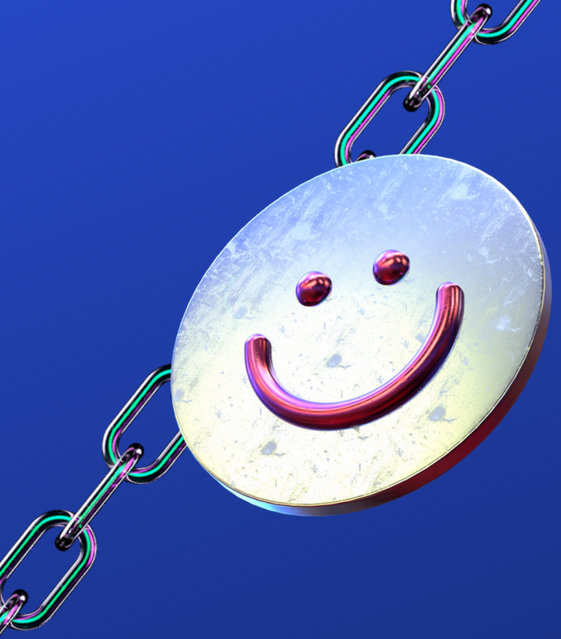
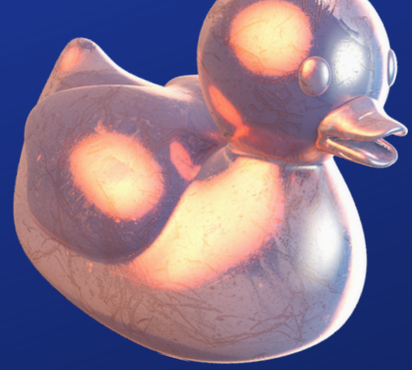
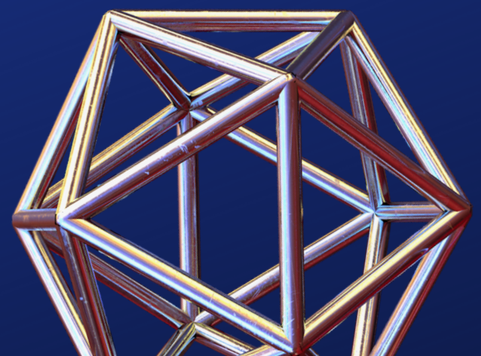
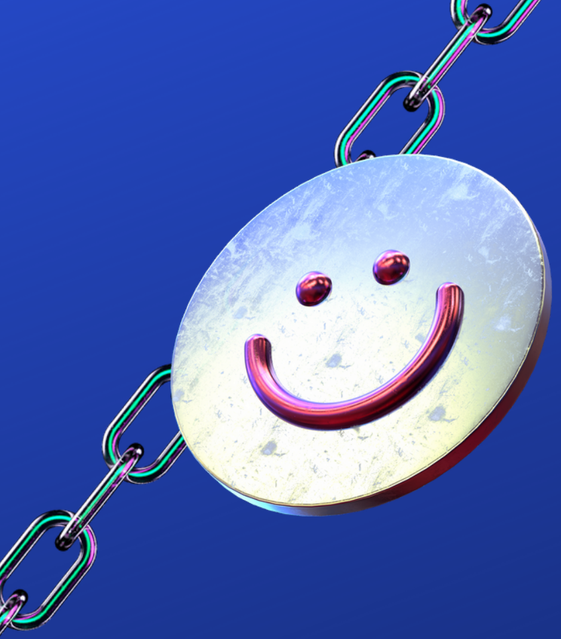


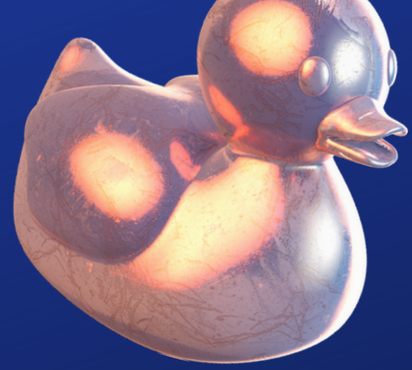


Mini Project

SKY RUSH



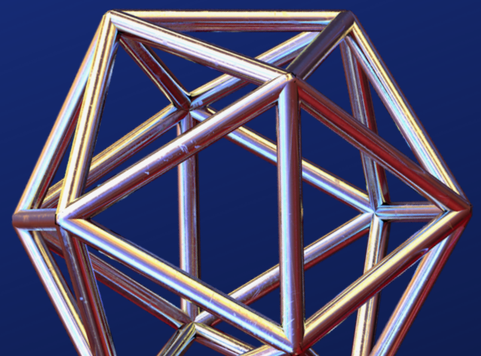




สมาชิก

นายณัฏฐนันธ์ ไทยสถิตย์ 65120501002

นายภคพล กลิ่นขจร 65120501014



โค้ดการทำงาน

```
import java.util.Iterator;

PImage bgImage;
PImage bgImage2;

Ship triangleShip, squareShip;
ArrayList<Bullet> bullets = new ArrayList<Bullet>();
ArrayList<Asteroid> asteroids = new ArrayList<Asteroid>();
ArrayList<SpecialItem> specialItems = new ArrayList<SpecialItem>();

boolean gameStarted = false;
boolean gameOver = false;
boolean showMenu = false;
int specialItemsCollected = 0;
int bulletSpeed = 5;
int Score = 0;
boolean collided = false;
float asteroidSpeed = 2.0;
int startTime;
int totalPlaytime;

void setup() {
  size(800, 600);
  triangleShip = new Ship(width / 4, height / 2, 30, 10);
  squareShip = new Ship(width * 3 / 4, height / 2, 30, 5);
  bgImage = loadImage("1.png");
  bgImage2 = loadImage("2.jpg");
}
```

```
void draw() {
  image(bgImage2, 0, 0, width, height);
  if (gameOver) {
    displayGameOverScreen();
    return;
  }

  if (!gameStarted) {
    if (showMenu) {
      displayCharacterSelection();
    } else {
      displayStartScreen();
    }
  } else {
    if (triangleShip.isSelected()) {
      triangleShip.display();
      createAsteroids(10);
    } else if (squareShip.isSelected()) {
      squareShip.display();
      createAsteroids(10);
    }

    for (Iterator<SpecialItem> itemIterator = specialItems.iterator(); itemIterator.hasNext(); ) {
      SpecialItem item = itemIterator.next();
      if (item.isActive()) {
        item.move();
        item.display();
        checkBulletSpecialItemCollision(itemIterator, item);
      } else {
        itemIterator.remove();
      }
    }

    for (Iterator<Bullet> bulletIterator = bullets.iterator(); bulletIterator.hasNext(); ) {
      Bullet bullet = bulletIterator.next();
      bullet.move();
      bullet.display();
      checkBulletAsteroidCollision(bulletIterator, bullet);
    }
    moveAsteroids();
    for (Asteroid asteroid : asteroids) {
      asteroid.display();
    }
    checkShipAsteroidCollision();

    displayScore();
    displayPlaytime();

    if (collided) {
      totalPlaytime = millis() - startTime;
      gameOver = true;
      displayGameOverScreen();
    }
  }
}
```

โค้ดการทำงาน

```
void displayStartScreen() {
  image(bgImage, 0, 0, width, height);
  fill(255);
  textAlign(CENTER, CENTER);
  textSize(50);
  text("Start Game", width / 2, height / 2 - 25);
  textSize(20);
  text("Press Spacebar to start ", width / 2, height / 2 + 25);
}

void displayCharacterSelection() {
  fill(255);
  textAlign(CENTER, CENTER);
  textSize(40);
  text("Select Your Ship", width / 2, height / 2 - 75);
  textSize(20);
  text("Press 1 for Triangle Ship", width / 2, height / 2 - 25);
  text("Press 2 for Square Ship", width / 2, height / 2);
}
void keyPressed() {
  if (!gameStarted) {
    if (showMenu) {
      if (key == '1') {
        triangleShipSelected();
        startTime = millis();
        gameStarted = true;
      } else if (key == '2') {
        squareShipSelected();
        startTime = millis();
        gameStarted = true;
      }
    } else if (key == ' ') {
      showMenu = true;
    }
  }

  triangleShip.move('W', 'S', 'A', 'D');
  squareShip.move('W', 'S', 'A', 'D');

  if (key == ' ') {
    if (triangleShip.isSelected()) {
      bullets.add(new Bullet(triangleShip.x, triangleShip.y, bulletSpeed));
    } else if (squareShip.isSelected()) {
      bullets.add(new Bullet(squareShip.x, squareShip.y, bulletSpeed));
      bullets.add(new Bullet(squareShip.x, squareShip.y, -bulletSpeed));
    }
  }

  if (gameOver && key == '5') {
    restartGame();
  }
}
```

```
void triangleShipSelected() {
  triangleShip.setSelected(true);
  squareShip.setSelected(false);
}

void squareShipSelected() {
  triangleShip.setSelected(false);
  squareShip.setSelected(true);
}

void createAsteroids(int count) {
  if (asteroids.size() < count) {
    float randomSize = random(20, 60);
    asteroids.add(new Asteroid(random(width), random(height), randomSize));

    if (random(1) < 0.05) {
      specialItems.add(new SpecialItem(random(width), random(height), 0.5));
    }
  }
}

void moveAsteroids() {
  for (Asteroid asteroid : asteroids) {
    asteroid.move();
  }
}

void checkBulletAsteroidCollision(Iterator<Bullet> bulletIterator, Bullet bullet) {
  Iterator<Asteroid> asteroidIterator = asteroids.iterator();

  while (asteroidIterator.hasNext()) {
    Asteroid asteroid = asteroidIterator.next();
    float d = dist(bullet.x, bullet.y, asteroid.x, asteroid.y);
    if (d < asteroid.size / 2) {
      asteroidIterator.remove();
      bulletIterator.remove();
      Score++;
      break;
    }
  }
}
```

โค้ดการทำงาน

```
void checkShipAsteroidCollision() {
  Ship ship = triangleShip.isSelected() ? triangleShip : squareShip;
  for (Asteroid asteroid : asteroids) {
    float d = dist(ship.x, ship.y, asteroid.x, asteroid.y);
    if (d < asteroid.size / 2) {
      collided = true;
      break;
    }
  }
}

void checkBulletSpecialItemCollision(Iterator<SpecialItem> itemIterator, SpecialItem item) {
  for (Iterator<Bullet> bulletIterator = bullets.iterator(); bulletIterator.hasNext(); ) {
    Bullet bullet = bulletIterator.next();
    float d = dist(bullet.x, bullet.y, item.x, item.y);
    if (d < 10) {
      bulletIterator.remove();
      item.setActive(false);
      specialItemsCollected++;

      bullets.add(new Bullet(bullet.x, bullet.y, bulletSpeed));
      bullets.add(new Bullet(bullet.x, bullet.y, bulletSpeed, radians(20)));
      bullets.add(new Bullet(bullet.x, bullet.y, bulletSpeed, radians(340)));

      break;
    }
  }
}

void displayScore() {
  fill(255);
  textSize(20);
  text("Score: " + Score, 50, 20);
}

void displayPlaytime() {
  int elapsedTime = millis() - startTime;
  int seconds = (elapsedTime / 1000) % 60;
  int minutes = (elapsedTime / 1000) / 60;
  fill(255);
  textSize(20);
  text("Time: " + nf(minutes, 2) + ":" + nf(seconds, 2), 50, 50);
}
```

```
void displayGameOverScreen() {
  int totalSeconds = (totalPlaytime / 1000) % 60;
  int totalMinutes = (totalPlaytime / 1000) / 60;

  image(bgImage, 0, 0, width, height);

  fill(255);
  textAlign(CENTER, CENTER);
  textSize(50);
  text("Game Over", width / 2, height / 2 - 75);
  textSize(20);
  text("Your time : " + nf(totalMinutes, 2) + ":" + nf(totalSeconds, 2), width / 2, height / 2 - 25);
  textSize(20);
  text("Your score : " + Score, width / 2, height / 2 );
  textSize(20);
  text("Your specialitems : " + specialItemsCollected, width / 2, height / 2 + 25);
  textSize(30);
  text("Press 5 to restart", width / 2, height / 2 + 65);
}

void restartGame() {
  specialItemsCollected = 0;
  Score = 0;
  gameOver = false;
  gameStarted = false;
  bullets.clear();
  asteroids.clear();
  collided = false;
  showMenu = false;
}
```

โค้ดการทำงาน

```
class Ship {
  PShape shape;
  float x, y;
  int speed;
  boolean selected;

  Ship(float x1, float y1, float size, int speed1) {
    x = x1;
    y = y1;
    speed = speed1;
    shape = createShape();
    shape.beginShape();
    if (speed >= 6) {
      shape.vertex(-size / 2, size / 2);
      shape.vertex(0, -size / 2);
      shape.vertex(size / 2, size / 2);
    } else {
      shape.vertex(-size / 2, size / 2);
      shape.vertex(-size / 2, -size / 2);
      shape.vertex(size / 2, -size / 2);
      shape.vertex(size / 2, size / 2);
    }
    shape.endShape(CLOSE);
  }

  void display() {
    shape(shape, x, y);
  }

  void move(int upKey, int downKey, int leftKey, int rightKey) {
    if (keyPressed) {
      if (keyCode == upKey && y - speed >= 0) y -= speed;
      else if (keyCode == downKey && y + speed <= height) y += speed;
      else if (keyCode == leftKey && x - speed >= 0) x -= speed;
      else if (keyCode == rightKey && x + speed <= width) x += speed;
    }
  }

  boolean isSelected() {
    return selected;
  }

  void setSelected(boolean selected) {
    this.selected = selected;
  }
}
```

```
class Bullet {
  float x, y;
  int speed;
  float angle;

  Bullet(float x1, float y1, int speed1) {
    x = x1;
    y = y1;
    speed = speed1;
    angle = 0;
  }

  Bullet(float x1, float y1, int speed1, float angle1) {
    x = x1;
    y = y1;
    speed = speed1;
    angle = angle1;
  }

  void move() {
    y -= speed * cos(angle);
    x += speed * sin(angle);
  }

  void display() {
    stroke(#F5E960);
    line(x, y, x + speed * sin(angle), y - speed * cos(angle));
  }
}
```

โค้ดการทำงาน

```
class Asteroid {
  float x, y;
  float size;
  float speedX;
  float speedY;
  PShape shape;

  Asteroid(float x1, float y1, float size1) {
    x = x1;
    y = y1;
    size = size1;
    speedX = random(-asteroidSpeed, asteroidSpeed);
    speedY = random(-asteroidSpeed, asteroidSpeed);
    shape = createShape();
    shape.beginShape();
    for (int i = 0; i < 8; i++) {
      float angle = TWO_PI / 8 * i;
      float xOff = cos(angle) * size / 2;
      float yOff = sin(angle) * size / 2;
      shape.vertex(xOff, yOff);
    }
    shape.endShape(CLOSE);
  }

  void move() {
    x += speedX;
    y += speedY;

    if (x < 0 || x > width) {
      speedX *= -1;
    }
    if (y < 0 || y > height) {
      speedY *= -1;
    }
  }

  void display() {
    fill(150);
    noStroke();
    shape.disableStyle();
    shape(shape, x, y);
  }
}
```

```
class SpecialItem {
  float x, y;
  float speed;
  boolean active;

  SpecialItem(float x1, float y1, float speed1) {
    x = x1;
    y = y1;
    speed = speed1;
    this.active = true;
  }

  void move() {
    x += speed;
  }

  void display() {
    fill(255, 0, 0);
    ellipse(x, y, 20, 20);
  }

  boolean isActive() {
    return active;
  }

  void setActive(boolean active) {
    this.active = active;
  }
}
```

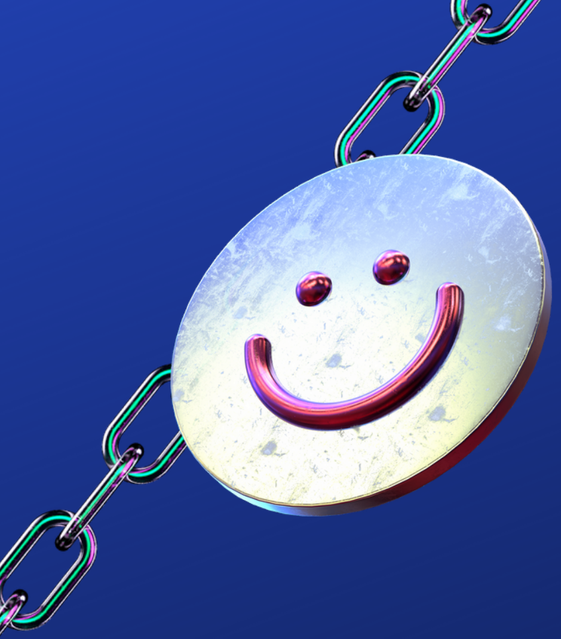

Start Game
Press Spacebar to start

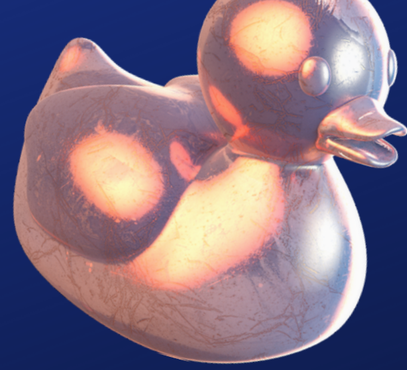
ปัญหาที่พบในการทำงาน

-ตัวยานและกระสุนชนกับอุกกาบาต โปรแกรมจะเกิด error ในตอนแรกที่เริ่มเขียนโค้ด

-ตัวเกมยังไม่ได้ตามที่ตั้งใจไว้ในตอนแรก

-ปัญหาเรื่องการสุ่มการเกิดของอุกกาบาต มีโอกาสสุ่มโดนผู้เล่น





THANK YOU