

Ultimate Pacman

Casey Hofland – 19/09/2019 – 3017348 – Game Development Kernmodule 1

Concept

Wat is Ultimate Pacman?! De inspiratie moet duidelijk zijn, maar waar komt het idee vandaan om Pacman te maken maar zonder de maze? Vrij makkelijk ben ik gewoon niet heel fantasierijk, en ben snel geneigd om een 'twist' te zien als "we nemen iets en veranderen er 1 ding aan". Maar wat mij ook heel erg aansprak aan het idee was dat het simpel was, en dat ik het idee had dat ik met iets simpels mij meer kon focussen op de code en de structuur voor dit project.

Van alle retrogames die ik in een artikel over "100 beste retrogames" had voorbij zien komen, waarom Pacman? Ik had ook plannen voor Rad Racer met jumping of een Samus collectaton. Maar die vroegen teveel van me op graphical / worldscale gebied om praktisch te zijn voor een 3-weeks project, daarom ben ik met het simpelste idee gaan rennen, nogmaals om me te kunnen focussen op de code.

Toelichting Code

Dit was het eerste project waarin ik echt mijn Framework ben gaan gebruiken, en ik was blij hoe snel ik ermee een prototype op touw kon zetten. Mijn Framework is een folder vol met scripts die allemaal 1 taak uitvoeren. Er zit bijvoorbeeld een Singleton in, een LookAtCamera, en een OnEvent die op events let (Start, Destroy, CollisionEnter) en een UnityEvent die invoked word wanneer dat gebeurt. Deze is vooral handig om snel connecties op te zetten en methodes aan te roepen. Ook heb ik er heel erg op kunnen verbeteren nu ik het in de praktijk gebruik, waardoor ik het nog efficiënter kan opzetten.

Ik heb voor de code specifiek aan dit spel geprobeerd om dingen goed apart van elkaar te houden. Dat is vooral te zien met het Movement2D script, die ervoor zorgt dat ik een bepaald stuk code maar 1 keer hoeft te schrijven om daarna te worden gebruikt door alle geesten en Pacman. Verder heb ik geprobeerd Singleton Managers op te stellen die hun eigen handelingen uitvoeren en speldata bijhouden, waarin ik denk ik nog wat oefening nodig te hebben om deze efficiënt op te stellen (ideaal zouden ze allemaal dontDestroyOnLoad moeten zijn en mogelijk kan ik ze beter met de Toolbox pattern opstellen).

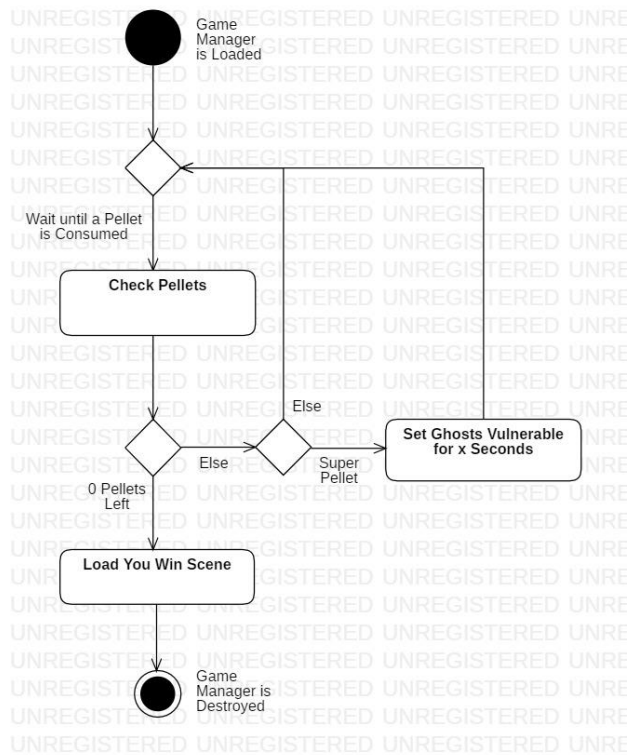
Ook ben ik geïntroduceerd aan interfaces. Ik vind het nog moeilijk om goede momenten te vinden waar je ze moet gebruiken, maar kan wel zien waarom ik ze zou willen hebben. Misschien met een groter spel komen ze meer van pas.

Als laatste heb ik een StateMachine gemaakt op niet 1, maar wel 2 manieren. De eerste gebruikt Unity's animator en StateMachineBehaviours om door states heen te switchen en andere behaviour te tonen (displayed door Blinky (red) en Pinky (pink)). De tweede gebruikt

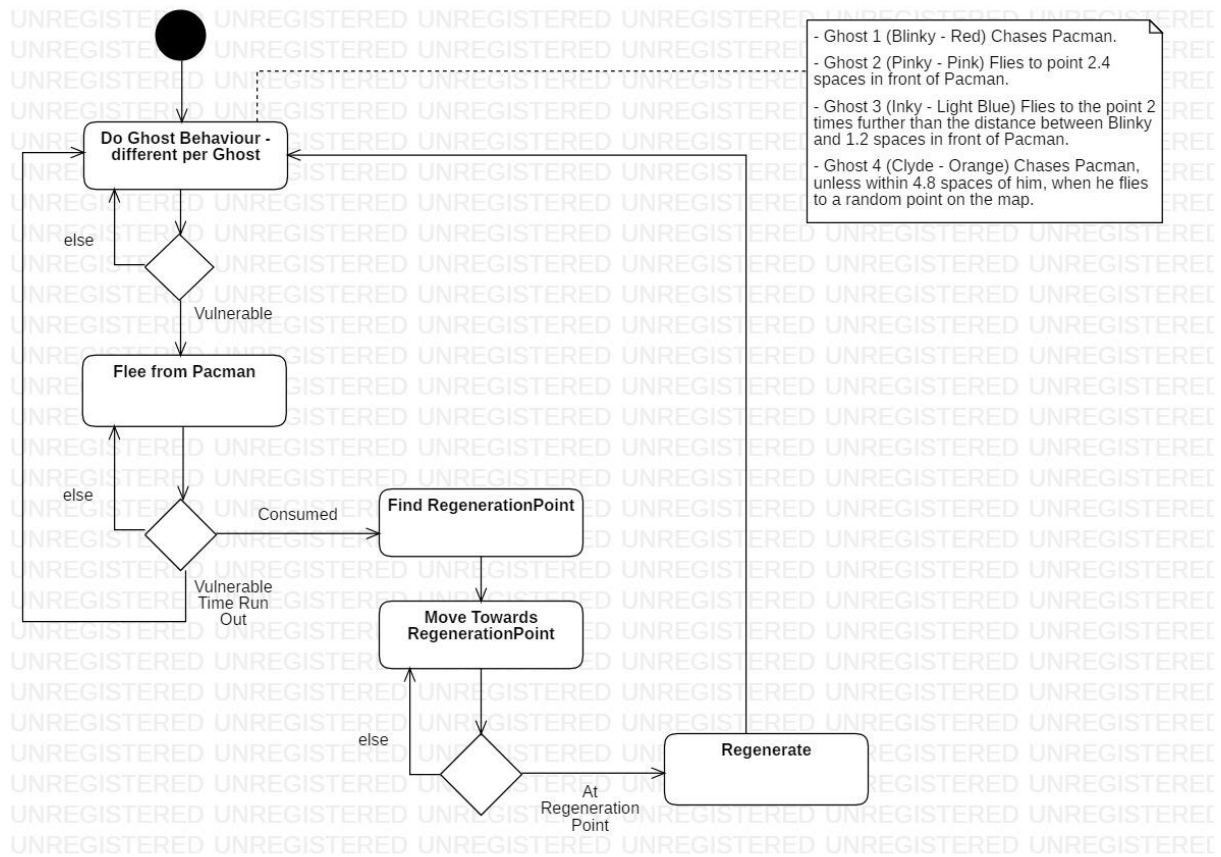
een zelfgeschreven StateMachine om ook dat te doen (al laat ik de animaties nog steeds in de animator gebeuren.) Mijn zelfgeschreven StateMachine vond ik een beetje tijdsverspilling omdat ik de Animator uitpluizen en ontdekken veel interessanter vond. Ik vind het nog tricky om er goed mee te werken, maar geloof dat het erg handig kan zijn in coalitie met programmeren en ga er in de toekomst zeker meer mee doen.

Diagrammen

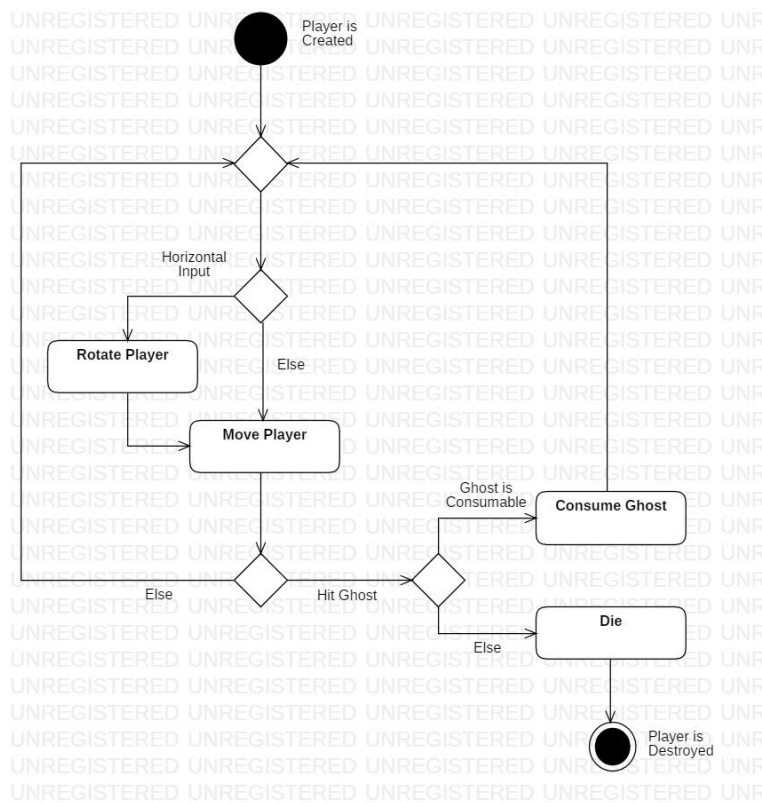
Game Manager Activity Diagram



Ghost Activity Diagram



Player (Pacman) Activity Diagram



Ultimate Pacman Class Diagram (StateMachineBehaviours omitted in favor of the NativeStateMachine)

