processingによる
タイピングゲームの作成

プログラミング基礎&演習I

情報工学科　22-442　山口　遼

ゲームの説明・特色

表示される文字を正しくタイピングしていき、より高いスコアを目指すゲームです。

-時間制限-

指定した時間内にタイピングをし、スコアを算出する。

-スコア-

正タイプ数と誤タイプ数の差がスコアとなる。

-効果音-

正解したときと誤タイプしたときに効果音がなる。

動作の流れ

タイトル画面

・スタートボタンを押してゲームをはじめる

・ボタンを押してる間だけボタンが灰色になり、わかりやすい動作になるように工夫した

動作の流れ

ゲーム画面

・表示された文字をタイピングしていく

・左上にTitleボタンが表示されていて、押すとタイトルに戻ることができる

（スコアは初期化される）

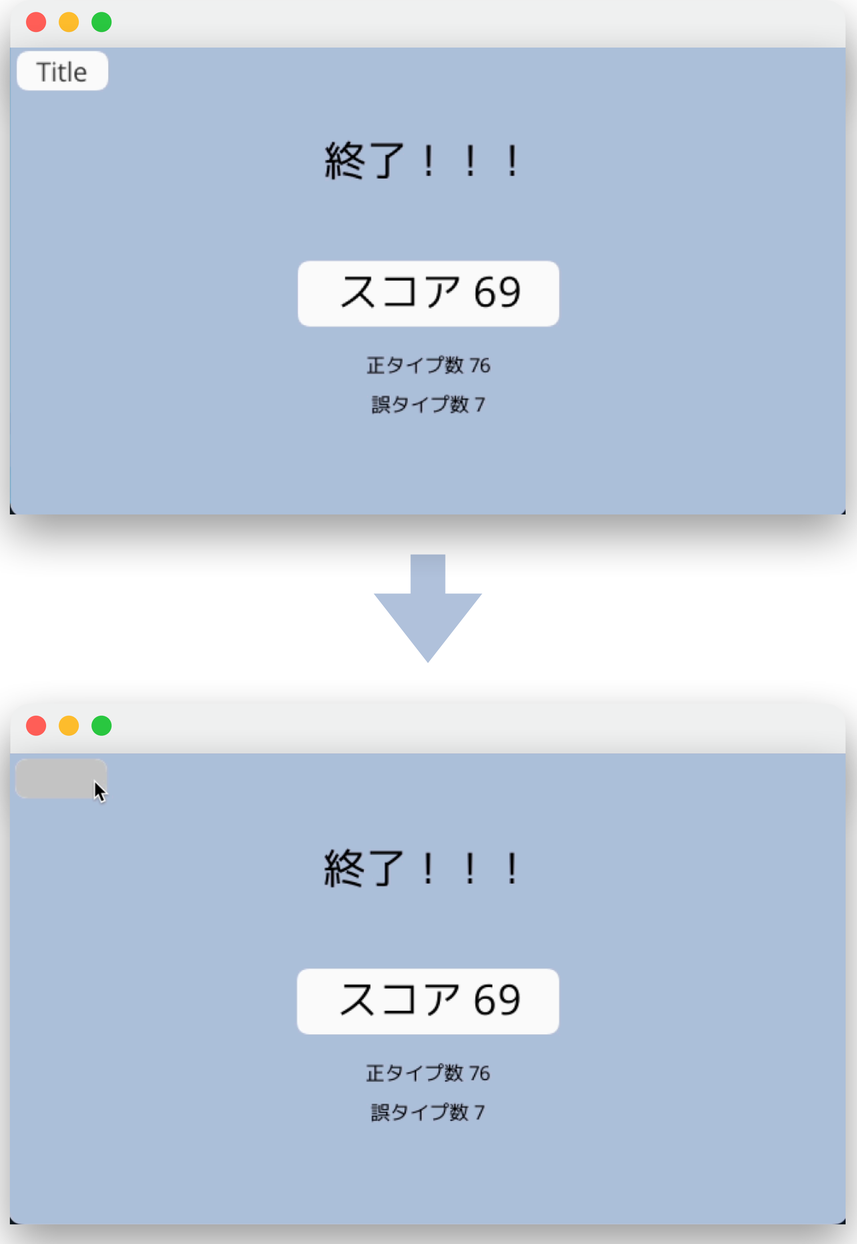
・右上に徐々に短くなるタイマーが表示されていて、なくなったらゲームが終了する

動作の流れ

リザルト画面

・スコア、正タイプ数、誤タイプ数が表示される

・ゲーム画面と同様に、Titleボタンを押すとタイトル画面に戻れる

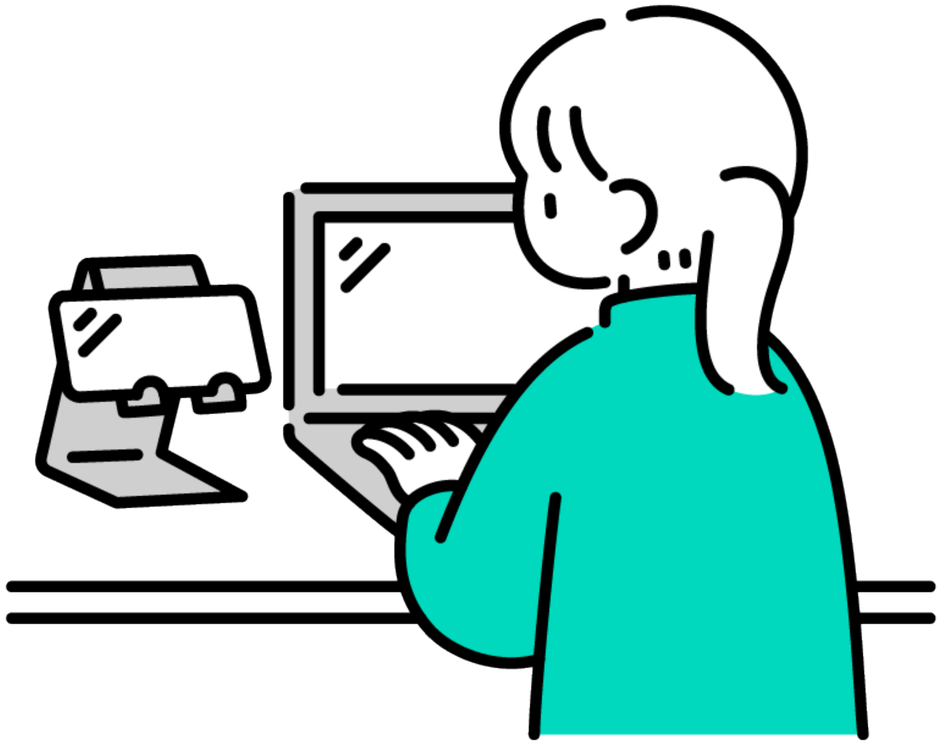


工夫した点

◯登録した語句が一周するまで重複しないようにした

◯入力した文字が誤っているとき、効果音だけがなり入力されないようにした

◯クラスを用いてボタンを作成した

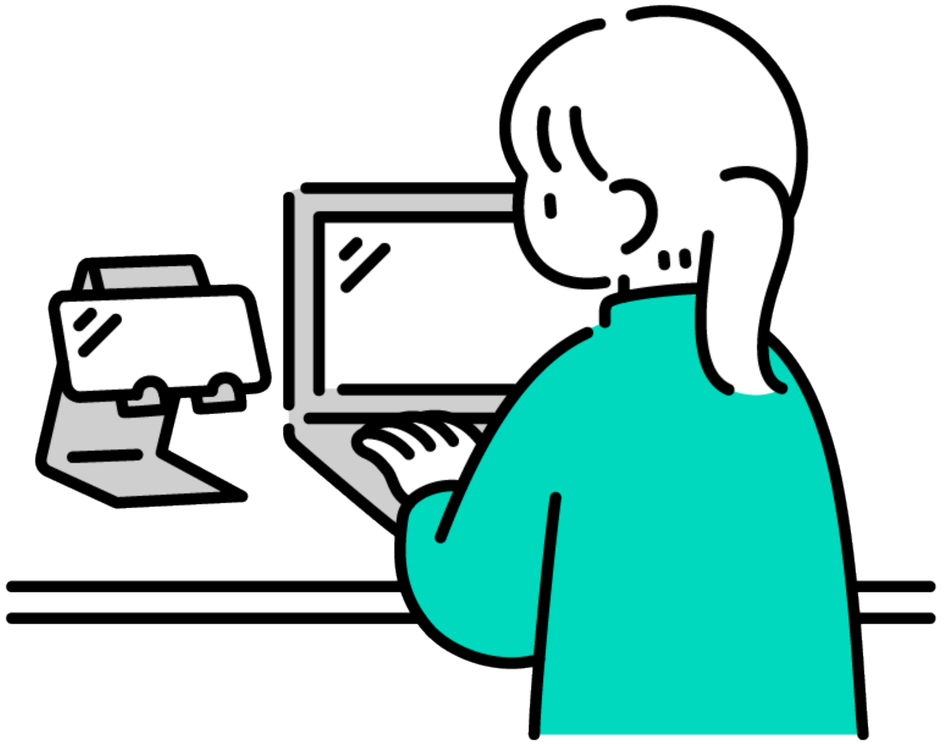


今後に向けた改善点

◯様々なゲームモードを追加する

◯より見やすいUIにする

◯時間、文字列をプログラム実行中に設定できるようにする



```
import ddf.minim.*;

Minim minim;//minimクラスのインスタンスを生成
//AudioPlayerクラスのインスタンスを生成
AudioPlayer correctingPlay;
AudioPlayer missingPlay;

void setup(){
  size(640,360);

  //音声関連
  minim = new Minim(this);
  correctingPlay = minim.loadFile("correcting.mp3");
  missingPlay = minim.loadFile("missing.mp3");

  //フォントの設定
  KoruriRegular = createFont("Koruri-Regular.ttf",32);
  MANIFESTO = createFont("MANIFESTO.ttf",32);
  OxygenMono = createFont("OxygenMono-Regular.otf",32);

  //各設定
  textAlign(CENTER);
  rectMode(CENTER);
  imageMode(CENTER);

  //画像の読み込み
  titleLogo = loadImage("Typing_Game_Logo.png");

  if(language == 0){
    japanese_strings();
  }
}

void draw(){
  background(180,192,217);//背景を設定

  //場面による動き
  if(gamemode == 0){
    title_scene();
  }else if(gamemode == 1){
    typing_scene();
  }else if(gamemode == 2){
    result_scene();
  }
}

```

```
47 void keyTyped(){
48   if(gamemode == 0){
49
50   }else if(gamemode == 1){
51     typing_key();
52   }else if(gamemode == 2){
53
54   }
55 }
56
57 void mousePressed(){
58   if(gamemode == 0 || gamemode == 1 ||gamemode == 2){
59     isMousePressed = true;
60   }
61 }
62
63 void mouseReleased(){
64   if(gamemode == 0){
65     if(normalModeB.isMouseOver()){//NormalModeが押されたとき
66     gamemode = 1;
67     startTime = millis();
68     timerStarted = true;
69
70     //正誤タイプ数初期化
71     trueTypeNum = 0;
72     falseTypeNum = 0;
73
74     //input初期化
75     input = "";
76     countW = 0;
77     }
78
79     isMousePressed = false;//長押しされてるかの判断初期化
80   }else if(gamemode == 1 ||gamemode == 2){
81     if(returnTitleB.isMouseOver()){//returnTitleが押されたとき
82       gamemode = 0;
83     }
84     isMousePressed = false;
85   }
86 }
87
88 //プログラムが終了する際に音声再生関連のリソースを正しく解放する
89 void stop() {
90   missingPlay.close();
91   minim.stop();
92   super.stop();
```

```
 93 }
 94
 95 class Button{
 96   float x,y,w,h;
 97   String dispText;
 98
 99   Button(float x,float y,float w,float h,String dispText){
100     this.x = x;
101     this.y = y;
102     this.w = w;
103     this.h = h;
104     this.dispText = dispText;
105   }
106
107   void display(){
108     fill(256,256,256);
109     noStroke();
110     rect(x,y,w,h,10);
111     textSize(20);
112     fill(50);
113     text(dispText,x,y+8);
114   }
115
116   boolean isMouseOver(){
117     return mouseX > x - w/2 && mouseX < x + w/2 && mouseY > y - h/2 && mouseY
118   }
119
120   void pressingMouse(){
121     if(isMouseOver()){
122       fill(200,200,200);
123       noStroke();
124       rect(x,y,w,h,10);
125     }
126   }
127 }
128
129 void japanese_strings(){
130   //stringsに代入
131   strings = new String[japanese.length][japanese[0].length];
132   for (int i = 0; i < japanese.length; i++) {
133     for (int j = 0; j < japanese[i].length; j++) {
134       strings[i][j] = japanese[i][j];
135     }
136   }
137 }
138
```

```
void title_scene(){
  image(titleLogo,widthS/2,100,846*0.3,350*0.3);

  textFont(MANIFESTO);

  normalModeB = new Button(widthS / 2, heightS / 2 + 50, 200, 50, "Start");
  normalModeB.display();

  if(isMousePressed){
    normalModeB.pressingMouse();
   }

   hasRun = false;//初期化
}

void typing_scene(){
  //タイトル表示
  image(titleLogo,widthS/2,60,846*0.2,350*0.2);

  //ランダムに問題を選ぶ
  while(flag == 1){
    while(flag == 1){
      flag = 0;
      randomQ = int(random(stringNum));
      for(int i=0;i<d;i++){
        if(randomQ == dupQ[i]){
          flag = 1;
        }
      }
    }
    dupQ[d] = randomQ;//重複の記録
    //dの処理（何周目か）
    if(d == stringNum - 1){
      d = 0;
      dupQ = new int[stringNum];
    }else{
      d++;
    }
  }

  //問題表示
  fill(50);
  textFont(KoruriRegular);
  question = strings[0][randomQ];
  textSize(50);
  text(question,widthS/2,170);
```

```
185  //ローマ字表示
186  textFont(OxygenMono);
187  roman = strings[1][randomQ];
188  textSize(20);
189  text(roman,widthS/2,200);
190  fill(0);
191  //入力表示
192  fill(256,256,256);
193  noStroke();
194  rect(widthS/2,260,400,40,28);
195  fill(0);
196  textSize(30);
197  text(input,widthS/2,270);
198
199  //正解したときの処理
200  if(input.equals(strings[1][randomQ])){
201    flag = 1;
202    input = "";
203    countW = 0;
204    //正解の音声を再生
205    correctingPlay.close();//前回の再生をclose
206    correctingPlay = minim.loadFile("correcting.mp3");//ロードのし直し
207    correctingPlay.rewind();//再生位置を最初に戻す
208    correctingPlay.play();//音声を再生
209  }
210
211  //タイマー処理
212  if(timerStarted){
213    int elapsedTime = millis() - startTime;
214    if(elapsedTime > interval){
215      gamemode++;
216      timerStarted = false;
217    }
218    //タイマー表示
219    fill(48,191,72);
220    noStroke();
221    float remain = float(interval - elapsedTime)/interval;
222    rect(550,20,150*remain,15,10);
223  }
224
225  //title画面に戻るボタン
226  returnTitleB = new Button(40,20,70,30,"Title");
227  returnTitleB.display();
228  if(isMousePressed){
229      returnTitleB.pressingMouse();
230  }
```

```
231 }
232
233 void result_scene(){
234   //各テキストの表示
235   textSize(20);
236   textFont(KoruriRegular);
237   fill(0);
238   text("終了！！！",widthS/2,100);
239   typingScore = trueTypeNum - falseTypeNum;
240   fill(256,256,256);
241   rect(width/2,190,200,50,10);
242   fill(0);
243   text("スコア "+typingScore,widthS/2,200);
244   textSize(15);
245   text("正タイプ数 "+trueTypeNum,widthS/2,250);
246   text("誤タイプ数 "+falseTypeNum,widthS/2,280);
247
248   //ランキングデータの保存
249   /*if(!hasRun){
250     currentRanking = 0;
251     for(int i = 0;rankingData[i] != "\0";i++){
252       currentRanking++;
253     }
254     rankingData[currentRanking] = str(typingScore);
255     rankingData[currentRanking + 1] = "\0";
256     saveStrings("ranking_data.txt", rankingData);
257     hasRun = true;
258   }*/
259
260   //title画面に戻るボタン
261   returnTitleB = new Button(40,20,70,30,"Title");
262   returnTitleB.display();
263   if(isMousePressed){
264       returnTitleB.pressingMouse();
265   }
266 }
267
268 void typing_key(){
269   char inputW = strings[1][randomQ].charAt(countW);
270   if(key == inputW){
271     input = input + key;
272     countW++;
273     trueTypeNum++;//正タイプ数加算
274   }else{
275     //不正解の音を再生
276     missingPlay.close();//前回の再生をclose
```

```
277     missingPlay = minim.loadFile("missing.mp3");//ロードのし直し
278     missingPlay.rewind();//再生位置を最初に戻す
279     missingPlay.play();//音声を再生
280
281     falseTypeNum++;//誤タイプ数加算
282   }
283 }
284
285 String[][] japanese =
286
287 //日本語
288 {{
289 哀悼,
290 斡旋,
291 亜流,
292 行脚,
293 安寧,
294 安穏,
295 },
296
297 //ローマ字
298 {
299 aitou,
300 assenn,
301 aryuu,
302 anngya,
303 annnei,
304 annnonn,
305 }};
306
307 int stringNum = 6;//問題数
308
309 //ints
310 int widthS = 640,heightS = 360;//縦と横のsize(size()は手動で変える必要あり)
311 int gamemode = 0;//ゲームモード判定（0:home,1:typing_scene）
312 int language = 0;//言語モード判定
313 int randomQ;//問題をランダムにする変数
314 int[] dupQ = new int[stringNum];//重複をカウントする配列
315 int d = 0;
316 int flag = 1;//解答の判定
317 int countW = 0;//inputの文字数カウント
318 int startTime;//タイマーがスタートした時間を入れておく変数
319 int interval = 20 * 1000;//制限時間（ミリ秒）
320 int typingScore,trueTypeNum,falseTypeNum;//合計スコア（正タイプ-誤タイプ
321 int currentRanking;//現在のランキングデータの記録数
322 //Strings
```

```
String question = "",roman = "";//問題の語とローマ字の表示用代数
String input = "";//入力されている文字
String[][] strings;//stringDBを代入する変数
//String[] rankingData = loadStrings("ranking_data.txt");//ランキングファイルの
//PImages
PImage titleLogo;
//PFont
PFont KoruriRegular;
PFont MANIFESTO;
PFont OxygenMono;
//booleans
boolean isMousePressed = false;//マウスが押されているかの判定
boolean timerStarted = false; // タイマーが開始されたかどうかのフラグ
boolean hasRun = false;//loop内で一度だけ動かしたいときのフラグ
//classes
Button normalModeB;//タイトル画面のボタンnormalMode
Button returnTitleB;//タイトル画面に戻るボタン
```