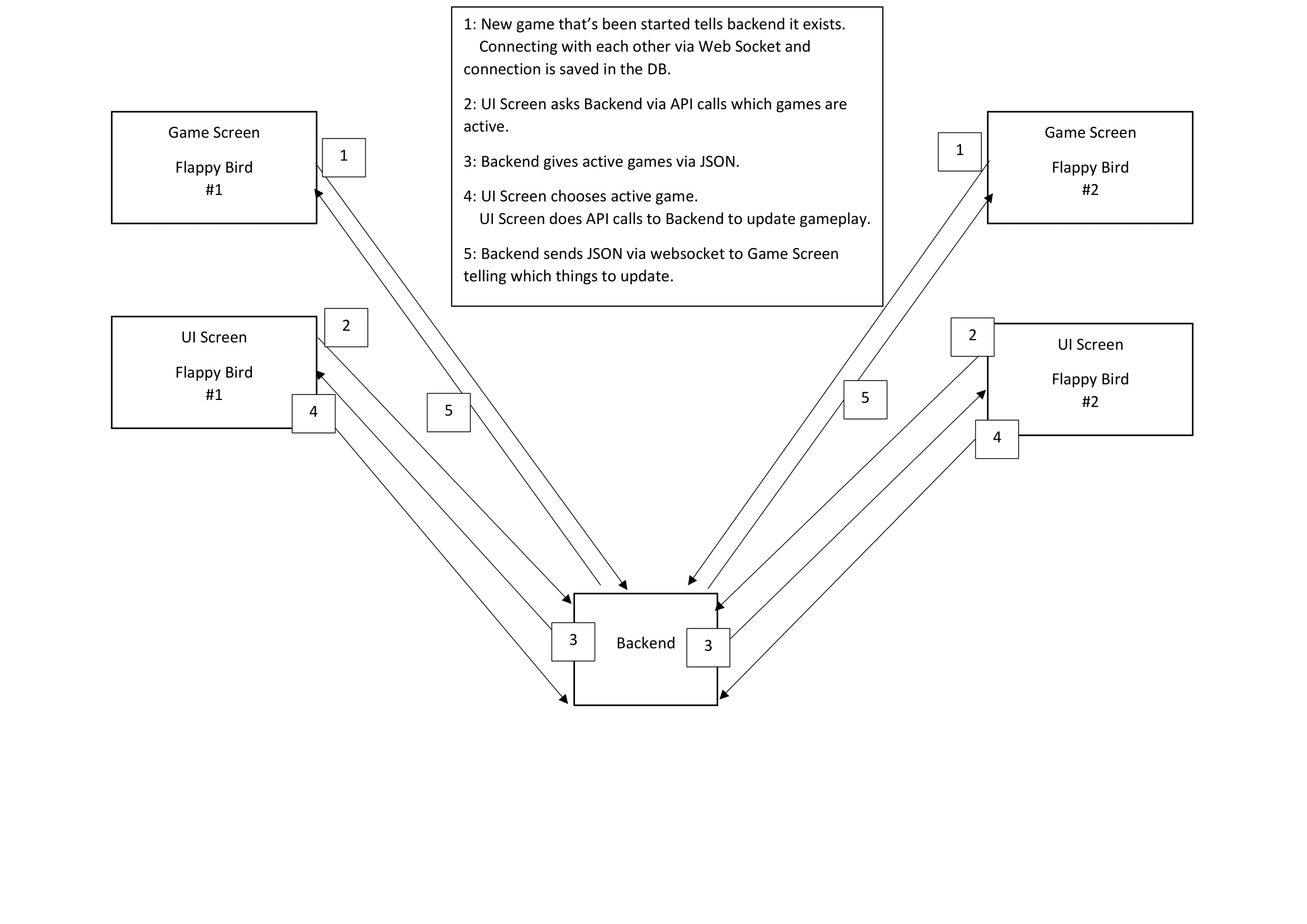
# Planning:

|  |  |
| --- | --- |
| 10/10/2019 | * Scope defining * Project document opstarten |
| 17/10/2019 | * Analyse * Wireframes * (Data modeling) * Features definen |
| 24/10/2019 | * Development opzoekingswerk * Start frontend * Start basic game |
| 4-8/11/2019 | * Frontend development * Backend development * Testing |
| 9-12/11/2019 | * Afwerken beide frontend systemen * Verschillende blokken code kiezen om te laten veranderen |
| 16-20/11/2019 | * Beide systemen met elkaar integreren * Code refactoring * Documentatie afwerken * Eventueel Brain Wave sensor integratie |

# Control flow

Zie ControlFlow.pdf document.

1. De eerste stap van de control flow begint wanneer een game wordt opgestart. Deze stuurt een request naar de backend waarbij dan ook telkens een web socket connectie wordt opgezet die dient als communicatie. Deze connectie wordt dan opgeslagen in de database zodat deze later gelinkt kan worden met de API-calls.
2. In de tweede stap vraagt de User Interface aan de backend via een API Call welke games er actief zijn.
3. Na stap twee stuurt de backend de actieve games terug aan de hand van een JSON-object.
4. Daarna selecteert de gebruiker op de User Interface de game die hij wil besturen. Om de game dan te laten veranderen stuurt de User Interface via API-calls, JSON-objecten door naar de backend waarin staat wat moet veranderen in de game.
5. Als laatste stuurt de backend de ontvangen JSON-objecten door via de websocket, die hij haalt uit de connections-tabel in de database, om zo de geselecteerde game te veranderen.

Stap 4 en 5 worden steeds herhaalt waardoor de game constant kan veranderd worden bij het “deployen” van de code.

# Database schema

Zie ERD.pdf document.

Highscores: Deze worden opgeslagen om zo een leaderboard te kunnen tonen van de beste scores.

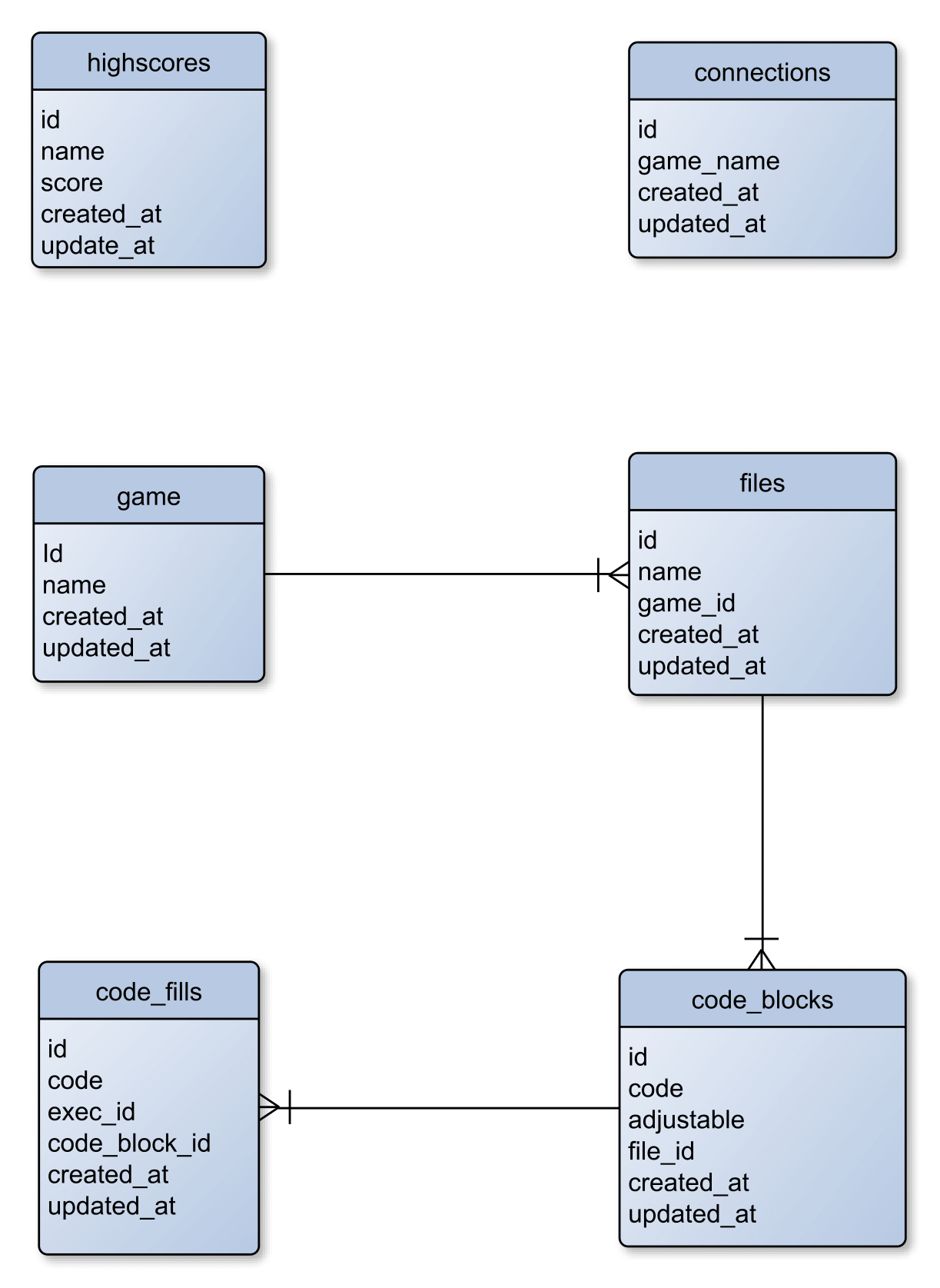
Connections: Hierin staat opgeslagen welke games aan de gang zijn en welke id ze hebben om deze zo te kunnen aanspreken in de backend wanneer er meerdere games aan de gang zijn.

Game: Een game heeft een id en een naam (bv. FlappyBird). Zo kan de speler verschillende games kiezen op het game scherm.

Files: Een game heeft meerdere Files. Een file is een representatief voorbeeld van de files die je zou kunnen terugvinden in de broncode van een spel (bv. game.js).

CodeBlocks: Een File bestaat uit meerdere code blokken. Dit zijn de blokken in de files die kunnen aangepast worden (bv. een functie om de kleur te veranderen van de achtergrond).

CodeFills: Een CodeBlock heeft meerdere CodeFills. Een CodeFill dient als invulling van een CodeBlock om zo het spel te veranderen (bv. een CodeFill die de achtergrondkleur op rood zet).



# Uitbreidingen

Tijdens het ontwikkelen van dit project hebben we ons voornamelijk beziggehouden met de standaard werking van het spel. Namelijk het laten veranderen van het spel, aan de hand van acties die uitgevoerd worden op het tweede scherm. Natuurlijk hebben we ook al gedacht aan een aantal uitbreiding die eventueel geïmplementeerd kunnen worden bij volgende projecten.

## Brain Wave sensor

Een van de uitbreidingen waar we persoonlijk graag aan zouden werken is het integreren van een Brain Wave sensor. Hiermee kunnen de spelers tijdens het spelen van Flappy Bird, te vogel controleren aan de hand van wat ze denken. Dit zou zowel een leuke als een uitdagende uitbreiding zijn voor ons project.

## Gamificatie

Gamificatie is het omzetten van niet game gerichte taken naar een game gerichte taak. Een voorbeeld hiervan kan zijn dat de speler eerst een bepaalde opdracht moet uitvoeren, bv. maak de vogel rood, waarna bij het juist uitvoeren van deze opdracht een nieuwe en iets moeilijkere opdracht vrijgegeven wordt. Hierdoor wordt de ervaring voor de gebruiker ook meer een spel en wordt hij of zij uitgedaagd om zo ver mogelijk te geraken. Dit zouden we graag integreren na de Brain Wave sensor integratie.

## Verschillende soorten User Interfaces

Als uitbreiding kunnen er ook verschillende user Interfaces ontwikkeld worden om zo het spel te besturen. Een voorbeeld hiervan kan zijn dat de gebruiker een bepaald QR-code moet inscannen, waarbij de bestaand API endpoints worden aangeroepen om zo het spel aan te passen.

## Nieuwe games ontwikkelen

De laatste uitbreiding waar we aan dachten is het ontwikkelen van verschillende nieuwe games die de gebruiker kan spelen. Zo kan bijvoorbeeld een simpel Mario Bros of een Donkey Kong spel ontwikkeld worden die dan zal samenwerken met de User Interface om zo het spel aan te passen. Zo kunnen de gebruikers uit verschillende games kiezen die ze willen spelen.

# Projectmatig werken

### 10/10/2019:

### 17-24/10/2019:

### 04-06/10/2019:

### 06-08/10/2019:

### 09-21/10/2019:

### 28-../10/2019:

