

Beschrijving City Check

- Spelers
 - 2+ teams
 - 1 of meer personen per team
- Tijdsduur
 - 3 uur (kan verander)
- Doel
 - Meest mogelijke punten verzamelen
 - Andere teams saboteren
 - Meest strategische route proberen uit te werken
- Werking van spel
 - Start op dezelfde locatie met x aantal punten
 - Vanaf bepaalde afstand van elkaar start van game
 - Vanaf dat punt wordt het traject gevolgd (Gps en netwerk functionaliteit moeten enabled zijn) (snake systeem achtig)
 - Meerdere Random locaties worden gegeven door de app bij start (relevant aan omgeving en trekpleisters (3 of meer, premium en normale locaties voor competitie te ondersteunen)) en updaten over tijd naarmate de wedstrijd vordert
 - puntensysteem
 - Per bezochte locatie wordt een bepaalde hoeveelheid punten toegekend (afhankelijk van de status van de locatie, premium of normale locatie)
 - Kruisen de trajecten elkaar op enig punt, dan wordt een bepaalde hoeveelheid punten afgetrokken van het team dat het traject kruist.
 - Andere extra spelelementen die de score kunnen beïnvloeden kunnen later worden toegevoegd.
 - VB: tower defense elementen (**nog steeds in concept & optioneel**)
 - Blokkades (doorbreken groot verlies van punten)
 - Static zone (AOE voor verlies punten)
- Uitbreidingen:
 - spectator/moderator toegang verschaffen voor veiliger verloop van spel en volgen van groepen
 - Recording van game.

User stories

Zie jira voor uitgewerkte versie.

Sprint 1 is volledig uitgewerkt met bijbehorende sub tasks, description en mockups , verdere user stories staan in de backlog klaar om een volgende sprint samen te stellen. De belangrijkste stories zijn uitgewerkt met extra informatie en mockups.

Link:

https://jira.ap.be/secure/RapidBoard.jspa?projectKey=CA18CC&rapidView=230&view=planning_nodetail

Sprint #1

We beginnen met de creatie van de eerste schermen.

Dit is het begin om teams en een spel te kunnen aanmaken via een gebruiksvriendelijke ui.

Alsook moeten we gebruik gaan maken van de google maps api om hun map te verwerken in ons spel view (in game scherm).

The screenshot shows a Jira RapidBoard interface for a sprint named "CA18CC Sprint 1" which contains 3 issues. At the top right, there is a "Start Sprint" button and a menu icon. Below this, a section labeled "Linked pages" is visible. The main area displays a list of three issues, each with a green icon, a priority indicator (up arrow), a title, and a status bar on the right. The status bar includes a button and a number in a circle. At the bottom left, there is a "+ Create issue" link. At the bottom right, a summary shows "3 issues" and an "Estimate" of "21".

Issue ID	Issue Title	Status	Estimate
CA18CC-1	Als user moet je een spel kunnen aanmaken	Opstart spel	8
CA18CC-2	Als user moet je een team kunnen aanmaken	Opstart spel	8
CA18CC-6	Als user wil ik een kaart zien van het speeldomein	Kaartsysteem	5

Uitwerking user storys binnen sprint #1

Story 1)

Description:

Create game: #UserStory1 Screenshot: (naam file). Dit gaat een simpel selectiescherm zijn met de mogelijkheid om te kiezen tussen 2 knoppen, het aanmaken van een game en het joinen van een game.

Als een game aanmaken word gekozen gaat men naar het volgende scherm Screenshot: (naam file). Hier komt men eerst bij een disclaimer pagina met wat extra info om dan door verwezen te worden naar de effectieve pagina Screenshot: (naam file). Om spelers te kunnen uitnodigen dmv een automatisch gegenereerde passcode.

Sub tasks:

UI opmaken vanuit mockups naar xml voor een gebruiksvriendelijke werking

UI voorzien van achterliggende werking (buttons code, ...)

Game class object kunnen aanmaken via de frontend

Communicatie functies voorzien om spel data te versturen en op te vragen met de backend

Tabellen opmaken voor data in te bewaren in de backend om een spel te maken

Opzetten van een asp.net core API om communicatie mogelijk te maken via http calls

Omzetten van de spel tabellen naar model classes voor het entity framework

Opstellen van een controller om functionaliteit en communicatie te voorzien

Story 2)

Description:

Create Team: #UserStory2 Screenshot: (naam file). Dit gaat een simpel selectiescherm zijn met de mogelijkheid om te kiezen tussen 2 knoppen, het aanmaken van een game en het joinen van een game.

Als het joinen van een game wordt gekozen gaat op het volgende scherm Screenhot: (naam file). Er gaat gevraagd worden om 1) Een team naam in te geven. 2) Een team kleur te kiezen 3) Een code in te geven die gegenereerd is bij de persoon die de huidige game heeft aangemaakt. Als deze code juist wordt ingevoerd gaat men de mogelijkheid krijgen om naar de volgende pagina doorverwezen te worden.

Sub tasks:

UI opmaken vanuit mockups naar xml voor een gebruiksvriendelijke werking

UI voorzien van achterliggende werking (buttons code, ...)

Team class object kunnen aanmaken via de frontend

Communicatie functies voorzien om team data te versturen en op te vragen met de backend

Tabellen opmaken voor data in te bewaren in de backend om een team te maken

Omzetten van de team tabellen naar model classes voor het entity framework

Opstellen van de controller functionaliteit voor een team aan te maken

Story 3)

Description:

View map: #UserStory3 Screenshot: (naam file). Op het moment dat een game wordt gestart krijgen alle spelers de map te zien met de eerste doelen random aangeduid op de map.

Sub tasks:

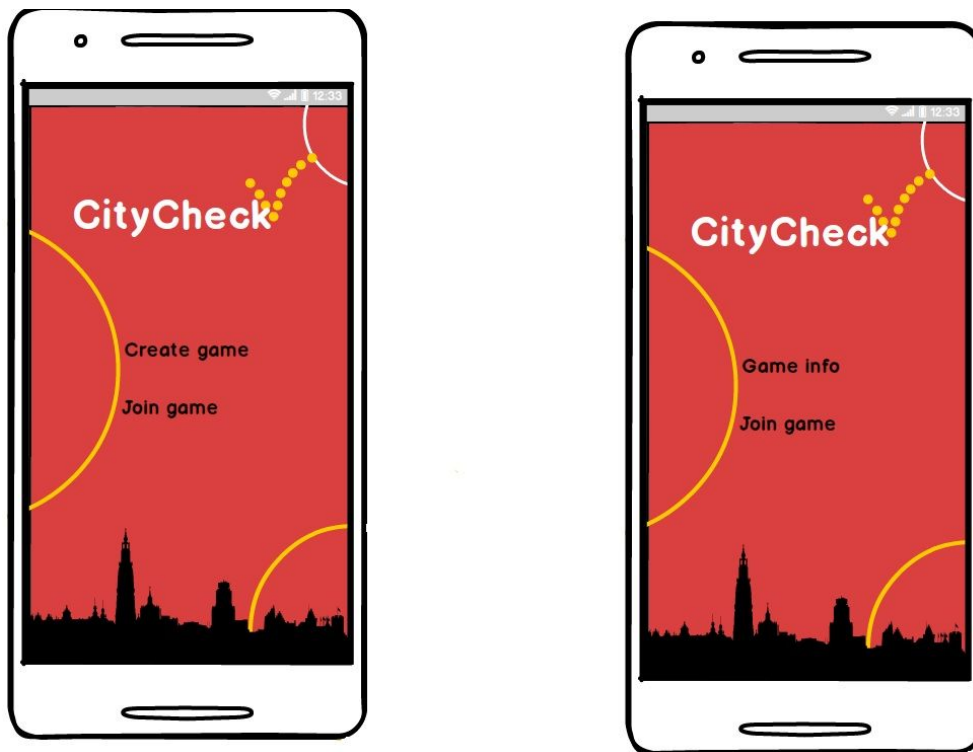
UI opmaken vanuit mockups naar xml voor een gebruiksvriendelijke werking

Inwerking google maps API in de app

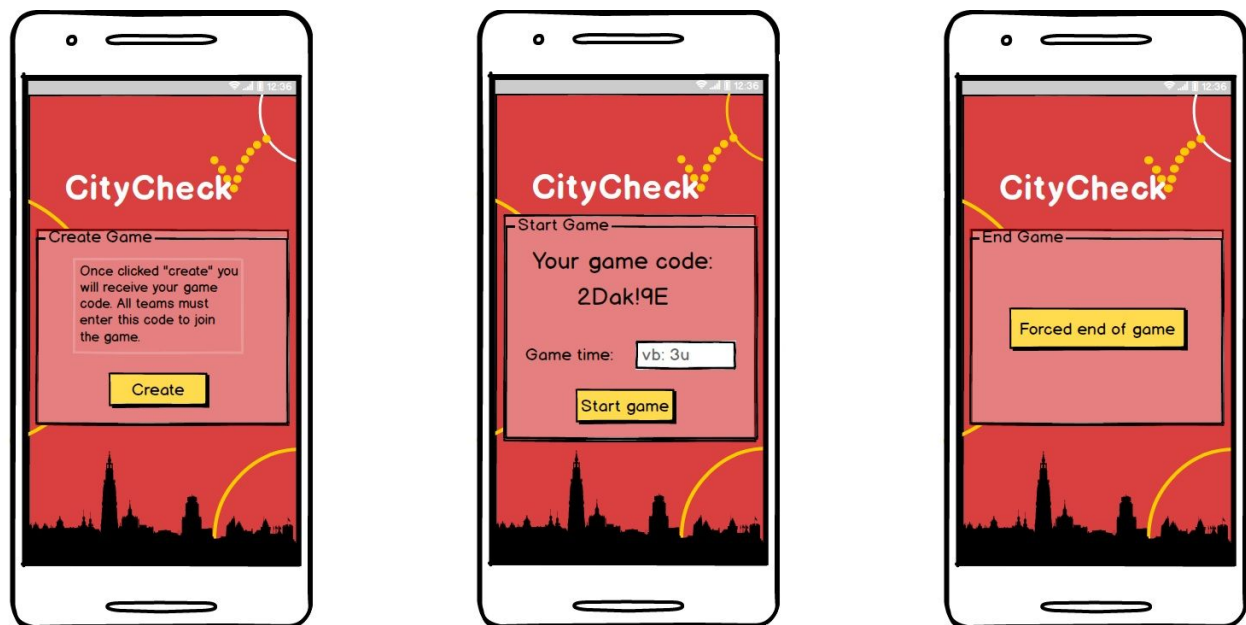
Map instellen op een overzicht van het speeldomein (Antwerpen) (border)

Mockups

Beginscherm voor en na het aanmaken van een spel.



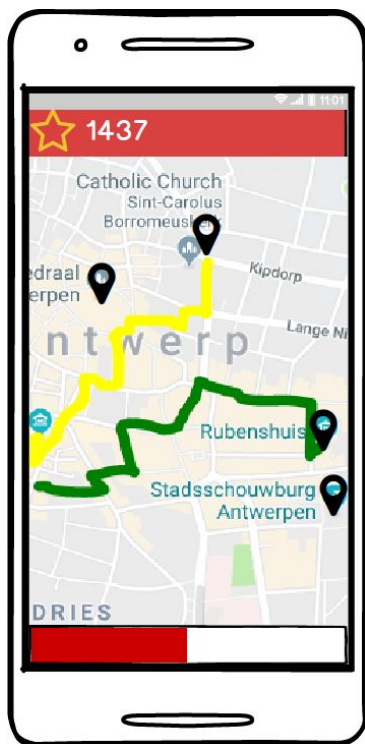
Aanmaken en beheren van een spel.



Team aanmaken en deelnemen aan een spel.



Kaart tijdens het spel en vraag op een locatie.



Einde spel.



Architectuur

Front-End

- Native android APP
 - =>XML,JAVA,GSON/JSON,...
- Google maps API
- Open data antwerpen (API)
- Sensor aanspraak (GPS) (binnen android java)

Back-End

- ASP.net core 2.0 framework
- AZURE (hosting) (mogelijkheid, nog niet zeker)