# C++ Fördjupning - Projektarbete Space Invaders

Er uppgift för projektarbetet är följande:  
Gör en variant av klassikern Space Invaders.  
All grafik ni behöver medföljer i den zippade filen.

Till denna uppgift får ni använda följande:  
- Power Points från lektioner  
- Kod från egna program och spel  
- TGE  
- Programmeringsböcker  
- Böcker på datorn  
- Internet

**Ni får ej använda:**    
- Mobiltelefon  
- Varandra

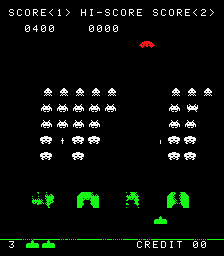
Ni kan plocka poäng genom att lägga till olika features i spelet. Alla features behöver inte implementeras för godkänt.   
Planera in och tänk igenom vilka features som passar er att implementera för att nå godkänt eller väl godkänt.

De olika features behöver inte göras i ordning, även om dom ligger i en viss logiskt ordning i listan.   
För att få poäng för en punkt så ska den vara väl fungerande, dvs. en buggig feature ger inga poäng.

I de fall information saknas så är det upp till egen tolkning hur spelet ska fungera.

**Inlämning**  
Sker senast **18:00 fredagen den 9 December**. Se till att få ihop ert projekt i god tid innan dess!!  
Liksom en vanlig uppgift måste både **källkod** och **exe** in.  
Ni skall även skriva vilka punkter ni har implementerat som en kommentar!!  
**Tips:** Är att spara undan en fungerande version då och då utifall ni skulle breaka allt i slutet.

**Godkänd max: (28 poäng totalt)**19 poäng krävs på Godkänt-delen.  
  
**Väl godkänt (8 poäng totalt)**  
5 poäng krävs på Väl Godkänt-delen.  
(och självklart minst 19 poäng från Godkänt-delen)



Referens video: <https://youtu.be/MU4psw3ccUI>

|  |  |  |
| --- | --- | --- |
| **Nr** | **GODKÄNT-NIVÅ FEATURES** | **Poäng** |
| **1** | Styra en spelare i botten av skärmen med **piltangenter.** Spelaren kan inte flytta utanför skärmen. | 1 |
| **2** | Spelaren kan avfyra godtyckligt antal skott från skeppet som färdas uppåt. När skottet åker utanför skärmen tas det bort och spelaren kan skjuta igen. | 1 |
| **3** | Det finns en begränsning på hur många skott som kan vara igång samtidigt (max 5st på skärmen samtidigt), samt en *reload time* så spelaren inte kan skjuta hur snabbt som helst. | 1 |
| **4** | Det finns fiender som går att skjuta ner. När dom träffas av ett skott så dör/försvinner dom. (ska tas bort korrekt från minnet). | 2 |
| **5** | Spelaren får poäng för varje fiende som dör. Poängen visas på skärmen. | 1 |
| **6** | (Nr 5 skall vara implementerad) Lägg till en High-Score som visar spelets högsta hi-score uppe i mitten på skärmen. Hi-Score skall börja med ett default värde (exempel 20 000). Lyckas spelaren samla ihop högre poäng än det som visas för Hi-Score skall denna också uppdateras på skärmen när spelaren får poäng. | 1 |
| **7** | Det finns olika typer av fiender. Några tål fler än en träffar och ger därmed också mer poäng. | 1 |
| **8** | Fienden rör sig i formation. Varje fiende rör sig åt höger ett steg, tills den mest högra har nått kanten. Då rör sig hela formationen ett steg neråt, och börjar gå åt vänster istället.  Observera att när de rör sig i sidled så rör sig endast en fiende åt taget. | 3 |
| **9** | När en fiende i formationen nått botten (samma y-led som spelaren) blir det game over. | 1 |
| **10** | Fienden kan avfyra skott neråt mot spelaren. Bara spelaren kan bli träffad av dessa skott. | 2 |
| **11** | För varje fiende som dödas så ökar hastigheten för de övriga (båda förflyttning och skottfrekvens om dessa är implementerade). | 1 |
| **12** | Om spelaren träffas av fiendens skott så förlorar spelaren ett liv. Spelaren börjar med tre liv och dessa visualiseras på skärmen. Om alla liv är slut så är det Game Over för spelaren. | 1 |
| **13** | Det finns statiska sköldar på banan (se bild) som skyddar spelaren mot fiendeskott. Spelarens skott förstörs också av sköldarna. | 1 |
| **14** | Sköldarna kan gå sönder och försvinner helt efter ett antal träffar. Hur många träffar det tål visualiseras med de bifogade bilderna ”shelter1(2, 3, 4).dds” | 1 |
| **15** | När en fiende dör spelas en explosionsanimation upp. Använd de bifogade bilderna. | 1 |
| **16** | Den bifogade bakgrunden ritas ut bakom allting. Hela bakgrunden ritas ut och är inte stretchad eller kapad. Spelets storlek är anpassad efter denna. | 1 |
| **17** | Det finns ett *mystery* ship som spawnar utanför skärmen längst upp ovanför fiendevågen (se röda ufot på bilden). Den åker sedan vågrätt över skärmen. Den tål två träffar och ger spelaren extra många poäng om den skjuts ner. Den dyker upp på random och försvinner när den åker utanför skärmen igen. | 1 |
| **18** | Spelet börjar med en titelskärm som presenterar spelets namn, vem som gjort det och instruktioner för styrning och andra knappar som går att använda. Spelet avslutas härifrån på escape och startas på enter (eller annan valfri knapp som presenteras). | 1 |
| **19** | När spelaren förlorar alla liv så kommer man till en G*ame Over-screen* som presenterar vilken level man kom till, hur många fiender som dödades samt hur många poäng man fick. När man trycker på någon tangent (välj själv) så återgår spelet till titelskärmen (om en sådan finns, annars startar spelet om direkt). | 1 |
| **20** | På titelskärmen så presenteras de tre högsta Hi-Score:sen från denna spelomgång. Alla hi-score platser skall ha default värden vid uppstart. | 1 |
| **21** | Hi-scores sparas även när man stänger ner och startar om spelet. Detta innebär att default värdena i punkt 21 ej skall sättas längre vid uppstart (så vidare det inte är första gången man startar spelet). | 1 |
| **22** | Spelaren kan pausa spelet genom att trycka på P. Spelen un-pausas även på P. När spelet är pausat skrivs texten ”paused” ut på skrärmen. | 1 |
| **23** | Följer kodstandard och har inga minnes läckor. | 2 |

|  |  |  |
| --- | --- | --- |
| **Nr** | **VÄL GODKÄNT-NIVÅ FEATURES** | **Poäng** |
| **1** | Objekt i spelet använder sig av Type object pattern så att all gemensamma data mellan objekt delas. | 1 |
| **2** | Alla objekt i spelet skapas och initieras korrekt av en factory/factories/factory functions. | 1 |
| **3** | Grafik och logik körs parallellt på varsin tråd, ska vara syncade och stängas korrekt. | 1 |
| **4** | Följande värde går att modifiera från en xml eller json fil utan att behöva kompilera om spelet:   * Spelarens hastighet (höger/vänster) * Spelarens antal liv * Projektilens hastighet för fiender & spelaren * Fiendernas hastighet * Default Hi-Score | 1 |
| **5** | All förflyttning i spelet görs med delta time (**er egen** timer och är oberoende av hårdvarans prestanda) och InputManager (ert bibliotek) används istället för TGE:s GetAsyncKeyState. | 1 |
| **6** | Om alla fiender dör så startar **banan** om, vilket innebär att en ny fiendevåg skapas och sköldarna återställs. Spelaren får en poängbonus. Fiendernas hastighet och skottfrekvens ökar lite efter varje bana. | 1 |
| **7** | När spelaren dör så reset:as man till mitten igen och alla fiender står stilla och väntar (varken rör sig eller skjuter) tills spelaren avfyrat det första skottet eller börja röra på sig | 1 |
| **8** | Spelaren får skriva in sitt namn vid hi-score, vilket även presenteras på titelskärmen. (Namnet får **ej** skrivas in i ett console-fönster) | 1 |