



Period 4 – Övningsuppgifter B

Objektorienterad spelprogrammering, 15,0 HP

Höstterminen 2015

1. Arvshierarki

Uppgiften bygger på Uppgift 7 – Period 3 Övningsuppgifter B.

I uppgiften står det

*Skapa ett nytt projekt i lösningen Övningar och kalla projektet Ö3B_8_11.
Sätt det nya projektet som startprojekt.*

Skapa en arvshierarki där GameObject är basklass och Mouse och Cheese är underklasser.

Basklassen skall innehålla en position och en textur. Vad som skiljer underklasserna från varandra är att musen skall gå att styra med hjälp av tangenter och osten skall försvinna om katten kolliderar med det.

- Modifiera basklassen så att den implementerar en Update-metod som förflyttar objektet åt höger på skärmen.
- Inga metoder skall vara virtual!
- Skapa en array med en katt och 10 ostar.
- Loopa runt arrayen och anropa Update för samtliga objekt i klassen.
Vad händer?
- Deklarera Update som virtual i basklassen och override metoderna i subklasserna.
- Kör programmet igen.
Vad händer?
- Gör basklassen abstrakt. Vad behöver modifieras för att programmet skall fungera som det är tänkt?
-

2. Kollision

Skapa ett nytt projekt i lösningen Övningar och kalla projektet Ö4B_2_3. Sätt det nya projektet som startprojekt.

- Lägg till en röd stridsvagn på spelplanen (Tank.png). Låt den förflytta sig från vänster till höger på skärmen (Enemy).
- Lägg till en blå stridsvagn på spelplanen. Denna stridsvagn skall kunna styras med tangentbordet (Player).
- Om bollarna går mot varandra skall de kollidera, dvs. den ena bollen skall inte kunna gå igenom/över den andra stridsvagnen. Den röda bollen skall stanna. Istället för att använda rektanglar skall du använda cirklar för att hantera kollisionerna.

3. Skjuta

- a. Skapa en klass Shot som använder texturen "smallShot.png".
- b. Uppdatera klassen Player med en variabel Shot. Det betyder att när du skapar en instans av Player så har spelarna var sitt ett skott.
- c. Låt spelaren skjuta i färdriktningen. Horisontellt räcker. Skottet skall ha en begränsad räckvidd.
- d. Gör en riktad missil. Den blåa stridsvagnen skall nu skjuta mot den röda stridsvagnen, men endast om den röda befinner sig framför den blåa. Ta ut den horisontella riktningen till den röda spelaren från den blåa.
- e. Använd pixelperfekt kollision.

4. Scollande bakgrund (parallax)

Skapa ett nytt projekt i lösningen Övningar och kalla projektet Ö4B_4. Sätt det nya projektet som startprojekt.

- a. Sätt fönsterstorleken till 1000x800 px.
Ladda in bilderna clouds_short, land_short och waves_short
Skapa en klass som representerar bakgrunden.
Skapa tre instanser av klassen med de olika texturen.
De olika objekten skall ha olika hastighet. Det längst bak skall ha hastighet 0,2, nästa 0,5 och det främre 1.
Startpositionen för bakre lagret skall vara 700, nästa 750 och det främre 800.
- b. Gör samma sak igen men den här gången skall du använda texturen med fulländ. Justera de värden som behövs justeras.
- c. Modifiera bakgrundsklassen så att när texturen tar slut, så börjar den om från början.
- d. Lägg till två lager till

5. Hoppa

Skapa ett nytt projekt i lösningen Övningar och kalla projektet Ö4B_5_7. Sätt det nya projektet som startprojekt

- a. Skapa ett nytt projekt.
Skapa en player klass.
Använd texturen IndianaJones.
Placera spelaren i övre delen av fönstret.
Spelaren skall kunna springa till vänster och höger.
Lägg till gravitation till spelarklassen så att spelaren startar med att falla.
Hantera att spelaren inte kan försvinna ut ur fönstret när spelaren faller.
- b. Låt nu spelaren hoppa när du klickar på uppåtpil.

6. Plattformer

Skapa en klass som representerar en plattform.

Skapa några plattformar som ligger på samma höjd och intill varandra.

7. Hoppa mellan plattformar

- a. Hantera kollision mellan spelare och plattform så att spelaren kan springa och hoppa på plattformarna.
- b. Placera plattformarna en bit från varandra så att spelaren kan falla ner emellan dem.