





Design Patterns en Functionaliteit:

My Inheritance:

Ik heb ervoor gekozen om voor een abstract class voor Enemies en Consumables te gaan omdat ze veel functionaliteit delen die ik makkelijk kan overreden of toevoegen. Dit scheelt dubbele code.

Events:

Ik heb er voor gekozen voor een paar major events te gaan, omdat als de speler doodgaat bijvoorbeeld de ghosts hierop moeten reageren door terug te gaan naar hun hok en de manager moet de speler respawnen. Of wanneer de speler een power pellet eet, dan moet alle ghosts eetbaar worden en wegrennen. Hiervoor is het handig als verschillende objects op dit soort grote events kunnen reageren zonder alles toe hoeven te weten.

State Machine:

Deze is vooral belangrijk voor de ghosts, die na het eten van een power pellet weg moeten rennen, maar ik wil ook een state machine gebruiken voor mijn bossfight zijn stages. Zo zijn deze verschillende states goed onderscheiden en kunnen makkelijk afgewisseld worden.

Singleton:

Ik wil een singleton gebruiken voor mijn game manager die levels in laad en de score bijhoud van een run. Deze zou dus ook overdragen tussen levels en aan het einde van een run je score opslaan.

Één Update Functie:

Ik wil alle code zo schrijven dat er maar een update functie is, die de OnUpdate van elke functie oproept. Dit doe ik voor controle over de volgorde waarin dingen geupdate worden maar voornamelijk om efficiënter te zijn.

Engine - Unity:

Ik ga Unity gebruiken voor dit project omdat ik vaardig ben met het programma, en grote community in de vorm van de asset store, forums en reference voor als ik ergens tegen aan loop. Ik ben ook van plan volop gebruik te maken van prefabs en de unity editor.