Day6: Greenfoot Demo Game – More Ideas

Pulling it all together!
Let’s run it!

• What do you notice?
Let’s look at the code!

• Look closely at **Player**
• Look at **MyPicture**
• Look at the **Levels**
• Look at **Gateway**
  • Resets the image
• What else do you notice?
public class SizeScalingCharm extends Reward {

    public void checkCaught() {
        Player actor = null;
        actor = (Player) getOneIntersectingObject(Player.class);
        if (actor != null) {
            this.getWorld().removeObject(this);
            actor.shrink();
        }
    }

    public void act() {
        super.act();
        checkCaught();
    }
}
Making the player shrink

```java
public void shrink()
{
    scale(20,20);
    isShrunken = true;
    fall();
}

public void unshrink()
{
    if(isShrunken)
    {
        load("pinkRobot.png");
        isShrunken = false;
    }
}
```
Add other transforms to MyPicture

```java
public void makePictureTransparent()
{
    for (Pixel p : getPixels())
    {
        if (p.getAlpha() != 0) {
            p.setAlpha(30);
        }
    }
}
```
public void alphaBlend(Picture top, int xStart, int yStart) {
    // the x and y work through the top picture
    for (int x = 0; x < top.getWidth() && x + xStart < this.getWidth(); x++) {
        for (int y = 0; y < top.getHeight() && y + yStart < this.getHeight(); y++) {
            int alpha = top.getPixel(x, y).getAlpha();
            System.out.println("alpha is "+alpha + "at x "+ x + " and y "+ y);

            int xPlacement = x + xStart;
            int yPlacement = y + yStart;
            int placementAlpha = this.getPixel(xPlacement, yPlacement).getAlpha();

            // Mix reds
            int topRed = top.getPixel(x, y).getRed();
            int bottomRed = this.getPixel(xPlacement, yPlacement).getRed();
            int red = topRed * alpha / 255 + bottomRed * (255 - alpha) / 255;

            // Mix greens
            int topGreen = top.getPixel(x, y).getGreen();
            int bottomGreen = this.getPixel(xPlacement, yPlacement).getGreen();
            int green = topGreen * alpha / 255 + bottomGreen * (255 - alpha) / 255;

            // Mix blues
            int topBlue = top.getPixel(x, y).getBlue();
            int bottomBlue = this.getPixel(xPlacement, yPlacement).getBlue();
            int blue = topBlue * alpha / 255 + bottomBlue * (255 - alpha) / 255;

            this.getPixel(xPlacement, yPlacement).setRed(red);
            this.getPixel(xPlacement, yPlacement).setGreen(green);
            this.getPixel(xPlacement, yPlacement).setBlue(blue);

            int maxAlpha = Math.max(placementAlpha, alpha);
            this.getPixel(x+xStart, y+yStart).setAlpha(maxAlpha);
        }
    }
}
Try adding more feature to your work from this morning and think about your final game!

- Image manipulation
- Level Changes
- Different platform and ladder configurations
- Charms with meaning