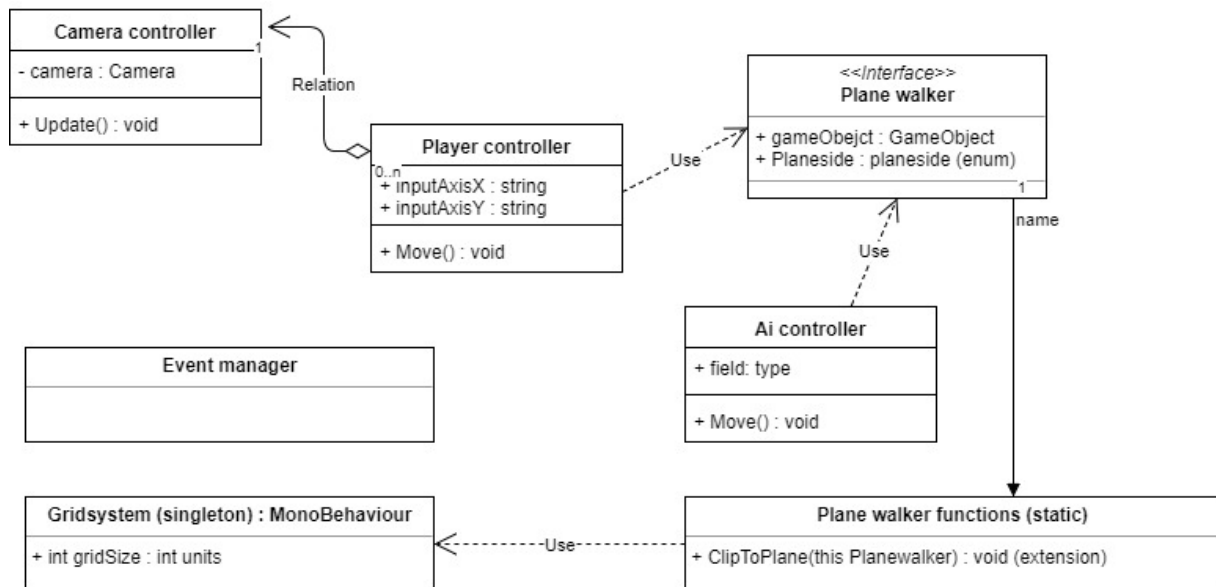
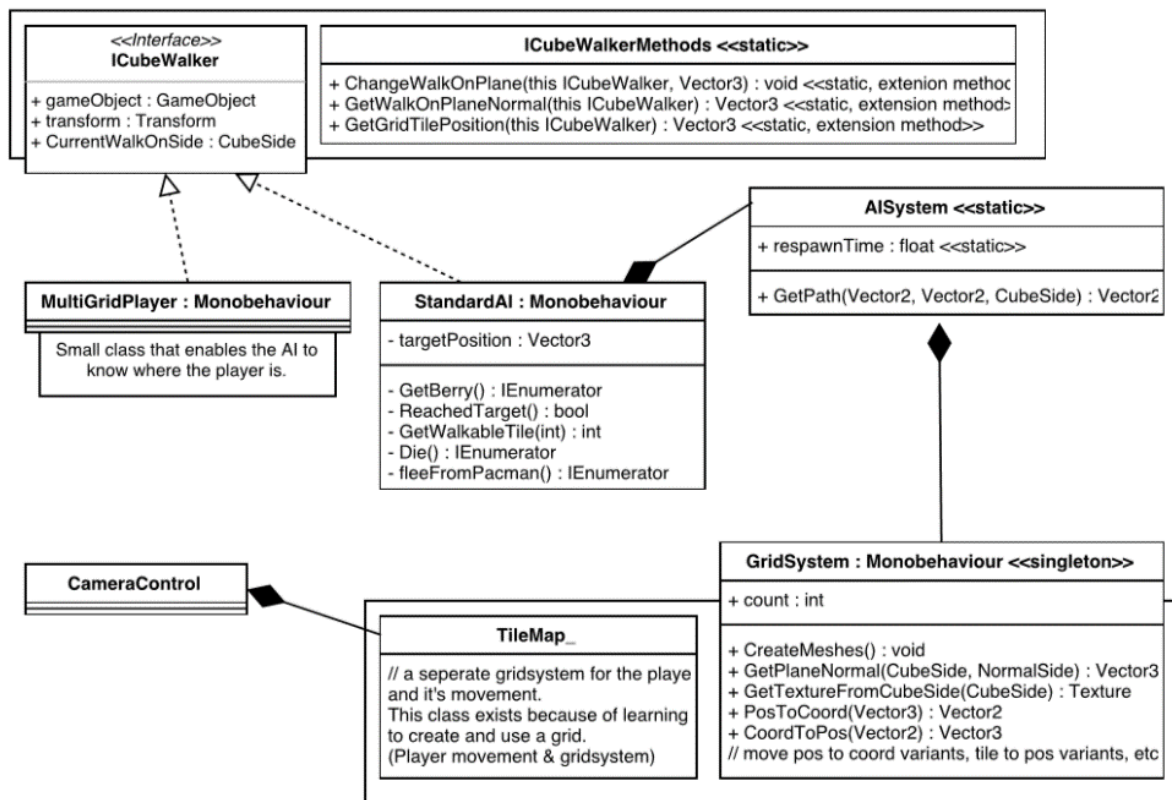


## UML uitleg

Oud:



Nieuw:



### **Uitleg:**

Omdat meerdere classes soms elkaar of een ander moeten kunnen bereiken, hebben we besloten om een aantal classes op te zetten als een (soort) singleton. Hierdoor kunnen classes makkelijk bij de gewenste class zonder direct alles te hoeven linken. Door de speler gameobject een singleton component mee te geven kon bijvoorbeeld alle AI altijd de plek van de speler makkelijk vinden via dit 'singleton'-achtige component.

Wij hebben twee grid-systemens, de reden hiervan is dat we allebei ons eigen leermoment wilde en dus een eigen variant ging schrijven om het leren van grids en het gebruiken daarvan. Hierdoor hebben we het MultiGridPlayer class het leven in geblazen. Deze class kan voor het AI system vertalen waar de speler zich bevindt aangezien de speler en de AI allebei een apart grid-systeem gebruiken om zich te bewegen en oriënteren.

Het ICubeWalker interface had als nut dat het bepaalde welke objecten op de muren, etc. zouden kunnen lopen en zou eigenlijk net zo goed een abstracte base-class kunnen zijn. Het leek echter handiger om zo min mogelijk nested inheritance in alle classes te hebben. Hierdoor is besloten om in plaats van standaard functies en een base class te gebruiken, een interface te gebruiken en deze functionaliteiten mee te geven via extension methods.

### **Veranderingen:**

Er zijn best een aantal dingen veranderd of toegevoegd door het maken van het project heen. Meesten zo klein dat die niet waardig zijn om hier te benoemen.

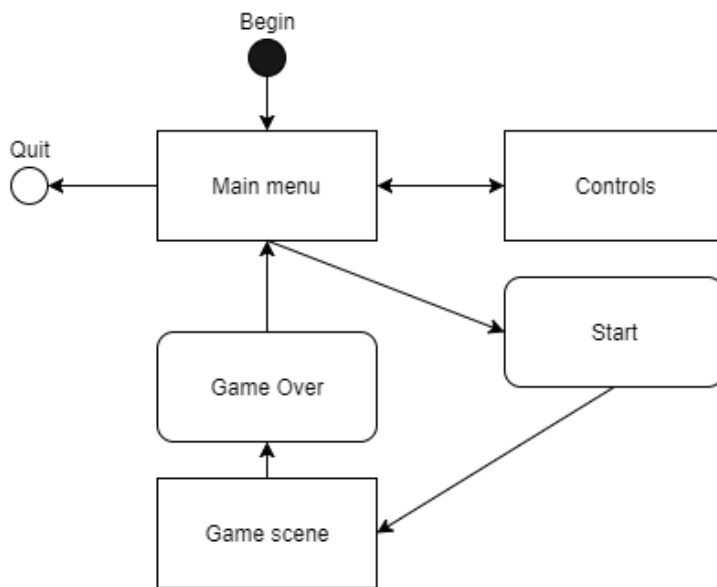
Een van de belangrijkste veranderingen is het feit dat we nu twee grid-systemen hebben, dit hadden we van te voren niet bedacht, maar heeft geen problemen met zich mee gebracht en is ook niet ontstaan uit een probleem. We hebben alleen wat kleine aanpassingen moeten maken zoals een offset toevoegen omdat links onder bij beide grids 0.5 verschilden op elke as. Dit kwam omdat bij één van de systemen links onder echt het hoekje was van het grid en bij de ander links onder het middelpunt van het vlak links onder was.

### **Volgende keer anders:**

De AI was gebouwd op het idee dat het ook in een compute shader berekend kon worden, maar hierdoor werd de code erg slecht. Dit is dus iets wat ik volgende keer anders zou doen, de AI zo opstellen dat het beter is en niet op GPU werkt of zeker weten dat we ook daadwerkelijk de code moeten aanpassen zodat hij zou kunnen werken op de GPU.

Ook zouden we naar een andere manier naar de player willen kijken. Nu kan je switchen van zijdes. Op dit moment doet hij wat hij moet doen maar zodra we het level (cube) veranderen van vorm dan werkt hij niet meer. Misschien dat ik vanaf het begin wil kijken naar een player dat op "muren" kan lopen ongeacht hoe scheef of recht deze staat. Nu is er gekeken wat hij voor deze opdracht moet doen en na meerdere pogingen werkt het het best door middel van een grid.

**Activity diagram:**



Scene 1:  
Main menu

Scene 2:  
Game Scene