Clock Blockers

The revolutionary new First-Person Shooter that focuses on foresight and tactical thinking.

# Forord

Det er noen andre dokumenter hvor jeg har skrevet litt mer informasjon. Et simpel Game Design Document, og et dokument jeg bare skrev på gøy som jeg hadde tenkt til å skrive litt mer i, men enn så lenge bare omhandler tilgjengelighetsnivået til klasser. Begge disse ligger under ‘Design’ mappen. Jeg har også skrevet en liten ‘testing guide’ i Readme.md filen.

# Prosessen

Mot slutten av februar, før jeg fant ut at eksamensoppgaven var lagt ut, fikk jeg en idé om å gjenskape [Clock Blockers](https://www.youtube.com/watch?v=CBawCe6du3w) av Corridor Digital. Jeg hadde ikke sett denne videoen på flere år, så jeg er ikke helt sikker på hvorfor jeg plutselig kom på det, men det var da sånn det gikk. Dette er grunnen til at jeg hadde utviklet det i beta-versjonen 2020.1(Som da akkurat hadde kommet ut i Alpha), og ikke 2018.4 (Som jeg tror var den anbefalte versjonen, men nå som jeg prøver å finne informasjon om det, så finner jeg ingenting).

Det startet med at jeg prøvde å lage en versjon som registrerte hver eneste kommando jeg gjorde (<Move left with x-input of 0.8, y-input of 0.4, 14.5seconds after start>, <Rotate 0.15 degrees to the right, 14.55 seconds after start>, etc… ), men etter et par dager med å implementere denne fant jeg noen åpenbare problemer med denne løsningen; Hvis en eneste kommando ikke ble gjort i riktig frame (Kanskje en lag-spike eller noe sånt) så ble det en liten desync, og fra og med da ville den alltid være feil resten av livet til den klonen.

Etter jeg hadde kommet til konklusjonen at denne løsningen ikke kom til å skaleres bra, tenkte jeg at jeg i hvert fall kunne skrive ned en GDD, noe som jeg tror var den beste tingen jeg gjorde under utviklingen av dette spillet. Det gjorde flere ting mye mer åpenbart for meg. Hovedsakelig det om at; Klonene trenger å kunne gjøre egne valg etter at de har blitt ferdig med å kopiere mine handlinger, og hvis dette er sant (som det er), så er det relativt åpenbart at jeg burde implementere handlings-kopiering med samme «boundaries» som valg-AIen kommer til å måtte bruke, så da begynte jeg på en ny iterasjon: Ta opp translasjons-dataene til spilleren hvert x sekund, samt alle handlinger den gjør. (Skyter, kjøper våpen, etc …). Så å si etter jeg hadde kommet til den konklusjonen så stoppet jeg med å utvikle spillet til cirka 2 måneder senere (Vet ikke helt hvorfor jeg stoppet).

Da jeg kom tilbake, mot slutten av april, så brukte jeg mesteparten av tiden min på å få en A\* Pathfinder til å fungere på auto-genererende markører rundt på kartet. I retrospekt burde jeg definitivt hatt ikke brukt markører, da jeg visste egentlig allerede da at dette ikke kom til å være den mest optimale løsningen, og at den må byttes ut før eller senere. Det beste med den løsningen var det at jeg fikk den til å fungere i Editor-view, slik at den ser fin ut. «Baby’s first tool».

Etter jeg hadde implementert markør-generasjon, samt pathfinding mellom to+ markører, var det på tide å implementere en FSM til AI. Jeg hadde to versjoner av denne også. I den første versjonen så var det selve State’n som bestemte når den skulle byttes, nesten som hvordan gamle operativ-systemer fungerte. Det var noen ganske store problemer med dette; Jeg måtte, på en eller annen måte, «innsprøyte» bytte-betingelsene til alle mulige state-transitions som state’n hadde når den ble laget, og hver enkel state måtte forandre seg dersom jeg la til nye states, og det bryter jo good ol’ SOLID principles. Den andre implementasjonen, som jeg byttet til relativt raskt, hadde bare dumme states, og dumme betingelser, og det var jobben til selve State Machine’n å bytte mellom dem.

## Features fra pensum

* Agenter
  + Spillet mitt er basert på bruken av kloner av spiller-karakteren, og disse klonene tar sine egne valg, ved hjelp av:
* FSM, Finite State Machine
  + Agentene som ble nevnt tidligere bruker en FSM for å bestemme valg.
* Unity
  + Ganske åpenbart; Spillet er utviklet i Unity (2020.1)
* Steering Behaviour
  + Agentene mine tar et valg (Enn så lenge er alle valgene eksterne, så det eneste valget dem tar er ‘Følge valgene som blir laget for meg’, men *:shrug:*
  + Dersom de velger å bevege seg, så finner de en rute ved hjelp av A\*, og deretter..
  + Bevege seg langs ruten.
* Pathfinding
  + Jeg bruker A\* Pathfinding, ved hjelp av markører som blir laget automatisk, til å finne rute mellom to steder på kartet.
  + Pathfindingen er helt ‘concurrent’, det vil si at den ikke fryser spillet når den kjører (Det vil ikke si at den er parallelisert derimot – alt går på en kjerne enn så lenge).

## Kjente bugs

* Det er to bugs, som jeg bare har opplevd i Editor Mode, en av dem har jeg rapportert til Unity (uten svar enn så lenge).
  + Den første, og den verste av dem, er det at iblant så krasjer Unity helt. Dette skjer som oftest etter jeg lager en klone. Jeg er ganske sikker på at det er en memory-issue (Plutselig går Unity fra 3-4GB ram til >20GB), men Profileren krasjer, og Memory Profileren klarer ikke å åpne snapshot’et som blir laget når minne-bruken blir høy, men uten å krasje.
  + Den andre er en mye mindre relevant issue, og jeg vet hvorfor det skjer, men vet ikke hvordan jeg kan fikse det. Hvis det plutselig blir en lag-spike (som skjer hver frame så å si i Play Mode, i editoren (d.v.s, ikke i en build)) vil kameraet gå crazy. (Jeg flyttet musen x cm, det burde flytte musen y pikseler på skjermen, men siden den er basert på frametime, hvis frametime plutselig øker voldsomt på grunn av spike, vil kameraet bevege seg mye mye lengre enn det burde.
* Du kan ikke hoppe dersom du har hoppet siden du sist var nære bakken. Det vil si at, dersom du faller uten å hoppe, så kan du hoppe mid-air. Jeg har prøvd å fikse dette på flere vis, men collision detection er så utrolig buggy i Unity virker det som (‘isGrounded’) spazzer hele tiden dersom du går framover.

* Det er en annen bug som jeg både vet hvorfor skjer, og hvordan jeg kan fikse, men ikke har kommet til å fikse enda er:
  + AI’en til klonen er veldig dum når det kommer til å hoppe (og alt annet for den del ...). Den vil alltid hopper dersom den kolliderer med en vegg. Antagelsen er da at den prøver å gå framover, men treffer en vegg. Problemet kommer da den går rundt hjørner – den vil alltid hopper når den går rundt et hjørne, eller når den er i luften, og prøver å gå under en «bru», og den kommer fra luften. Grunnet buggen jeg nevnte tidligere (den med hopping), så vil den hoppe dersom den kommer fra oven og treffer en vegg, selv om han skulle gå under.

## Kildehenvisninger

Jeg bruker noen packages til Unity, hovedsakelig Power Inspector og Graphy (samt flere packages laget av Unity, som Burst, Input System, og EditorCoroutines)

De stedene hvor jeg har brukt kode vil jeg hatt laget en link til hvor jeg fant koden. For eksempel:

BetweenNames\CustomEditors\Editor\Attributes\ReadOnlyFieldAttribute.cs

Har en kommentar ‘// <http://answers.unity.com/answers/801283/view.html>’. Dette henviser jo da til en link til hvor jeg fikk den koden.

Jeg beklager om dette ikke var det formatet dere foretrekker, men personlig føler jeg at dette er mer hjelpsomt en liste over kilder som ikke har et åpenbart bruksområde «Hva fikk du ut av denne kilden?»