



| | | |
|---|---------------------------------------|---|
|  Institut Sabadell | UF4. Programació Orientada a Objectes |  Curs: 2019/2020 |
| | <i>Crea el teu Videojoc. Part 1</i> | |

En aquest projecte crearàs el teu propi videojoc mitjançant la programació orientada a objectes.

Punt A Configuració

Descarrega de git <https://gitlab.com/malbareda/graphicboard-v3-junity> el projecte GraphicBoard v3. La part important es el paquet de Core. Crea un nou projecte i copia el paquet de Core. Després crea un nou paquet anomenat Part1, que serà el que faras servir.

Mai modifiquis el projecte GraphicBoard v3. Si no, no podràs fer pulls desde git quan hi hagin actualitzacions del projecte.

En el paquet Core hi ha 4 classes de les que veurem 3 avui. Totes elles tenen Javadoc, però les descriurem a grans trets

Window: La classe que descriu la finestra que obrirà. El nou programa obrirà una nova finestra a on operará i es jugarà el joc en aquesta nova finestra.

Field: Field seria el llenç a on es representaran tots els píxels. Evidentment anirà dins de la Window

Sprite: Sprite formarà cada un dels objectes diferents que farem servir en el joc.

D'aquí podem intuir que en el nostre joc hi haurà una Window, un Field, i múltiples Sprites. Com que només hi haurà una Window i un Field, es poden declarar estàticament. Crea una classe nova Joc1 amb un main i declara les següents variables estàtiques



```
//declaracio de field
static Field f = new Field();

//declaracio de window
static Window w = new Window(f);
```

Com pots veure, al constructor de Window li pasarem el field que hem creat. Per què creus que pot ser això?

Quan ho intentis importar per primera vegada t'apareixerà un error de que no ho troba. Això és perquè estan a paquets diferents. Java automàticament importa tot allò que té al mateix paquet, pero al estar Field a Core i la teva classe a Part1, hauràs d'importar Field i Window com a Core.Field i Core.Window

Al executar aquest codi se t'hauria d'obrir una finestra buida.

| | | |
|---|---------------------------------------|---|
|  Institut Sabadell | UF4. Programació Orientada a Objectes |  Curs: 2019/2020 |
| | <i>Crea el teu Videojoc. Part 1</i> | |

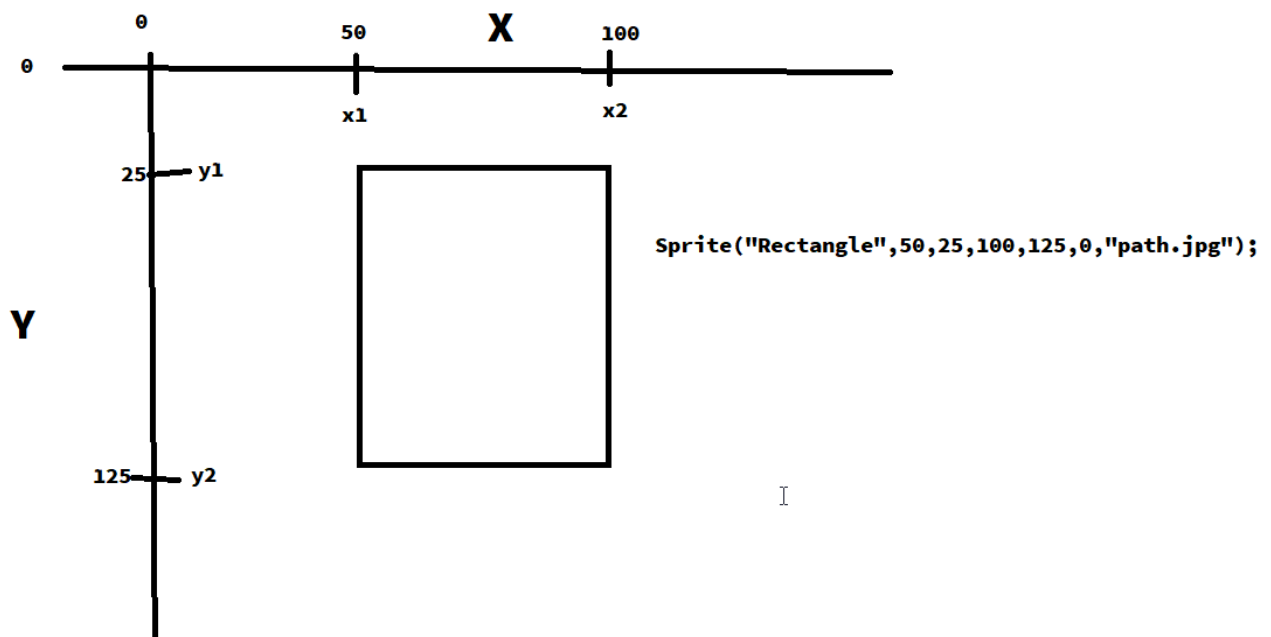
Punt B Dibuix

Bé, una finestra buida no aporta massa. Anem a fer el nostre primer Dibuix, que serà mitjançant un Sprite. Per a fer això haurem de cridar al constructor de Sprite. Un constructor es una funció que es fa servir per crear un objecte. De fet, ja coneixes molts constructors, com `new Scanner(System.in)`, `new Random()`, o `new ArrayList<>()`. Tots ells duen el `new`, seguit per el nom de la classe.

Si mires tots els constructors de Sprite (els pots mirar escrivint `new Sprite` i apretant `ctrl-espai`) veuràs que n'hi ha diversos. Farem servir el default. Les instruccions del constructor estan al Javadoc per li farem una ullada

```
Sprite roca = new Sprite("Roca", 50, 50, 100, 150, 0, "resources/rock1.png", f);
```



Mirem que fa aquest constructor. El primer camp és el nom del Sprite. Després les coordenades, que es donen com a 4 enters. Les coordenades funcionen com a `x1,y1,x2,y2`. Mirem aquest exemple.



Aquí tenim un rectangle que té el costat superior esquerra a 50,25 i el cantó inferior dret a 100,125.

La següent xifra es un doble (nombre amb decimals) que indica l'angle. És per fer rotacions i de moment es pot deixar a 0.

El següent camp es el path. El path és el que es coneix com una ruta relativa, que comença en la carpeta del projecte. L'últim camp es el field que ja havíem creat

| | | |
|---|---------------------------------------|---|
|  Institut Sabadell | UF4. Programació Orientada a Objectes |  Curs: 2019/2020 |
| | <i>Crea el teu Videojoc. Part 1</i> | |

Un cop creat l'objecte, l'hem de dibuixar. Veiem com es fa. En el main, fes servir aquest codi

```

boolean sortir = false;

while (!sortir) {
    f.draw();
    Thread.sleep(30);
}

```



Aquest codi serà el nostre bucle principal. Cada iteració del bucle la coneixerem com “un frame”. En cada frame, crearem una ArrayList de Sprites, i afegirem els Sprites que volem dibuixar, en el nostre cas, roca.

sprites es una ArrayList. La majoria ja coneixereu l'ArrayList o les llistes de Python. Per els que no, una ArrayList és un array dinàmic que canvia el seu tamany. el mètode add afegeix un nou sprite a la llista de sprites.

El següent mètode es el draw, un mètode de Field. A aquest mètode li pots passar una llista amb tots els sprites que vols dibuixar i els dibuixarà. Si algun sprite no està a la llista, no el dibuixarà. També pots cridar a draw() sense passar-li una llista, com és el cas, aleshores dibuixarà tots els Sprites que hagi creat anteriorment.

L'últim mètode es un Thread.sleep. Això fa una pausa de 30 milisegons, o dels milisegons que li passis. Fixa't que $1000/30 = 33.33$, és a dir, el nostre joc anirà a 33.33 Frames per cada segon, o 33.33 FPS.

Executa el codi. T'hauria de sortir la roca que has pintat.

| | | |
|---|---------------------------------------|---|
|  Institut Sabadell | UF4. Programació Orientada a Objectes |  Curs: 2019/2020 |
| | <i>Crea el teu Videojoc. Part 1</i> | |

Punt C Herència i Constructors



Ja podem dibuixar Sprites. No obstant, si has vist la classe Sprite per dintre sembla extraordinàriament complexa. Si has d'afegir funcionalitat als teus dibuixos com moviment, no sembla que sigui massa fàcil. Farem servir herència per això. Crea una classe Roca, que sigui filla de Sprite (una classe filla es aquella que en la seva declaració de classe seria "public class Roca extends Sprite"). En aquesta classe crea una variable privada int que sigui "accionsDisponibles".

El compilador et demanarà que posis a Roca un constructor que sigui compatible amb Sprite. Fixa't que si fas l'autocompletar (fes servir el String, int, int, int, int, double, String, Field) et farà servir la funció super(). super fa referència al teu pare (així com this fa referència a un mateix) i super() crida al constructor del pare, en el nostre cas, el constructor de Sprite. Hauries de tenir el següent:

```
//classe Roca que és filla de Sprite
public class Roca extends Sprite {

    //la variable no sera estatica, ja que no és una classe amb main
    private int accionsDisponibles;

    //constructor generat automàticament
    public Roca(String name, int x1, int y1, int x2, int y2, double angle, String
path, Field f) {
        //crida al constructor del pare, com a primera linea.
        //Per tant cridara a new Sprite() amb els paràmetres passats
        super(name, x1, y1, x2, y2, angle, path, f);
        //aqui faras altres coses que vulguis fer al crear una nova Roca.
    }
}
```

| | | |
|---|---------------------------------------|---|
|  Institut Sabadell | UF4. Programació Orientada a Objectes |  Curs: 2019/2020 |
| | <i>Crea el teu Videojoc. Part 1</i> | |

Apart del constructor per defecte, crea els següents constructors. Nota. la mateixa classe pot tenir diversos constructors, sempre que tinguin paràmetres diferents



- Un que els hi passes tots els paràmetres més el int d'accions disponibles. farà servir el constructor del pare i després inicialitzarà accions disponibles. Aquest te'l donem fet

```
//constructor amb accions disponibles. Aquest es un constructor diferent al primer i
has de tenir els dos!
    public Roca(String name, int x1, int y1, int x2, int y2, double angle, String
path, Field f, int accions) {
        //crida al constructor del pare, com a primera linea.
        //Per tant cridara a new Sprite() amb els paràmetres passats
        super(name, x1, y1, x2, y2, angle, path, f);
        //aquí faras altres coses que vulguis fer al crear una nova Roca.
        //this. fa una referencia al propi objecte. this.accionsDisponibles es
refereix a la variable accionsDisponibles global a l'objecte
        this.accionsDisponibles = accions;
    }
```

- Un constructor que només els hi passes les 4 variables de localització, el field i l'int d'accions disponibles i assumeix que la roca es dirà roca i tindrà l'imatge de la roca que hem fet servir al punt anterior.
- Un constructor al que li passes el field i 3 ints, x1, y1, i tamany, i crea una roca quadrada amb aquest tamany de costat, amb 50 accions disponibles
- Un constructor buit, que crea una roca que va de (0,0) a (50,50), amb 50 accions disponibles. En aquest cas el constructor agafa el Field estàticament de la classe Joc1 (és a dir, Joc1.f)

Crea una roca de cada tipus, afegeix totes a la llista de sprites, i visualitza-les.

Potser estàs pensant, “Com puc posar objectes de classe Roca a una llista que es de Sprite?”. Pots fer-ho degut a la propietat de la polimòrfia. Els objectes de classe Roca, al ser Roca filla de Sprite, també són de classe Sprite.

| | | |
|---|---------------------------------------|---|
|  Institut Sabadell | UF4. Programació Orientada a Objectes |  Curs: 2019/2020 |
| | <i>Crea el teu Videojoc. Part 1</i> | |

Punt D Variables estàtiques

En Orientació a objectes, una variable estàtica és aquella que és la mateixa per a tots els elements de la classe. És a dir, quan la canvies per a un objecte, canvia per a tots els objectes de la classe.

Per provar-ho pots fer una variable

```
public static int comptador=0;
```

a la classe Roca.

Prova creant dos objectes de classe Roca, incrementa el comptador a una, i prova d'imprimir el comptador de l'altra. Què passa?

```
Roca r1 = new Roca("roca1", 50, 50, 100, 100, 0, "imatges/roca1.png", f);
Roca r2 = new Roca("roca1", 50, 50, 100, 100, 0, "imatges/roca1.png", f);
r1.comptador++;
System.out.println(r2.comptador);
```

Aprofita aquesta propietat per fer un comptador autonumèric per a les roques. Per fer això necessitaràs dues variables, un comptador que serà estàtic i una id que no ho serà. Els dos seran paràmetres de la classe Roca. Fes que el comptador augmenti al cridar un constructor (qualsevol d'ells) i la id es sumi al nom de la Roca.

Per tant si fas 3 roques, la primera tindrà id 1, la segona tindrà id 2, i la tercera tindrà id 3, mentre que el comptador, al ser estàtic, serà igual per a tothom.

Pensa que no podràs afegir la id al nom de la Roca quan crides al pare, així que hauras de modificar el nom després d'haver cridat al constructor del pare.