**Documentatie GDEV2 multiplayer/networking**

Daan van Westerlaak – GDEV2

**Concept**

Ik had dit jaar voor het eerst een Fire Emblem game gespeeld, nadat ik er vorig jaar door klasgenoten aan werd geïntroduceerd.

Van dit spel vind ik het combat systeem en dan vooral de skill activatie erg interessant, dus leek mij dit een leuke uitdaging om te maken.

Wat ik zo interessant vond aan de skills is dat ze eigenlijk op alles kunnen reageren en allemaal effect op elkaar kunnen hebben, dus dit leek mij een interessant iets wat betreft networking.

**Uitkwering**

Het is mij niet gelukt om de daadwerkelijke game te bouwen in verband met tijd en andere deadlines, maar ik heb eigenlijk wel alle mechanics (zoals skills, characters en wapens) al gemaakt.

Deze skills en combat werken wel al in het “PrototypeCombatTest” script. Hier worden twee prototype characters aangemaakt met twee verschillende skills en de zelfde wapens. Dat wordt de combat tussen deze twee karakters gestart en gelogd in de console.

**Ontwerp**

Toen ik na ging denken over hoe ik de structuur van de networking communicatie ging opzetten, bedacht ik mij dat het handig was om de server alles te laten doen en alleen visuele feedback te sturen naar de cients.

Toen kwam ik er echter achter dat alleen het player object commands kan sturen naar de server en daar was ik geen fan van, omdat dan de player class erg vol zou raken. Hierdoor besloot om de player class (die ik al had geschreven met gedachte dat het meer communicatie code was dan echt een player) een functie te geven waarbij je een delegate als parameter mee kon geven zodat de player class deze dan op de server kon uitvoeren.

Hierbij liep ik tegen een potentieel probleem aan. Dit probleem is dat de parametertypes van commands geen ‘object’ mogen bevatten. Hierdoor zou je dus een hoop overloads moeten schrijven om te zorgen dat je elke mogelijkheid door kan sturen naar de server, hierbij was echter nog een probleem; het totaal aantal van deze mogelijkheden liep in de miljoenen.

Ik besloot toen, omdat ik die overloads ook niet echt nodig had, om alleen parameter loze delegates te accepteren en door te sturen naar de server.

Ik heb het zelfde concept ook gebruikt om vanaf de server delegates naar alle clients of één specifieke client te sturen.