```java
import java.awt.Color;
import java.awt.Font;
import java.awt.Graphics;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import java.awt.event.KeyEvent;
import java.awt.event.KeyListener;
import java.awt.event.MouseEvent;
import java.awt.event.MouseListener;
import java.awt.image.BufferedImage;
import java.io.File;

import javax.sound.sampled.AudioSystem;
import javax.sound.sampled.Clip;
import javax.swing.ImageIcon;
import javax.swing.JFrame;
import javax.swing.JPanel;
import javax.swing.Timer;


public class MarbleManiac extends JPanel {

	private static final int WIDTH = 1600;
	private static final int HEIGHT = 900;

	private double time = 0;
	// advancing
	private double advance = 0;

	private double endingAdvance = 0;
	private int whichFace = 0;
	private int hackiaStep = 0;

	private Hackia hackia;

	// emotion signals
	public boolean jewelry = false;
	public boolean pool = false;
	public boolean church = false;
	public boolean ending = false;

	private BufferedImage image;
	private Graphics g;
```

```
public Timer timer;
private Ball clicker;

static File theme = new File("AXT High School OST.wav");
static File song = theme;
static Clip clip;

public MarbleManiac() {
	// buffered image for animation
	image = new BufferedImage(WIDTH, HEIGHT, BufferedImage.TYPE_INT_RGB);
	g = image.getGraphics();

	timer = new Timer(0, new TimerListener());
	timer.start();
	addKeyListener(new Keyboard());
	setFocusable(true);
	addMouseListener(new Mouse());

	clicker = new Ball(200, 350, 5, Color.white);
	hackia = new Hackia(200, 350);
}

static void ChangeSong(File newSong) {
	if (song != newSong) {
		clip.stop();
		song = newSong;
		PlaySound();
	}

}

static void PlaySound() {
	try { // file sound is viewed as a clip
		clip = AudioSystem.getClip(); // gets the clip
		clip.open(AudioSystem.getAudioInputStream(song)); // opens the clip
		clip.start(); // starts the clip
		clip.loop(999);
	} catch (Exception e) { // catches exceptions

	}
}

public static void run(File sound) {
		while (true) // continuously plays the audio file
		{
			PlaySound();
		}
	}

	private class Mouse implements MouseListener {

		@Override
		public void mouseClicked(MouseEvent e) {
			if (e.isMetaDown()) {
				clicker.setX(e.getX());
				clicker.setY(e.getY());
				System.out.println(clicker.getX() + " " + clicker.getY());
			}

		}

		@Override
		public void mouseEntered(MouseEvent arg0) {
			// TODO Auto-generated method stub

		}

		@Override
		public void mouseExited(MouseEvent arg0) {
			// TODO Auto-generated method stub

		}

		@Override
		public void mousePressed(MouseEvent arg0) {
			// TODO Auto-generated method stub

		}

		@Override
		public void mouseReleased(MouseEvent arg0) {
			// TODO Auto-generated method stub

		}

	}
```

```
// keyboard input
private class Keyboard implements KeyListener {

	@Override
	/**
	 * keyPressed checks what key is pressed and allows the bumper to move to that
	 * position
	 */
	public void keyPressed(KeyEvent e) {

		if (e.getKeyCode() == KeyEvent.VK_W) {
			hackia.setY(hackia.getY() - 10);
			whichFace = 1;
			hackiaStep++;
		}
		if (e.getKeyCode() == KeyEvent.VK_S) {
			hackia.setY(hackia.getY() + 10);
			whichFace = 4;
			hackiaStep++;
		}
		if (e.getKeyCode() == KeyEvent.VK_A) {
			hackia.setX(hackia.getX() - 10);
			whichFace = 3;
			hackiaStep++;
		}
		if (e.getKeyCode() == KeyEvent.VK_D) {
			hackia.setX(hackia.getX() + 10);
			whichFace = 2;
			hackiaStep++;
		}

		if (e.getKeyCode() == KeyEvent.VK_BACK_SPACE) {
		}

		if (e.getKeyCode() == KeyEvent.VK_ENTER) {
			advance++;
			if (ending == true) {
				endingAdvance++;
			}
		}

		if (e.getKeyCode() == KeyEvent.VK_SPACE) {
				}
			}

			@Override
			public void keyReleased(KeyEvent e) {
				// TODO Auto-generated method stub

			}

			@Override
			public void keyTyped(KeyEvent e) {

			}

		}

		public void drawMain() {
			GraphicsUtilities.drawPicture(g, "main.png", 0, 0, WIDTH, HEIGHT);
		}

		public void drawMain2() {
			GraphicsUtilities.drawPicture(g, "main2.png", 0, 0, WIDTH, HEIGHT);
		}

		public void drawRoom() {
			GraphicsUtilities.drawPicture(g, "first.png", 0, 0, WIDTH, HEIGHT);
		}

		public void drawDownstairs() {
			GraphicsUtilities.drawPicture(g, "second.png", 0, 0, WIDTH, HEIGHT);
		}

		private class TimerListener implements ActionListener {
			public void actionPerformed(ActionEvent e) {

				drawMain();
				if (ending == true) {
					drawMain2();
				}

				if (advance == 0) {
					g.setColor(Color.white);
					g.setFont(new Font("Poor Richard", Font.BOLD, 90));
```

```
                                        // change at end (the number after %)
                                        if (time % 50 <= 25) {
                                                drawMain();
                                        } else {
                                                g.drawString("Press Enter To Advance", 360, 700);
                                        }
                                } else {
                                        g.setFont(new Font("Poor Richard", Font.BOLD, 30));
                                        if (advance == 1) {
                                                g.drawString("Marbles, the elements that individualize
each being,", 0, 700);
                                        }
                                        if (advance == 2) {
                                                g.drawString("Have been the key to existence since the
beginning of time", 0, 700);
                                        }
                                        if (advance == 3) {
                                                g.drawString(
                                                                "They each portray the inner thoughts and
beliefs of a person and are vital to mankind", 0,
                                                                700);
                                        }
                                        if (advance == 4) {
                                                g.drawString("Each having its own counterpart,", 0, 700);
                                        }
                                        if (advance == 5) {
                                                g.drawString("They include...", 0, 700);
                                        }
                                        if (advance == 6) {
                                                g.drawString("Love - Hate", 0, 700);
                                        }
                                        if (advance == 7) {
                                                g.drawString("Peace - Violence", 0, 700);
                                        }
                                        if (advance == 8) {
                                                g.drawString("Happiness - Sadness", 0, 700);
                                        }
                                        if (advance == 9) {
                                                g.drawString("These emotions, each symbolized by
marbles, are identified by their color", 0, 700);
                                        }
                                        if (advance == 10) {
```

```
                                        g.drawString("Blue conveying emotions associated with
positivity", 0, 700);
                                }
                                if (advance == 11) {
                                        g.drawString("Red representing emotions associated with
negativity", 0, 700);
                                }
                                if (advance == 12) {
                                        g.drawString(
                                                        "The ultimate goal of a human’s life is to
channel his/her energy and have all five emotions be blue.",
                                                        0, 700);
                                }
                                if (advance == 13) {
                                        g.drawString("As of today, May 18th 2019,", 0, 700);
                                }
                                if (advance == 14) {
                                        g.drawString("You, Hackia, have been poisoned and
stripped of all charisma", 0, 700);
                                }
                                if (advance == 15) {
                                        g.drawString("You must find and restore all five of your
marbles to blue before it is too late.", 0,
                                                        700);
                                }
                                if (advance == 16) {
                                        g.drawString("Wake up, your time starts now", 0, 700);
                                }
                                if (advance == 17) {
                                        drawRoom();
                                }
                                if (advance == 18) {
                                        drawRoom();
                                        g.drawString("Was that all a dream?", 0, 700);
                                }
                                if (advance == 19) {
                                        drawRoom();
                                        g.drawString(
                                                        "Precious stones and gems are what a
pirate loves to find in this type of “treasure chest”",
                                                        0, 700);
                                }
                                if (advance == 20) {
```

```
                    drawRoom();
                    g.drawString(
                            "The locations you can go to include the
jewlery box, neighborhood pool, or neighborhood church",
                            0, 700);
                }
                if (advance == 21) {
                    drawRoom();
                    g.drawString("Your quest has just begun...", 0, 700);
                }
                if (advance == 22) {
                    clicker.draw(g);
                    drawRoom();
                    g.drawString("Go right to continue", 0, 700);
                    whichToDraw();
                    borderPatrol();
                    if (hackia.getX() >= WIDTH - hackia.getWidth() + 50) {
                        hackia.setX(WIDTH - hackia.getWidth() + 50);
                        //set background to outside
                    }
                }

                if (endingAdvance == 0 && ending == true) {
                    drawMain2();
                    g.setFont(new Font("Poor Richard", Font.BOLD, 50));
                    g.drawString("Congratulations! You have triumphed.", 365,
700);

                } else {
                    g.setFont(new Font("Poor Richard", Font.BOLD, 30));
                    if (endingAdvance == 1) {
                        g.drawString("You have successfully collected all
three marbles", 0, 700);
                    }
                    if (endingAdvance == 2) {
                        g.drawString(
                                "Using your capabilities and energy,
you transformed your cynical emotions to radiate optimism",
                                0, 700);
                    }
                    if (endingAdvance == 3) {
                        g.drawString("You have reclaimed love, peace, and
happiness", 0, 700);
```

```
                    }
                    if (endingAdvance == 4) {
                        g.drawString("You are officially filled with blue
colored emotions rather than red ones", 0,
                                700);
                    }
                    if (endingAdvance == 5) {
                        g.drawString("Thanks for playing.", 0, 700);
                    }

                }
            }

//          System.out.println("" + endingAdvance);
//          System.out.println(advance);
            time++;
            repaint();

        }

    }

    public void borderPatrol() {
        if (hackia.getX() >= WIDTH - hackia.getWidth() + 50) {
            hackia.setX(WIDTH - hackia.getWidth() + 50);
        }

        if (hackia.getX() <= -50) {
            hackia.setX(-50);
        }

        if (hackia.getY() >= HEIGHT - hackia.getHeight()) {
            hackia.setY(HEIGHT - hackia.getHeight());
        }

        if (hackia.getY() <= 300) {
            hackia.setY(300);
        }
    }

    public void whichToDraw() {
        if (whichFace == 3 && hackiaStep % 2 == 1) {
            hackia.drawLL(g);
```

```
		} else if (whichFace == 3 && hackiaStep % 2 == 0) {
			hackia.drawLR(g);
		} else if (whichFace == 2 && hackiaStep % 2 == 1) {
			hackia.drawRL(g);
		} else if (whichFace == 2 && hackiaStep % 2 == 0) {
			hackia.drawRR(g);
		} else if (whichFace == 1 && hackiaStep % 2 == 1) {
			hackia.drawBL(g);
		} else if (whichFace == 1 && hackiaStep % 2 == 0) {
			hackia.drawBR(g);
		} else if (whichFace == 4 && hackiaStep % 2 == 1) {
			hackia.drawFR(g);
		} else {
			hackia.drawFL(g);
		}
	}

	public void paintComponent(Graphics g) {
		g.drawImage(image, 0, 0, getWidth(), getHeight(), null);
	}

	public static void main(String[] args) {

		// frame settings
		JFrame frame = new JFrame("marbles...");
		frame.setSize(WIDTH + 18, HEIGHT + 47);
		frame.setLocation(150, 50);
		frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
		frame.setContentPane(new MarbleManiac());
		frame.setVisible(true);

		ImageIcon img = new
ImageIcon("C:\\Users\\797439\\ITCS\\ACLHackathon\\icon.png");
		frame.setIconImage(img.getImage());

		PlaySound();
	}

}
```