IKT103-prosjekt

# Super Mario 2D spill

## Før jeg startet

Før jeg startet med prosjektet, brukte jeg god tid på å finne ut hvilket prosjekt som kunne gi meg best utbytte med tanke på læring, og gi meg best mulig karakter. Database hadde jeg allerede laget, så jeg bestemte meg for å lage et spill. Vi hadde 3 ulike spill å velge mellom. Pac-Man, Tetris, og Super Mario. Det endte til slutt med Super Mario.

Det er flere grunner jeg valgte Super Mario:

* Det krever å få Mario til å bevege seg venstre, høyre, opp og ned i en spesiell fart.
* Det er flere bakgrunner/sprites i bildet hvor noen bare er bakgrunner, mens andre spiller en viktig rolle i spillet.
* Jeg får designe kule animasjoner og figurer.

Etter jeg hadde valgt Super Mario, så jeg mye på Youtube for å gi meg et innblikk over hvordan man burde organisere koden, hva som skal til, og hvilke verktøy man kan bruke. Kodingen utførte jeg på CLion, bakgrunnen tegnet jeg på Paint, og figurer og diverse brukte jeg PixilArt: <https://www.pixil.art/draw/16x16-among-us-9c4c51d18fad35a>.

## Når jeg begynte

Det tok lang tid å finne ut små ting. Et av problemene var trappen. Først startet jeg å lage trappen og bakken som en sammensatt sprite, men da er det vanskelig å finne ut av hvor i spriten Mario kolliderer. Da valgte jeg å lage hver enkel trappelevel som egen sprite, i tillegg til bakken som egen sprite. Da ble det lettere å sjekke hvor Mario kolliderte med de forskjellige spritsene.

## Hoveddel

Som en førsteåring så er jeg kjent med mine egne svakheter innenfor koding. Derfor startet jeg med å få inn de grunnleggende funksjonalitetene som hopping, gå venstre og høyre, at bakgrunnen beveger seg sånn at banen blir lengre, kollisjon, og å sette inn sprites. Jeg la fort merke til at det ble mye kode, og begynte å lage funksjoner for forskjellige texturene.

## Sluttdel

Mot slutten finpusset jeg koden og laget klasser til diverse entiteter. Jeg laget klasser, og passet på at alt fungerte som det skulle.

## Tiltak som går utover requirements

Design:

* Først og fremst synes jeg designet på banen er ryddig og fin i sin helhet. De forskjellige spritsene passer godt sammen, samtidig som bakgrunnen er tegnet selv. Størrelsene i forhold til skjerm og spritsene passer også sammen. Mario er et eksempel, han er hverken for liten eller for stor.

Marios/mappet sin bevegelse:

* For å kunne lage et større map som dekker mer enn 1 skjermlengde, implementerte jeg at bakgrunnen beveget seg istedenfor Mario. På den måten står Mario på en fast x-koordinat, mens mappet og resten av spritsene blir flyttet. Dette gjelder hvis Mario er innenfor en viss x-intervall. Hvis Mario er utenfor dette intervallet (som er i starten og slutten av mappet) så er det Mario som beveger seg. I tillegg kan ikke Mario gå forbi mappet.

Pole:

* Jeg har også implementert en sluttdel hvor flagget og Mario drar ned i stangen mot bakken sammen. Her sjekker jeg om flagget og Mario har samme y-koordinat. Om de ikke har samme, og flagget er høyere, så vil flagget gå ned helt til de har samme y-koordinat. Når de har samme vil de gå sammen ned med samme fart.

Musikk og lyd:

* Programmet har 3 sanger som går i loop som gamemusic. Hvis Mario dør, stopper musikken og det spilles en «Mario død» lyd. Hvis Mario fullfører mappet, vil loopen stoppe og en «winner» sang vil bli spilt. I tillegg har jeg lagt inn lyd for hopping, når han blir stor, og når han tar coins.

Coin counter:

* For hver coin Mario tar, vil Coincounteren inkrementere med 1 oppe på skjermen. Coin Counteren mener jeg er også «estetisk» designet, og passer fint inn i designet i mappet som en helhet.

Diverse:

* I requirements står det at Mario kan ødelegge blokker. I dette programmet kan Mario først delvis ødelegge, deretter ødelegge bricks. Dette gjør jeg ved å lage variabler som holder styr på om bricken er hel, delvis ødelagt, og ødelagt.
* Laget vector som inneholder alle sprites som skal bevege seg ved trykk av venstre eller høyre. Problemet er at når man bruker pushback på et sprite objekt, så er det ikke den orginale spriten som ligger i vectoren, men en kopi. Dette betyr at endringer som skjer etter pushback med spriten, vil ikke skje med spriten i vectoren, og vice versa. Derfor er det en pointer til spritsene i vectoren, og når jeg itererer gjennom vectoren så derefererer jeg dem.

VIKTIG:

For å åpne UML diagrammet må man åpne hos <https://app.diagrams.net/>.

Diagrammet ligger også som bilde-vedlegg i Diverse filen