305 Mini Project Short Report

Game Strategy

The game we are creating is called Flappy Plane Our game takes inspiration from Flappy Bird, swapping the bird for a plane and pipes for clouds. There are two game modes - **Training** and **Singleplayer**. Training mode stays on the lowest difficulty level until all three lives are depleted. The singleplayer mode is continuous, where the difficulty will increase after the plane has passed 10 clouds. The increasing horizontal scroll speed determines three difficulties - “easy”, “medium”, and “hard”, remaining “hard” after 30 clouds.

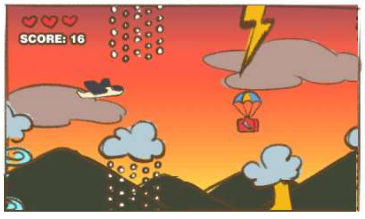
Design Specifications

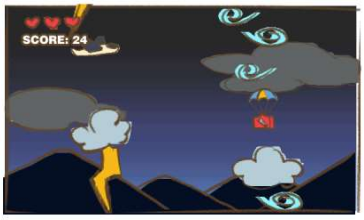
For user convenience, our approach was to create a simple, self-explanatory game that only utilised controls on-screen and straightforward gameplay. Players start with three lives and gain more by collecting toolboxes. Getting hit by clouds loses health, with options to retry or return home upon reaching game over.

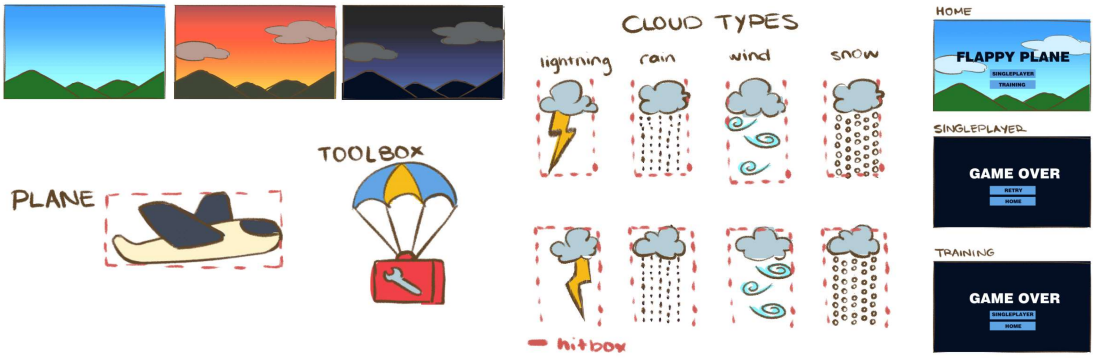
- **Hardware** - DIP switches, push-buttons and seven-segment displays will only be used during the debugging process
- **UI Controls** - All control elements in the game will be on the screen.
- **Plane Movement** - The plane can only move up and down
- **Obstacles** - Clouds (akin to Flappy Bird pipes)
- **Lives System** - The plane will have three lives at the beginning, increased by 1 per obtained toolbox in between clouds; losing all lives will result in game over.
- **Respawn Behaviour** - When the plane loses a life, it flickers and respawns to centre before resuming
- **Game Over Options** - Once the plane dies (all lives lost) in Singleplayer mode, the player can retry or return home, and once the plane dies in Training mode, the player can try Singleplayer mode or return home

Plans

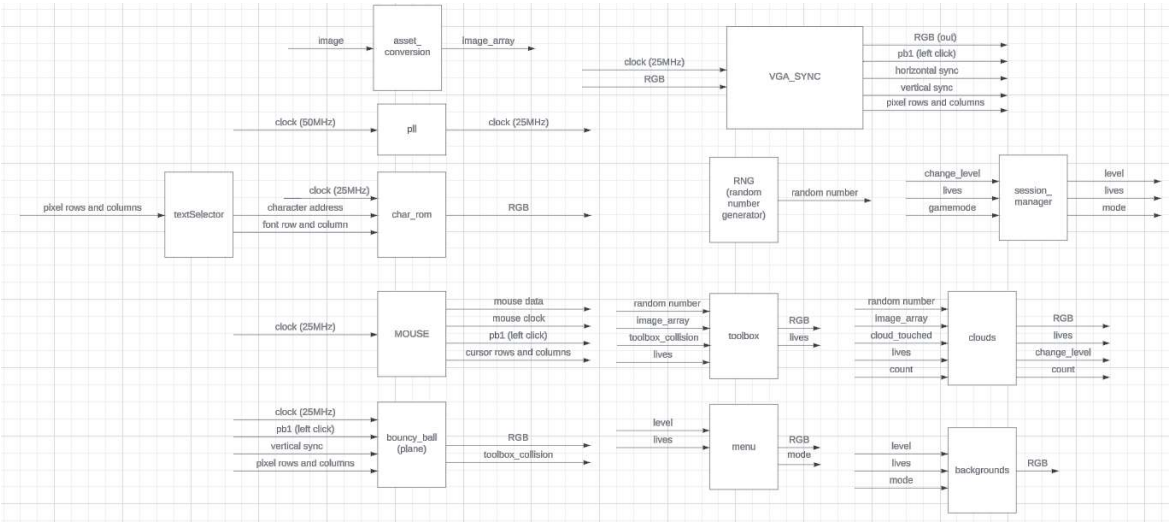
We plan to recreate the below designs for Flappy Plane, detailing the position of game elements and overall appearance. The main tasks to create the game are programming session logic (such as mouse control, RNG, difficulty, etc.), adding assets (including their behaviours) and testing. We will divide the tasks into three and delegate accordingly. Quartus Prime 18.1, DE0-CV development board and project management tools such as github will be used to complete this project by the end of week 11.





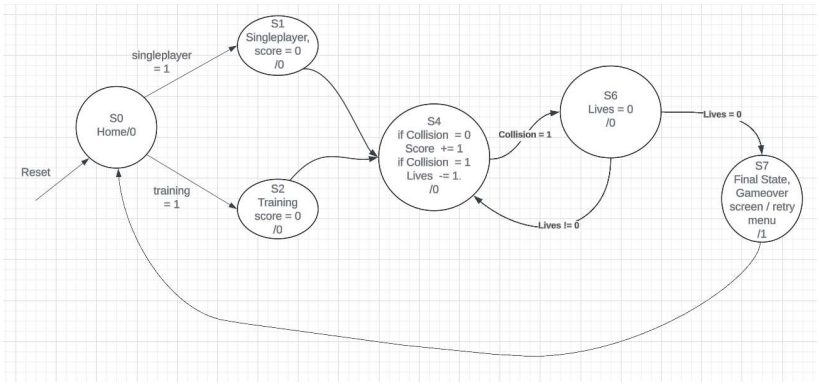


Block Diagram



High-level State Machine

At the start, the player will be presented with a home screen where they can choose between the game modes “Single-Player” or “training”. Once a mode has been selected the game will begin. When the player collides with an obstacle a “heart” will be deducted and once all hearts are depleted, the player will arrive at the final state which is the “game over” screen.



The user will now be able to “retry” the game again or go back to the home screen.