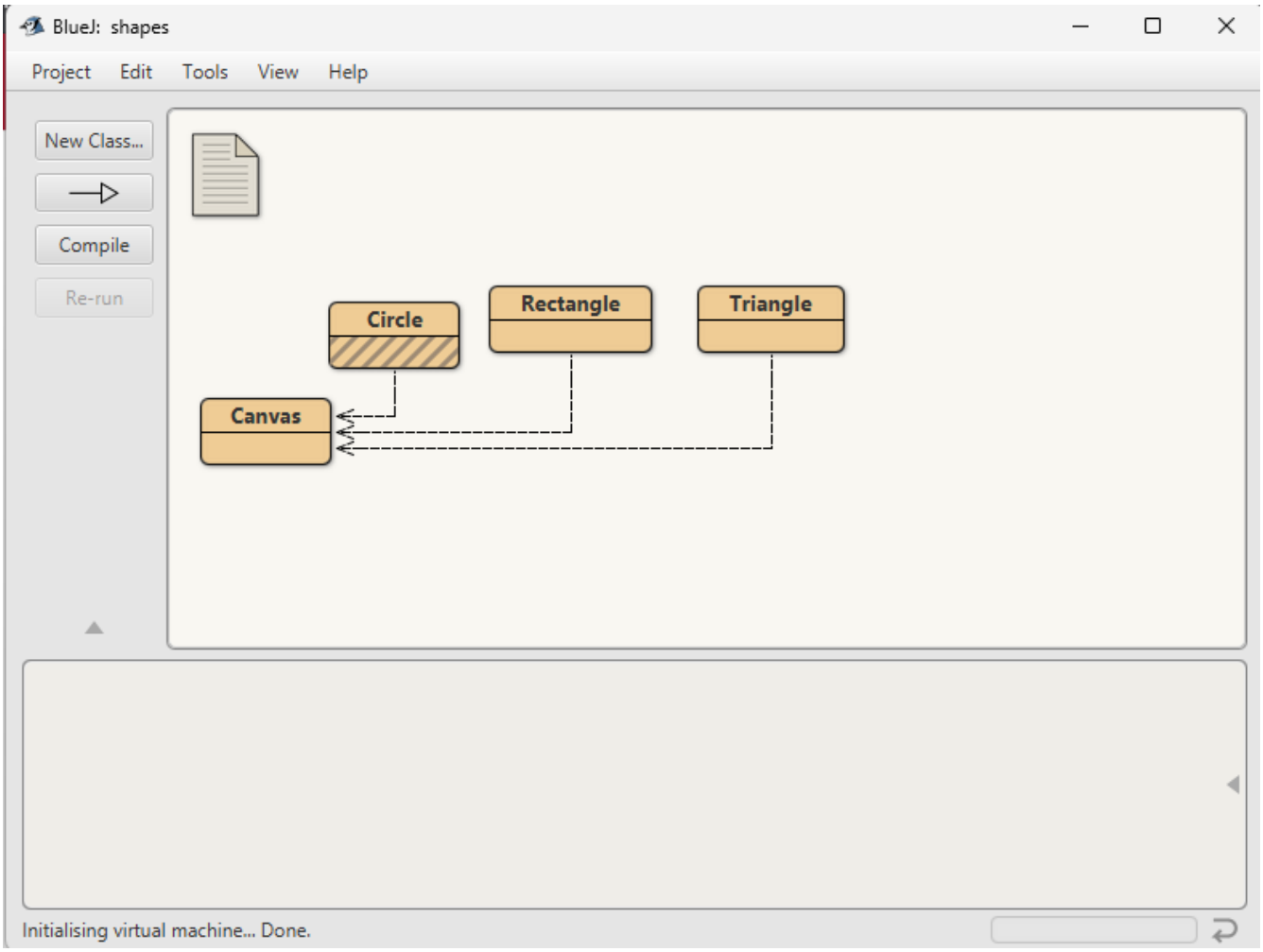
Lab 01 POOB

Juan Diego Patiño Muñoz
Hever Barrera Batero

1. Shapes

A. Conociendo el proyecto shapes

1.



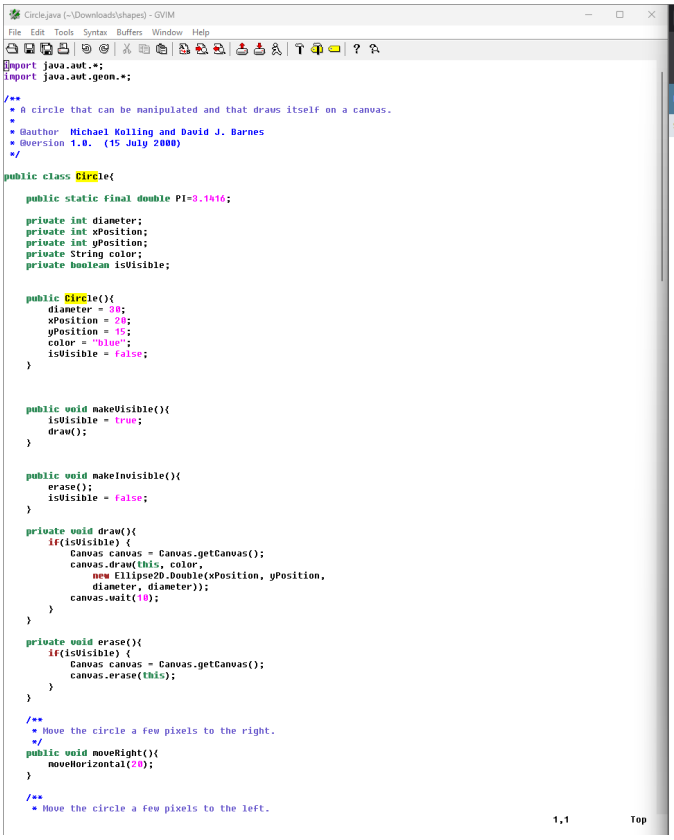
2.a. Hay cuatro clases, Circle, Rectangle, Triangle y Canvas
2.b. La relacion es que Circle, Rectangle, Triangle dependen de Canvas

3.a. Hay cuatro clases, Circle, Rectangle, Triangle y Canvas
3.b. la constante PI

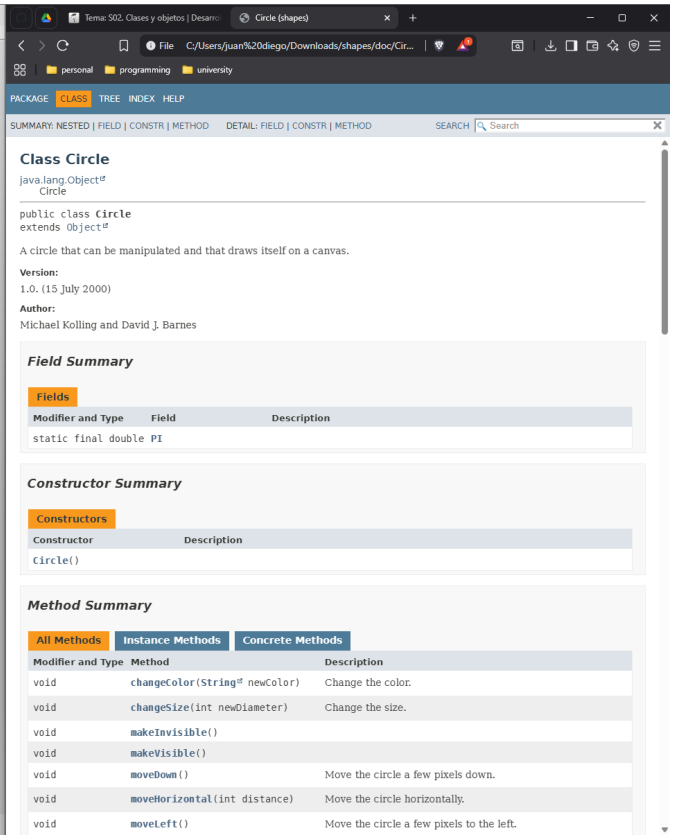
3.c. 12
3.d. changeSize

4.a. 6
4.b. Las instancias de la clase
4.c. 14
4.d. Solo los puede usar la clase

5.a



```
import java.awt.*;
import java.awt.geom.*;

/**
 * A circle that can be manipulated and that draws itself on a canvas.
 *
 * @author  Michael Kolling and David J. Barnes
 * @version 1.0.  (15 July 2000)
 */

public class Circle{

    public static final double PI=3.1416;

    private int diameter;
    private int xPosition;
    private int yPosition;
    private String color;
    private boolean isVisible;


    public Circle(){
        diameter = 30;
        xPosition = 20;
        yPosition = 15;
        color = "blue";
        isVisible = false;
    }


    public void makeVisible(){
        isVisible = true;
        draw();
    }


    public void makeInvisible(){
        erase();
        isVisible = false;
    }

    private void draw(){
        if(isVisible) {
            Canvas canvas = Canvas.getCanvas();
            canvas.draw(this, color,
                new Ellipse2D.Double(xPosition, yPosition,
                diameter, diameter));
            canvas.wait(10);
        }
    }

    private void erase(){
        if(isVisible) {
            Canvas canvas = Canvas.getCanvas();
            canvas.erase(this);
        }
    }

    /**
     * Move the circle a few pixels to the right.
     */
    public void moveRight(){
        moveHorizontal(20);
    }

    /**
     * Move the circle a few pixels to the left.
```



Class Circle

java.lang.Object
Circle

public class Circle
extends Object

A circle that can be manipulated and that draws itself on a canvas.

Version:
1.0. (15 July 2000)

Author:
Michael Kolling and David J. Barnes

Field Summary

Modifier and Type	Field	Description
static final double	PI	

Constructor Summary

Constructor	Description
Circle()	

Method Summary

Modifier and Type	Method	Description
void	changeColor(String newColor)	Change the color.
void	changeSize(int newDiameter)	Change the size.
void	makeInvisible()	
void	makeVisible()	
void	moveDown()	Move the circle a few pixels down.
void	moveHorizontal(int distance)	Move the circle horizontally.
void	moveLeft()	Move the circle a few pixels to the left.

5.b. Para ocultar como esta implementado el codigo realmente

6.a. Que puede ser accedido por la instancia
6.b. Que puede ser accedido sin una instancia
6.c. Que su valor no va a cambiar nunca

7.a. Que es un valor entero
7.b. pi * (256/2)**2 = 51471.85
7.c. 3.5232768214520586e+38
7.d. Que solo pudiera ser positivo

8.a. Solo habria uno y habria el mismo numero de circulos, es decir 100
8.b. Para cualquier circulo PI es el mismo mientras que el diametro no lo es necesariamente

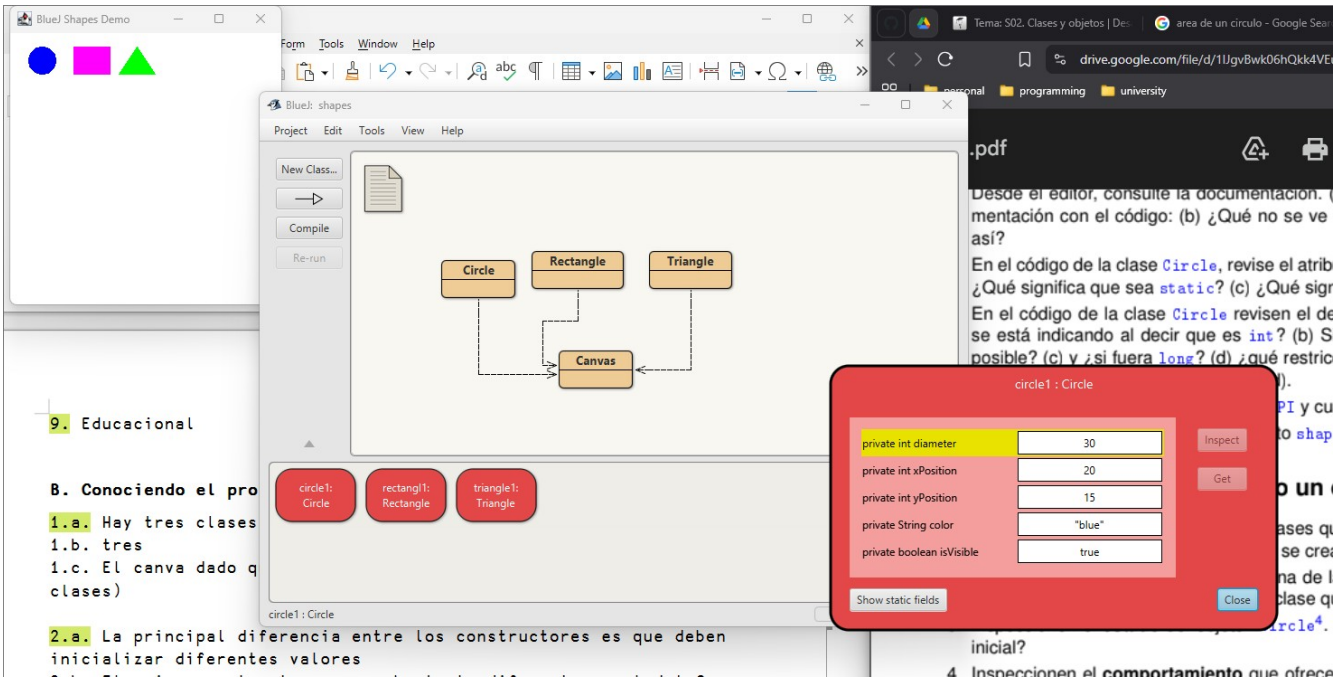
9. Educacional

B. Conociendo el proyecto shapes

1.a. Hay tres clases triangulo, rectangulo y circulo
1.b. tres
1.c. El canva dado que ahi es donde se dibujaran las figuras (otras clases)

2.a. La principal diferencia entre los constructores es que deben inicializar diferentes valores
2.b. El unico constructor que es bastante diferente es el del Canva el cual es privado a diferencia de los otros que son publicos

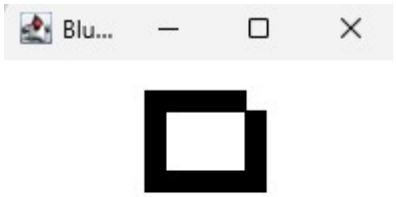
3.a



3.b. Azul

4.a. misma captura que en 3.a
4.b. El unico que no aparece es PI y debe ser dado a que PI es independiente del circulo

5.a. Se necesitaron 4 clases rectangulo
5.b. Se necesitan 4 objectos
5.c.



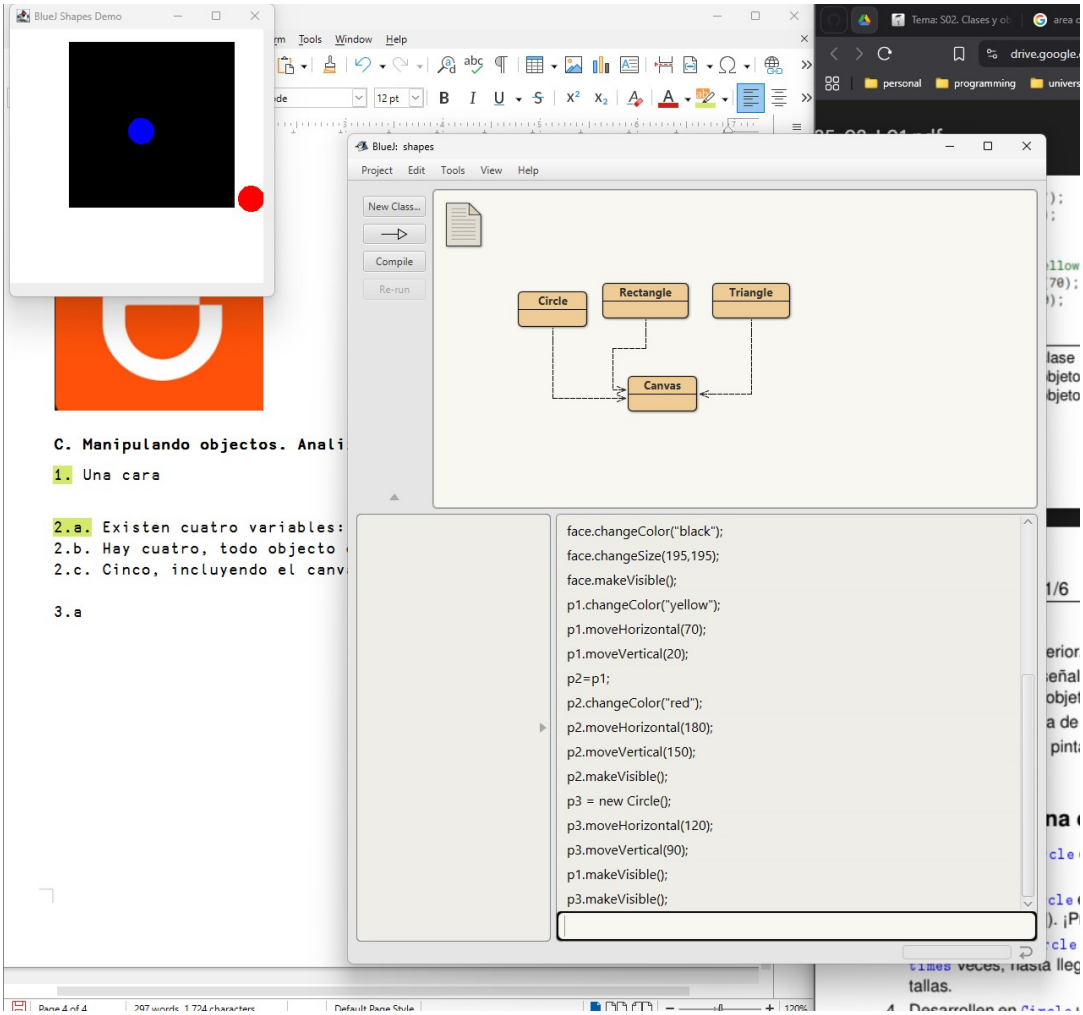
5.d

C. Manipulando objectos. Analizando y escribiendo codigo

1. Una cara

2.a. Existen cuatro variables: face, pointOne, pointTwo, pointThree
2.b. Hay cuatro, todo objecto es una variable
2.c. Cinco, incluyendo el canva

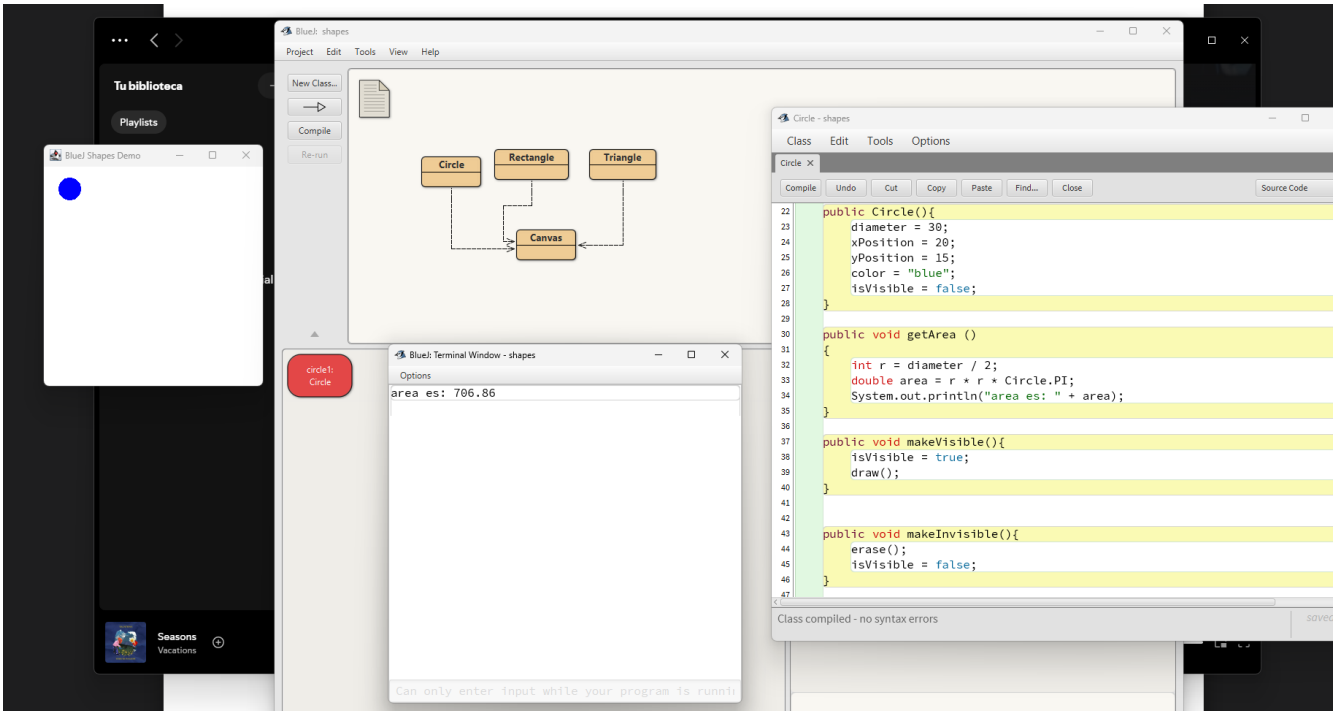
3.a



4.a. No tengo idea de que es la segunda imagen
4.b. No tiene sentido

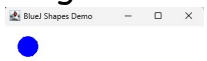
D. Extendiendo una clase

1.

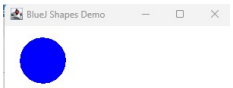


2.

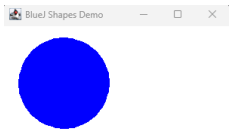
original



aumentado 50%



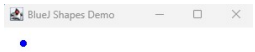
aumentado 100%



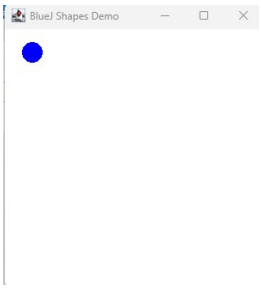

```
public void bigger (int per)
{
    changeSize(diameter + ((int) diameter * (per / 100)));
}
```

3.

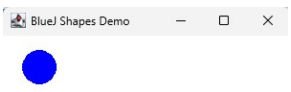
area <= 50



area <= 400



area <= 1000

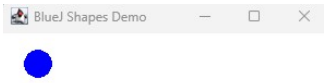


```
    public void shrink (int times, int area)
    {
        while (getArea() > area)
        {
            diameter -= 1;
        }
        changeSize(diameter);
    }
```

Para esto hacemos que getArea no imprima sino que retorne el area

4.

para un area de 500




```
    public Circle (int area)
    {
        diameter = (int) (2 * Math.sqrt(area / Circle.PI));
        xPosition = 20;
        yPosition = 15;
        color = "blue";
        isVisible = true;
    }
```

5. Un metodo que retorne el radio

```
public double getRadius ()
{
    return diameter / 2;
}
```

6.

Modifier and Type	Method	Description
void	**bigger**(int per)	
void	**changeColor**(String newColor)	Change the color.
void	**changeSize**(int newDiameter)	Change the size.
double	**getArea**()	
double	**getRadius**()	
void	**makeInvisible**()	
void	**makeVisible**()	
void	**moveDown**()	Move the circle a few pixels down.
void	**moveHorizontal**(int distance)	Move the circle horizontally.
void	**moveLeft**()	Move the circle a few pixels to the left.
void	**moveRight**()	Move the circle a few pixels to the right.
void	**moveUp**()	Move the circle a few pixels up.
void	**moveVertical**(int distance)	Move the circle vertically.
void	**shrink**(int times, int area)	
void	**slowMoveHorizontal**(int distance)	Slowly move the circle horizontally.
void	**slowMoveVertical**(int distance)	Slowly move the circle vertically

Methods inherited from class java.lang.Object

clone, equals, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

3. Marble game

A. creando una nueva clase

```
/* estos valores establecen la posicion dentro de una matriz de celdas, mas
 * no la posicion donde la celda sera dibujada en el canva; para obtener esos
 * valores se debe multiplicar col y row por el mxwidth
 * */
private int col;
private int row;

/* se refiere a si la celda puede alojar a una canica sin importar
 * su color, si permissive es true entonces cualquier color puede permanecer aqui,
 * si es false solo celdas de color 'onlyAccepts' podran
 * */
private boolean permissive;

/* este atributo solo tiene sentido si permissive es false y se refiere
 * al color que debera ser la canica para exitar perder
 * */
private String   onlyAccepts;

/* dado que el fondo del juego es blanco haremos uso de esto para no crear
 * N * N rectangulos, sino mas bien solo los que no son permisivos, si esta
 * celda es no permisiva entonces el attrb floor tendra un valor; null en
 * caso contrario
 *  */
private Rectangle floor;

/* se refiere si hay una canica la cual esta sobre esta celda, claramente
 * esto solo puede pasar si la celda es permisiva o si la canica y el
 * color de la celda son el mismo
 * */
private boolean beingPressed;
```

dado a limitaciones de tiempo no se pudo completar este punto, sin embargo el juego es totalmente funcional y con todo lo que se pedia, esperamos que en la sustentacion se pueda resolver cualquier tipo de pregunta de este estilo para terminal el laboratorio de una manera satisfactoria.

RETROSPECTIVA

1. ¿Cuál fue el tiempo total invertido en el laboratorio por cada uno de ustedes? (Horas)

Aproximadamente unas 13 horas

2. ¿Cuál es el estado actual del laboratorio? ¿Por qué?
Parcialmente completo, aproximadamente un 95%, dado que las preguntas teoricas del punto 3 no se alcanzaron a resolver debido a problemas de tiempo trabajando en el juego

3. Considerando las prácticas XP del laboratorio, ¿cuál fue la más útil? ¿por qué?

Los unit tests; aunque no se hicieron propiamente aquí, si se testeo cada funcion que se iba definiendo

4. ¿Cuál consideran fue el mayor logro? ¿Por qué?

Tener el juego totalmente funcional y sin mayor dependencia, funciona perfectamente y cumple todo lo solicitado

5. ¿Cuál consideran que fue el mayor problema técnico? ¿Qué hicieron para resolver?

El mayor problema tecnico fue el tema de entender como funcionaba Jpanel y se resolvio con la JavaAPI

6. ¿Qué hicieron bien como equipo? ¿Qué se comprometen a hacer para mejorar los resultados?
La distribucion del trabajo, si bien no terminamos las preguntas teoricas el mayor reto fue cumplido a cabalidad

7. ¿Qué referencias usaron? ¿Cuál fue la más útil? Incluyan citas con estándares adecuados.
Los dos mayores recursos fueron JavaAPI y chatGPT sin uso excesivo del mismo dado que no le pediamos codigo sino mas bien como debiamos buscar la informacion

https://docs.oracle.com/javase/8/docs/api/

conversacion con ChatGPT durante todo el proyecto:
https://chatgpt.com/share/68b3b416-fa8c-8001-bef6-fe732705725f