Projectplan Prototype Simmac

Versie 1.3.7

Contact gegevens van betrokken mensen

- Nolan Bijmholt (Ontwikkelaar):
  - Telefoon: +31623101156
  - Email: nm.bijmholt@student.alfa-college.nl
- Douwe Westerdijk (Ontwikkelaar):
  - Telefoon: +31622177314
  - Email: dh.westerdijk@student.alfa-college.nl
- Rik Teerling (Klant)
  - Email: r.teerling@alfa-college.nl

Versiebeheer

Het versiebeheer van dit project wordt geregeld met Git en GitHub. Voor iedere nieuwe feature maken wij een aparte branch aan in de Git-repository.

De GitHub-repository is te vinden via deze link*. Deze repository is privé en alleen toegankelijk voor bevoegde gebruikers. Als je geen toegang hebt en dit graag wilt, stuur dan een e-mail naar (een van) de ontwikkelaars.

Software

Voor de ontwikkeling van Simmac maken wij gebruik van de Unity Game Engine, versie 6000.0.40f1, op Linux. Wij kiezen voor Unity omdat het gebruiksvriendelijk is voor eenvoudige 2D-projecten zoals dit, en omdat alle ontwikkelaars ervaring hebben met Unity. We gebruiken versie 6000.0.40f1 omdat dit de meest recente versie was toen de ontwikkelfase van Simmac begon.

Microsoft Visual Studio Code wordt gebruikt als ontwikkelomgeving. Deze IDE is simpel en lichtgewicht, maar biedt ook veel functionaliteit voor Unity, dankzij de extensies die toegevoegd kunnen worden. Hierdoor werkt het goed samen met de Unity Game Engine en onze ontwikkelomgeving.

Daarnaast gebruiken we VSCode om onze documentatie in Markdown te maken, omdat het eenvoudig maar praktisch is. We exporteren de documentatie vervolgens als PDF.

Voor versiebeheer hebben we gekozen voor GitHub, vanwege de gebruiksvriendelijke interface en de uitgebreide versiebeheerfuncties. Bovendien draagt het bij aan het GitHub-portfolio van de ontwikkelaars.

Voor het maken van klassendiagrammen en flowcharts gebruiken wij yEd. Dit programma is gratis en biedt alle functionaliteit die wij nodig hebben voor het creëren van klassendiagrammen.

Eisen en wensen

Omdat het doel van dit project het testen van het minigame-gedeelte van Simmac is, is het de bedoeling dat wij de gameloop van klanten die een bestelling plaatsen en de speler die de bestellingen maakt, af hebben.

Het is dus de bedoeling dat wij goed werkende demo's hebben van de vier spellen en ook hun functionaliteit implementeren, zodat deze tijdens het testen beoordeeld kunnen worden.

Planning

De planning wordt bijgehouden met google sheets in een stroken planning, deze is te vinden met deze link**, deze spreadsheets is privé en alleen toegangkelijk voor bevoegden. Als je geen toegang hebt en dit graag wil, stuur een email naar (een van) de ontwikkelaars.

Folderstructuur

Alle bestanden die door de gebruiker zijn aangemaakt, in plaats van door Unity of VSCode zelf, worden opgeslagen in de Assets-map. Dit is de folderstructuur die wij gebruiken:

```
./Assets/
├── Plugins
├── Resources
│   ├── AchievementSprites
│   ├── Animations
│   ├── Player
│   ├── Sprites
│   └── Tilemap
├── Scenes
│   ├── GameScene
│   │   ├── Prefabs
│   │   └── Scripts
│   │       ├── Events
│   │       ├── Orders
│   │       └── Stations
│   ├── GlobalScripts
│   ├── Minigames
│   │   ├── BurgerStack
│   │   │   ├── Prefabs
│   │   │   ├── Scripts
│   │   │   └── Sprites
│   │   ├── PFIB
│   │   │   ├── Prefabs
│   │   │   ├── Scripts
│   │   │   └── Sprites
│   │   ├── ShakeShifter
│   │   │   ├── Prefabs
│   │   │   ├── Scripts
│   │   │   └── Sprites
│   │   └── SniperShowdown
│   │       ├── Prefabs
│   │       ├── Scripts
│   │       └── Sprites
│   └── UiScenes
│       └── Scripts
├── Settings
│   └── Scenes
├── Sprites
```

```
└── TextMesh Pro
    ├── Fonts
    ├── Resources
    │   ├── Fonts & Materials
    │   ├── Sprite Assets
    │   └── Style Sheets
    ├── Shaders
    └── Sprites
```

Wanneer er scènes worden toegevoegd, krijgen alle benodigde assets binnen dat script een map met de naam van de scène. Uitzondering hierop zijn bestanden die uit de Unity Resources-map moeten worden gehaald. Wanneer assets nodig zijn in meerdere scènes, worden deze opgeslagen in de Global-variaties van de map die zich in de Scenes-map bevindt.

Testplan

We gaan de speelervaring testen door proefpersonen het spel 15 minuten te laten spelen en hen daarna een vragenlijst voor te leggen. Dit formulier bevat vragen over de individuele minigames en de algehele spelbeleving, plus ruimte voor aanvullende opmerkingen.

De testgroep bestaat uit mensen met uiteenlopende achtergronden: sommigen zijn game-ontwikkelaars, anderen hebben maar beperkte ervaring met spellen. Zo kunnen we beoordelen hoe het spel aanslaat bij verschillende doelgroepen.

Hierna verzamelen we alle ingevulde formulieren en op basis van de testresultaten bepalen we de verdere ontwikkeling van het spel.

Taken

Dit zijn de taken die uitgevoerd moeten worden voor dit project, door wie deze taken uitgevoerd worden staat in de strokenplanning. Maar hier is een gedetaileerder overzicht.

- Documentatie
    - Projectplan
    - Voorstel
    - GDD
    - Testplan
    - Technisch Ontwerp
    - Flowchart

Features

- **Stations**
    - Interactie met speler
    - Opent scene
    - Verschillende station types (keuken, service, etc.)
    - Verbind stations met producten

- **Minigames**
    - Burger Stack
    - Sniper Showdown
    - Shake Shifter
    - Put The Fries In The Bag
- **GameManager**
    - Class GameManager
    - Save States
    - Orders opslaan
    - Verbinden met stationen
    - Verbinden met klanten
    - Verbinden met UI
    - Dynamische tijdsinstellingen (dag cyclus)
    - Beheren van personeel en klanten
    - Prioriteit van bestellingen
- **Customers**
    - Order logica
    - Satisfaction Handling
- **Event Handler**
    - Event class
    - Event aanroepen
    - Random gebeurtenissen
- **Movement**
    - Kan bewegen
    - Sprite verandert als je van richting verandert
- **Camera**
    - Speler volgen
    - In- en uitzoomen
    - Overzicht modus
- **Producten**
    - Verschillende producttypes
    - Kwaliteitscontrole voor bestellingen
    - Effecten van productkwaliteit op klanttevredenheid
- **UI**

    - Dynamische UI-elementen afhankelijk van de situatie
    - Bestellingen markeren in de UI
    - Real-time weergave van de voortgang van bestellingen
    - Tijd en geld aangeven in UI

- **Systeem**

    - Autosave-systeem
    - Speler kan stoppen en later verder gaan via autosave

- **Testen**

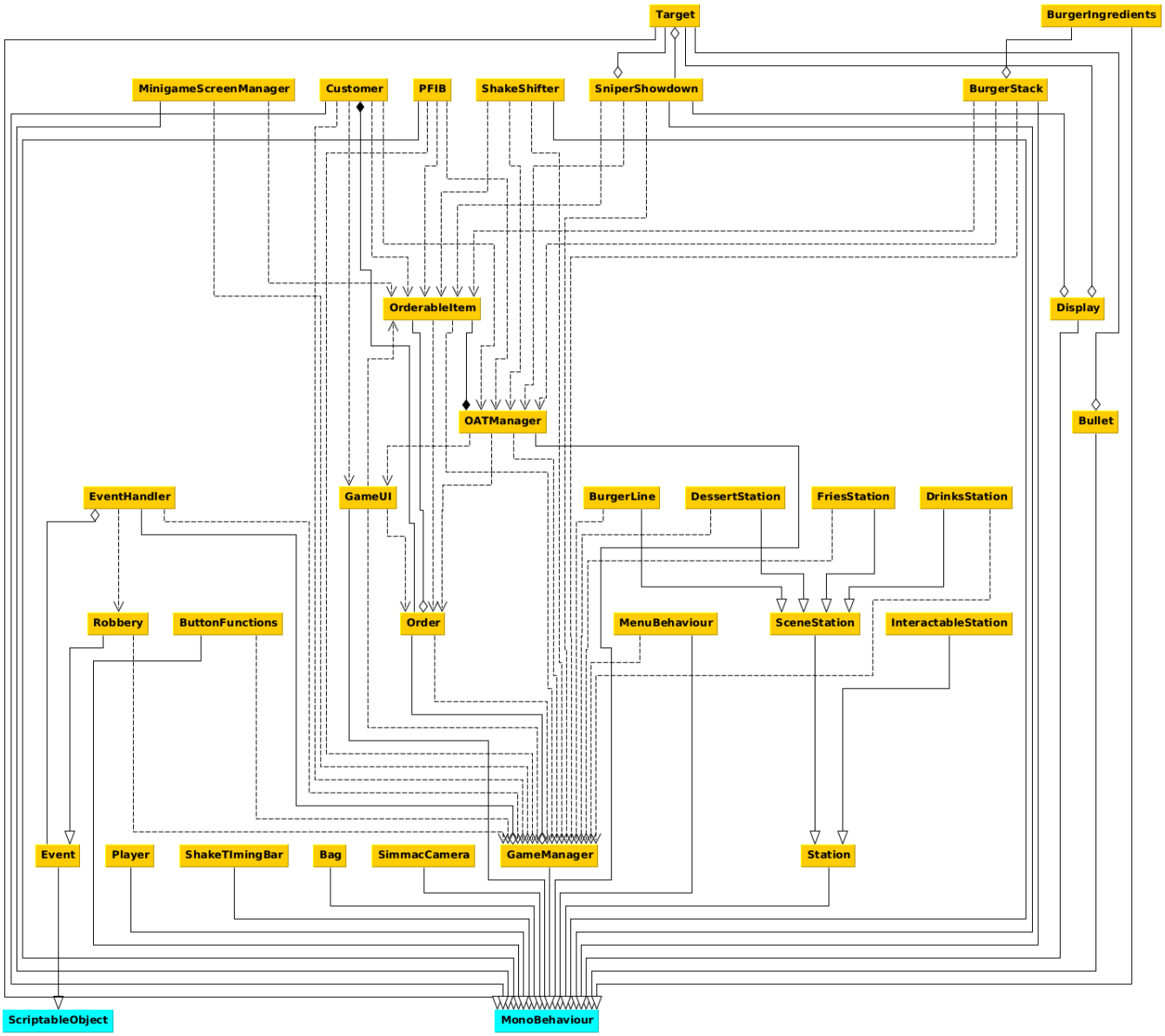
    - Testformulier ontwerpen en maken
    - Testen uitvoeren
    - Test resulaten analyseren

- **Assets**

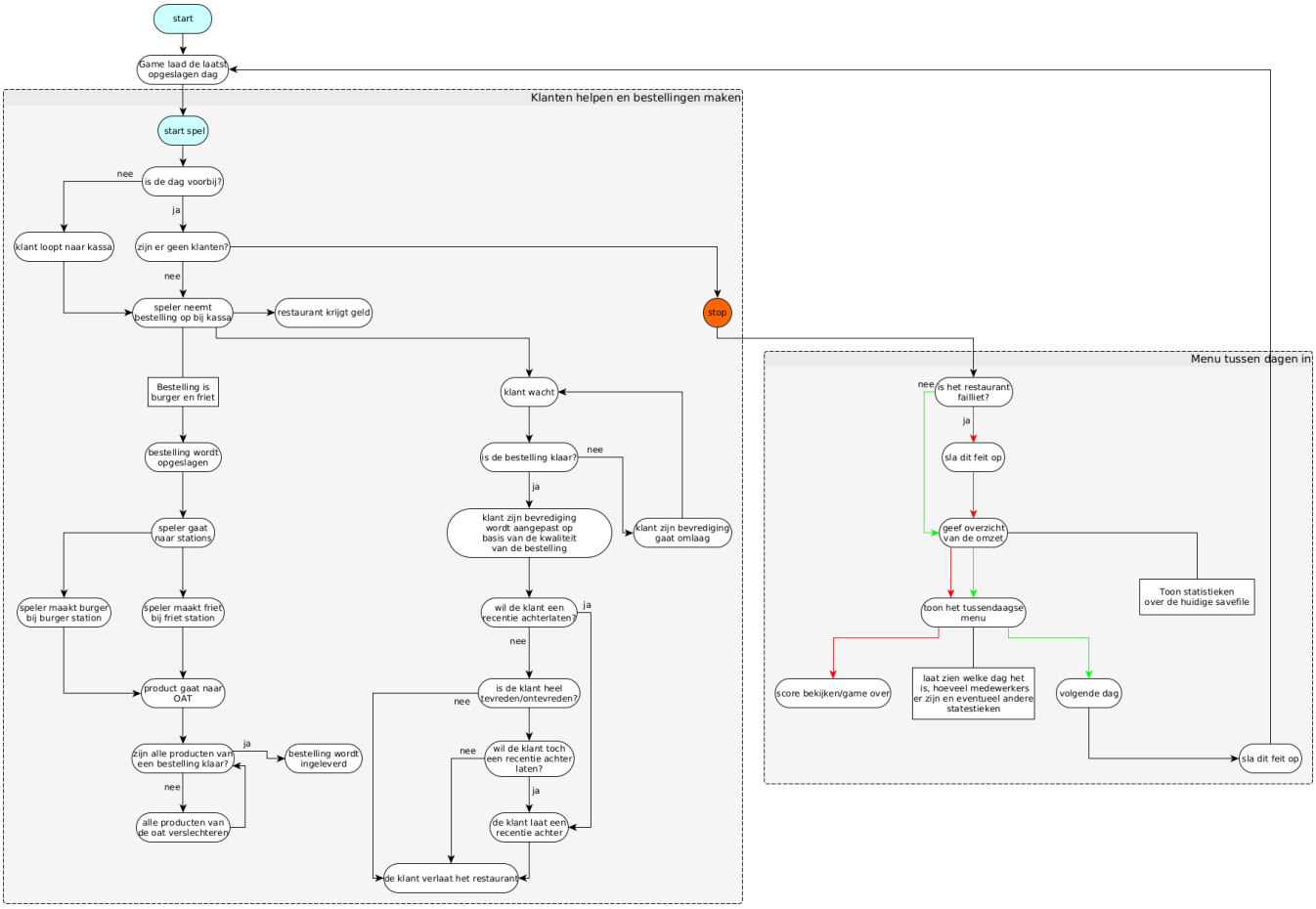
    - Tilemap
    - Assets voor minigames

Technisch ontwerp

- Klassendiagram



- Flowchart van de gameloop



Links

- GitHub repository*

https://github.com/Medikenji/Meesterproef

- Planning**

https://docs.google.com/spreadsheets/d/1UAxxr9gHdqxGKr9qrRfpPmZMmMEpT43-x3lhe_vaecc/edit?usp=sharing