

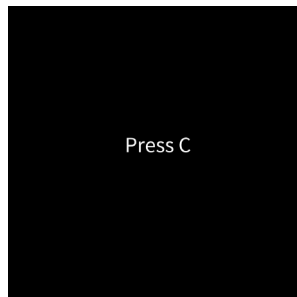
改良・開発  
アプリ名

# 東〇風シューティングゲーム

学籍番号:2022531033  
氏 名: 関川 謙人

コンセプト 東方リスペクト!!弾幕をよけて敵をぶっ倒せ。全4ステージ。

## アプリの概要



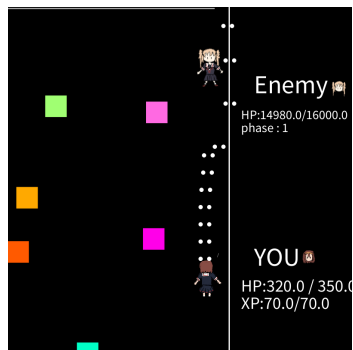
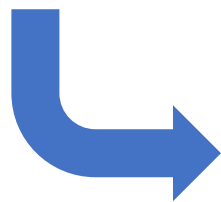
Cを押して  
スタート

### 操作方法

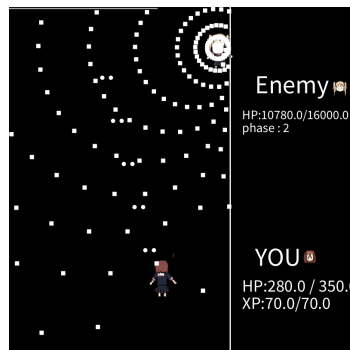
Cキー:スタート、コンティニュー  
Zキー:通常弾発射  
Xキー:ボム  
十字キー:プレイヤー操作  
Shiftキー:低速移動  
Eキー:終了(コンティニュー画面)

## 製作秘話

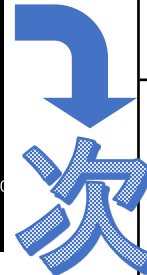
最初にいいアイデアが思い浮かばなかったため東方にしたのだが、意外と始めてみたらいいものが作れたので感動している。制作もそうだが、友達とTwitterスペースで会話しながら家で作業をしていたことが一番楽しかった。ゲームを作ってみて、あれだけのゲームを一人で作って見せたZUN氏は凄い人だと改めて思わされた。



第1フェーズでは  
色とりどりの四角形が  
落ちてくるぞ!!



第2フェーズは円状に  
弾が飛んでくるぞ。軌  
道も変わるから難しい!!

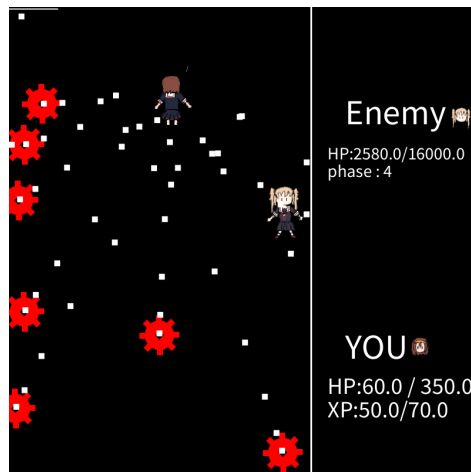
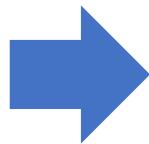
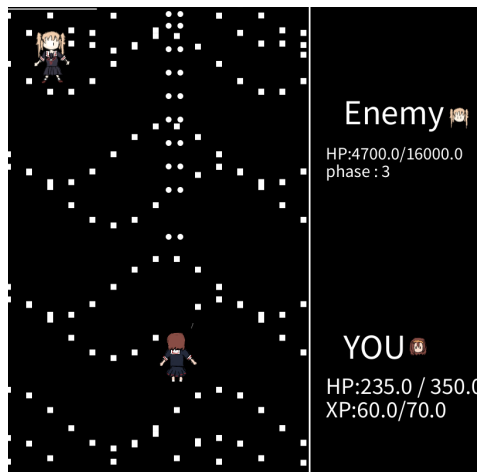


次

## 販売価格とその根拠

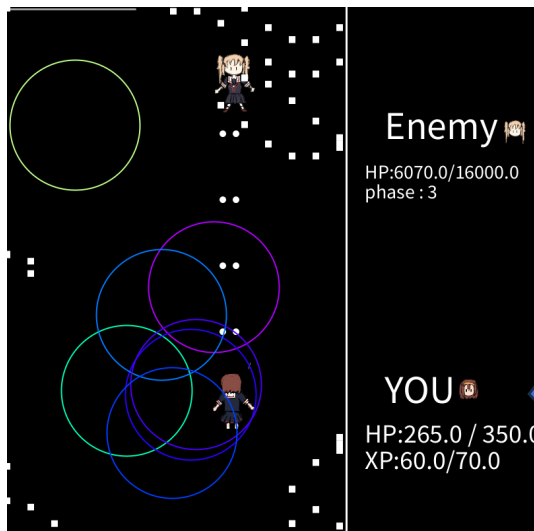
0円

東方リスペクトであることと、艦これキャラを参考にして自機、敵機のイラストを作ったことから。商用利用はさすがにまずいので。



最終フェーズ。  
歯車がプレイヤーに向かって弾を発射してくるぞ！  
クリアできた  
らすごい！

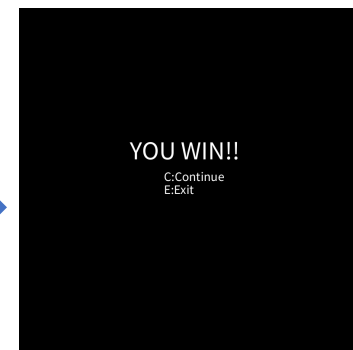
第3フェーズでは  
上下からSin,Cos型の  
弾幕が出てくるぞ!!



Xキーを押すと  
超強力な必殺技  
が出せるぞ！敵  
弾が消えると同  
時に敵にダメー  
ジを与えよう！



ゲームオーバー画面



クリア画面

# アプリの仕様

## 動作条件

- ・このアプリを実行するには以下のファイルをプログラムと同じフォルダに入れておく必要がある

shiratuyu\_dot.png  
murasame.png  
shiratuyu\_head.png  
Murasame\_head.png

## Classの使い方

- ・自機、敵機、弾を定義するためにクラスを利用した。
- ・自機のクラスについて

float型でプレイヤーの位置、移動スピード、HP、最大HP、XP、最大XP（XP：技を出すためのポイント）、bool形でプレイヤーの進行方向、動作の可否のを定義している。また、display関数でプレイヤーを表示、movement関数でプレイヤーを動かしているという作りである。

- ・敵機のクラスについて

```
class Enemy{
//フィールドの宣言
float x;
float y;
float xsize;
float ysize;
float xmove = Game_width / 2;
float ymove = height / 4;
float HP;
float Max_HP;
boolean is_boss;
Enemy(float x, float y, float xsize, float ysize,
float HP, float Max_HP, boolean is_boss) {
    this.x = x;
    this.y = y;
    this.xsize = xsize;
    this.ysize = ysize;
    this.HP = HP;
    this.Max_HP = Max_HP;
    this.is_boss = is_boss;
}

void display() {
    noStroke();
    image(murasame, x - 50, y - 45);
}

void enemy_move() {
    if (phase == 1 && frameCount % 1000 == 0) {
        xmove = random(512);
        ymove = random(200);
    }
    else if (phase == 2 && frameCount % 500 == 0) {
        xmove = random(512);
        ymove = random(200);
    }
}
```

- ・敵機のクラス定義
  - ・左の図はEnemyクラス定義箇所の一部。
  - ・基本仕様はプレイヤーと似通っているが、敵がランダムに位置を変えるようにしている。ウィンドウの上に敵のHPゲージを表示するように設定。gear変数は最終フェーズで出てくる歯車の動きを定義している。
- ちなみに歯車であるが、画像を読み込んでおらず、すべてellipseとrectで表現している。回転しながら弾を発射する拳動を再現するのはなかなか大変であった。

```
class Player{
//フィールドの宣言
//プレイヤー表示
//プレイヤーの座標。
float x;
float y;
//キネキーが押されているかの判定
boolean is_up;
boolean is_down;
boolean is_right;
boolean is_left;
//-----
boolean movable = true;
float speed; //プレイヤーの移動スピードを定義。
float HP;
float Max_HP;
float XP;
float Max_XP;
//コンストラクタを宣言
Player(float position_x, float position_y, boolean up, boolean down, boolean right, boolean left,
float speed_arg, float health, float Max_HP, float XP, float Max_XP) {
    x = position_x;
    y = position_y;
    is_up = up;
    is_down = down;
    is_right = right;
    is_left = left;
    speed = speed_arg;
    HP = health;
    this.Max_HP = Max_HP;
    this.XP = XP;
    this.Max_XP = Max_XP;
}
```

Playerクラス定義箇所の一部。長くなるため以降は省略する。

```

class Weapon{
  //フィールドの宣言
  float x;
  float y;
  float speed;
  float atk;
  float XP;
  float vecX;
  float vecY;
  float r = 0; //r : 半径
  float biase = 0;
  //Weaponクラスのコンストラクタ
  Weapon(float xpos ,float ypos,float sped, float attack,
  float XP,float vecX,float vecY) {
    x = xpos;
    y = ypos;
    speed = sped;
    atk = attack;
    this.XP = XP;
    this.vecX = cos(radians(vecX));
    this.vecY = sin(radians(vecY));
  }
  // 使用関数宣言
  // 球を動かす
  void movebullet() {
    fill(0,0,400);
    noStroke();
    y -= speed;
    ellipse(x,y,10,10);
  }
  //四角形を落とす
  void rect_fall() {
    noStroke();
    fill(x,y,400);
    y += speed;
    rect(x,y,50,50);
  }
  //円状に弾を発射する
  void cbullet() {
    noStroke();
    fill(0,0,400);
    x = xpos + r * cos(radians(vecX));
    y = ypos + r * sin(radians(vecY));
  }
}

```

- ・弾のクラス定義
- ・左の図Weapon（弾）クラス定義箇所の一部。
- ・floatで弾の位置、スピード、消費XP、攻撃力、進行方向を定義している。
- ・このクラスとArrayListを利用して弾の管理を行った。
- ・movebullet()は自機弾、rect\_fall()は第一フェーズでの敵弾、cbullet()は第二フェーズでの敵弾、control\_tetra()で第三フェーズでの敵弾を定義している。

### 既存の製品との対応

- ・アイデア
  - 東方を参考にしているため完全オリジナルではない。
  - しかし、敵の弾幕、ボムの仕様に関してはすべて自分で発想したうえで実装したため、そこは完全オリジナル。

### ・開発

- 本アプリはすべて1人で開発した
- 開発の際に参考にしたのはProcessingのreference [1]、Google検索[2]である。

## 付録

アプリケーションのプログラムの  
重要箇所（関数や変数の役割，アルゴリズム，複雑な条件設定など）が  
わかるような資料があると良い

- ・当たり判定はプレイヤー側に合わせて範囲指定を用いて実装した。
  - ・プレイヤーを狙う弾、あるいは円状の攻撃弾などは最初に要素を追加する際（要素名.add<Weapon>()）の際に最初から方向を指し示している。
  - ・この開発で初めて利用した関数、キーワードは、dist,ArrayList,atan2,random,keycodeである
  - ・また、弾の数はすべてint型のNumOf（弾の名前）という形の変数で管理をしている。
  - ・弾の当たり判定やダメージなどの挙動についてはすべてcontrol\_オブジェクト名()という形でグローバル関数として定義している。
  - ・フレーム数に応じた動作の実装のため、frameCount変数を利用している。
  - ・status()関数で横のステータス画面を表示している
  - ・judge()で勝敗の判定、judge\_phase()関数で敵の残りHPを利用したフェーズ管理を行っている。
  - ・continue\_judge()でゲーム終了画面でのキー入力を受け取り、ゲーム終了、または続行をつかさどっている。
- 続行の場合、初期化を行うのもこの関数である。