

# Heavy OOP

## Advanced

## Inhoud

Unreal en Visual studio .....	3
Create class wizard .....	3
Pawns en Controllers.....	3
Default Pawn en Controller .....	3
Bug in Unreal .....	3
SquadGame .....	4
ASquadGameCharacter .....	4
ASquadGamePlayerController.....	4
ASquadGameAIController .....	4

## Unreal en Visual studio

Net als anderen heb ik veel problemen gehad met de samenwerking tussen unreal en visual studio. Visual studio start vaak niet op of juist meerdere keren. Visual studio werkt heel erg traag samen met unreal. Daarnaast is mijn project meerdere keren zo erg gebroken dat ik het hele project opnieuw moest beginnen. Hier ben ik helaas erg veel tijd aan kwijt geraakt.

### Create class wizard

Via een tutorial had ik ontdekt dat er een wizard is om c++ classes mee aan te maken. Deze is erg simpel, maar als je op de knop 'Show all Classes' klikt krijg je een overzicht van alle standaard classes in Unreal. Voor mijn AI class kon ik simpelweg zoeken naar 'AI', en de AIController selecteren. Er wordt dan automatisch een cpp en header file aangemaakt met de juiste base class en includes.

## Pawns en Controllers

Voor mijn project wilde ik een squad mechanic maken waarbij de speler gevolgd wordt door een of meerdere AI characters en waarbij de speler ook kan switchen tussen deze characters. De manier waarop Unreal zijn classes heeft opgebouwd zorgt er voor dat dit heel makkelijk gemaakt kan worden. Maar dan moet je wel eerst deze classes leren kennen. De twee belangrijkste classes zijn APawn en AController.

APawn is de class voor alles wat bestuurd kan worden door spelers of AI. De class heeft een heleboel functies voor physics die op de pawn uitgevoerd kan worden. De AController class heeft twee child classes; APlayerController en AAIController. Een controller heeft een referentie naar een pawn die hij bestuurd. Een pawn kan maar door één controller bestuurd worden, en een controller kan maar een pawn besturen. Dit kan aangepast worden door de Possess(APawn\* Pawn) class in de AController. Om de controller uit te wisselen tussen een AIController en een Playercontroller moet je dus bij allebei de Possess functie gebruiken.

## Default Pawn en Controller

In Unreal heb je een GameMode class. Hierin staan een aantal standaard instellingen beschreven. De template van Unreal maakt hier ook gebruik van. De blueprint die als default pawn is geselecteerd spawn hierdoor automatisch op het spawnpunt. Ik ben er nog niet helemaal uitgekomen hoe dit precies werkt. Ik kan me voorstellen dat dit voor blueprints makkelijk is om te gebruiken, maar voor mijn project lijkt het mij handiger om de objecten gewoon handmatig in de scene te zetten.

### Bug in Unreal

Blijkbaar zit er een bug in Unreal waardoor het hele programma meerdere keren bij mij gecrasht is. Dit gebeurt wanneer ik een kopie van de default pawn blueprint in de scene zet, en deze vervolgens probeer te koppelen aan de PlayerController doormiddel van de Possess functie. Om dit op te lossen heb ik gewoon een kopie gemaakt van deze blueprint (TopDownCharacter) . De squadmembers maken gebruik van de andere blueprint (SquadGameCharacter). De blueprints zijn identiek en maken allebei gebruik van dezelfde c++ class (ASquadGameCharacter). Wanneer ik het op deze manier opbouw crasht Unreal niet. Misschien doe ik iets heel erg doms omdat ik het concept van de Default Pawn nog niet helemaal begrijp, maar van wat ik er op internet van terug heb kunnen vinden is dit een bekend probleem in Unreal.

# SquadGame

## ASquadGameCharacter

Dit is de APawn class voor het character en is grotendeels hetzelfde gebleven als de standaard vlass van de TopDown template. Het enige verschil zit in de functie die de muisklik uitvoert. Wanneer de speler op een ander SquadGameCharacter klikt loopt hij daar niet naar toe, maar switch hij van character door de pawns van beide controllers uit te wisselen.

## ASquadGamePlayerController

Deze class is ook grotendeels hetzelfde gebleven als de TopDown class. Het enige dat ik veranderd heb is dat ik de Possess functie override. Nadat de Super::Possess functie wordt aangesproken wordt ook de static member value voor de PlayerCharacter in de ASquadGameAIController aangepast naar de nieuwe pawn die gepossessed wordt door de PlayerController

## ASquadGameAIController

Dit is de nieuwe class die ik geschreven heb. Zoals hierboven al stond heeft deze class een static pointer naar de pawn die bestuurd wordt door de speler. Daardoor volgen alle AI characters hetzelfde character. De class heeft ook een possess(APawn\* pawn) functie om een nieuwe pawn te besturen. Daarnaast heeft de class een Tick functie die elk frame wordt uitgevoerd. Wanneer de afstand tot de speler groter is dan een bepaalde waarde dan loopt hij naar de speler toe.

## Unreal en Github

Het is me niet gelukt om het hele project op GitHub te krijgen. De classes die hierboven beschreven zijn staan wel op mijn repository. Ik ga kijken of ik het project nog op een andere manier kan delen.