Plane game

Game Style: 2D WW2 Plane simulation

Developer: Stephan Vitalyeivitch Kashkarov

# Overview

Plane game is a 2D simulation game based around ww2 plane combat. The game consists of two phases, “patrol” and “battle”.

The patrol phase is a large not to scale map that shows the player’s location in the world and in relation to other objects. Other objects in the world map include: tanks and planes. Planes track the player across the world map, upon collision with the player they trigger a battle. Tanks just sit on the world map at randomized locations and wait for the player to initiate the battle.

The battle phase is a side on arena “battle”. The player sprite can change rotation and throttle and shoot. The player will spawn at the left side of the screen. this will be mirrored by an AI controlled vehicle. This AI will attempt to land a hit from its bullets onto the player sprite and vice versa. The aim of this mode is to either kill or flee. The screen in the battle screen doesn’t move so if the player reaches the edges of the screen then the player will return to the patrol phase.

## Win Condition:

The player has cleared the patrol map of all enemies.

## Lose Condition:

Get hit by a single bullet

# Game Screens

## Screen 1 – Start Screen

|  |  |
| --- | --- |
|  | Events, Transitions and Callbacks:   * Click:   Continue to menu screen |

## Screen 2 – Menu Screen

|  |  |
| --- | --- |
|  | Events, Transitions and Callbacks:   * Play – go to HQ screen   Objects and Variables:   * Logo\_img and two buttons |

## Screen 3 – HQ level

|  |  |
| --- | --- |
|  | Events, Transitions and Callbacks:   * Go - puts the player into the patrol * Upgrade - upgrades the plane * Back – takes you back the menu   Objects and Variables:   * Go button * Back button * Upgrade button |

## Screen 4 – Patrol level

|  |  |
| --- | --- |
|  | Events, Transitions and Callbacks:   * Mouse movement – player follows * Battle - Triggered when player collides with enemy runs battle * Exit – land at your base   Objects and Variables:   * Player Object – position, lives and score * Hostile objects * Base |

## Screen 5 – Battle sequence

|  |  |
| --- | --- |
|  | Events, Transitions and Callbacks:   * Collision with Wall of Death   + Player loses a life   Objects and Variables:   * Walls – list of coordinates required to deal with complex shapes; property to determine if Wall of Death vs. regular wall.   Notes:   * Death due to collision with Wall of Death does NOT reset time, only player’s position to start of level. |

## Screen 6 – Win/Lose Screen

|  |  |
| --- | --- |
| << image goes here >> | Events, Transitions and Callbacks:   * Y, click on Yes   + Play Again – back to Level 1 with Score of 0 * N, click on No, Esc   + Exit the game   Objects and Variables:   * Win State – used to determine if the player won or lost, and to display an appropriate message. |

# Supporting Information

## Development Environment

The game will be developed in Python using Pygame. Some game assets (such as sprites and obstacles) will be created by the designer – some attribution will be required for assets owned by other people. (found in attributes.txt)

## Key Features

### Enemy AI

* In patrol will move in the direction of player until collision
* In battle the AI will just shoot at the player