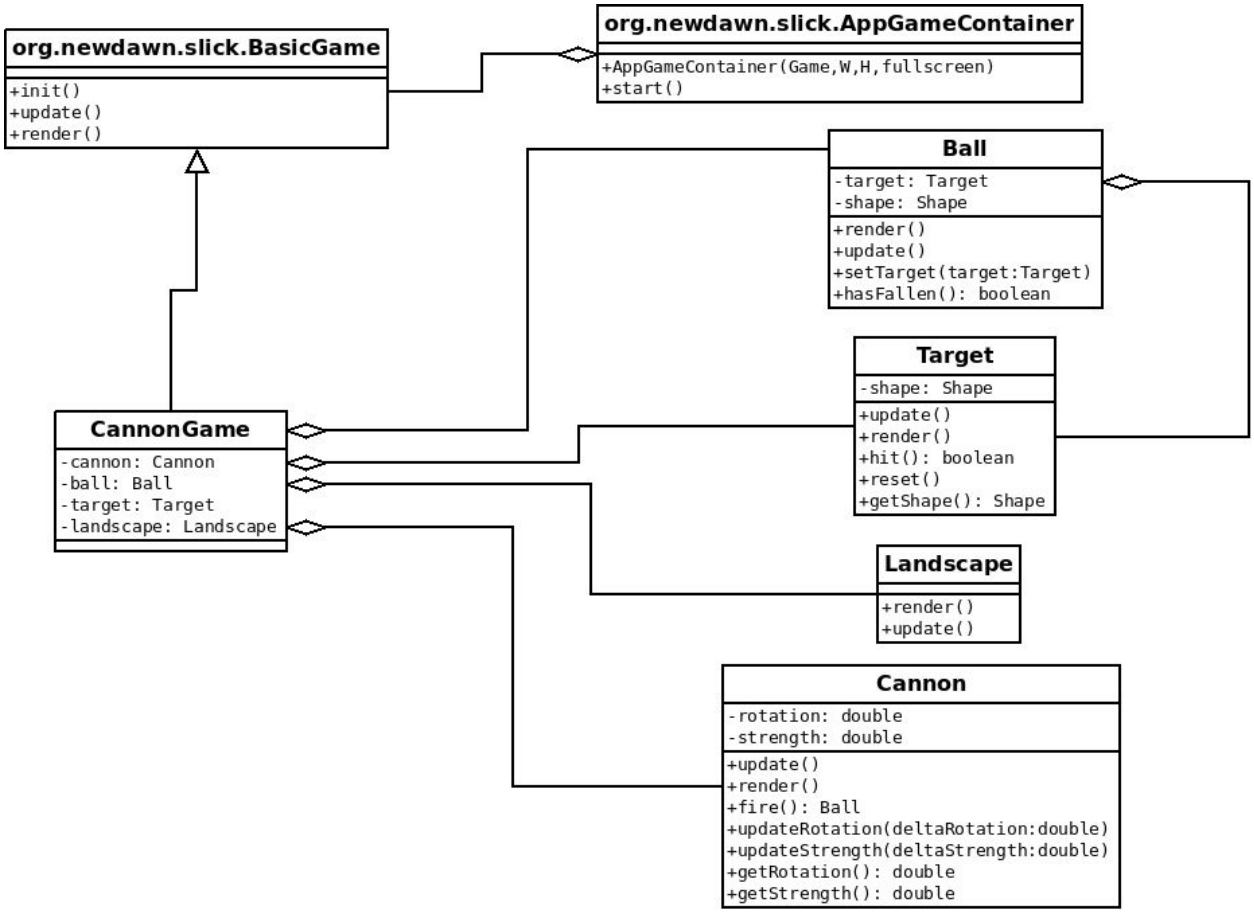
Pràctica CANNON

La pràctica proposada consisteix en implementar un joc gràfic, on hi ha un canó que dispara un projectil fins que s'encerta un objectiu. Es pot ajustar l'angle del canó i la força del dispar.

Aquí teniu el diagrama de classes que heu d'implementar:



Inicialització

Emprarem la llibreria "slick2d", que podeu trobar a http://slick.ninjacave.com/. Aquesta llibreria ens proporcionarà tot el que necessitem per desenvolupar el nostre joc. Podeu trobar informació en el javadoc oficial: http://slick.ninjacave.com/javadoc/ i també a la seva wiki: http://slick.ninjacave.com/wiki/index.php?title=Main_Page

Funcionament

El loop principal es basa en dos mètodes fonamentals:

- update() es crida cada vegada que el joc ha d'ésser actualitzat. Aquesta funció s'encarrega de mantenir l'estat del joc.
- render() és el mètode on hi hem de posar el codi que dibuixarà cada frame del joc. El número de vegades que s'invoca aquest mètode per segon determina el FPS.

A més dels mètodes descrits anteriorment, n'hi ha un altre també de molt important: init(). Aquest s'invoca només un cop, i la seva funció és carregar les imatges i recursos a la memòria.

La classe "CannonGame" deriva de la classe "BasicGame". Això ja ens proporciona una implementació del loop principal i la inicialització dels objectes per obtenir l'entrada per teclat, entre altres coses.

Per arrancar el nostre joc, doncs, el que hem de fer és:

1. Crear un objecte de tipus CannonGame.
2. Crear un objecte de tipus AppGameContainer. El constructor d'aquest requereix que li proporcionem un objecte de tipus Game.
3. Cridar al mètode "start" de l'objecte anterior

```
CannonGame cg = new CannonGame();
AppGameContainer app = new AppGameContainer(rg);
app.start();
```

Gràfics

El mètode "render" obté una referència a un objecte de tipus "Graphics". Mitjançant aquest, podeu dibuixar diferents formes a la pantalla. Heu de recordar que l'origen de coordenades és al costat de l'esquerra a dalt de la pantalla. Exemple:

```
public void render(GameContainer gc, Graphics g) {
      Rectangle r = new Rectangle(0,0,50,50);
      r.setX(100);
      r.setY(100);
      g.draw(r);
}
```

Podeu trobar molta més informació al javadoc oficial.

Input

El mètode "update" que hereteu de la classe BasicGame reb un objecte de tipus "GameContainer", que podeu utilitzar per obtenir un altre objecte de tipus "Input". Exemple:

```
Input i = gameContainer.getInput();
if (i.isKeyDown(Input.KEY_SPACE) {
      dosomething();
}
```

Recordeu que la funció "update" automàticament serà cridada unes 60 vegades per segon (en un joc això és el més usual).

Compilació i execució

Per poder compilar i executar el vostre programa, heu d'indicar a l'entorn on es troben les llibreries, tant les natives del sistema operatiu que empreu com les de java.

El primer pas és configurar el vostre projecte (project structure…) i indicar el directori on hi ha els fitxers .jar de slick2d.

Un cop feu això, el programa hauria de compilar correctament. Però no ho podreu executar, perquè també es necessiten els fitxers amb les llibreries natives. Això es pot solucionar indicant el paràmetre "-Djava.library.path=/on/hi/ha/els/dll/o/els/fitxers/so" a les opcions de execució de la màquina virtual.

Requeriments del vostre programa

El programa tindrà una pantalla de presentació, on surt el títol del joc. L'usuari ha de pitjar la tecla "ENTER" per començar.

S'utilitzaran les tecles següents: Cursors UP/DOWN per augmentar o disminuir la força del dispar (de 10 a 100). Cursors LEFT/RIGHT per variar l'angle de rotació del canó. I en voler disparar, es pitja la barra d'espai.

Si s'encerta l'objectiu, augmentem la puntuació en +100. Aleshores es torna situar l'objectiu en una posició aleatòria.

Si no encertem i la puntuació és major que 0, es resten 30 punts. L'objectiu es mou a una altra posició. Inicialment, es disposa de 5 dispars. Si s'encerten tres vegades seguides, s'afegeix un dispar més. El joc s'acaba quan no hi ha més dispars restants.

Aquí teniu una captura de pantalla: