|  |  |
| --- | --- |
| Symphogear (1101行) | GUNGNEAR (547行) |
| Game g;  Ship ship;  boolean tubasa = false;  boolean kurisu = false;  boolean Burst = false;  int qw;  float qwe;  float light;  int bosshp, a, beftime, nowtime, accumulate;  void **setup**() {    frameRate(1);    size(800, 400);    background(255);    smooth();    imageMode(CENTER);    textSize(30);    textAlign(CENTER);    qw = -1;  }  void **draw**() {    if (key == &apos;a&apos;) {      tubasa = false;      kurisu = false;    }    else if (key == &apos;s&apos;) {      tubasa = true;      kurisu = false;    }    else if (key == &apos;d&apos;) {      kurisu = true;      tubasa = false;    }    if (qw == -1) {      background(255);      qwe = 0;      light = 0;      bosshp = 100000000;      g = new Game();      qw = 0;    }    else if (qw == 0) {      textAlign(CENTER);      fill(255, 0, 255);      if (Burst == false) {        a=130;      }      else if (Burst == true) {        a=80;        textSize(60);        text("Extra Mode", width/2, 150);      }      textSize(80);      text("Symphogear", width/2, a);      textSize(30);      text("Press Enter Key", width/2, 350);      text("a = Hibiki, s = Tubasa, d = Kurisu", width/2, 250);      text("z = Shot", width/2, 200);      text("Ready?", width/2, 320);        accumulate = 0;      if (keyCode == ENTER) {        beftime = millis();        qw = 1;      }    }    else if (qw == 1 || qw == 5 || qw == 6) {      nowtime = millis();      accumulate += nowtime - beftime;      beftime = nowtime;      g.update();    }    else if (qw == 4) {      fill(255, 0, 0);      textSize(90);      text("Warning", width/2, height/2);     textSize(30);      if (keyCode == ENTER)qw = 5;    }    else if (qw == 2) {      fill(255, 255, 0);      background(0);      text("Clear", width/2, height/2+40);      text("Point "+g.P, width/2, height/2);      text("Shift = Restart", width/2, height/2 +150);       text(nf((accumulate/60000)%60, 2) + ";" + nf((accumulate/1000)%60, 2) + ";" + nf((accumulate/10)%100, 2), width/2, height/2+80);      if (keyCode == SHIFT) qw = -1;    }    else if (qw == 3) {      background(50, 50, 0);      text("Game Over", width/2, height/2+40);      text("Point"+g.P, width/2, height/2);      text("Shift = Restart", width/2, height/2 +150);       text(nf((accumulate/60000)%60, 2) + ";" + nf((accumulate/1000)%60, 2) + ";" + nf((accumulate/10)%100, 2), width/2, height/2+80);      if (keyCode == SHIFT) qw = -1;    }    else if (qw == 7) {      background(0);      text("End", width/2, height/2+40);      text("Point"+g.P, width/2, height/2);      text("Shift = Restart", width/2, height/2 +150);       text(nf((accumulate/60000)%60, 2) + ";" + nf((accumulate/1000)%60, 2) + ";" + nf((accumulate/10)%100, 2), width/2, height/2+80);      if (keyCode == SHIFT) qw = -1;    }  }  void **keyPressed**() {    if (key == CODED)g.input.keyPressedc(keyCode);    else g.input.**keyPressed**(key);  }  void **keyReleased**() {    if (key == CODED)g.input.keyReleasedc(keyCode);    else g.input.**keyReleased**(key);  }  void **keyTyped**() {    g.input.**keyTyped**(key);  } | Game g;  void **setup**() {    size(800, 400);    g = new Game();  }  void **draw**() {    g.update();  }  void **keyPressed**() {    if (key == CODED)g.input.keyPressedc(keyCode);    else g.input.**keyPressed**(key);  }  void **keyReleased**() {    if (key == CO「ED)g.input.keyReleasedc(keyCode);    else g.input.**keyReleased**(key);  }  void **keyTyped**() {    g.input.**keyTyped**(key);  } |

if else の関数は「processingをはじめよう」を参照

booleanでon,offの切り替えを行い、自機の変更をできるようにした。

imageModeで画像の中心ををx,yの始まりにした。これで当たり判定の簡略化。

g.input.keyPressedなどはclassを別に用意している。

qwを使い、

スタート画面、プレイ画面、クリアー画面、エンド画面、ゲームオーバー画面

と表示を変えられるようにした。qw=-1はゲームをリセットするためにある。

nowtime = millis() ; accumulate += nowtime – beftime ; beftime = nowtime;

は時間を表示するためにあり、nowtimeでその時の時間を記憶し、beftimeでその前の時間を記憶している。accumulateは今と前の差を足し続けている。それでゲームのプレイ時間を表示できるようにした。(父親のアドバイスを受けた)

nf()は表示するとき、常に二桁で表示するためにしている。

