Secondary Technical School of Electrical Engineering

Information technology

Ječná 517, 120 00 Nové Město

Pong

Arcade Game

Oleg Goshovskyy

Information technology

2024

Table of contents

Table of Figures

1 Project goal

The goal was to program a classic arcade game Pong where players control a paddle to deflect a ball at his enemy. The game can be played in two players where both control paddle on different sides, or alone where player plays against a computer.

2 Software

The game was made in IntelliJ IDEA 2023.2.2 (Ultimate Edition), using Java SE Development Kit 16.0.2. No external libraries were used.

3 Game description

This game is about player-controlled paddles deflecting ball. The game runs in a window.

3.1 Mechanics

Method for title screen:

```java
/**
 * Draws the title screen with menu options.
 *
 * @param g2 The Graphics2D object.
 */
private void drawTitleScreen(Graphics2D g2) {  1 usage  oleg +2
    String text = "PONG";
    drawTitle(text, g2);
    g2.setFont(g2.getFont().deriveFont(Font.BOLD,  size: 24f));
    text = "Player vs Player";
    int line = 0;
    if (commandNum == line) {
        drawChoice(text, g2, commandNum);
    }
    line = drawMenu(text, g2, line);
    text = "Player vs Computer";
    if (commandNum == line) {
        drawChoice(text, g2, commandNum);
    }
    line = drawMenu(text, g2, line);
    text = "Settings";
    if (commandNum == line) {
        drawChoice(text, g2, commandNum);
    }
    line = drawMenu(text, g2, line);
    text = "Quit";
    drawMenu(text, g2, line);
    if (commandNum == line) {
        drawChoice(text, g2, commandNum);
    }
}
```

Figure 1: drawTitleScreen method in UI

This menu has 4 choices leading into other menus. Other menus are built on the same principle.

```java
/**
 * Updates the computer paddle's position based on the ball's position.
 * The paddle moves towards the ball's Y-coordinate at ySpeed and if the ball is 25 pixels or further then it moves at yMaxSpeed + ySpeed.
 */
@Override 8 usages  Oleg
public void update() {
    directions = Directions.NONE;
    if (y + height / 2 - 25 > gp.getBallY() + gp.getBallHeight() / 2) {
        if (y >= 0) {
            y -= yMaxSpeed;
            directions = Directions.UP;
        }
    } else if (y + height / 2 > gp.getBallY() + gp.getBallHeight() / 2) {
        if (y >= 0) {
            y -= ySpeed;
            directions = Directions.UP;
        }
    }
    if (y + height / 2 + 25 < gp.getBallY() + gp.getBallHeight() / 2) {
        if (y <= gp.getScreenHeight() - height) {
            y += yMaxSpeed;
            directions = Directions.DOWN;
        }
    } else if (y + height / 2 < gp.getBallY() + gp.getBallHeight() / 2) {
        if (y <= gp.getScreenHeight() - height) {
            y += ySpeed;
            directions = Directions.DOWN;
        }
    }
}
```

Figure 2: update method in Computer

Method used for updating position of paddle controlled by computer.

```java
/**
 * Handles key press events in the play state.
 *
 * @param code the key code of the pressed key
 */
private void playState(int code) {  1 usage  Oleg +1
    switch (code) {
        case KeyEvent.VK_W -> leftPlayerUpPressed = true;
        case KeyEvent.VK_S -> leftPlayerDownPressed = true;
        case KeyEvent.VK_UP -> rightPlayerUpPressed = true;
        case KeyEvent.VK_DOWN -> rightPlayerDownPressed = true;
        case KeyEvent.VK_P -> gp.setGameState(GameState.PAUSE_STATE);
        case KeyEvent.VK_ESCAPE -> gp.setGameState(GameState.MENU_STATE);
    }
}
```

Figure 3: playStateMethod in KeyHandler

This method checks if player is pressing any buttons affecting the game.

Method for running the game:

```java
/**
 * Main game loop runs at the specified FPS.
 * Updates and renders the game state.
 */
@Override  Oleg
public void run() {
    double drawInternal = 1000000000.0 / FPS;
    double delta = 0;
    long lastTime = System.nanoTime();
    long currentTime;
    long timer = 0;
    while (gameThread.isAlive()) {
        currentTime = System.nanoTime();
        delta += (currentTime - lastTime) / drawInternal;
        timer += currentTime - lastTime;
        lastTime = currentTime;
        if (delta >= 1) {
            update();
            drawToTempScreen(); // Draw everything to the buffered image
            drawToScreen(); // Draw the buffered image to the screen
            delta--;
        }
        if (timer >= 1000000000) {
            timer = 0;
        }
    }
}
```

Figure 4: run method in GamePanel (Game loop)

This method checks how much time has passed, divides that by drawInterval and adds that to delta. If delta is bigger or equal to 1 then the game updates and repaints, delta has 1 removed from it and loop repeats.

4 Manual

The player on the left side can control the left paddle with keys “W” and “S”, and player on the right controls his paddle with arrow up “↑” and arrow down “↓”, players can go in menu by pressing "ESC". The same buttons are used along with "Enter" to navigate in the menu. Game can be paused anytime while in game by pressing “P”.

5 Conclusion

The development of this project was smoother than I expected. I managed to implement most of functions that I wanted my game to include.

6 Sources

Font used ingame: https://www.fontspace.com/press-start-font-f5841