Title: Flyer

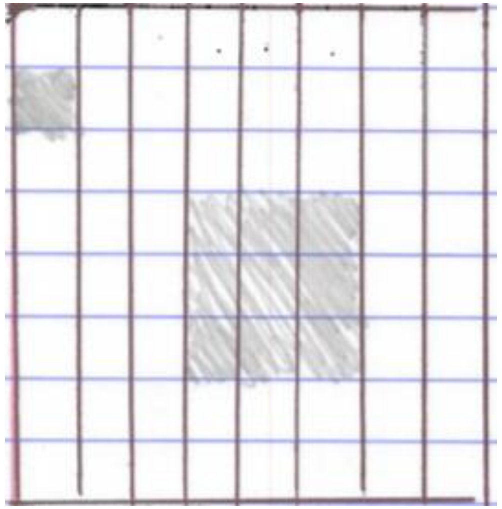
Logline: Fly around in a sky full of other pilots!

Team: Joseph Swetz

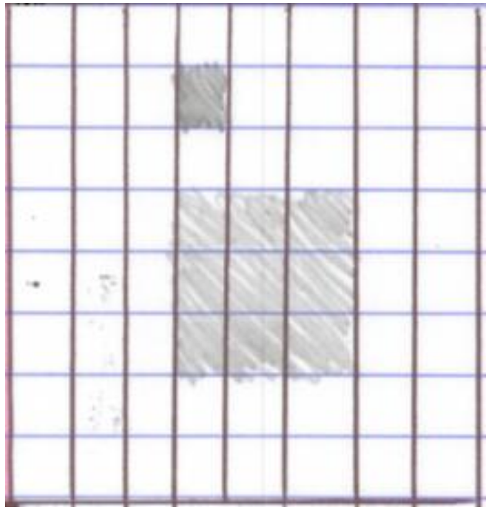
Activity Description:

The user moves a yellow square around the grid, with other random pilots (red beads) moving across the grid. The pilots will continuously appear along random edges of the grid and begin moving to a random spot on the opposite side of the grid. If a collision takes place, thus destroying a red bead, a sound is played and a visual appears.

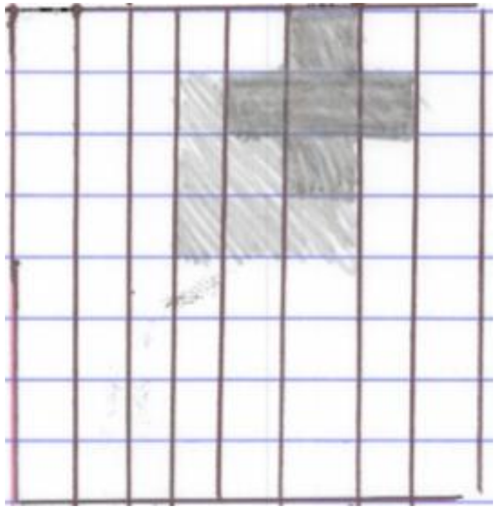
Mockups:



Toy starts with user in the middle. Red beads appear around the edges of the grid.



Red beads move across the grid.



Should the user choose to collide with a Red bead, a sound will occur and a visual will appear.

Essential Features:

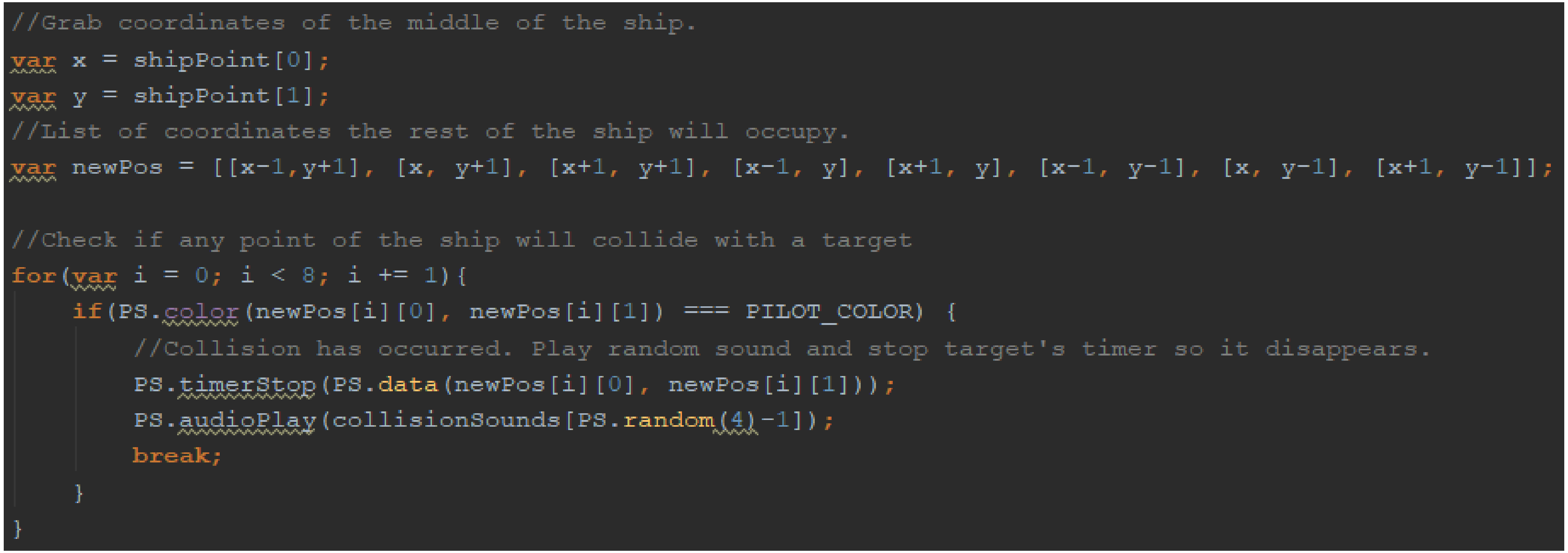
- Movement of yellow square with the arrow keys.
- Other pilots randomly appear at the edges of the grid.
  - Pilots fly across the grid to a random position on the opposite side of the grid.
  - Having new pilots to interact with helps to keep the toy interesting.
- A random sound (out of a set of possible sounds) is played and an impact visual appears when collision occurs.
  - Useful feedback so the user knows that they just collided with another pilot.
- A visually pleasing color scheme.
  - If no one wants to look at it, they won’t want to play with it, either.
  - Potential color scheme

🔵 RGB: 0, 0, 255

🔴 RGB: 255, 0, 0

🟡 RGB: 255, 205, 54

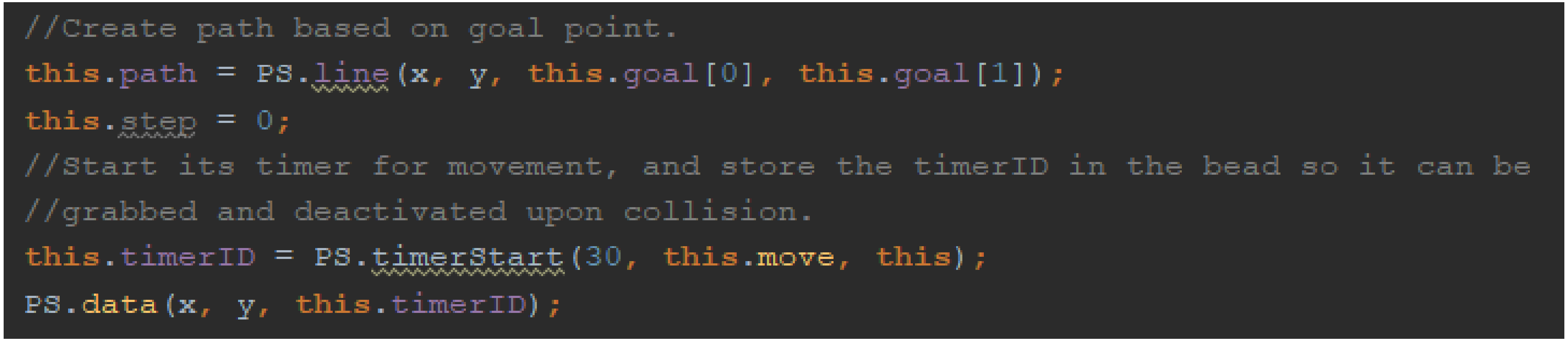
Demo Code: Movement/Collision



```
//Grab coordinates of the middle of the ship.
var x = shipPoint[0];
var y = shipPoint[1];
//List of coordinates the rest of the ship will occupy.
var newPos = [[x-1,y+1], [x, y+1], [x+1, y+1], [x-1, y], [x+1, y], [x-1, y-1], [x, y-1], [x+1, y-1]];

//Check if any point of the ship will collide with a target
for(var i = 0; i < 8; i += 1){
    if(PS.color(newPos[i][0], newPos[i][1]) === PILOT_COLOR) {
        //Collision has occurred. Play random sound and stop target's timer so it disappears.
        PS.timerStop(PS.data(newPos[i][0], newPos[i][1]));
        PS.audioPlay(collisionSounds[PS.random(4)-1]);
        break;
    }
}
```

Demo Code: Pilot Movement



```
//Create path based on goal point.
this.path = PS.line(x, y, this.goal[0], this.goal[1]);
this.step = 0;
//Start its timer for movement, and store the timerID in the bead so it can be
//grabbed and deactivated upon collision.
this.timerID = PS.timerStart(30, this.move, this);
PS.data(x, y, this.timerID);
```