Gruppe 33

Intro

Gjennom denne perioden har vi lært oss å lage et spill ved hjelp av JavaFX og bruken av FXML.

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.

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. Har lært oss veldig mye, ikke bare om java og fxml, men også om hvor viktig det er å ha faste dager hvor man jobber med faget og har god struktur hele veien på hva som må fikses til når.

Spillmessig så føler vi at vi har gjort en god jobb, men føler at vi kunne ha implementert mer kode enn å prøve å lære ved å lese seg opp. 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.

//at vi kunne ha brukt mer tid på å gjøre om til Android. Hadde vi brukt mer tid på det fra starten av kunne vi ha implementert det tidligere til spillet.

Hvem Som Gjorde Hva:

Begge av oss har kodet og gått gjennom hverandres koder, for å sørge for at alt stemmer med det vi har gjort fra før, og minimerer sjansen for komplikasjoner senere. Vi sørger for at begge av oss går gjennom hele koden flere ganger, så vi begge kjenner til hele spillet vårt.

Furkan:

FIENDER:

Furkan har for det meste implementert de Fiendtlige objektene, hvor de skal forekomme i tilfeldig posisjon ved hjelp av «angle» og Math.random(), og ulike sannsynligheter på hva en fiende vil droppe av ulike powerups ved hjelp av IF-tester.

En løkke sjekker antall nåværende fiender og når fiender er lik null, så starter en ny wave, med bestemte antall nye fiender, og i ulike vannskelighetsgrader. Vannskelighetsgraden til en enemy gir den en ny størrelse og flere liv. Dersom en fiende har vanskelighetsgrad større enn en og blir skutt, så dukker det opp flere nye svakere fiender i samme posisjon.

I tillegg, har spillet automatisk animere fiendtlige objekter. Hvis et fiendtlig objekt "treffer" avataren til spilleren, vil spilleren miste et liv, her går vi gjennom en FOR-løkke som sørger for at ulike kollisjoner skjer kun hvis Enemy sin x- og y-posisjon og Player sin x-og y-posisjon, er like. Dersom

og dersom spilleren er tom for liv, avsluttes spillet med en melding til brukeren og 2 knapper,” Restart” og ”Main Menu”.

FILEHANDLING:

Har fått til Filbehandling med lagring av spillets tilstander. Spilleren har mulighet til å lagre spillet til fil slik at man fortsetter der man slapp, blant annet lagring av hvor mye liv man har igjen(Health), poengsum(Score), hvilket brett man er på(Wave), og nivå på ulike krefter man har(PowerLevel).  
Ved hjelp av en if-test kan vi se om, filen ikke eksisterer eller er korrupt, og vil gi beskjed til brukeren om dette.

Pedram:

BEVEGELSE:  
Vi har laget et skytespill som beveges med piltastene hvor vi kan bevege fritt på en avatar ved hjelp av KeyEvents hvor vi bruker boolean. Vi starter først med FALSE boolean verdier, og ved hjelp IF-setninger blir boolean verdiene TRUE dersom man trykker på knappen, og da vil avataren bevegelse etter den TRUE-verdien, som tilsvarer retningen. Det samme gjelder for skytingen, hvor vi har en boolean verdi på SPACE-knappen. Skuddene har en fart og bestemt retning mot høyre, hvor startposisjonen til skuddene tilsvarer avatarens nåværende posisjon, ved hjelp av avatarens x og y GET-metoder.

Vi har klart å lage et spill der brukeren kan bevege en avatar i en 2-dimensjonal verden. Avataren er lett gjenkjennelig av spilleren, ved hjelp av et bilde av et romskip. Brukeren kan bevege denne avataren ved input, som tastetrykk av enten pil-taster eller «WASD».

FXML:  
Har fikset FXML, og css.  
Har laget ulike fxml-filer med knapper.

Deler av koden

OOP:

Deler som kunne vært bedre

Hva vi har lært ( intro ++)

Ikke en dritt

Planlegge til neste store prosjekt

Dette prosjektet har gitt oss en forståelse på hvor viktig det er å ha en klar og tydelig plan på hvordan prosjektet skal utføres.

Fordele koder og deler.

Et treff kan for eksempel bety at de 2-dimensjonale koordinatene for avataren og fienden er identiske. Spilleren skal ha muligheten til å sette spillet på pause og starte på nytt.

Grupperapport  Gruppen må sammen utarbeide en grupperapport for semesteroppgaven. Denne rapporten skal belyse de delene av oppgaven dere er spesielt fornøyd med. I tillegg, skal dere diskutere elementer som dere mener kunne ha blitt gjennomført bedre og/eller aspekter i besvarelsen dere ikke er helt fornøyd med. Utdyp hva dere har lært i henhold til prosjektarbeid og hva dere vil gjøre annerledes for deres neste store gruppeprosjekt. Til slutt, dokumenter arbeidet som har blitt uført for hvert medlem, med beskrivelse for medlemmets tekniske bidrag.

Individuell rapport  Den individuelle rapporten skal inneholde følgende elementer:

* 1. Generell refleksjon av din erfaring med semesteroppgaven.
  2. En oppsummering av dine individuelle bidrag til gruppearbeidet.
  3. Evaluering av bidragene gjort av hver av de andre medlemmene i gruppen.
  4. Et utvalg av kildekode med forklaring som belyser dine viktigste tekniske bidrag.
* Evaluering av tekniske bidrag  Evaluering av programkoden følger pensum i Programutvikling, som bygger på emnet Programmering. Det refereres dermed til semesterplanen og materialet fra forelesningene for hvilke tekniske temaer som tas hensyn til i evalueringen. En anbefaling er å bruke dokumentasjonen aktivt til å begrunne ulike valg dere gjør i utarbeidelsen av koden. Javadokumentasjon skal først og fremst beskrive klasser, men vi tillater i dette emnet at dere også diskuterer problemstillinger her.  Når det gjelder karaktersetting vil dette, som alltid, følge de generelle kvalitative beskrivelsene for de ulike karakterene fra universitets- og høgskolerådet:  http://www.uhr.no/utdanning/karakterpanel\_1