# Opdracht 3: Technical plan

**Opbouw:**
Het ingewikkelde in mijn spel zit hem ongetwijfeld in zowel de pathfinding en in de characters die zich juist moeten gedragen.
Daarom zal ik deze twee onderdelen apart behandelen. Ik noem eerst de functionaliteiten die bij beide onderdelen horen en vervolgens ga ik dieper in op de aanpak.

**Character:**
Voordat ik kan bepalen hoe ik dit spel het best kan maken, lijkt het mij handig om de verschillende functionaliteiten van de verschillende characters op een rijtje te hebben.

*Picaman: wordt bestuurd op basis van player input*
- lopen
- koekjes eten
- doodgaan
- enemy AI onschadelijk maken

*Good AI:*
- lopen
- koekjes eten
- doodgaan

Deze AI zoekt soms een koekje, en anders loopt hij heen een weer

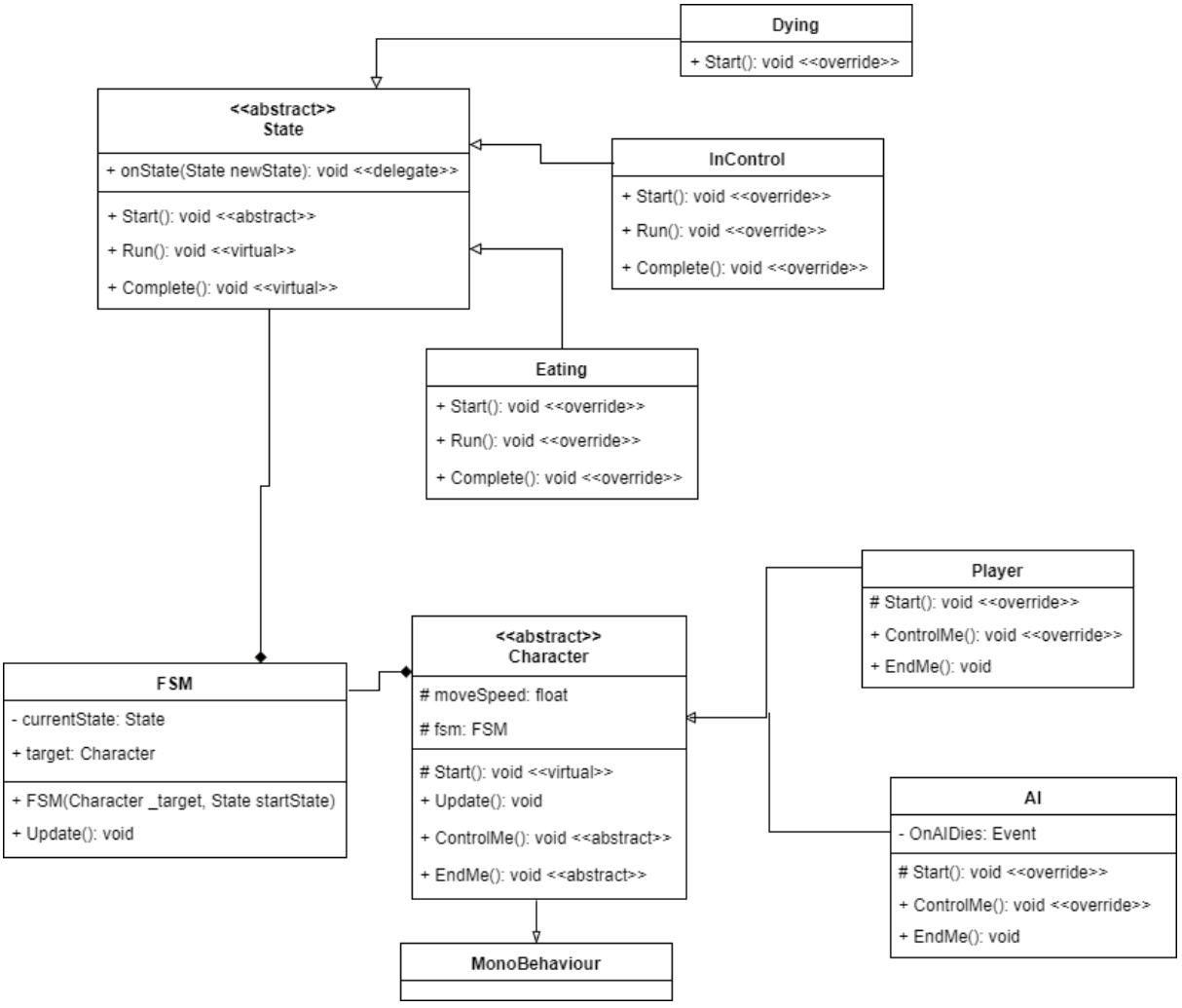
*Evil AI:*
- lopen
- good AI's opeten
- een 'good AI' worden

Deze AI loopt naar de speler als deze binnen een bepaalde range is, anders patrolled hij tussen 2 punten.

**FSM:**
Om te beginnen lijkt het mij goed dat alle characters gebruik maken van een FSM om de code gestructureerd te houden.
Ook wil ik dat de AI's en de speler van dezelfde (abstract) class inheriten: de Character class. Dit lijkt mij handig omdat ik op deze manier veel code duplicaties kan voorkomen. Alle characters moeten namelijk kunnen lopen, eten en hebben een FSM nodig om te zorgen dat ze maar één ding tegelijk kunnen doen.
Belangrijk om hierbij te vermelden is dat de FSM alleen met de base class kan communiceren aangezien ik anders 3 verschillende State Machines zou moeten maken.

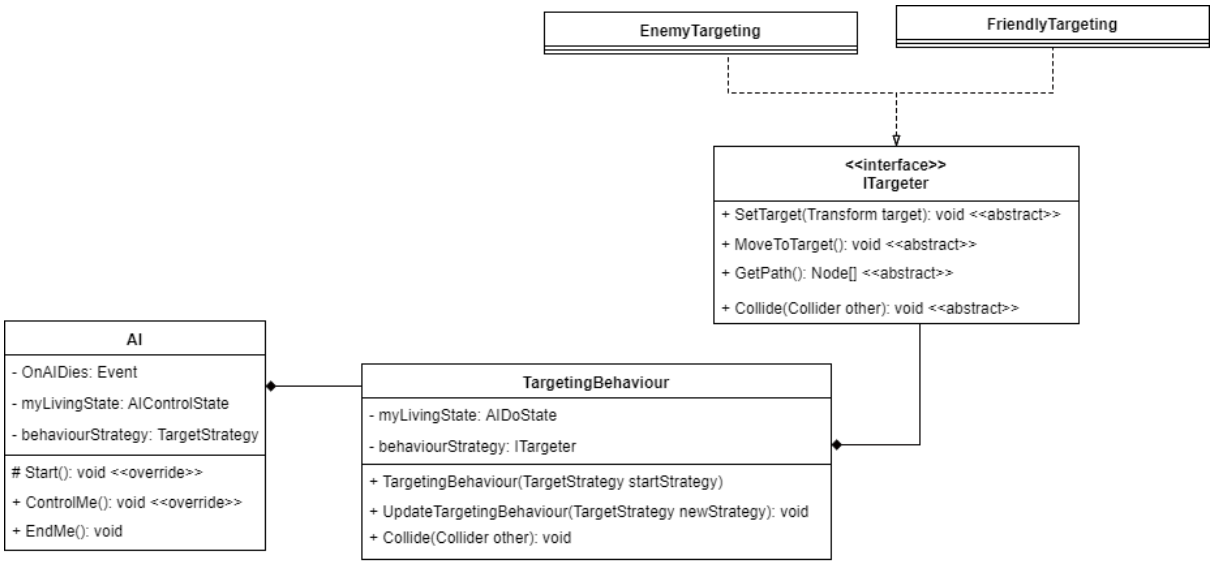
Het lijkt mij verder handig om binnen deze FSM 3 states te maken: eating(aangezien dit niet onderbroken kan worden, behalve door dood te gaan), inControl(een state waarin de AI zijn

ding doet of de speler de character bestuurt) en een dying state(die op het laatst uitgevoerd wordt als een character sterft).
Voor nu hebben we dus de volgende structuur gecreëerd:




**AI:**
Een volgend dilema waar we nu op komen is de AI: deze moet eigenlijk 2 gedragingen hebben en het liefst wisselen hiertussen zonder dat er een nieuwe AI geinstantiate hoeft te worden. Het belangrijkste verschil tussen deze 2 AI's is dat een evil pacman de speler zoekt als deze binnen zijn bereik is, terwijl een bevriende pacman naar koekjes zoekt.
Een manier om dit gedrag te laten veranderen heb ik gevonden in het strategy pattern.
Dit zou ik kunnen toepassen door een ITargeter interface te maken die door zowel een EnemyTargeting als door een FriendlyTargeting class geïmplementeerd wordt.
Vervolgens heb ik nog een TargetingBehaviour class nodig die het juiste gedrag aan kan roepen.

In UML zou deze structuur er als volgt uitzien:

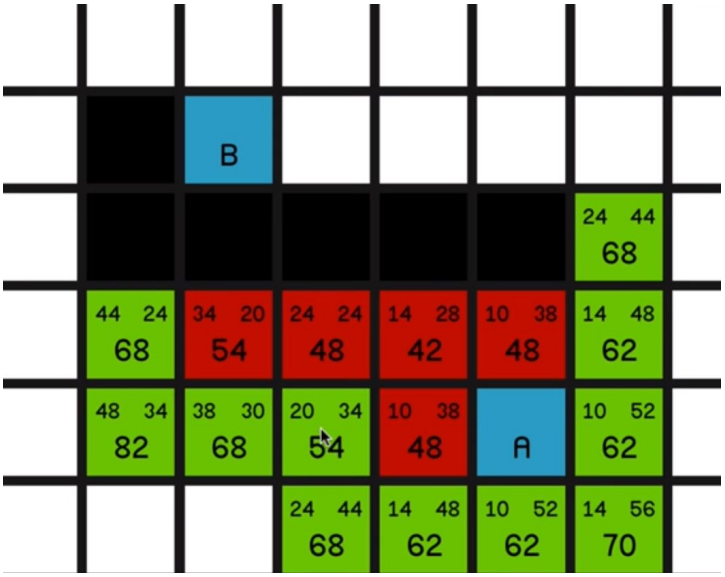



Ook kan ik voor de AI gebruik maken van een EnemyManager. Dit kan mij helpen om makkelijk het overzicht te behouden over de verschillende AI's. Dit script bevat dan zowel een Singleton als een Observer Pattern.

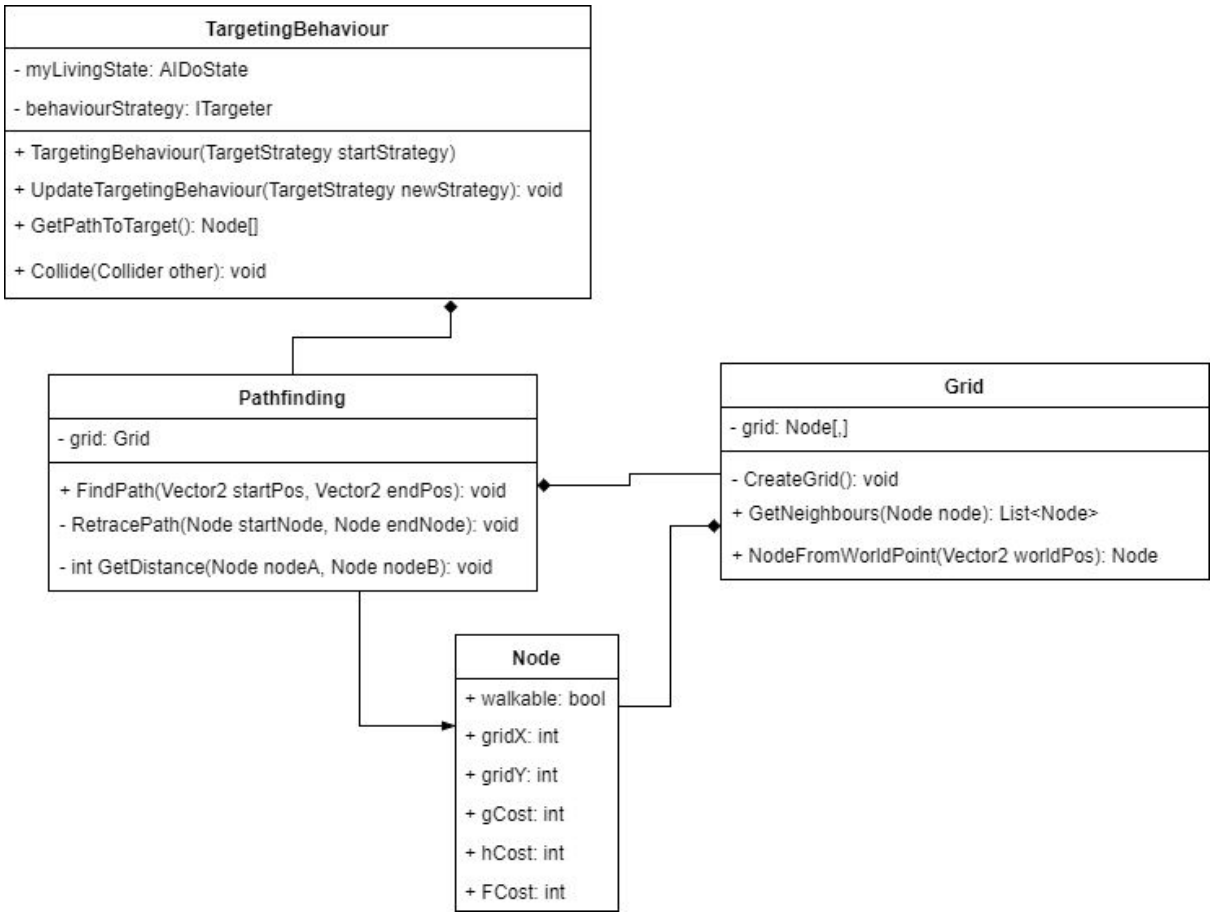
**Pathfinding:**

De TargetingBehaviour class moet gebruik maken van een pathfinding systeem om het pad naar de target kunnen vinden. Hiervoor heb ik dus een pathfinding systeem nodig. Het systeem dat ik gekozen heb is A*, omdat dit een systeem is wat geschikt is voor 2D en omdat ik hier een hele goede tutorial-serie* voor gevonden heb.

De basis van dit systeem is dat er een grid gegenereerd wordt van Nodes en dat er steeds per tile wordt bekeken of die geschikter is om dichter bij de targetNode te komen. Deze Nodes worden vanaf de startNode(= pacman positie) opgezocht.



De UML van de pathfinding ziet er dan als volgt uit:



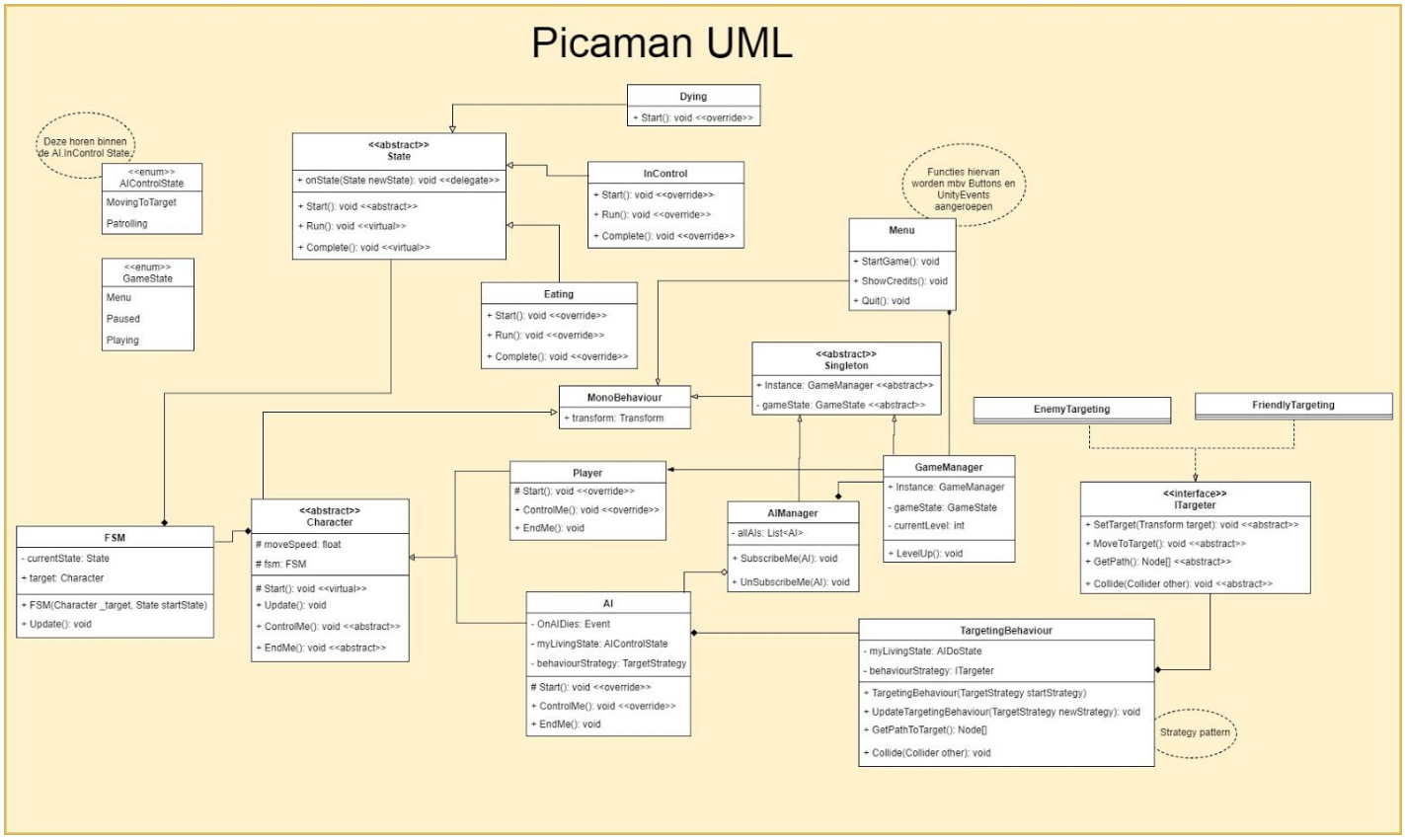

Verder ga ik voor dit spel een Singleton gebruiken voor de GameManager. Dit script bestaat gedurende het hele spel en wordt gebruikt om levels te laden, het spel op te starten en om het spel af te sluiten.

**Conclusie:**
De design patterns die ik in dit spel ga toepassen zijn in ieder geval:

- strategy pattern
- Finite State Machine
- Singleton
- Observer pattern

De grootste uitdaging voor mij zit hem vooral in het toepassen van de Strategy Pattern, de FSM en in het maken van het pathfinding systeem. Ik heb al deze dingen nog nooit eerder gemaakt namelijk.

Uiteindelijk zal dit ongeveer de structuur van het spel worden:




**Bronnen:**
*Tutorial-serie A* pathfinding:*
*https://www.youtube.com/playlist?list=PLFt_AvWsXl0cq5Umv3pMC9SPnKjfp9eGW*