```
PFont myFont;
//-----------player-----
int playerX = 240;
int playerY = 400;
int playerSize = 20;
//shot
int shotPx = playerX;
int shotPy = playerY;
int shotPVy = -5;
int shotPr = 7;
//flag
boolean playerFlag = true;
boolean moveLeft = false;
boolean moveRight = false;
boolean shotPFlag = false;
```

敵の共通項目をまとめて、元々いた敵はenemy0、2体目の敵はenemy1で表した

```
//-----------enemy common-------
PImage img;
int enemyR = 40;
int moveEVy = 30;
int moveCount = 0;
//shot
int shotEr = 7;
int shotEVx = 0;
int shotEVy = 5;
int shotECount = 0;
//-----------enemy0-------
int enemyX0 = 40;
int enemyY0 = 30;
int moveEVx0 = 10;
boolean enemyFlag0 = true;
//enemy0 shot
int shotEx0;
int shotEy0;
boolean shotEFlag0 = false;
//-----------enemy1-------
int enemyX1 = 90;
int enemyY1 = 30;
int moveEVx1 = 10;
boolean enemyFlag1 = true;
//enemy1 shot
int shotEx1;
int shotEy1;
boolean shotEFlag1 = false;
```

オープニングとエンディングの変数を宣言した

```
// opening ending
boolean openingFlag = true;
boolean endingFlag = false;
```

```
int opSelectedItem = 0;
int edSelectedItem = 0;

void setup() {
  size(480, 480);
  background(0);
  fill(0, 0, 0);
  smooth();
  noStroke();
  frameRate(50);
  myFont = createFont("MS Gothic", 24, true);
  img = loadImage("enemy.png");
```

変数を初期化した

```
  initValues();
}

void initValues() {
  moveCount = 0;
  // player
  playerX = 240;
  playerY = 400;
  playerFlag = true;
  moveLeft = false;
  moveRight = false;
  shotPFlag = false;
  // enemy0
  enemyX0 = 40;
  enemyY0 = 30;
  moveEVx0 = 10;
  enemyFlag0 = true;
  shotEFlag0 = false;
  // enemy1
  enemyX1 = 90;
  enemyY1 = 30;
  moveEVx1 = 10;
  enemyFlag1 = true;
  shotEFlag1 = false;
}

//------------------------
boolean collisionCheck(int myX, int myY, int myR, int youX, int
youY, int youR) {
  return (pow(myX - youX, 2) + pow(myY - youY, 2)) <= pow(myR / 2 +
youR / 2, 2);
}

//-----------PLAYER--------------------------------
void playerMove() {
  if (playerFlag) {
    if (moveLeft) {
```

```
      playerX -= 7;
    }
    if (moveRight) {
      playerX += 7;
    }
    fill(255);
    ellipse(playerX, playerY, playerSize, playerSize);
  }
}

void playerShot() {
  fill(255, 0, 0);
  if (shotPFlag) {
    shotPy += shotPVy;
    // atarihantei enemy0
    if (collisionCheck(shotPx, shotPy, shotPr, enemyX0, enemyY0,
enemyR)) {
      shotPFlag = false;
      enemyFlag0 = false;
      shotEFlag0 = false;
    }

    // atarihantei enemy1
    if (collisionCheck(shotPx, shotPy, shotPr, enemyX1, enemyY1,
enemyR)) {
      shotPFlag = false;
      enemyFlag1 = false;
      shotEFlag1 = false;
    }

    if (shotPy < 0) {
      shotPFlag = false;
    }
    ellipse(shotPx, shotPy, shotPr, shotPr);
  }
}
//-----------------------------------------------

//------------------ENEMY-----------------------
void enemyMove() {
  // enemy0
  if (enemyFlag0) {
    if (moveCount % 10 == 0) {
      enemyX0 += moveEVx0;
      if (enemyX0 < 20 || 460 < enemyX0) {
        enemyY0 += moveEVy;
        enemyX0 -= moveEVx0;
        moveEVx0 *= -1;
      }
    }
    fill(0, 255, 0);
    image(img, enemyX0 - (enemyR / 2), enemyY0 - enemyR / 2, enemyR,
enemyR);
  }
```

```
  // enemy1
  if (enemyFlag1) {
    if (moveCount % 10 == 0) {
      enemyX1 += moveEVx1;
      if (enemyX1 < 20 || 460 < enemyX1) {
        enemyY1 += moveEVy;
        enemyX1 -= moveEVx1;
        moveEVx1 *= -1;
      }
    }
    fill(0, 255, 0);
    image(img, enemyX1 - (enemyR / 2), enemyY1 - enemyR / 2, enemyR,
enemyR);
  }

  if (enemyFlag0 || enemyFlag1) {
    moveCount++;
  }
}

void enemyShot() {
  // enemy0 shot
  if ((shotECount = ++shotECount % 35) == 0 && enemyFlag0 && !
shotEFlag0) {
    shotEFlag0 = true;
    shotEx0 = enemyX0;
    shotEy0 = enemyY0;
  }

  // enemy shot1
  if ((shotECount = ++shotECount % 35) == 0 && enemyFlag1 && !
shotEFlag1) {
    shotEFlag1 = true;
    shotEx1 = enemyX1;
    shotEy1 = enemyY1;
  }

  // atarihantei enemy shot0
  if (shotEFlag0) {
    shotEy0 += shotEVy;
    if (collisionCheck(shotEx0, shotEy0, shotEr, playerX, playerY,
playerSize)) {
      playerFlag = false;
      shotEFlag0 = false;
      shotPFlag = false;
    }
    if (shotEy0 > 480) {
      shotEFlag0 = false;
    }
    ellipse(shotEx0, shotEy0, shotEr, shotEr);
  }

  // atarihantei enemy shot1
```

```
  if (shotEFlag1) {
    shotEy1 += shotEVy;
    if (collisionCheck(shotEx1, shotEy1, shotEr, playerX, playerY,
playerSize)) {
      playerFlag = false;
      shotEFlag1 = false;
      shotPFlag = false;
    }
    if (shotEy1 > 480) {
      shotEFlag1 = false;
    }
    ellipse(shotEx1, shotEy1, shotEr, shotEr);
  }
}
//----------------------------------------------------
```

オープニングとエンディング画面になる条件をかいた
オープニングが始まり、プレイヤーが死んだ場合エンディングが始まる
敵が２体とも死んだ場合もエンディングが始まる

```
void draw() {
  background(0);
  if (openingFlag) {
    opening();
  } else if (playerFlag == false) {
    endingFlag = true;
    ending();
  } else if (enemyFlag0 == false && enemyFlag0 == false) {
    endingFlag = true;
    ending();
  } else {
    playerMove();
    playerShot();
    enemyMove();
    enemyShot();
    printPlayTime();
  }
}
```

プレイ時間を表示した

```
void printPlayTime() {
  textSize(32);
  fill(255);
  text("time : " +moveCount, width - 200, height - 30);
}
```

オープニング画面

題名と操作説明とゲームを始めるか終わるかの選択肢が書いてある

```
void opening() {
  textSize(32);
  fill(255);
  text("invader", 10, 30);
  textSize(20);
  text("user guide",10,90);
  text("UP:shot",40,130);
  text("RIGHT,LEFT:move",40,160);
  text("ENTER:decide",40,190);
  textSize(32);
  text("->", 10, 300 + 30 * opSelectedItem);
  text("start", 60, 300);
  text("end", 60, 330);
}
```

エンディング画面
プレイヤーが勝った場合はenemy down!負けた場合はyour diedと表示される
さいごにはプレイ時間ともう一度始めるか終わるかの選択肢が書いてある

```
void ending() {
  textSize(32);
  fill(255);
  if (playerFlag == false) {
    text("your died", 10, 30);
    text("play time : " + moveCount, 10, 100);
  } else if (enemyFlag0 == false || enemyFlag1 == false) {
    text("enemy down!", 10, 30);
    text("play time : " + moveCount, 10, 100);
  }
  text("->", 10, 300 + 30 * edSelectedItem);
  text("re start", 60, 300);
  text("end", 60, 330);
}
```

startをENTERキーで選択するとオープニングが終わりゲームが始まる
それが当てはまらない場合はゲームが終わる

```
void keyPressed() {
  if (openingFlag) {
    if (keyCode == ENTER) {
      if (opSelectedItem == 0) {
        openingFlag = false;
      } else if (opSelectedItem == 1) {
        exit();
      }
    }
```

オープニングで上か下キーを押すと矢印を動かせる
restartをENTERキーで選択するとエンディングが終わりゲームが始まる
それが当てはまらない場合はゲームが終わる
エンディングで上か下キーを押すと矢印を動かせる

```
    if (keyCode == UP || keyCode == DOWN) {
      opSelectedItem ++;
      opSelectedItem %= 2;
    }
  } else if (endingFlag) {
    if (keyCode == ENTER) {
      if (edSelectedItem == 0) {
        endingFlag = false;

        setup();
      } else if (edSelectedItem == 1) {
        exit();
      }
    }
    if (keyCode == UP || keyCode == DOWN) {
      edSelectedItem ++;
      edSelectedItem %= 2;
    }




  } else {
    if (key == 'a' || keyCode == LEFT) {
      moveLeft = true;
    }
    if (key == 'd' || keyCode == RIGHT) {
      moveRight = true;
    }

    if (!shotPFlag && playerFlag) {
      if (key == 'f' || keyCode == UP) {
        shotPFlag = true;
      }
      shotPx = playerX;
      shotPy = playerY;
    }
  }
}

void keyReleased() {
  if (key == 'a' || keyCode == LEFT) {
    moveLeft = false;
  }
  if (key == 'd' || keyCode == RIGHT) {
    moveRight = false;
```

```
    }
}
```