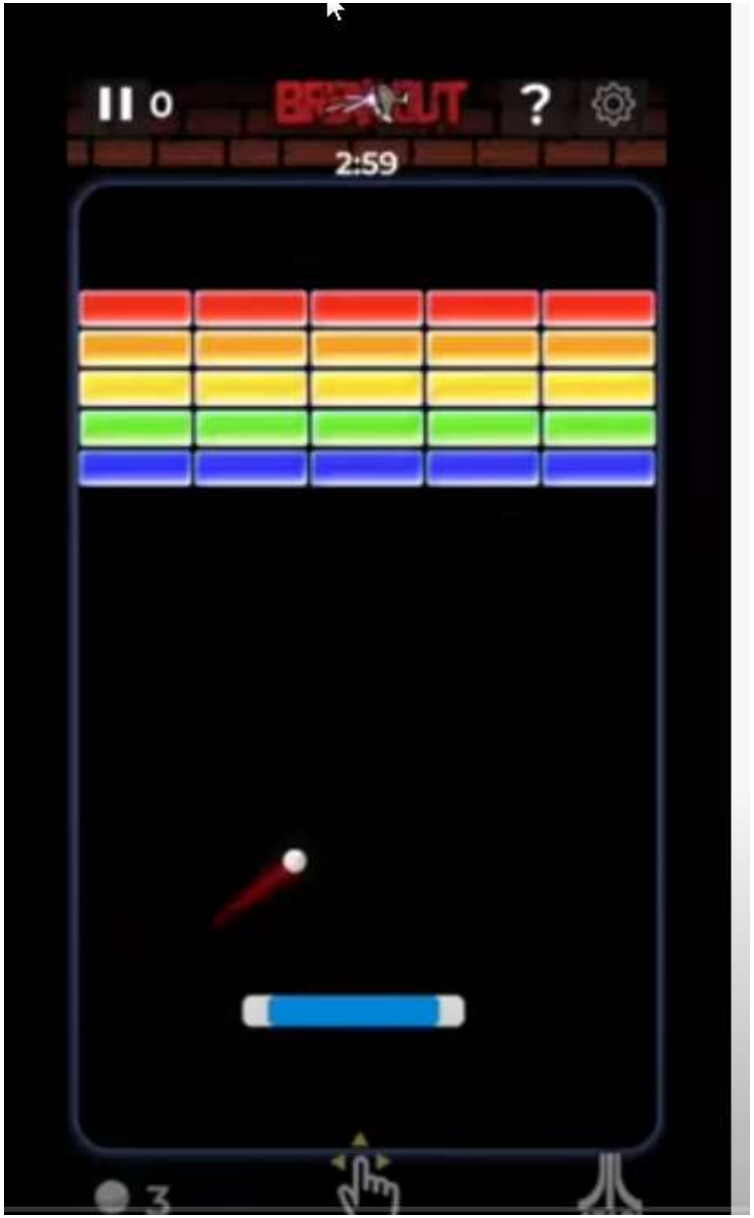
Greenfoot Breakout

By Derek Peacock

Atari Breakout

https://www.youtube.com/watch?v=JruswoS8mPU&ab_channel=PuzzlingGames



ShapeSprite

```java
/**
 * This is a basic Actor where the image
 * is a drawn shape such as a Rectangle or Oval.
 *
 * @author Derek Peacock
 * @version 0
 */
public class ShapeSprite extends Actor
{
    protected Shapes shape;

    protected int width;
    protected int height;

    protected int speed = 4;

    protected GreenfootImage image;

    public ShapeSprite(Shapes shape, int width, int height)
    {
        this.width = width;
        this.height = height;
        this.shape = shape;
```

By default a Red Rectangle
(the paddle) or a Blue Circle (the ball)
are created as images.

```java
public ShapeSprite(Shapes shape, int width, int height)
{
    this.width = width;
    this.height = height;
    this.shape = shape;

    image = new GreenfootImage(width, height);

    if(shape == Shapes.Rectangle)
    {
        setColor(Color.RED);
    }
    else setColor(Color.BLUE);

    setImage(image);
}
```

Shape colour & speed

```
public void setColor(Color color)
{
    image.setColor(color);

    if(shape == Shapes.Rectangle)
        image.fill();
    else
        image.fillOval(0, 0, width, height);
}

public void setSpeed(int speed)
{
    this.speed = speed;
}
```

```
 */
public enum Shapes
{
    Rectangle, Oval
}
```

The Ball (a kind of ShapeSprite)

```
11 public class Ball extends ShapeSprite
12 {
13     private GameWorld game;
14
15     // Current velocity change in x and y
16
17     private int dx;
18     private int dy;
19
20     public Ball(int width, int height)
21     {
22         super(Shapes.Oval, width, height);
23
24         dx = speed; dy = speed;
25         turn(45);
26     }
27
```

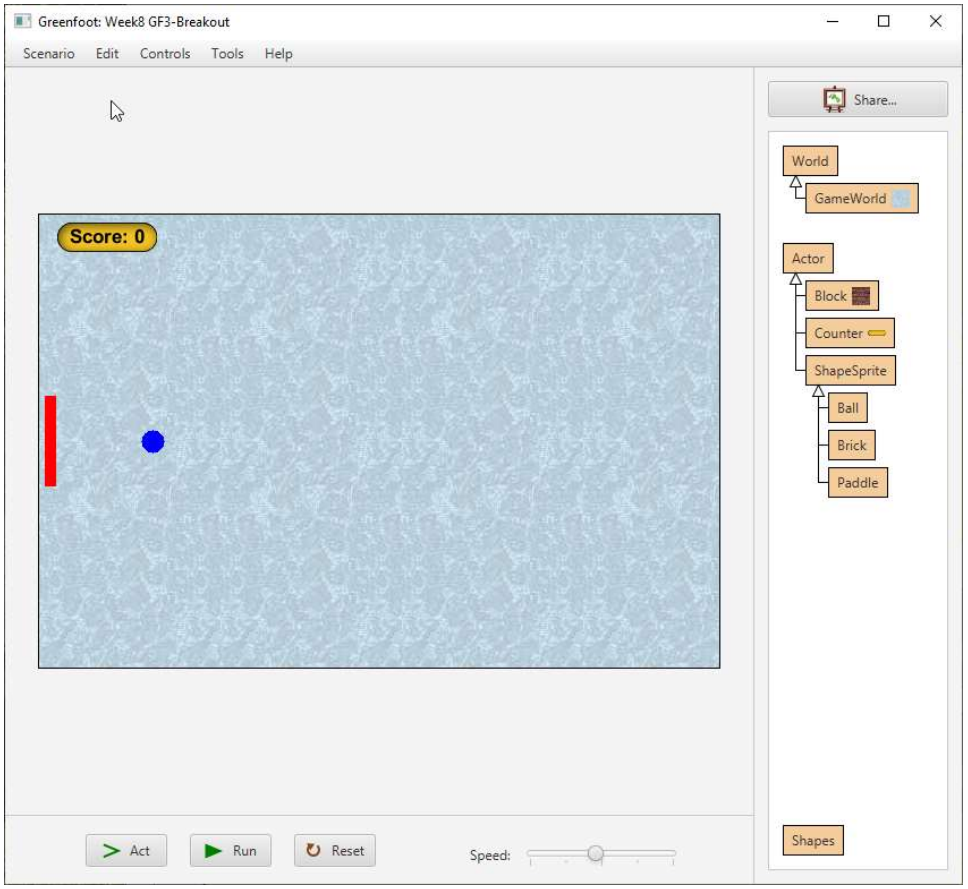
The Paddle (a kind of ShapeSprite)

```
/**
 * This method moves the paddle up or down at the
 * current speed.
 */
public void move()
{
    int x = getX(); int y = getY();

    if(Greenfoot.isKeyDown("down") && !isAtEdge())
    {
        y += speed;
    }

    if(Greenfoot.isKeyDown("up") && y > speed)
    {
        y -= speed;
    }

    setLocation(x, y);
}
```



GameWorld

```
public class GameWorld extends World
{
    private static final int SCREEN_WIDTH = 600;
    private static final int SCREEN_HEIGHT = 400;

    private Paddle paddle;
    private Ball ball;

    private Counter score;

    /**
     * Set the screen size, then create a paddle, ball
     * and a score button. Setup lines of bricks.
     */
    public GameWorld()
    {
        super(SCREEN_WIDTH, SCREEN_HEIGHT, 1);
```

```
public GameWorld()
{
    super(SCREEN_WIDTH, SCREEN_HEIGHT, 1);

    paddle = new Paddle(10, 80);
    ball = new Ball(20,20);

    addObject(paddle, 10, 200);
    addObject(ball, 100, 200);

    score = new Counter("Score: ");
    addObject(score, 60, 20);

    setupBricks();
}
```

Bouncing the Ball

```
public void act()
{
    game = (GameWorld)getWorld();
    move();
}

private void move()
{
    int x = getX(); int y = getY();

    if(x >= game.getWidth() - width)
        dx = -speed;

    if(y >= game.getHeight() - height)
        dy = -speed;

    if(x <= 0)
    {
        dx = 0; dy = 0;
         game.endGame(false);
    }

    if(y <= 0)
        dy = speed;
```

If the ball hits the right side or left side then reverse the x speed

If the ball hits the bottom side or top side reverse the y speed

Check for collisions with brick or paddle

If the ball hits the left side end the game

```
    if(y <= 0)
        dy = speed;

    checkCollisions();

    setLocation(x + dx, y + dy);
}
```

Check for collisions

```
private void checkCollisions()
{
    if(getOneIntersectingObject(Paddle.class) != null)
    {
        dx = speed;
        Greenfoot.playSound("pong.wav");
        return;
    }

    if(getOneIntersectingObject(Brick.class) != null)
    {
        removeTouching(Brick.class);
        game.increaseScore();
        Greenfoot.playSound("pong.wav");

        dx = -dx;
    }
}
```

If the ball collides with the **paddle** play sound and move right

If the ball collides with a **brick**, remove it play a sound and bounce back

Practical Exercise

- Get the ball bouncing around the screen
- Get the ball bouncing off the paddle
- Setup at least one line of bricks
- Get the ball removing bricks
- Get win game and lose game working
- Add another line of bricks
- https://www.greenfoot.org/doc/joy-of-code (see smoke and mirrors)
- https://www.youtube.com/user/18km (Greenfoot Channel)