

Om spillet

Den opprinnelige planen var å lage en eksakt kopi av Space Invaders. Men etter å ha fulgt en del tutorials og eksperimentert litt fant jeg at en variasjon med mer action og fart var mer spennende og utvikle fra.

Jeg startet utviklingen i Linux og møtte raskt på en del mindre snags. Som for eksempel at icon ikke vises i game vindu når man kjører spillet. Ingen kritiske feil, men allikevel nok til at jeg bestemte meg for å fortsette utviklingen i Windows. Dette fordi meningen med oppgaven er å sette seg inn i python / libraries og ikke feilsøke operativsystem (jeg hadde ikke kunnet la være og prøve å finne løsninger).

En av de første utfordringene jeg møtte var en udefinert framerate. Noen som hadde gjort at spillet ville kjørt i forskjellige hastigheter avhengig av hvor rask pc man har. Etter noe søk (der jeg fant noen «skittene» løsninger) fant jeg raskt ut at pygame har en innebygget funksjon for å kontrollere nettopp dette.

Deretter var det en del plankekjøring før jeg kom frem til rakettene og hvordan disse beveger seg. Med plankekjøring mener jeg hvor lett det var å «bygge» de primære elementene i spillet. Pygame gjør det lett å legge til bakgrunner, «sprites» (ikoner i mitt tilfelle) samt få disse til å bevege seg med enkel logikk og en enkel event loop. Det å plukke taster fra event loopen var også grei skuring.

Men så kommer man til selve spill logikken der man må regne ut kollisjoner og holde styr på elementene og hvor de er i forhold til hverandre. Er det noe jeg gjerne skulle ha gjort, men ikke fått tid til er det å endre struktur på hvordan UFO dataen lagres i lister. Her burde jeg laget / kommet opp med en bedre data struktur.

Det er her jeg hadde undervurdert hvor komplisert et spill (selv et lite) kan bli om man ikke holder tungen rett i munn med variabler og posisjoneringer av elementer. Jeg fant heldigvis en god artikkel på utregning av kollisjonskurs via en tutorial. Etter dette følte jeg det var en del «tygging», eller programmering som det også kalles 😊.

Om Koden

I begynnelsen var jeg veldig opptatt av at koden skulle være generell og gjerne «cleane» funksjoner uten bruk av globale variabler. Det viste seg fort å stå i veien for lett og lesbar kode, fremdrift på utviklingen og jeg måtte inngå kompromiss med meg selv for å ha fremgang. Jeg føler at jeg selv har en tendens til å prøve å gjøre effektivisering kontinuerlig selv om det kanskje ikke er nødvendig. Det kom klart frem for meg i denne oppgaven da det og faktisk levere noe er viktigere enn å overtenke / effektivisere.

Noen eksempler på dette:

Jeg har valgt og ikke ha en main funksjon. Måten spillet er programmert på nå er lett å lese og det å flytte deler av koden inn i en main funksjon vil gjøre det mindre lesbart og oversiktlig da man nå kan lese fra topp til bunn.

Jeg har derfor også valgt å beholde «spillet» i en fil. Vi er vel akkurat i grenseland for hva jeg føler er greit å holde i en fil. Men så lenge elementene ikke «trenger» klasser og funksjonaliteten er begrenset føler jeg dette er bra nok. Som nevnt tidligere skulle jeg ønske at jeg hadde laget en bedre datastruktur, og egne klasser for UFO med posisjon som attributter hadde vært en bedre løsning.

Jeg har valgt å bruke et eksternt LIB får å hente user input fra pygame da det viste seg ikke å være noe innebygget og jeg følte jeg ikke hadde tid til å programmere det. En «kanskje» dirty løsning jeg har brukt er å slice brukerinput til 3 istedenfor å redigere library direkte. Men det tenker jeg at dette biblioteket burde hatt et argument for max_characters men det hadde jeg ikke tid til å se på.

Jeg vil avslutte med å si at dette er det første spillet jeg noensinne har laget og jeg føler det har gitt meg mye nyttig informasjon, spesielt om x / y axis, rendering, movement av elementer og sikkert mye mer.