



Made by: Geoffrey Hendrikx

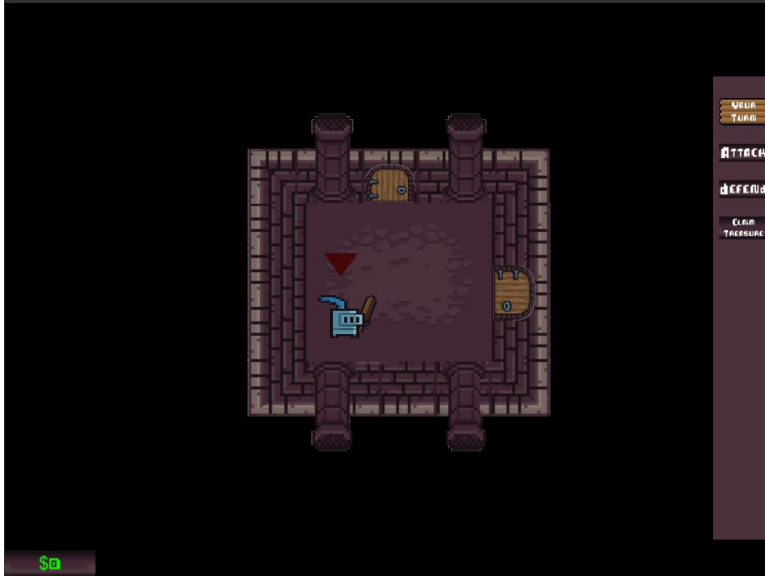
Intro	3
Game Flow	3
Code Structuur	5
PMI	8

Intro

Deze kernmodule moesten we een online multiplayer game maken. Hierbij kregen we de opdracht de scores in de database op te slaan.

Ik had hier nog geen ervaring in, maar ik vind dat er echt wel een mooi spel uit is gekomen.

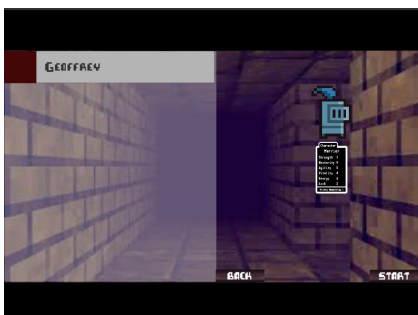
Het spel dat we moesten maken had een dungeon crawler genre. Dus ik wou er graag een medieval style van maken. Hieronder kun je zien hoe mijn design uitpakte in dit spel.



Ik ben super blij met de uitkomst. Ik wou in dit spel ook mijzelf uitdagen met een mooie style en dat is mij gelukt.

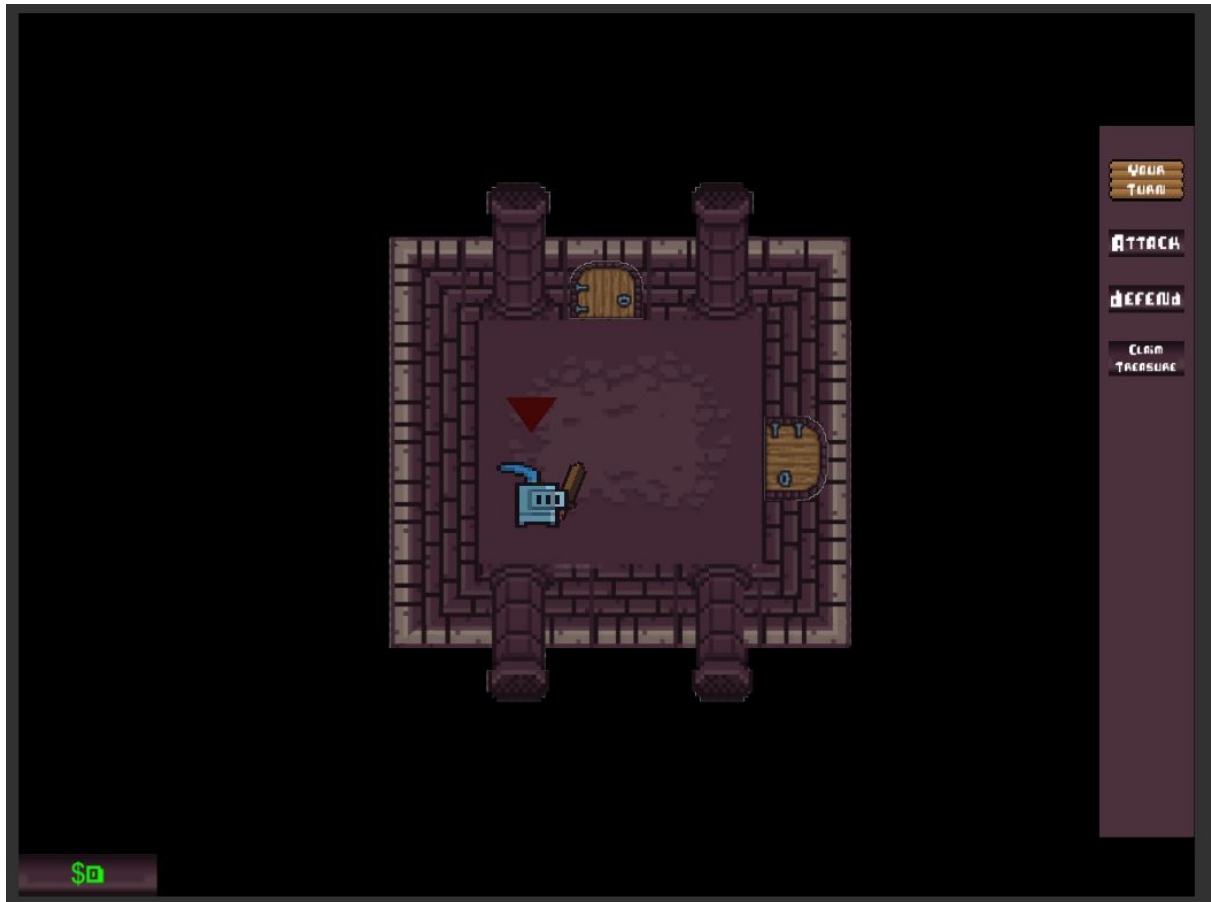
Game Flow

De game begint met een inlogscherm waar je inlogt als een gebruiker. Hierna kan je kiezen voor host of connect en kom je in het volgende scherm terecht. *Afbeelding hieronder*



Dit scherm is zo ingesteld dat de Host wel een start button heeft, maar de Client heeft dit niet.

Als je een Client bent krijg je het “ClientBehaviour” script en zal er een player voor je worden aangemaakt. De server heeft zowel een “ClientBehaviour” als een Serverbehaviour script. De Host kan het spel starten en je krijgt dan het volgende scherm voor je.



Zoals je ziet heb je rechts een menu wat aangeeft dat het “Your Turn” is. In dit menu kan je 3 dingen doen attacken, defenden en een treasure claimen. Deze knoppen returnen soms een request denied. Je kan namelijk niet een monster attacken als er geen monster is bijvoorbeeld.



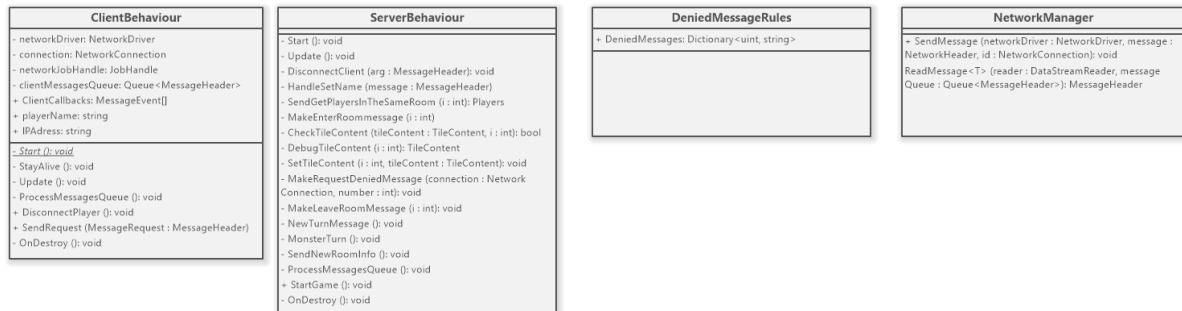
Als alle spelers uit de dungeon zijn stopt het spel en ga je naar de statistics screen.

Code Structuur

Elke Message heeft een eigen class, omdat elke message andere dingen moest serialize/deserialize. Achteraf had ik de None message offerwijl de “StayAliveMessage” vaker kunnen gebruiken. Deze had ik kunnen gebruiken en de messagetype kunnen veranderen voor messages zonder variables. Het was echt een gedoe om zoveel Messages te handlen namelijk. Messages werden op verschillende plaatsen gemaakt en ik wil het over 1 specifiek hebben. Dat is de “RoomInfoMessage” Ik heb hier erg veel moeite mee gehad en het wordt ook wel op een rare plaats gemaakt. Hij wordt namelijk gemaakt in de Game Manager. Ik heb dit gedaan, omdat ik dacht dat het managed wat in de kamer zit, dus ik kan het in de UIManager doen, maar dit maakt de kamer basically. Dit was de reden dat ik dacht de GameManager is een goeie plaats. Als ik dit nog een keer zou doen zou ik het waarschijnlijk in de UIManager maken. Het uitlezen werd gedaan met een queue, dit heb ik gedeeltelijk gedaan, maar sommige dingen had ik de iterator zelf nodig en heb ik het gewoon in de switch case gedaan. Dit was misschien niet de beste oplossing.



NetworkMessage UML Ik heb het project in meerdere UML's verdeeld.



Uit deze UML wil ik graag een belangrijke class bespreken genaamd de DeniedMessageRules. Deze class ziet er als het volgende uit:

```

4 references
public static class DeniedMessageRules
{
    public static Dictionary<uint, string> DeniedMessages = new Dictionary<uint, string>
    {
        { 0, "Cannot connect because the game has already started." },
        { 1, "Too many people has joined the server. Try again in a couple of minutes." },
        { 2, "Can't Escape from a room with an monster in it." },
        { 3, "It's not your turn." },
        { 4, "Can't Attack when there isn't a monster." },
        { 5, "Can't Claim a treasure when there isn't a treasure." },
        { 6, "Can't Claim a treasure with an monster in it." }
    };
}

```

Ik heb dit zo opgelost, want als er nu iets fout gaat moet ik alleen het cijfer te pakken en dan geeft die die foutmelding weer. Zo maak ik de request denied in de server manager.

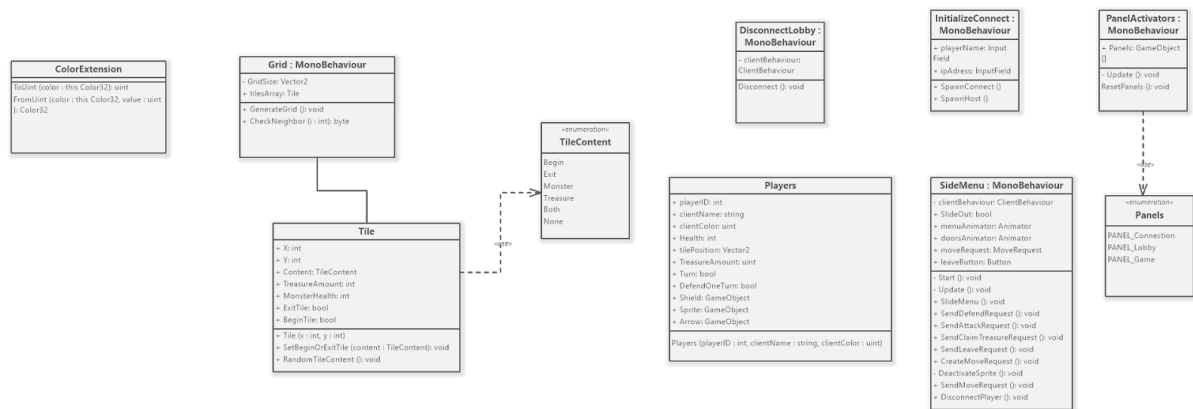
```

4 references
private void MakeRequestDeniedMessage(int i, int number)
{
    var requestDenied = new RequestDeniedMessage()
    {
        requestDenied = (uint)number
    };

    NetworkManager.SendMessage(networkDriver, requestDenied, connections[i]);
}

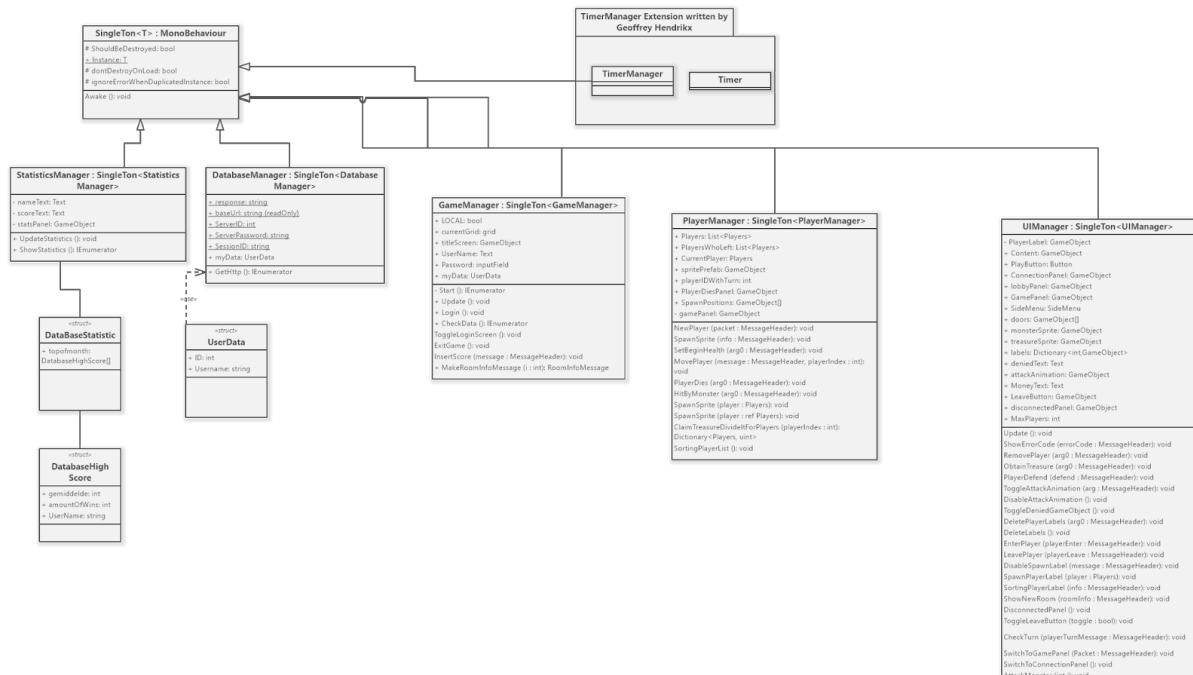
```

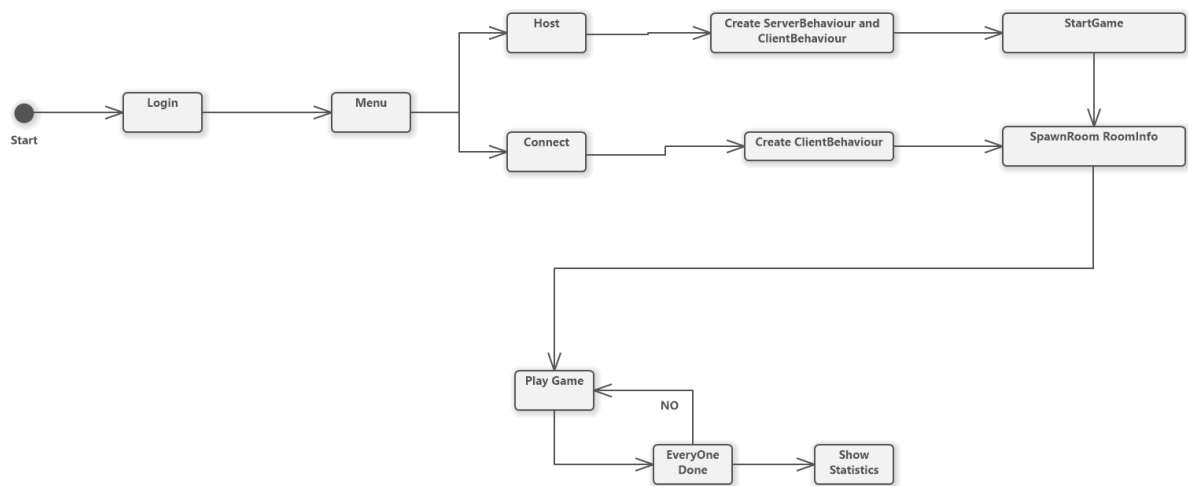
GameFlow UML



De SideMenu Class is eigenlijk de base game class. Voor als je de beurt hebt is dat de class die alles regelt en de Request verzendt.

Hieronder zie je mijn Manager Overview





PMI

Positief:

- Ik heb super veel geleerd en ik vond dat Vincent het erg duidelijk uitlegde.
- Ik moet zeggen dat ik nu eerder een Network project zou maken, omdat ik nu weet hoe het werkt.
- Ik heb er zoveel tijd aan besteed dat ik supertrots op mezelf ben.
- Ik heb ook veel over level design geleerd en over designen.

Min:

- Corona dus thuis werken. Al was dit soms best fijn, het was vervelend in de lessen qua internet connecties etc etc.
- Erg veel tijd kwijt voor zowel de kernmodule als project vrij.
- Het verzenden/ophalen van de statistieken is me helaas niet meer gelukt vanwege tijd.

Interessant:

- ik heb erg veel geleerd en ik heb gelezen dat er nog veel meer mogelijkheden zijn in online.
- Ik vind het nu erg interessant om ook nu te weten hoe andere developers dit aanpakken.
- Ik vind het nu ook interessant dit soort artikelen te lezen.