

# Flyer

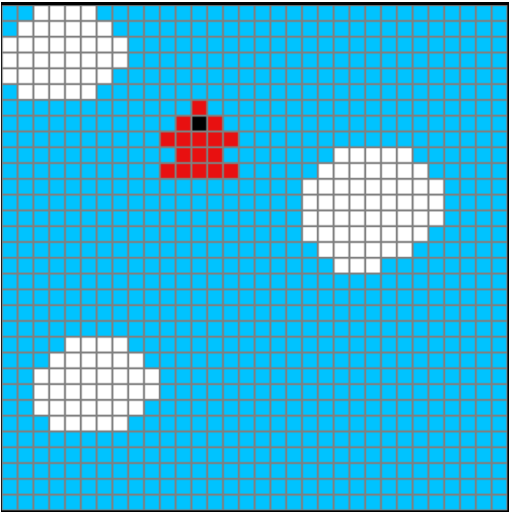
Logline: Embrace the Anarchy

Team: Joseph Swetz,  
Login: jsswetz

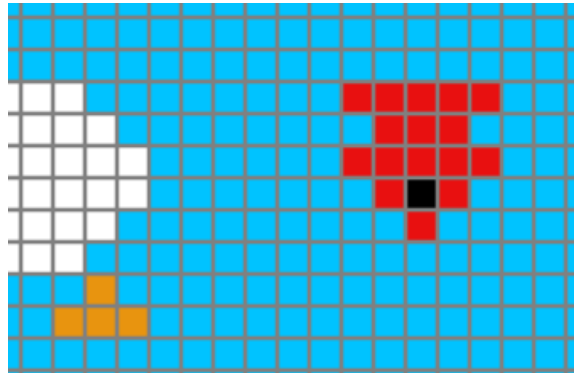
# Activity Description:

The user moves a plane around the grid with the arrow keys or WASD. Orange planes will continuously appear along random spots on the 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 an orange plane, a sound is played and the orange plane will fade away.

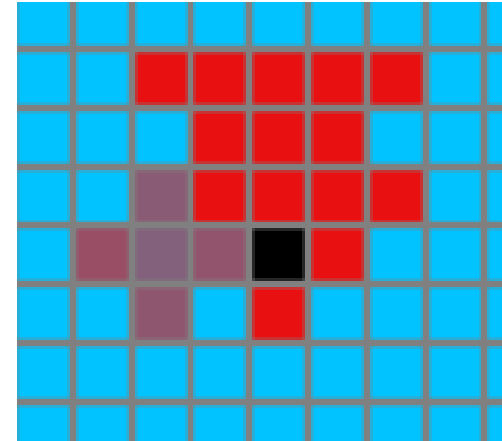
## Screenshots:



Toy starts with user in the middle moving upwards.



Red planes fly across the grid.



Should the user choose to collide with an orange plane, a sound will occur and the plane will turn to red and fade away.

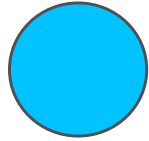
# Essential Features:

- Movement of plane with the arrow keys / WASD.
- 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 visual effect occurs upon collision.
  - Four possible sounds that can occur upon collision.
  - Orange planes turns to red upon collision.
    - This red will fade away into transparency.
  - 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.

# Significant Changes

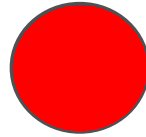
- Changed color scheme:

Background:



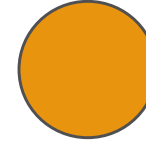
RGB: 0, 195, 255

Player:



RGB: 255, 0, 0

Other  
pilots:



RGB: 232, 148, 15

- Player movement more closely resembles an airplane than before.
  - No stopping in mid-air: plane constantly moves in the current direction.
  - Cannot turn in the opposite direction.
  - When changing direction, must wait a short span of time before turning again.
- Sprites are now used for the player and the other planes.
  - Improved collision detection using `PS.spriteCollide()`
  - 5x5 sprite for the player's plane means that it doesn't have to look square-shaped, but can collide with pilots in corners.

# Code: Collision

```
//Play collision sound and visual if s2 is a valid target sprite.
var collide = function(s1, p1, s2, p2, type){
    var collisionSounds = ["fx_blast1", "fx_blast2", "fx_blast3", "fx_blast4"];
    if(targetPlanes[p2]){//Check that there is a valid sprite on this plane.
        //Stop the target's movement timer.
        var timerID = targetPlanes[p2].timerID;
        PS.timerStop(timerID);
        PS.audioPlay(collisionSounds[PS.random(4)-1]); //Play explosion sound
        PS.spriteShow(s2, false); //Hide sprite (will delete itself once path ends)
        targetPlanes[p2] = null; //Free up plane for another target.
        numOfTargets -= 1; //Decrement total count of targets on the grid.
        var s2Coordinates = PS.spriteMove(s2); //Grab location coordinates of explosion.
        explode(s2Coordinates.x, s2Coordinates.y, p2); //Play explosion animation
    }
};
```

# Code: Explosion Animation

```
//Play and explosion animation at the given coordinates on the given plane.
var explode = function(x, y, plane){
    PS.gridPlane(plane); //Switch to the grid plane of the sprite that is exploding.
    var coordinates = [[x, y], [x-1, y], [x+1, y], [x, y-1], [x, y+1]];
    for(var i = 0; i < 5; i += 1){
        //Grab x and y coordinates
        var x = coordinates[i][0];
        var y = coordinates[i][1];

        //If x and y are on the grid, include them in the explosion effect.
        if((x >= 0) && (x < ROOM_WIDTH_HEIGHT) && (y >= 0) && (y < ROOM_WIDTH_HEIGHT)) {
            PS.alpha(x, y, 255); //Set bead to be fully visible.
            PS.color(x, y, PS.COLOR_ORANGE); //Set to explosion color.
            PS.fade(x, y, 60, {onEnd: function(x, y){
                PS.fade(x, y, 0); //Disable fader once effect is done.
            }, params: [x, y]
            });
            PS.alpha(x, y, 0); //Fade away to transparency.
        }
    }
    PS.gridPlane(0); //Switch back to bottom grid plane.
};
```

# Demo

<https://users.wpi.edu/~jsswetz/portfolio/Flyer/game.html>