バックアップtex

なにんなにんが

2019年7月23日

1　main

```
  import ddf.minim.*;
import ddf.minim.analysis.*;
import ddf.minim.effects.*;
import ddf.minim.signals.*;
import ddf.minim.spi.*;
import ddf.minim.ugens.*;
import javax.swing.*;
import java.awt.*;

Minim minim = new Minim(this);
AudioPlayer[] Answer;
Stairs s;
Quiz q;
Scene o;

JLayeredPane pane;
JTextField field;
JTextArea area;
JScrollPane scrollPane;

int goal = 0, game = 0, timer = 0;
int stairs = 10;
float n = 1; 何段目の階段のクイズか//
float n0 = n; 直前の//n
int scene = 0; シーン切り替え値//
PFont font;
int sec = 10; 制限時間//
int quizLimit = sec;

int trueButton = ○ボタン0;//
int falseButton = 0;//\UTFボタン{2613}
int judge;
int done = 0, isPush = 0; 押せるか押せないか//
int mcnt = 0;

String[] data = nullデータ読み込み;//
String[] questionクイズ問題格納;//
int quesTime = 0;
int[] answer正誤格納;//
int ansTime = 0;クイズ出題用

//
int[] number(int number) {
  IntList nums = new IntList(number);
  for (int i = 0; i < number; i++) {
    nums.append(i);
  };
  nums.shuffle();
  int[] result = nums.array();
  return result;
}
int qCount = 出題済み総数0;//
int[] num;
int t = 0;ここまで
//雲

//
color c1 = color(0, 0, 0);
color c2 = color(0, 255, 255);
PVector nOffset1, nOffset2;
float nScale = 0.007;
float tScale = 0.03;
float step = 5.0;
```

```
int br = 300;
int bx = int(random(300, width - 300));
int by = int(random(300, width - 300));
int bsX = 5;
int bsY = 5;

//revue
String fileName;
String[] item;
int commentCount= 0;


void setup() {
  size(700, 500);
  font = createFont("myFont.vlw", 64);
  textFont(font);
  textAlign(LEFT, TOP);
  o = new Scene();
  s = new Stairs();
  q = new Quiz();データの読み込み

  //
  data = loadStrings("quiz.txt");
  if (data == null) {
    print開発者に問い合わせてください("");
    exit();
  }
  question = new String[data.length / 2];
  answer   = new int[data.length / 2];
  for (int i = 0; i < data.length; i++) {
    if (i % 2 == 0) {
      question[ansTime] = data[i];
      //println(question[quesTime]);
      quesTime++;
    } else {
      answer[ansTime] = int(data[i]);
      if (answer[ansTime] == 0) {
        t++;
      }
      //println(answer[ansTime]);
      ansTime++;
    }
  }ここまで
  //
  num = number(data.length / 2 - 1);雲

  //
  nOffset1 = new PVector(random(10000), random(10000));
  nOffset2 = new PVector(random(10000), random(10000));正誤効果音

  //
  Answer = new AudioPlayer[5];
  Answer[0] = minim.loadFile("BGM/incorrect1.mp3");
  Answer[1] = minim.loadFile("BGM/correct1.mp3");
  Answer[2] = minim.loadFile("BGM/info-girl1-zannen1.mp3");
  Answer[3] = minim.loadFile("BGM/plane-cloud.mp3");
  Answer[4] = minim.loadFile("BGM/wind1.mp3");


  // の親の親にあたるを取得SmoothCanvasJLayeredPane
  Canvas canvas = (Canvas) surface.getNative();
  pane = (JLayeredPane) canvas.getParent().getParent();

    // 複数行のテキストボックスを作成
  area = new JTextArea();
  area.setLineWrap(true);
  area.setWrapStyleWord(true);
  scrollPane = new JScrollPane(area);
  fileName = "data/sample.txt";
  item = new String[10];
}

void draw() {
  o.scene(scene);
  if (scene == 3) {
    time();
    if (n == 0) {
      Answer[2].play();
      Answer[2].rewind();
      scene = 5;
    }
    if (game == 0 && n > 0) {
      s.stairs();
      s.gate();
      s.player();
      if (n > 0 && n <= stairs) {
        goal = 0;
        Answer[4].play();
      } else if (n == stairs + 1) {
        goal = 1;
        Answer[3].play();
      }
    } else if (game == 1) {
      Answer[4].pause();
      Answer[4].rewind();
      judge = -1;
      if (n == 1) {
        judge = q.quiz1(quizLimit);
      } else if (n == 2) {
        judge = q.quiz2(quizLimit);
      } else if (n == 3) {
        if (done == 0) {
          done = 1;
          isPush = 1;
        }
        judge = q.quiz3(quizLimit);
      } else if (n == 4) {
        if (done == 0) {
          done = 1;
          isPush = 1;
        }
        judge = q.quiz4(quizLimit);
      } else if (n == 5) {
        judge = q.quiz5(quizLimit);
      } else if (n == 6) {
        judge = q.quiz6(quizLimit);
      } else if (n == 7) {
        if (done == 0) {
          done = 1;
          isPush = 1;
        }
        judge = q.quiz7(quizLimit);
      } else if (n == 8) {
        if (done == 0) {
          done = 1;
          isPush = 1;
        }
        judge = q.quiz8(quizLimit);
      } else if (n == 9) {
        judge = q.quiz9(quizLimit);
      } else if (n == 10) {
        if (done == 0) {
          done = 1;
          isPush = 1;
        }
        judge = q.quiz10(quizLimit);
      }
      q.gimmick();
      action();
    }
  }
}

void keyPressed() {
  //main command
  /*
  if (keyCode == ENTER) {
   if (scene <= 2) {
   scene ++;
   } else if (scene == 3) {
   if (game == 0 && n <= 10) game = 1;
   } else if (scene == 4) {
   reset();
   n = 1;
   scene = 0;
   } else if (scene == 5) {
   reset();
   n = 1;
   scene = 0;
   }
   }
   */

  if (game == 1) {
    q.isButtonPushed();
  }

  //working command
  if (keyCode == UP) {
    n ++;
    reset();
  } else if (keyCode == DOWN) {
    n --;
    reset();
  } else if (key == 'd') {
    isPush = 0;
  } else if (key == 'D') {
    isPush = 1;
  } else if (key == 'b'){
    scene = 4;
  }
}

void mousePressed() {
  //main command
  if (game == 1) {
    q.isButtonPushed();
    game = 0;
  }
  if (isPush == 1) q.isClear();
  if (scene <= 2) {
    scene ++;
  } else if (scene == 3) {
    if (game == 0 && n <= stairs) game = 1;
  } else if (scene == 4) {
    if (o.push() == 1) {
      scrollPane.setBounds(width / 2 - 100, height / 2 - 50, 200, 100);
      pane.add(scrollPane);
      scene = 6;
    } else {
      reset();
      n = 1;
      scene = 0;
    }
  } else if (scene == 5) {
    reset();
    n = 1;
    scene = 0;
  } else if (scene == 6) {
    scrollPane.setBounds(-100, -100, 10, 10);
    pane.add(scrollPane);
    commentCount++;
    area.setText("");
    reset();
    n = 1;
    scene = 0;
  }
  if (scene == 3 && n == 0) {
    reset();
    n = 1;
    scene = 0;
  }
}

void time() {
  timer ++;
  if (game == 1) {
    if (timer % 60 == 0) quizLimit--;
    if (quizLimit <= 0) {
      qCount++;
      n -= int(random(1, n));
      reset();
      Answer[0].play();
      Answer[0].rewind();
    }
  }
}

void action() {
  int fall = int(random(1, n));
  if (fall > 3) fall = 3;
  if (judge == 0) {
    reset();
    Answer[0].play();
    Answer[0].rewind();
    n -= fall;
  } else if (judge == 1) {
    reset();
    Answer[1].play();
    Answer[1].rewind();
    n ++;
  }
}

void reset() {
  game = 0;
  timer = 0;
  trueButton = ○ボタン0;//
  falseButton = 0;//\UTFボタン{2613}
  done = 0;
  isPush = 0;
  quizLimit = sec;
  mcnt = 0;
  br = 300 - 28*int(n);
  bx = int(random(300, width - 300));
  by = int(random(300, width - 300));
  Answer[3].pause();
  Answer[3].rewind();
}
```

```
void stop() {
  for (int i = 0; i < Answer.length; i++) {
    Answer[i].close();
  }
  minim.stop();
  super.stop();
}
```

2 Quiz

```
  class Quiz {
  int flag = -1;

  void quizTimer(int t) {
    smooth();
    stroke(1);
    fill(255,255,0);
    rect(width - 255, 30, 250,48,30);
    noStroke();
    fill(255);
    rect(width-30,40,-200,25,30);
    fill(255, 0, 255);
    rect(width-30, 40, -t*20, 25,30);
  }

  void button() {
    rectMode(CENTER);
    noStroke();
    flag = -1;土台

    //
    fill(255, 0, 0);
    ellipse(width / 6, height * 4/5, 150, 50);
    rect(width / 6, height * 4/5 - 25/ 2, 150, 25);
    fill(0, 0, 255);
    ellipse(width * 5/6, height * 4/5, 150, 50);
    rect(width * 5/6, height * 4/5 - 25/2, 150, 25);
    fill(250, 255, 0);
    ellipse(width / 6, height * 4/5 - 25, 150, 50);
    ellipse(width * 5/6, height * 4/5 - 25, 150, 50);○ボタン

    //
    if (trueButton == 1) {
      fill(150);
      ellipse(width / 6, height * 4/5 - 25, 75, 25);
      fill(255, 0, 0);
      ellipse(width / 6, height * 4/5 - 25, 65, 15);
      fill(200);
      ellipse(width / 6, height * 4/5 - 25, 60, 10);
      flag = 1;
    } else {
      fill(0);
      ellipse(width / 6, height * 4/5 - 25/2 -5, 75, 25);
      rect(width / 6, height * 4/5 - 30, 75, 25);
      fill(200);
      ellipse(width / 6, height * 4/5 - 43, 75, 25);
      fill(255, 0, 0);
      ellipse(width / 6, height * 4/5 - 43, 65, 15);
      fill(200);
      ellipse(width / 6, height * 4/5 - 43, 60, 10);
    }

    //\UTFボタン{2613}
    if (falseButton == 1) {
      fill(150);
      ellipse(width *5/6, height * 4/5 - 25, 75, 25);
      stroke(0, 0, 255);
      strokeWeight(3);
      line(width *5/6 - 33, height * 4/5 - 30, width *5/6 + 30, height * 4/5 - 30 + 10);
      line(width *5/6 - 33 + 65, height * 4/5 - 30, width *5/6 + 30 - 65, height * 4/5 - 30 + 10);
      flag = 0;
    } else {
      fill(0);
      ellipse(width *5/6, height * 4/5 - 25/2 -5, 75, 25);
      rect(width *5/6, height * 4/5 - 30, 75, 25);
      fill(200);
      ellipse(width *5/6, height * 4/5 - 43, 75, 25);
      stroke(0, 0, 255);
      strokeWeight(3);
      line(width *5/6 - 33, height * 4/5 - 50, width *5/6 + 30, height * 4/5 - 36);
      line(width *5/6 - 33 + 65, height * 4/5-50, width *5/6 + 30 - 65, height * 4/5-36);
    }
    stroke(0);
    strokeWeight(1);
    rectMode(CORNER);
  }
```

```
void isButtonPushed() {
  if (isPush == 0) {
    int w = 75, h = 50;
    int tx = width/6 - w/2, ty = height * 4/5 - 25 - w/2;
    int fx = width *5/6 - h/2, fy = height * 4/5 - 25/2 -5 - h/2;
    if (((mouseX >= tx && mouseX <= tx + w) && (mouseY >= ty && mouseY <= ty + h)) || key == 'o') {
      trueButton = 1;
      falseButton = 0;
    } else if (((mouseX >= fx && mouseX <= fx + w) && (mouseY >= fy && mouseY <= fy + h)) || key == 'x') {
      trueButton = 0;
      falseButton = 1;
    } else {
      trueButton = 0;
      falseButton = 0;
    }
  }
}

void gimmick() {
  if (isPush == 1) {
    fill(255, 200, 150);
    ellipse(bx, by, br, br);
    if (bx < br / 2 || bx > width - br / 2) {
      bsX *= -1;
    }
    if (by < br / 2 || by > height - br / 2) {
      bsY *= -1;
    }
      bx += bsX;
      by += bsY;
    }
}

void isClear(){
  if ((mouseX-bx)*(mouseX-bx) + (mouseY-by)*(mouseY-by) <= br*br) isPush = 0;
}

void quizWindow() {
  smooth();
  background(255);
  noStroke();
  fill(30, 0, 156);
  rect(5, 5, width - 10, height * 4/5-50,10);
  fill(250, 225, 0);
  rect(10, height / 3, width - 20, 50,10);
}

void questionText() {
  int s = question[num[qCount]].length();
  fill(0);
  if (s < 15) {
    textSize(50 - s);
  } else {
    textSize(45 - s);
  }
  text(int(n) + ":" + question[num[qCount]], 30, height/3);
}
int quiz1(int time) {
  quizWindow();
  quizTimer(time);
  questionText();
  button();
  if (flag == answer[num[qCount]]) {
    qCount++;
    return 1;
  } else if (flag != answer[num[qCount]] && flag != -1) {
    qCount++;
    return 0;
  } else {
    return -1;
  }
}
int quiz2(int time) {
  quizWindow();
  quizTimer(time);
  questionText();
  button();

  if (flag == answer[num[qCount]]) {
    qCount++;
    return 1;
  } else if (flag != answer[num[qCount]] && flag != -1) {
    qCount++;
    return 0;
  } else {
    return -1;
  }
}
int quiz3(int time) {
  quizWindow();
  quizTimer(time);
  questionText();
  button();

  if (flag == answer[num[qCount]]) {
    qCount++;
    return 1;
  } else if (flag != answer[num[qCount]] && flag != -1) {
    qCount++;
    return 0;
  } else {
    return -1;
  }
}
int quiz4(int time) {
  quizWindow();
  quizTimer(time);
  questionText();
  button();

  if (flag == answer[num[qCount]]) {
    qCount++;
    return 1;
  } else if (flag != answer[num[qCount]] && flag != -1) {
    qCount++;
    return 0;
  } else {
    return -1;
  }
}
int quiz5(int time) {
  String [] back = question[num[qCount]].split("");
  quizWindow();
  quizTimer(time);
  int s = question[num[qCount]].length();
  fill(0);
  if (s < 15) {
    textSize(50 - s);
  } else {
    textSize(45 - s);
  }
  text(5, 30, height/3);
  for (int i = back.length-1; i  >= 0; i--) {
    text(back[i], 30 + (back.length - i)*20, height/3);
  }
  button();

  if (flag == answer[num[qCount]]) {
    qCount++;
    return 1;
  } else if (flag != answer[num[qCount]] && flag != -1) {
    qCount++;
    return 0;
  } else {
    return -1;
  }
}
int quiz6(int time) {
  quizWindow();
  quizTimer(time);
  questionText();
  button();

  if (flag == answer[num[qCount]]) {
    qCount++;
    return 1;
  } else if (flag != answer[num[qCount]] && flag != -1) {
    qCount++;
    return 0;
  } else {
    return -1;
  }
}
int quiz7(int time) {
  quizWindow();
  quizTimer(time);
  questionText();
  button();

  if (flag == answer[num[qCount]]) {
    qCount++;
    return 1;
  } else if (flag != answer[num[qCount]] && flag != -1) {
    qCount++;
    return 0;
  } else {
    return -1;
  }
}
int quiz8(int time) {
    quizWindow();
    quizTimer(time);
    questionText();
    button();

    if (flag == answer[num[qCount]]) {
      qCount++;
      return 1;
    } else if (flag != answer[num[qCount]] && flag != -1) {
      qCount++;
      return 0;
    } else {
      return -1;
    }
  }
  int quiz9(int time) {
    quizWindow();
    quizTimer(time);
    questionText();
    button();

    if (flag == answer[num[qCount]]) {
      qCount++;
      return 1;
    } else if (flag != answer[num[qCount]] && flag != -1) {
      qCount++;
      return 0;
    } else {
      return -1;
    }
  }
  int quiz10(int time) {
    quizWindow();
    quizTimer(time);
    questionText();
    button();

    if (flag == answer[num[qCount]]) {
      qCount++;
      return 1;
    } else if (flag != answer[num[qCount]] && flag != -1) {
      qCount++;
      return 0;
    } else {
      return -1;
    }
  }
}
```

3 Scene

```
  class Scene {
  String data[] = null;
  PImage img;
  int move;
  PImage backgroundOcean = loadImage("image/background_blue_ocean.png");
  PImage nightSea                  = loadImage("image/nightSea.jpg");
  int revueX  = width * 1/2;
  int revueY  = height - 160;
  int revueW = 120;
  int revueH  = 40;


  void scene(int s) {
    data = loadStrings("ending.txt");
    background(255);
    if (s == 0) {
      move = 0;
      mcnt++;
      textSize(90);
      img = loadImage("image/sora.jpeg");
      image(img, 0, 0, width, height);

      fill(0, 0, 255);
      textクイズの都へ("", 93, 133);
      fill(255, 0, 0);
      textクイズの都へ("", 90, 130);
      textSize(35);
      fill(0);
      if ((mcnt % 60) < 30) {
        text("Mouse Click", 240, 360);
      }
    }
    if (s == 1) {
      mcnt++;
      fill(0);
      textSize(30);
      img = loadImage("data/image/kanban.jpg");
      image(img, 0, 0, width, height);
      textクイズの都・タンブリンに入るには("", 85, 60);
      text(" 『クイズの階段』の試練に挑まねばならない。", 25, 150);
      text(" 間違えれば階段から転げ落ちるだろう・・・。 ", 30, 240);
      text(" 見事クイズに正解し、階段を駆け上がれ！！ ", 40, 330);
      if ((mcnt % 60) < 30) {
        text("Click で次へ", 270, 420);
      }
    } else if (s == 2) {
      mcnt++;
      tint(255, 255, 255, 165);
      img = loadImage("image/sora.jpeg");
      image(img, 0, 0, width, height);
      tint(255, 255, 255, 255);
      q.button();
      fill(0);
      textSize(40);
      text操作説明("", width/2 - 80, 40);
      textSize(30);
      text("をすると問題が出題されます。Click", 110, 110);
      text下のような○("ボタンがでますので、X", width/2 - 230, 150);
      text正解と思う方をクリックしてください。("", width/2 - 250, 190);
      text正解すると階段を登り、("", 200, 230);
      text間違えると階段を滑り落ちていきます。("", 100, 270);
      if ((mcnt % 60) < 40) {
        text("でClick", width/2-100, 450);
        fill(255, 0, 0);
        textSize(50);
        textスタート("", width/2, 430);
      }
    } else if (s == 3) {
      //game playing
    } else if (s == 4) {
      fill(0);
      textSize(30);

      int x = 150;
      int y = height + move;
      image(backgroundOcean, 0, 0, width, height);
      for (int i = 0; i < data.length-1; i++) {

        text(data[i], x, y);
        y += 30;
      }
      textSize(50);
      text(data[data.length-1], 0, y);
      if (y < height/2) {
        textSize(30);
        text("Click to Restart", width*3/4-55, height-30);
        fill(0, 255, 255);
        rectMode(CENTER);
        rect(revueX, revueY, revueW, revueH,2);
        rectMode(CORNER);
        fill(0);
        textレビュー("", revueX - revueW / 2, revueY - revueH / 2);
      } else {
        move -= 30;
      }
    } else if (s == 5) {
      mcnt++;
      fill(255, 100, 50);
      img = loadImage("image/haka.png");
      image(img, 0, 0, width/2, height);
      textSize(80);
      text残念！！("", width/2, 30);
      textSize(30);
      textあなたは道半ばで("", width/2, height/2 - 30);
      text息絶えてしまったようだ・・・("", width/2-60, height/2);
      if ((mcnt % 60) < 20) {
        text("Click to Retry", width*3/4 - 100, height-35);
      }
    } else if (s == 6) {
      image(nightSea, 0, 0, width, height);
      println(area.getText() );
      item[commentCount] = area.getText();
      saveStrings( fileName, item);
      fill(255);
      rectMode(CENTER);
      rect(revueX , height / 2 + 80, 100, 50,5);
      fill(0);
      text送信("", revueX - 30, height / 2 + 65);
      rectMode(CORNER);
    }
  }
  int push() {
    if (revueX - revueW / 2 <= mouseX&& revueX + revueW / 2 >= mouseX&& mouseY >= revueY - revueH / 2 && mouseY <= revueY + revue
      return 1;
    } else {
      return 0;
    }
  }
}
```

4 Stairs

```
  class Stairs {
  int stages = stairs;
  float shrinkX = (width/3)/stages;
  float shrinkY = (height*3/4)/stages;

  float len = width/3/4;
  float open = width/2 - len - len/2;
  float close = width/2 - len;
  float gateX1 = width/2 - len;
  float gateX2 = width/2;
  float gateY = height/8;

  float pY = height-shrinkY/2;
  float playerX = width/2, playerY1 = pY-shrinkY*(n0-1), playerY2;
  int diff = int(shrinkY*3/2); 画像をキャラに使用しているために生じるズレ//
  float top = pY-shrinkY*(stages);

  PImage avatar = loadImage("image/character2.png");
  float sizeX = 50, sizeY = 80;
  PImage backgroundOcean = loadImage("image/background_blue_ocean.png");
  PImage city            = loadImage("image/city.png");

  void cloud() {
    background(255);
    noStroke();
    float time = frameCount * tScale;
    for (int w = 0; w < width; w += step) {
      for (int h = 0; h < height; h += step) {
        float n = noise(w * nScale + nOffset1.x + time, h * nScale + nOffset1.y - time, time);
        float r = map(n, 0, 1, red(c1), red(c2));
        float g = map(n, 0, 1, green(c1), green(c2));
        float b = map(n, 0, 1, blue(c1), blue(c2));
        fill(r, g, b);
        rect(w, h, step, step);
        fill(255, 255, 255, map(abs(n - 0.5) + 0.5, 0, 1, 255, 0));
        rect(w, h, step, step);
      }
    }
  }

  void backDisplay() {
    cloud();
    tint(255, 255, 255, 165);
    image(backgroundOcean, 0, 0, width, height);
    image(city, 0, 0, width, 150);
    tint(255, 255, 255, 255);
    stroke(0);
  }

  void stairs() {
    backDisplay();
    float rectX = width/3;
    float rectY = height-shrinkY*stages;
    float rectLen = width/3;

    fill(150);
    for (int i = 0; i < stages; i++) {
      rect(rectX - shrinkX*i, rectY+shrinkY*i, rectLen + shrinkX*(i*2), rectY);
    }
  }

  void gate() {
    if (goal == 1) {
      if (gateX1 > open) {
        if (playerY1 <= top) {
          gateX1 --;
          gateX2 ++;
          sizeX --;
          sizeY --;
        }
      } else {
        scene = 4;リスタート時のエラーをなくす

        //
        gateX1 ++;
        gateX2 --;
        sizeX = 50;
        sizeY = 80;
        playerY1 = pY;
      }
    } else if (goal == 0) {
      if (gateX1 < close) {
        gateX1 ++;
        gateX2 --;
      }
    }
    fill(100);
    rect(width/3, height/8, width/3, height/7);
    fill(255, 255, 200);
    rect(width/2 - len/2, gateY, len*3/2, height/7);
    fill(#C1191F);
    rect(gateX1, gateY, len, height/7);
    rect(gateX2, gateY, len, height/7);
  }

  void player() {
    playerY2 = pY-shrinkY*(n-1) - diff;
    textSize(20);
    if (n <= stairs) {
      fill(255);
      rect(width/6 -5, height/4, 75, 20 +2);
      fill(255, 150, 200);
      text(int(n) + 段目"", width/6, height/4);
    } else if (n == stairs + 1) {
      fill(255);
      rect(width/3 -5, 30, 210, 20 +2);
      fill(255, 0, 0);
      textクイズの都へようこそ("", width/3, 30);
    }
    if (playerY1 < playerY2) {
      playerY1 += 7;
    } else if (playerY1 > playerY2) {
      if (playerY1 > top) playerY1 -= 4;
    }
    imageMode(CENTER);
    image(avatar, playerX, playerY1, sizeX, sizeY);
    imageMode(CORNER);
  }
}
```