

Documentatie GDEV1 - Sam Walet

Mijn Game:

Mijn game is Rogue-Like Pac-Man. Het is Pac-Man met elementen van rogue-like games zoals The Binding Of Isaac. Met name meerdere kamers met een boss op het einde.

Het spel speelt gewoon als Pac-Man je moet de pellets opeten terwijl de spookjes achter je aan zitten. Het verschil is er zijn twee doolhoven die aan elkaar zijn gelinkt als kamers. Zodra 2500 punten hebt behaald gaat de deur naar de final boss open. Dit is het mega spook die ghosts summont terwijl jij een spoor van pellets moet opeten totdat de ultra pellet verschijnt. Hiermee word je groot genoeg om de mega ghost op te eten. Verder kan je natuurlijk power pellets eten om de spookjes op te kunnen eten.

Controls:

A of < = verander richting naar links.

W of ^ = verander richting naar boven.

S of v = verander richting naar beneden.

D of > = verander richting naar rechts.

Design Patterns en andere zooi:

Singleton:

Ik heb de singleton pattern gebruikt voor mijn event en game manager, omdat deze beide informatie en functies bevatten die ik graag vanaf overal wil oproepen. In het geval van de game manager gaat het om de game herstarten, score en levens bijhouden. Bij de event manager gaat het om het oproepen van events zoals het eten van een power pellet of het doodgaan van Pac-Man. Deze zitten dan ook allebei op een gameObject dat consistent is door alle scènes (DontDestroyOnLoad).

Finite State Machine:

Ik gebruik een finite state machine voor de spookjes omdat deze complex gedrag hebben bestaande uit drie states: achtervolgen, wegrekken en dood zijn. Het grootste verschil tussen deze states is waar ze naartoe gaan met hun pathfinding. Ze gaan naar Pac-Man, zo ver mogelijk weg van Pac-Man en naar de kooi. Verder bestuur ik vanuit de states de animaties en variabelen die hier betrekking toe hebben.

Observer:

Ik gebruik een observer pattern voor wanneer pacman doodgaat omdat er hier veel gebeurt. Wanneer pacman doodgaat wordt dat event opgeroepen en: wordt pacman gerespawned, gaan de spookjes terug naar hun startplek en base state, als het het laatste leven is wordt het spel gereset. Ik pas hetzelfde toe voor het begin en einde van blue mode wat wordt aangeroepen door de power pellet en de wegren state van ghosts.

A* Pathfinding:

Ik heb A* pathfinding toegepast voor de pathfinding van de ghosts, met name de eerste 4 delen van deze tutorial:

https://www.youtube.com/watch?v=-L-WgKMFuhE&index=1&list=PLFt_AvWsXI0cq5Umv3pMC9SPnKjfp9eGW

Ik heb vervolgens wel veel aanpassing gemaakt om de pathfinding real-time te laten updaten en niet diagonaal te laten gaan. Ik heb ook volgens de tutorial heap optimization gebruikt voor de pathfinding. Ik heb het grid van de pathfinding vervolgens ook gebruikt voor de beweging van pac-man zodat hij netjes binnen de hokjes blijft.

Generics:

De state machine en heap class zijn beide generic zodat deze gebruikt kunnen worden voor elke class. Dit is handig want als ik ooit een state machine nodig heb of heap optimization op iets anders wil toepassen kan dat heel makkelijk.

Reflectie:

Ik ben best tevreden mijn uiteindelijke game, ik heb delegates, events, singletons, a* pathfinding, generics en finite state machines gebruikt allemaal dingen die ik nog nooit eerder had gedaan (behalve de animator van unity dat is technisch gezien een state machine). Verder is het me zowaar gelukt te maken wat ik origineel voor ogen had. Ik heb veel geleerd en ben zeer tevreden met mijn eindproduct alleen doen doe spookjes zo af en toe nog niet wat ik wil, maar je kan niet alles hebben.