Projekt – Robotron

Under större delen av resten av kursen kommer ni arbeta med ett projekt. Projektet går ut på att skapa ett spel och syftet är framförallt att ni ska lära er mer om objektorientering. Förhoppningsvis har ni även roligt på vägen.

Ett filmklipp över hur spelet ser ut och fungerar finns här:
http://www.youtube.com/watch?v=940WwmYSYLE

Betygskriterier

	G	VG	MVG
Funktionalitet	- En spelare som kan styras och skjuta skott - En sorts monster som jagar spelaren, dör av skott och dödar spelaren	- Åtminstone två sorters monster, som beter sig olika - Mjuk och kontinuerlig rörelse hos objekt genom användning av Timer. - Oberoende kontroll av åtta tangenter för styrning av spelare och skott	- Ett välfungerande spel med bl.a. poäng, extraliv, omgångar, ökande svårighetsgrad
Struktur och objekt-orientering	- Vettig(a) kod, variabelnamn, kommentarer och metoduppdelning - Uppdelning i åtminstone 3 klasser	- Välstrukturerad och lättläst kod, med bra variabelnamn, kommentarer och metoduppdelning - Uppdelning i åtminstone 5 klasser - Användning av lämpliga variabeltyper (int, boolean, …) och datastrukturer ([], ArrayList, … - Användning och förståelse av (kan förklara) arv (för t.ex. ”Sprites”) - Användning och förståelse av gränssnitt - Kunna förklara och använda händelsehantering - Användning och förståelse av static metoder samt static och final variabler	- Detaljerad förståelse av klasser och objekt/instanser, bl.a. medlemsvariabler och -metoder och statiska d:o - Detaljerad förståelse av arv, bl.a. kunna förklara hur konstruktorer anropas i arvshierarkier, kunna förklara och använda överlagring och överskuggning av metoder - Detaljerad förståelse av gränssnitt, bl.a. deras syfte och skillnaden mot arv - Förståelse och användning av polymorfism - Användning och förståelse av Design Patterns, t.ex. Observer – Observable och Singleton

Tillvägagångssätt

Börja med att **skapa ett fönster** som spelet spelas i. För att göra det:

- Skapa en klass (t.ex. Robotron) som ärver av JFrame.
- Skapa en konstruktor i klassen som innehåller metodanropen
  setSize(800, 600);
  setVisible(true);
- Skapa en main-metod, som skapar en instans av klassen
- Om du kör programmet bör du nu få upp ett tomt fönster

Nästa steg blir att **skapa en yta (JPanel) som grafiken kan ritas på**. Gör följande:

- Skapa en till klass (kanske GameArea) som ärver av JPanel.
- Ersätt metoden som sköter uppritningen[1] av panelen:

```
public void paintComponent(Graphics g) {
    g.setColor(Color.BLACK);
    g.fillRect(0, 0, getWidth(), getHeight());
    g.setColor(Color.YELLOW);
    g.fillRect(100, 100, 10, 30);
}
```

- Om du kör programmet nu bör du fått ett svart fönster med en gul rektangel (spelaren?) i.

Sen är det dags att spelaren får en **ikon / sprite**.

- Ladda ner ”player.png” från Fronter och lägg till i (roten på) projektet.
- Skapa en ikon med
  `new ImageIcon("player.png");`
- Lagra referensen i en variabel och rita upp den med
  `g.drawImage(<din variabel>.getImage(), 200, 200, this);`
- Nu bör det se ut så här =>

[1] Förklara paintComponent

Nästa steg är att få spelaren att **röra sig**. För att få det att fungera vill du lyssna på tangenttryckningar och reagera på dem. Du måste även se till att fönstret ritas om, så att det syns att spelaren flyttat på sig.

- Skapa en konstruktor i GameArea(?) och i denna
  `setFocusable(true);`[2]
  `addKeyListener(<din lyssnare>);`
- <din lyssnare> ska vara en instans av en klass som kan hantera tangenttryckningar, dvs. implementerar gränssnittet KeyListener. Googla på ”Java KeyListener” och läs på om interface i boken (eller på nätet) vid behov.
  [keyTyped, ke.getKeyChar()]
- Lägg till ett anrop till
  `repaint();`
  för att skärmen ska ritas om när du vet att spelaren har flyttat på sig.

Tillvägagångssätt
Timer
GameObject
Fiender
Skott
Korrekt rörelse (flera steg)

Övrigt
Objektorientering: klasser, arv, gränssnitt
Objektorienterad design
Klasser: Robotron, GameArea, GameObject…
Datastrukturer
Betygskriterier
Läsanvisningar (boken, tutorials)
Tidsplan

```
setDefaultCloseOperation(EXIT_ON_CLOSE);
setLocationRelativeTo(null);
```

[2] En panel kan normalt inte få fokus och därför heller inte fånga tangenttryckningar. setFocusable(true) fixar detta.