**KRAVSPESIFIKASJON**

**Hensikt**: Visuell interaktiv simulering/spill av en rakett.

**Behov**: En enkel og morsom måte for elever å visualisere og bygge intuisjon for hvordan fysikkformler utspiller seg i virkeligheten.

**Brukere**: Naturfag/fysikkelever fra hele verden.

**Produktdokumentasjon**

**Funksjonelle krav:**

- Interaktiv simulering av rakettfysikk

- Visuell representasjon

- Landingssystem

- Mulighet for å endre miljøvariabler

- Vise informasjon om simuleringen for brukeren

- Anvendbart av et internasjonalt publikum

- Tilfeldig generert terreng

**Ikke funksjonelle krav:**

- Hurtig

- Reagere i realtime på miljøendringer (masse, gravitasjon, etc.)

- Stabil fysikk

- Være ferdig innen fredag

**Brukerveiledning og tekniskdokumentasjon**

Brukerveiledning og tekniskdokumentasjon finner du på Github- siden til applikasjonen.

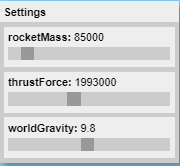
**PLANLEGGING OG UTVIKLING**

**Funksjonalitet:**

1. Applikasjonen skal være visuell og interaktiv.
2. Spillet begynner med at raketten faller og målet er å lande raketten på riktig måte, slik at den ikke krasjer.
3. Terrenget skal være tilfeldig generert.
4. For å styre raketten, kan brukeren svinge med "A" og "D" tastene.
5. For å fyre rakettmotoren trykker man på "SPACE".
6. Mulighet for å overstyre satte simuleringsvariabler
7. Simuleringen lar brukeren fly så høyt de vil. Dersom brukeren flyr utenfor skjermen på høyre eller venstre, skal simuleringen bli nullstilt.

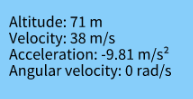
**Brukergrensesnitt**

**Innstillinger:**



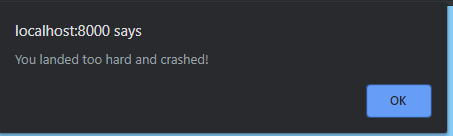
Dette vinduet viser og tillater manipulasjon av miljøvariabler. Her kan brukeren enkelt lese av masse, motorkraft og gravitasjon. I tillegg kan han endre de til det han ønsker med gliderne. Masse er i kilogram, motorkraft er i newtons og gravitasjon er m/s^2. Gliderne har minimum og maksimum-verdier, som forhindrer at brukeren skal gjøre simuleringen ustabil.

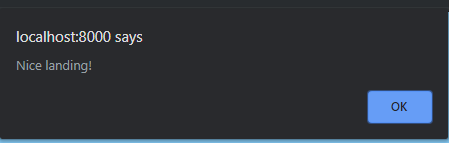
**Informasjon:**



Her vises informasjon om ulike variabler fra simuleringen. Alle verdiene er avrundet.

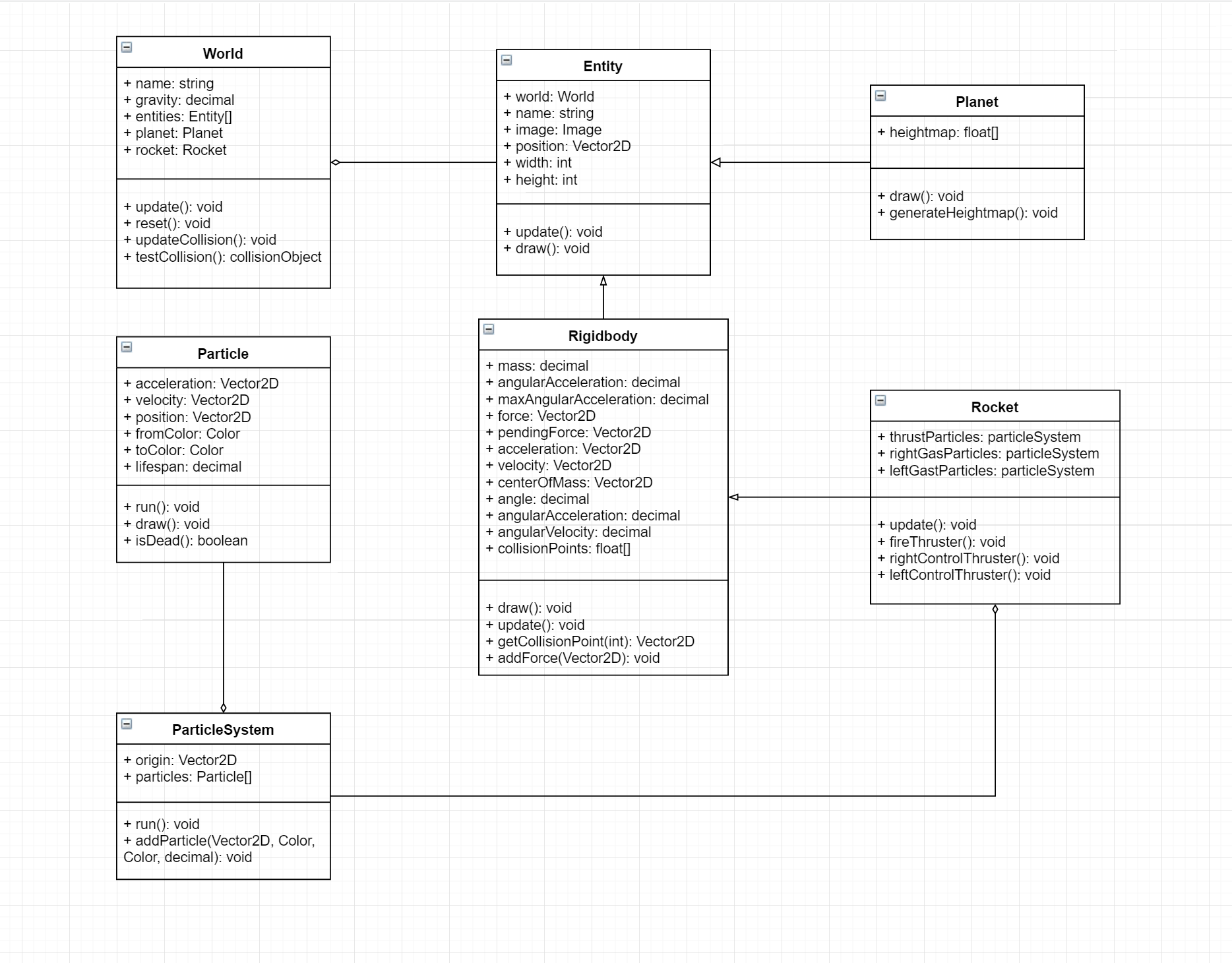
**Landingsmelding:**



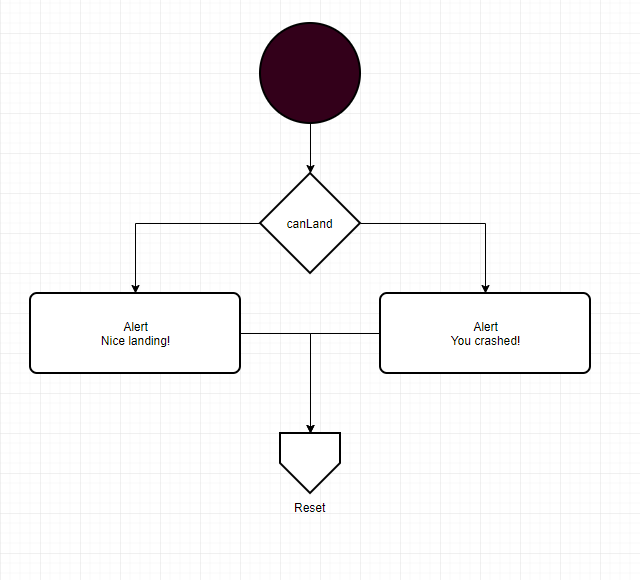


Når brukeren lander raketten, vil det en av meldingene ovenfor bli vist. Hvilke melding som blir vist, avhengiges av om raketten krasjet eller ikke.

**Kode**

****

Klassediagrammet over er i UML format. Her ser vi en oversikt over klassene og relasjonene mellom dem.

****

Flytskjemaet ovenfor viser valgprosessen som gjennomgås, når raketten treffer bakken.

**Testing**

Applikasjonen har ingen brukerinput utenom glidere og styring av rakett. Gliderne har minimum og maksimum verdier, som passer på at simuleringen blir værende stabil. Derimot er det muligheter for «bugs», når det kommer til selve simuleringen og hva brukeren kan gjøre med raketten få fram disse.

**Tester:**

Får å teste applikasjonen, har jeg prøvd ulike måter jeg kan tenke meg kan framstille bugs. Jeg to bugs, relatert til brukerinput.

Det første jeg prøvde var å skalere vinduet, slik at oppløsningen ble endret. Jeg regnet med at dette ville skape problemer, siden jeg ikke hadde tatt hensyn til dette når jeg programmerte applikasjonen. Det viste seg at raketten og terrenget, ble værende der det var og det ble hvit skjerm det vinduet hadde ekspandert.

Den andre jeg testet var å fly utenfor skjermen på høre og venstre. Jeg trodde dette ville skape problemer, fordi terrenget ikke er generert utenfor skjermen. I tillegg var jeg usikker på om jeg skulle tillate brukeren å fly utenfor, fordi man uansett ikke kan se raketten der. Som antatt, var det ingen kollisjon på grunn av manglende terreng, og dette tillot raketten å falle for alltid.

Til slutt, har jeg testen om fysikk utregningene er riktige. Jeg har ikke gjort noen manuell utregning. Derimot ved å se på tallene for høyde, fart, akselerasjon, ser jeg at de stemmer. Når akselerasjonen er -9.81, synker farten ganske nøyaktig med det samme per sekund.