

Game engine INDA24++

Vad ska ni göra?

Vad är projektet?

Vårt projekt är ett spel och tillhörande spelmotor. Spelet är ett multiplayer dungeon crawler roguelike med party game element. Spelarna tar sig mellan olika rum och slåss mot fiender och samlar på sig items tills de hittar bossrummet där bossen besegras genom ett minigame i stilen av ett party game. Spelmotorn ska tillhandahålla alla funktioner som spelet behöver. Detta inkluderar rendering av low poly modeller i top down perspektiv, ECS, procedural generation av kartor och multiplayer.

Scope och begränsningar?

Vi vill inte göra en helt färdig spelmotor som är till för alla möjliga spel, utan vi begränsar oss till en motor som klarar av att hantera det vårt spel kommer kräva. Det innefattar bland annat 3D-rendering och multiplayer. Det vill säga, den minimum viable produkten vi kan få fram är en motor som klarar av att hantera en demo på det spel vi tänkt skapa. Vi begränsar oss även till Windows och Linux, där endast Wayland kommer att stödjas på Linux.

Vi kommer därmed inte att skapa varken en helt färdig spelmotor med massvis av funktioner, och begränsar oss till de funktioner vi anser nödvändiga av en motor för att skapa ett inte heller helt färdigt spel.

Vem ska göra vad? (Alla skriver sin egna)

Physics and Math, Skye

Arbetet med Physics & Math innefattar att implementera kollision, acceleration och dylikt. Om tid finns även vektorer och matriser från grunden.

Player Controller, Oscar

Arbetet med Player Controller innefattar att designa och prototypa samt implementera det som gör att spelarna kan styra sina individuella karaktärer. Det innefattar spelarrörelse, interaktioner med spelvärlden, använda föremål och attackera fiender.

Arbetet börjar med att iterera designer och prototyper utanför den egna spelmotorn. För det kommer en avancerad spelmotor användas som t.ex Unity eller Godot. När spelmotorn är tillräckligt klar för att användas går arbetet över till att implementera spelarkontrollern i den egna spelmotorn.

World Generation, Lukas och Theodor

Arbetet med world generation innefattar att skapa de algoritmer som genererar spelvärlden. Det kommer bestå av olika typer av rum som kopplas ihop med hjälp av korridorer.

Arbetet kommer börja utanför själva huvudprojektet då det är betydligt lättare att utveckla om man kan visualisera algoritmerna, vilket då kräver att rendering är klar. Eftersom detta kan komma att ta ett tag påbörjas arbetet i exempelvis raylib och kommer sedan föras in i projektet när rendering är redo. Arbetet kommer innefatta att prototypa olika algoritmer för att generera rummen, samt hur de kan anslutas på ett sätt som gör att alla rum går att nå, vart spawnables (fiender, items, etc) kan genereras och så vidare.

ECS, Dexter och Zimon

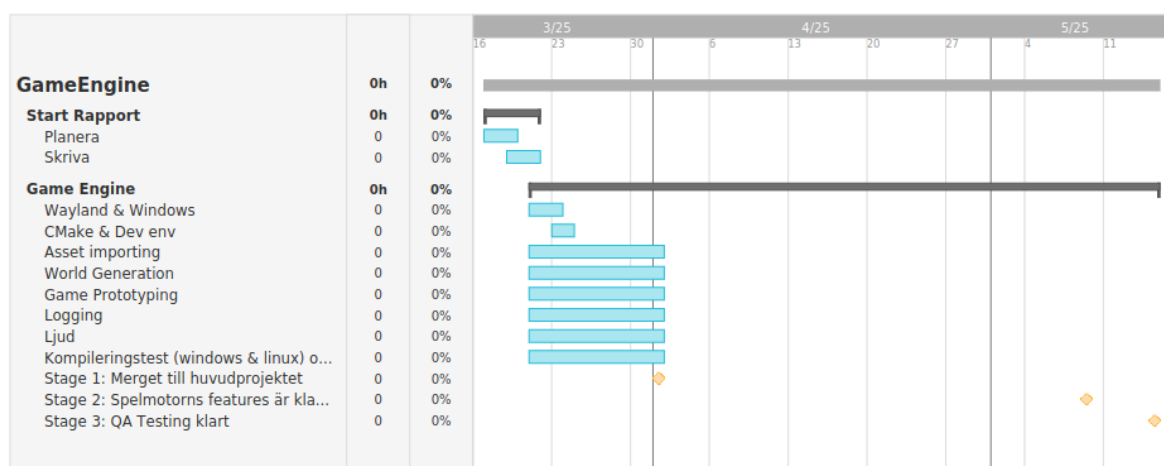
Ett Entity Component System (ECS) är en viktig del av vår spelmotor för att det ger skalbarhet och flexibilitet. Samtidigt är det enklare att debugga och testa delar av motorn och spel för att systemen är isolerade.

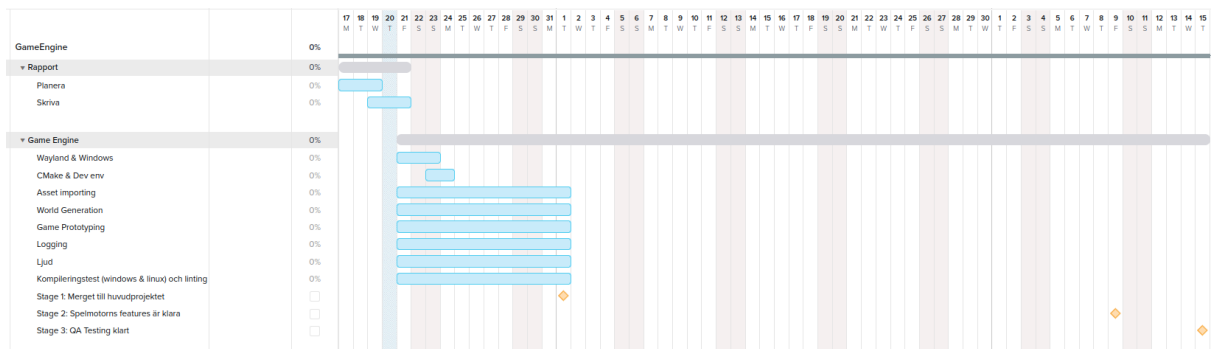
Systemet kommer att påbörja utveckling utanför huvudprojektet och kommer sedan att läggas ihop med de andra delarna av projektet som samtidigt utvecklas av gruppmedlemmarna.

Network, Viktor och Dexter

Vi vill skapa ett spel som klarar av att hantera flera spelare, vilket kräver någon form av nätverkskod. Då nätverkskoden kommer behöva anpassas väldigt mycket beroende på vad spelet kräver, kommer det att arbetas med parallellt med utvecklingen av spelet.

Tidsplan





Hur ska ni samarbeta?

Hur ska ni använda git/github? (Code reviews, Möten)

Vi ska använda ett github repo och dela in varje del vi jobbar på i olika branches. Planen är att ha en main-branch som alltid är stabil och fungerande: Den ska då också gå att kompilera både på linux och windows. Utveckling av nya funktioner kommer ske på separata branches, och sedan kommer det att göras pull requests till main när de är redo. Då kommer koden behöva kollas igenom av minst en annan person som har god erfarenhet i C++ och det måste ses till att den kan kompileras i både windows och linux..

Ska ni ha möten?

Vi kommer ha möten 1-2 gånger i veckan, ett fysiskt efter eller innan veckans lektion beroende på tid, samt ett digitalt under veckan. Detta för att se till att saker blir gjorda och att alla kan få hjälp om de kört fast.