

One_Player_Pong

//One Player Pong Game

//Mr. H.

```
import ddf.minim.*;
import ddf.minim.analysis.*;
import ddf.minim.effects.*;
import ddf.minim.signals.*;
import ddf.minim.spi.*;
import ddf.minim.ugens.*;
Minim minim;
AudioPlayer soundFile;
AudioPlayer backgroundMusic;
AudioInput input;
```

```
int x = 10;
int y = 10;
int bounceX = 1;
int bounceY = 1;
int score = 0;
int highScore = 0;
int lives = 3;
int stage = 0;
boolean run = true;
boolean scoreEnable = true;
int scroll = -350;
```

```
void setup() {
  size(600, 600);
  background(255);
  noCursor();
  minim = new Minim(this);
  soundFile = minim.loadFile("tennis.mp3");
  input = minim.getLineIn();
  soundFile.play();
  backgroundMusic = minim.loadFile("Song2.mp3");
  input = minim.getLineIn();
  backgroundMusic.loop();
}
```

```
void draw() {
  if (stage == 0) {
    cursor();
    background(180);
    fill(255);
    rect(200, 200, 200, 100);
    fill(0);
```

```

textSize(60);
text("Classic Pong", 120, 120);
textSize(20);
text("Press Here", 250, 245);
text("To Start", 260, 275);
text("Instructions:", 100, 400);
text("1. Use the Mouse to move up and down", 100, 430);
text("2. Don't let the ball get by you!", 100, 460);
if (mouseX>=200 && mouseX<=400 && mouseY>=200 && mouseY<=400 && mousePressed)
{
  stage = 1;
}
}

if (run == false) {
  noCursor();
  background(255);
  fill(0);
  text("Score = ", 20, 20);
  text(score, 100, 20);
  text("Lives = ", 20, 40);
  text(lives, 100, 40);
  text("High Score = ", 200, 20);
  text(highScore, 340, 20);
  if (lives == 0) {
    textSize(60);
    fill(255,0,0);
    text("Game Over", scroll, 400);
    scroll = scroll + 3;
    if (scroll >= 675) {
      scroll = -350;
    }
    fill(0);
    textSize(20);
    text("Press Mouse for New Game", 177, 248);
    if (mousePressed){
      run = true;
      lives = 3;
      score = 0;
      stage = 1;
      bounceY = 1;
    }
  }
  if (lives <= 3 && lives >= 1) {
    text("Press Mouse To Continue", 180, 248);
    if (mousePressed) {
      run = true;
      bounceY = 1;
    }
  }
}

```

```
}  
}  
fill(255, 0, 255);  
rect(580, mouseY, 10, 50);  
}
```

```
if (stage == 1 && run == true) {  
  noCursor();  
  background(255);  
  fill(0);  
  text("Score = ", 20, 20);  
  text(score, 100, 20);  
  text("Lives = ", 20, 40);  
  text(lives, 100, 40);  
  text("High Score = ", 200, 20);  
  text(highScore, 340, 20);  
  fill(0, 255, 0);  
  ellipse(x, y, 20, 20);  
  x = x + (int(score/5) - lives + 7) * bounceX;  
  y = y + 1 * bounceY;
```

```
  if (x <= 10) {  
    bounceX = bounceX * -1;  
    scoreEnable = true;  
  }  
  if (y >= 580) {  
    bounceY = bounceY * -1;  
  }  
  if (y <= 10) {  
    bounceY = bounceY * -1;  
  }  
}
```

```
if (x >= 590) {  
  lives = lives - 1;  
  run = false;  
  x = 10;  
}  
fill(255, 0, 255);  
rect(580, mouseY, 10, 50);  
if (x >= 580 && y >= mouseY && y <= mouseY + 50) {  
  minim = new Minim(this);  
  soundFile = minim.loadFile("tennis.mp3");  
  input = minim.getLineIn();  
  soundFile.play();  
  bounceX = bounceX * -1;  
  bounceY = (int(random(16)) - 8);  
  if (scoreEnable == true){  
    score = score + 1;
```

```
background(random(255),random(255),random(255));  
}  
scoreEnable = false;  
if (score >= highScore){  
  highScore = score;  
}  
}  
}  
}
```