```
//--------------------PEDRO-----------------------
import processing.video.*;
import processing.sound.*;
//-------------------ANDREY----------------------
int showScreen = 1; //É uma variável inteira que é usada para controlar qual tela deve ser
exibida no momento.
Button playButton;
Button helpButton;
Button exitButton;
Button backButton;
//-------------------GUSTAVO----------------------
int ballSize = 15;
float maxSpeed = 3;
float ballX = 300;
float ballY = 450;
float ballSpeed = 1;
float vecX = random(0.2,1.5);
float vecY = -3;
//--------------------PEDRO-----------------------
PImage bg;
Movie video;
SoundFile bgm;
SoundFile ping;
//--------------------KEVIN-----------------------
int brickWidth = 50; // largura de cada brick
int brickHeight = 25; // altura de cada brick
int numRows = 10; // número de linhas de bricks
int numCols = 10; // número de colunas de bricks
int[][] bricks = new int[numRows][numCols]; // matriz para armazenar os bricks
float xOffset = 50; // x-offset da matriz dos bricks
float yOffset = 100; // y-offset da matriz dos bricks
int[] colors = {
1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
9, 9, 9, 9, 2, 2, 9, 9, 9, 9,
9, 3, 3, 3, 3, 3, 9, 3, 3, 3,
9, 4, 4, 4, 4, 4, 9, 4, 4, 4,
9, 9, 9, 9, 5, 5, 9, 5, 5, 5,
6, 6, 6, 9, 6, 6, 9, 6, 9, 9,
7, 7, 7, 9, 7, 7, 9, 7, 7, 9,
8, 8, 8, 9, 8, 8, 9, 8, 8, 9,
9, 9, 9, 2, 2, 2, 9, 9, 9, 9,
1, 1, 1, 1, 1, 1, 1, 1, 1, 1
};
int[] brickColors = new int[numRows * numCols];
int count = 0;

void setup() {
  size(600, 900);
  //--------------------ANDREY------------------------
  playButton = new Button(width / 2, height / 2 - 50, "Jogar");
  helpButton = new Button(width / 2, height / 2, "Créditos");
  exitButton = new Button(width / 2, height / 2 + 50, "Sair");
  backButton = new Button(width / 2, height - 50, "Voltar");
  //---------------------KEVIN------------------------
  textAlign(CENTER, CENTER);
  ellipseMode(CENTER);
  imageMode(CENTER);
  //---------------------PEDRO------------------------
  ping = new SoundFile(this, "ping.wav");
  bgm = new SoundFile(this, "bgm.mp3");
  video = new Movie(this, "bgmv.mp4");
  //video.loop(); // Inicie a reprodução em loop do vídeo
  video.play();
  bg = loadImage("bg.png");
  //---------------------KEVIN------------------------
  frameRate(60);
  for (int row = 0; row < numRows; row++) {
    for (int col = 0; col < numCols; col++) {
      bricks[row][col] = colors[count]; //preenche a matriz de bricks com valores da variável
"colors" 1 a 9
      count++;
    }
  }
}
//--------------------ANDREY-----------------------
void draw() {
  drawScreen();
}

void drawScreen() {
  switch (showScreen) {
  // Tela do Menu
  case 1:
    image(bg,width/2,height/2);
    fill(0,100);
    rect(width/2, height/2, 160, 190);
    playButton.draw();
    helpButton.draw();
    exitButton.draw();
    if (playButton.isMouseOver()) {
      playButton.hover();
    }
    else if (helpButton.isMouseOver()) {
      helpButton.hover();
    }
    else if (exitButton.isMouseOver()) {
      exitButton.hover();
    }
    break;

  // Tela do Jogo
  case 2:
    //---------------------PEDRO-----------------------
    if (video.available()) {
      video.read();
    }
    //bgm.play();
    image(video, width/2 , height/2);
    strokeWeight(1);
    rectMode(CORNER);
    //---------------------KEVIN-----------------------
    // desenha os bricks na tela
    for (int row = 0; row < numRows; row++) {
      for (int col = 0; col < numCols; col++) {
        if (bricks[row][col] > 0) { // se o valor do brick for maior que zero
          // calcula a posição x e y do brick na tela
          float brickX = col * brickWidth;
          float brickY = row * brickHeight;
          // define a cor do brick com base no seu valor
          if (bricks[row][col] == 1) {
            fill(#CC0001); // vermelho
          } else if (bricks[row][col] == 2) {
            fill(#FB940B); // laranja
          } else if (bricks[row][col] == 3) {
            fill(#FFFF00); // amarelo
          } else if (bricks[row][col] == 4) {
            fill(#01CC00); // verde
          } else if (bricks[row][col] == 5) {
            fill(#03C6C6); // ciano
          } else if (bricks[row][col] == 6) {
            fill(#0000FE); // azul
          } else if (bricks[row][col] == 7) {
            fill(#762CA7); // roxo
          } else if (bricks[row][col] == 8) {
            fill(#FE98BF); // rosa
          } else {
            fill(#FFFFFF); // qualquer outro valor, branco
          }
          // desenha o brick na tela
          pushMatrix();
          stroke(#000000);
          translate(xOffset, yOffset);
          rect(brickX, brickY, brickWidth, brickHeight, 5);
          popMatrix();
        }
      }
    }

    //-------------------GUSTAVO----------------------
    fill(#FFFFFF); //Branco
    noStroke();
    ellipse(ballX, ballY, ballSize, ballSize);
    stroke(0);
    quad(mouseX + 50, height - 40, mouseX + 50, height - 50, mouseX - 50, height - 50,
  mouseX - 50, height - 40);
    ballX = ballX + vecX * ballSpeed;
    ballY = ballY + vecY * ballSpeed;
    if (ballX + ballSize > width) {
      vecX *= -1;
      ping.play();
      if (ballSpeed < maxSpeed);
      ballSpeed += 0.001;
    }
    else if (ballX - ballSize < 0) {
      vecX *= -1;
      ping.play();
      if (ballSpeed < maxSpeed);
      ballSpeed += 0.001;
    }
    else if (ballY + ballSize > height - 45) {
      if (ballX + 50 > mouseX) {
        if (ballX - 50 < mouseX) {
          vecY *= -1;
          ping.play();
          if (ballSpeed < maxSpeed);
          ballSpeed += 0.001;
        }
        else if (ballY + ballSize > height) {
          showScreen = 1;
          bgm.stop();
          ballX = 300;
          ballY = 450;
          vecX = 2;
          vecY = -3;
        }
      }

      else if (ballY + ballSize > height) {
        showScreen = 1;
        bgm.stop();
        ballX = 300;
        ballY = 450;
        vecX = 2;
        vecY = -3;
      }
    }
    else if (ballY - ballSize < 0) {
      vecY *= -1;
      if (ballSpeed < maxSpeed);
      ballSpeed += 0.001;
    }

    //---------------------KEVIN-----------------------
    for (int row = 0; row < numRows; row++) {
      for (int col = 0; col < numCols; col++) {
        if (bricks[row][col] > 0) { // se o valor do brick for maior que zero
          // calcula a posição x e y do brick na tela
          float brickX = (col * brickWidth) + xOffset;
          float brickY = (row * brickHeight) + yOffset;
          // verifica se a bola atingiu o brick
          if (ballX + ballSize / 2 > brickX && ballX - ballSize / 2 < brickX + brickWidth && ballY +
  ballSize / 2 > brickY && ballY - ballSize / 2 < brickY + brickHeight) {
            // diminui o valor do brick em um (ou o remove completamente se o valor for menor
  que 3)
            if (bricks[row][col] > 8) {
              bricks[row][col]--;
            }
            else {
              bricks[row][col] = 0;
            }

            //--------------------GUSTAVO----------------------
            if (ballX > brickX + brickWidth) {
              vecX *= -1;
              if (ballSpeed < maxSpeed);
              ballSpeed += 0.01;
              ping.play();
            }
            else if (ballX < brickX) {
              vecX *= -1;
              if (ballSpeed < maxSpeed);
              ballSpeed += 0.01;
              ping.play();
            }
            if (ballY > brickY + brickHeight) {
              vecY *= -1;
              if (ballSpeed < maxSpeed);
              ballSpeed += 0.01;
              ping.play();
            }
            else if (ballY < brickY) {
              vecY *= -1;
              if (ballSpeed < maxSpeed);
              ballSpeed += 0.01;
              ping.play();
            }
          }
        }
      }
    }
    break;
  // Tela de Crédtos
  case 3:

    backButton.draw();
    if (backButton.isMouseOver()) {
      backButton.hover();
    }
    fill(255);
    ballSpeed = 3;
    ellipse(ballX, ballY, ballSize, ballSize);
    ellipse(-ballX, -ballY, ballSize, ballSize);
    ballX = ballX + vecX * ballSpeed;
    ballY = ballY + vecY * ballSpeed;
    if (ballX + ballSize > width) {
      vecX *= -1;
    }
    else if (ballX - ballSize < 0) {
      vecX *= -1;
    }
    else if (ballX + ballSize > width) {
      vecX *= -1;
    }
    else if (ballY - ballSize < 0) {
      vecY *= -1;
    }
    else if (ballY + ballSize > height) {
      vecY *= -1;
    }

    fill(ballX, ballY,0);
    textSize(40);
    text("Kevin Henriques",width/2, height/6*1);
    text("Gustavo Cardoso",width/2, height/6*2);
    text("Andrey Bonat",width/2, height/6*4);
    text("Pedro Lyra",width/2, height/6*3);
    text("Alexandre Bueno",width/2, height/6*5);
    break;
  }
}

void movieEvent(Movie m) {
  m.read();
}
//--------------------ANDREY-----------------------
void mousePressed() {
  if (showScreen == 1 && mouseX > playButton.x - playButton.width / 2 && mouseX <
playButton.x + playButton.width / 2 && mouseY > playButton.y - playButton.height / 2 &&
mouseY < playButton.y + playButton.height / 2) {
    showScreen = 2;
    bgm.amp(0.1);
    bgm.loop();
  } else if (showScreen == 1 && mouseX > helpButton.x - helpButton.width / 2 && mouseX <
helpButton.x + helpButton.width / 2 && mouseY > helpButton.y - helpButton.height / 2 &&
mouseY < helpButton.y + helpButton.height / 2) {
    showScreen = 3; // se o mouse estiver sobre o botão na tela 2, atualiza para a tela 3
  } else if (showScreen == 1 && mouseX > exitButton.x - exitButton.width / 2 && mouseX <
exitButton.x + exitButton.width / 2 && mouseY > exitButton.y - exitButton.height / 2 &&
mouseY < exitButton.y + exitButton.height / 2) {
    exit(); // se o mouse estiver sobre o botão na tela 2, atualiza para a tela 3
  } else if (showScreen == 3 && mouseX > backButton.x - backButton.width / 2 && mouseX <
backButton.x + backButton.width / 2 && mouseY > backButton.y - backButton.height / 2 &&
mouseY < backButton.y + backButton.height / 2) {
    ballX = 300;
    ballY = 450;
    vecX = 2;
    vecY = -3;
    ballSpeed = 1;
    showScreen = 1; // se o mouse estiver sobre o botão na tela 2, atualiza para a tela 3
  }
}

class Button {
  float x;
  float y;
  String label;
  float width;
  float height;

  Button(float x, float y, String label) {
    this.x = x;
    this.y = y;
    this.label = label;
    width = 120;
    height = 40;
  }
  void draw() {
    stroke(0);
    strokeWeight(2);
    fill(255);
    rectMode(CENTER);
    rect(x, y, width, height);
    textSize(16);
    fill(0);
    text(label, x, y);
  }

  boolean isMouseOver() {
    return mouseX > x - width / 2 && mouseX < x + width / 2 && mouseY > y - height / 2 &&
mouseY < y + height / 2;
  }

  void hover() {
    fill(150,60,60, 100);
    rect(x, y, width, height);
    fill(255,0,0);
    text(label, x, y);
  }
}
```