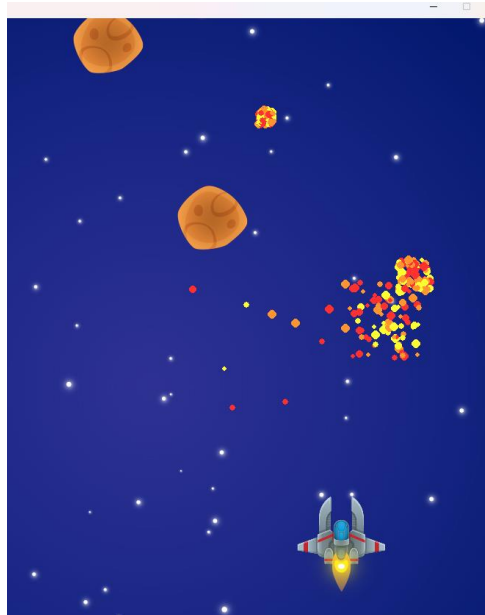


Projekt: Spaceshooter

Ni ska göra ett rymdskjutarspel där man ska kunna styra ett rymdskepp och skjuta på fiender. Spelet ska vara en bana långt. Har man klarat banan har man vunnit spelet, blir ens rymdskepp förstört på vägen mot målet har man förlorat. Alla gör först ett grundspel motsvarande E-nivå, sedan utökar du ditt spel med så många extrafunktioner som du kan från förslagslistan.



Grundspelet ska innehålla detta:

- Ett rymdskepp som du kan förflytta vänster, höger, upp och ner.
- Skott som kan flyga uppåt och som åker snabbare än rymdskeppet.
- En asteroid som slumpmässigt skapas utanför skärmens överkant och som åker rakt fram och försöker krascha in i spelaren. Ska försvinna när den hamnar utanför skärmen.
- Spelaren kan skjuta på asteroiden som då ska explodera om den träffas.
- Har spelaren träffats av en asteroid exploderar spelarens rymdskepp. Då avslutas spelet.
- Det ska spelas bakgrundsmusik och det ska finnas ljudeffekter för t.ex. när spelaren skjuter, spränger fiender och när det egna skeppet exploderar.

Extrafunktioner att välja mellan:

Extrafunktioner att välja mellan (sorterade efter svårighetsgrad):

Lättare:

- **Samla poäng:** Det ska gå att samla poäng i spelet, t.ex. när fiender skjuts ner. Poängen ska kunna visas för spelaren.
- **Visa en Pygame Powered logotype:** I början av spelet kan du visa Pygames logga med någon fadein, fadeout-effekt och en kort ljudeffekt innan man kommer till huvudmenyn. Flera varianter av logotypen finns här: <https://www.pygame.org/docs/logos.html>
- **Berättelse med berättarröst:** Ett intro i början av spelet och/eller en slutskärm när man vinner spelet med berättelse och voice-over effekt (antingen spelar du själv in din röst eller använder text-to-speech) och tillhörande bilder (som du t.ex. kan AI-generera om du inte vill rita själv).
- **En enkel menyskärm:** T.ex. Starta (startar spelet), Poäng (visar vilken spelare som fått högst poäng), Hjälp (visar hjälpinformation), Avsluta (avslutar spelet). Menyskärmen skulle t.ex. kunna ha ikoner/menyvalsalternativ som spelaren klickar på med musen/väljer med piltangenterna. Den skulle också kunna ha en separat bakgrundsmusik.
- **Game Over-text:** En enkel "Game Over"-skärm som visas när spelet är slut och ger spelaren en chans att börja om eller gå till menyn.

Medelsvåra:

- **Asteroiderna ska kunna skjutas sönder i mindre bitar:** Första gången den träffas delas den upp i två mindre bitar. Andra gången den träffas delas den upp i två ännu mindre bitar. Tredje gången de träffas exploderar de och försvinner ur spelet. Detta kan t.ex. lösas med partikeleffekter eller animerad sprite.
- **PowerUPs:** Spelaren kan samla genom att åka in i dem, t.ex. extra hälsa, mer eldkraft, mer sköld.
- **Kraftfältsmätare eller sköldgrafik:** Ett sätt att visa när spelarens sköld börjar ta slut, t.ex. en rektangulär energi/sköldmätare i ena hörnet av skärmen som minskar varje gång rymdskeppet tar skada.
- **Ett extra vapen:** T.ex. laser, raket eller plasma med unik funktionalitet och utseende (kan ha en annan ljudeffekt, ge mer skada eller skjuta snabbare).
- **Hjälpare kanon:** En extra kanon som sitter fast vid sidan av spelarens rymdskepp och skjuter.
- **Fiender som skjuter tillbaka mot spelaren:** Fiender med annat rörelsemönster än asteroiderna.
- **Större fiendevåg på slutet:** I stället för en boss, skapa en utmanande våg med flera fiender som dyker upp samtidigt i olika kombinationer med ökad svårighet (mer hälsa, snabbare rörelse, mer angrepp etc.).

Svårare:

- **Tvåspelarläge:** Ett extra rymdskepp med egna tangenter så att spelaren kan spela tillsammans med en kompis på samma dator. Då behöver interfacet anpassas så att det är lätt att se statusen på båda skeppen.
- **En boss i slutet av banan:** En unik fiende som dyker upp på slutet av spelet och har speciella attacker och rörelsemönster.

Betygskriterier

För att nå ett E:

- **Krav:** Gör klart grundspelet.
- **Kodkrav:** Spelet ska vara korrekt och använda konsekvent kodningsstil med enkel kommentering. Använd lämpliga namn för variabler, funktioner och klasser.
- **Förklaring:** Att bara genomföra grundspelet ger en bra förståelse för spelets funktionalitet, även utan extrafunktioner.

För att nå ett C:

- **Läser enbart programmering 1:** Gör klart grundspelet. *Två extrafunktioner* (minst en från den medelsvåra eller svåra listan).
- **Läser programmering 1 och tillämpad programmering:** *Tre extrafunktioner* (minst en från den medelsvåra eller svåra listan).
- **Kodkrav:** Koden ska vara noggrant kommenterad och tydligt strukturerad.
- **Förklaring:** Att lägga till extrafunktioner ger en extra utmaning som kan visa på en djupare förståelse och kreativitet

För att nå ett A:

- **Läser enbart programmering 1:** Gör klart grundspelet. *Tre extrafunktioner* (minst en från den svåra listan).
- **Läser programmering 1 och tillämpad programmering:** *Fyra extrafunktioner* (minst en från den svåra listan).
- **Kodkrav:** Koden ska vara utförligt och pedagogiskt kommenterad, med tydliga förklaringar av algoritmer och lösningar. På den här nivån förväntar jag mig att du använder dig av klasser och metoder.
- **Förklaring:** Genom att inkludera en extrafunktion från den svåra listan kan visa på förståelse för mer avancerade koncept, t.ex. att hantera mer komplexa strukturer som fiender med rörelsemönster eller utveckla ett tvåspelarläge.

Välj en av följande målgrupper till ditt spel:

För Hardcore-spelare (Skicklighetsbaserat spel)

- Svårighetsgrad: Svårare och mer utmanande fiender som inte släpps på förrän spelaren har klarat de föregående fienderna. Fiender med komplexa rörelsemönster och mer precisa träffar som kräver skicklighet.
- Skjutmekanik: Skott och vapen kan vara långsammare eller ha begränsade användningar, så spelaren måste vara mer strategisk med sina skott.
- Hinder: Färre eller inga slumpmässiga element, till exempel asteroider som alltid rör sig på samma sätt.
- Tidsbegränsningar: Tidsgränser eller andra mekanismer som gör att spelaren måste spela snabbt och exakt för att avancera.
- Belöning: Enkla belöningar för skicklig prestation, som exempelvis poäng eller uppgraderingar för att öka svårigheten.
- Game Over: "Permadeath" eller med ett hårdare straffsystem där spelaren måste börja om från början om de förlorar.

För Casual-spelare (Slumptungt och lättillgängligt spel)

- Svårighetsgrad: Lättare fiender som dyker upp i enklare mönster, men kanske med fler tillfällen att döda. Högre frekvens av fiender och enklare att överleva.
- Skjutmekanik: Skott och vapen kan vara snabbare och lättare att använda. Mängden ammunition kan vara oändlig, eller spelaren kan få extra ammunition ofta.
- Hinder: Mer slumpmässiga element som gör spelet mindre förutsägbart, t.ex. asteroider som dyker upp på oväntade ställen.
- Tidsbegränsningar: Färre tidsbegränsningar, så att spelaren inte känner sig stressad. Möjlighet att pausa spelet och fortsätta vid ett senare tillfälle.
- Belöning: Frekventa belöningar som ger en känsla av framgång och progression, t.ex. power-ups, extra liv eller en visuell förbättring av skeppet.
- Game Over: En mjukare "Game Over" där spelaren får möjlighet att fortsätta från där de var, eller en "Continues"-funktion som gör att de inte behöver börja om helt.

Planeringen

Jag vill att du har en tydlig plan för spelet innan du programmerar det. Den ska bestå av ett designdokument som du skriver i Google Docs där du beskriver din speldesign och ger ett förslag på teknisk lösning.

Några frågor du kan ställa till dig själv:

- Vad händer i spelet från att du startar det tills att det slutar?
- Vilka hinder och fiender ska du sätta i vägen för spelaren?
- Hur länge kommer spelet att pågå innan man kommer till slutet?
- Vad ska hända om spelaren förlorar spelet?
- Vad ska hända när spelaren vinner spelet?
- Ska ditt spel ha en berättelse? Om vad? Hur ska den visas för spelaren?
- Vilka extrafunktioner ska du ha med? Hur ska de implementeras? När ska de vara klara?
- Hur väl anpassat är ditt spel till den avsedda användaren (som kan vara antingen casual- eller hardcore spelare)?

Tidsplanering för hela projektet

Vecka 8 - 11: Grundspelet

- **Vecka 8:** Steg 1-3 (skärm, rymdskepp, stjärnbakgrund, jetmotorer)
- **Vecka 9:** Steg 4-5 (skott och asteroider)
- **Vecka 10:** Steg 6-8 (kollisioner och explosionseffekter)
- **Vecka 11:** Steg 9-10 (skjuta ner asteroider + ljud och musik)

Vecka 12 - 15: Extrafunktioner

- **Vecka 12:** Första extrafunktionen (+ 1 till för Tillämpad programmering)
- **Vecka 13:** Andra extrafunktionen (+ 1 till för Tillämpad programmering)
- **Vecka 14:** Tredje extrafunktionen (+ 1 till för Tillämpad programmering)
- **Vecka 15:** Finputsning, testning och redovisning

Inlämningsdetaljer:

- Din planering
- Ett färdigt spel (via GitHub)
- En utvärdering (kommer som en separat uppgift)