Gruppe 33

Gjennom denne perioden har vi lært oss veldig mye, vi har lært objekt-orientert programmering og sett mer av sammenhengen mellom Programmeringsfaget i høsten og programutvikling i år. Vi har klart å lage et spill, med hjelp av forelesningene, oracle-nettsidene, student assistenter og masse google søk.

På starten var vi helt nye når det gjelder programmering, men etter dette prosjektet føler vi oss mer rustet til java-relaterte fag og andre store prosjekter. Vi har lært oss veldig mye, ikke bare om java og fxml, men også om programmerings logikk og sammenhengen med matten.

Vi er svært fornøyd med innsatsen vår gjennom dette semesteret, vi har møtt flere ganger i uken siden prosjektet startet og jobber effektivt. Vi er stolte med vårt sluttresultat.

Med hensyn til spillet så føler vi at vi har gjort en god jobb, men føler at vi kunne ha kodet mer enn å bare lese kode teori. Vi skjønte sent at den beste måten å faktisk lære å kode på, er å bare prøve og feile om og om igjen. Men det positive i det er at når vi først kom i gang så ble det lettere å kode og prøve på ulike metoder og funksjoner på å løse ulike hindringer. Samtidig ble det veldig lett å bruke sider som f.eks. Oracle.

Et av de tingene vi kunne ha gjort bedre er å fokusere på utvidelser som mobil applikasjon, noe som også kan hjelpe oss videre i ingeniør utdanningen. Når vi startet på dette så vi sammenhengen mellom javaFX og Gluon, og fikk mye av det til, men ikke nokk til å fullføre det feilfritt til fristen. Dette er noe vi selv frivillig kommer til å jobbe med etter å ha levert applikasjonen som en fremtidig patch.

I fremtiden vil vi først og fremst se helheten i et slikt prosjekt mye lettere. Vi har også nå lært at en oppskrift/mal er ganske viktig når du tar for deg slike store prosjekter. Konkrete mål og strukturert arbeid er et nødvendig krav.

Når vi lagde dette spillet kan vi si at vi alltid satt sammen, når vi kodet og implementerte spillet. Koding var helt nytt for oss begge, når det gjaldt et slikt prosjekt. Dermed følte vi at, vi forsto og lærte best når vi satt sammen. Teknisk sett lærte vi hverandre opp. Etterhvert ble det slik at vi selv kunne bytte og fikse på små ting, nettopp fordi vi sammen hadde laget spille. Men i fremtiden tenker jeg at det er lettere å dele opp oppgaver, da man vet at begge parter er erfaren nok innenfor koding.

Bra kode →

Kollisjon:

For kollisjon har vi brukt det samme for Player-Enemy, Bullet-Enemy og Player-PowerUp. Dette skjer i «gameUpdate» metoden, hvor objektet sitt siste posisjon blir lagret og sammenlignet med det andre objektet med en matematisk utregning. Siden vi bruker enkelte grafiske objekter , regner vi ut kollisjonen med ved å bruke ved pytagoras-formelen og sammenligner om radiusen til begge objektene er mindre enn distansen, altså om: radiusEnemy + radiusPlayer > nåværende posisjon(Math.sqrt(dx * dx + dy * dy)).

PowerUps:

Når en fiende dør, har vi laget en sannsynlighet for alle de ulike powerUps`ene til å dukke opp, på samme posisjon som fienden ble eliminert. Dette skjer i «GameUpdate» metoden og blir kjørt når en fiende dør og fjernes fra enemies-listen. Alle de ulike powerupsene har forskjellige drop-sjangser og styrker. Vi har implementert 4 forskjellige: Ekstra liv, Ekstra styrke, Ekstra dobbelStyrke og Slowmotion timer. Når du har fått 5 liv, blir drop-sjangsen til ekstra liv veldig lite. Når du plukker opp flere styrke-powerups, kan du øke din skytekraft, hvor du får flere skudd, og kan få tilgang til å bruke en Nuke som sprer skudd i for av en sirkel som øker i radius. Den siste powerupen er en slowmotion timer, som minsker alle enemies sin fart med en konstant i et bestemt tidsperiode. Enemies blir da treigere, mens player har samme fart.

Filbehandling:

Det er mulig å både Lagre og laste opp lagret fil i spillet. Til å lagre spillet bruker vi en Printwriter som setter opp bestemte variabler i en tekst fil. Vi bruker en FileChooser til å velge destinasjon til lagringsplassen og velge filnavn, mens all lagringen skjer i en egen klasse. Vi lagrer de viktigste egenskapene til spillet, som gir brukeren til å lagre og senere fortsette der han slapp.

For å loade spillet burker vi igjen en FileChooser hvor du selv kan velge filer som har blitt lagret. Dersom du velger en fil som er korrupt eller ikke har blitt lagret fra spillet, får du opp en feilmelding på skjermen. Vi har laget en Load() metode og bruker den sammen med en BufferedReader , hvor vi leser linje for linje i tekstfilen, og setter hver linje inn som en parameter i Load metoden.

Nivåer:

Vi har implementert ulike nivåer(waves) med ulike menger fiender. Nivåene går gjennom en if-test som sørger for at ingen nivåer starter før alle fiendene er eliminert. For hvert nivå som går, så blir det vanskeligere fiender som har blitt hardkodet i en løkke, og som sørger for at vanskelighetsgraden øker. For hver «wave» vil fiendene øke, men også deres nivå og rank.

Tråder:

For dette spillet har vi implementert bruken av tråder for bedre ytelse. Selve spill logikken benytter seg av «JavaFX Application Thread», men for å få en bedre ytelse og minimere belastning har vi implementert en egen tråd for alt av lydeffekter. Hvor vi bruker en animtaiontimer til å itterere på if-tester, som vil starte en lydeffekt om en bestemt hendelse skjer. For eksempel når en avataren blir truffet så dukker det opp en kræsj-lyd.

Resten av spill-logikken kjører i en egen animationTimer, hvor vi bruker to store metoder: gameUpdate og gameRender. Grunnen til at disse to kjører i samme loop, er fordi de er avhengig av hverandre. Vi bruker canvas og GrapichsContext til å implementere spillet. Dermed må vi hele tiden oppdatere posisjon, hendelser osv. Og deretter kjøre Draw-

metoden til hvert objekt. Slik har vi optimalisert den beste ytelsen for spillet, som gir minst belastning.

Resten:

Resten av koden er godt dokumentert ved bruk av JavaDOC. For mer innformasjon eller spørsmål til koden kan dette brukes.

Kort oppsummert har vi laget et spill som inneholder alt av kjernedelene: Avatar, Bevegelse, Fiender, Kollisjon, pause, filbehandling og videregående elementer som powerUps, lyd, musikk, flere nivåer, velge bane/nivå og bedre ytelse ved hjelp av tråder.