PacMan

Technische Uitwerkingsplan

**Code:**

In het spel speelt OncollisionEnter een grote rol sinds het daarvan af hangt of de speler verloren of gewonnen heeft.

Als de speler dus een collision heeft met een munt, krijgt hij 1 punt erbij.

Als dan de speler alle punten heeft weten te krijgen dan wint hij het spel. Maar als hij een collision heeft gemaakt met een vijand, dan gaat er een leven van hem af. Als de speler dan geen levens meer over heeft, dan heeft hij het spel verloren.

**Prioriteit:**

In het originele spel van pacman moest de speler alle punten in de map eten zonder zelf opgegeten te worden door de spoken om het spel te kunnen winnen. Dit deel van het spel zal dus ook de meeste prioritijd krijgen. De rest dat ook belangrijk is maar minder prioritijd heeft is de vijand script en verlies functie.

**Prioriteiten lijst:**

1. Speler controls script
2. De 2D map
3. Muntjes in de map

- munt eet functie

- Winscherm functie

1. Spook movement AI-script

* Speler levens-functie
* Spook collision met speler script
* Verlies scherm en speler verlies functie

1. Fruit script

* Speler onsterfelijkheid voor een aantal seconden
* Spook sterf script

**Features:**

Ik gebruik voor mijn spel unity 2018.2.9f1, ik zal deze engine gebruiken omdat het heel goed 2D games support, maar ook omdat ik al heel wat ervaring heb met unity.

Zoals ik boven al vermeld had zal ik vooral veel gebruik maken van de OncollisionEnter functie, want het spel kan gewonnen of verloren worden aan de hand van met wie je een collision hebt.

Voor de vijand script zal ik gebruik maken van unity ’s NavMesh. Met NavMesh kan ik ervoor zorgen dat de vijand random door de hele map kan lopen zonder recht door de muren heen te lopen.

Omdat ik meerdere vijanden ga maken met allemaal hetzelfde script, is het belangrijk dat hun niet allemaal hetzelfde pad. Daarom maak ik een script die telkens een random pad kiest voor de vijand.

UML & Activity Class



