

Assessment 4 Implementation Plan

Draft 2 – 30/Mar/14

Menu Screens

- Replace the single “Play” button with two new ones, for playing in single- and multi-player modes
- Remove the credits screen and replace with a link to the team website (as we did in the last assessment)
- Update the information on the controls screen, with details of the new keyboard control scheme
- Also consider whether to add a direct link to the user manual

States

- Fill out the `MultiPlayState` class
- Edit (and potentially rename) the `PlayState` class to be consistent with the new multi-player mode

Airspace

- Remove left-hand sidebar, and move the clock to the top-centre of the screen
- Change the background image to match our own design aesthetic
- Remove instantiation of `Controls` from the `Airspace` class and into the game states
- Edit both the `SeparationRules` and `MultiPlayState` classes to make collisions non-fatal in multiplayer games

Controls

- Remove the radial menu, which is the `FlightMenu` class
- Also remove all the mouse control functions in the `Controls` class
- New `util.KeyBindings` class, that will contain a `HashMap` linking game functions (e.g. “turn-left”) to keyboard buttons (e.g. `Input.KEY_A`)
- Change `Controls` class to allow multiple instances, one for each player in multiplayer
- Each instance will take a `KeyBindings` object on creation to define the buttons it will handle
- Implement commands for cycling between planes using the keyboard
- Investigate how we can meet/relax the requirement for being able to turn planes to an arbitrary heading

Scores

- Add capability for multiple instances to the `ScoreTracking` class
- Potentially have the `ScoreTracking` class render itself
- Add point additions and deductions for multi-player events:
 - Deduct a large number of points from the controllers of both planes involved in a collision
 - Deduct a smaller amount of points for each separation violation
 - Add points for a successful, planned hand-over
 - Deduct points for an unplanned handover

Airports

- Ensure the `Airport` class can cope with multiple instances
- Add functionality for storing planes, and for automated handover of landing planes
- Airport will also need an owner field
- Actually add a second airport (to both gameplay modes)

Flights

- Add a variable and getter/setter for the flight's owner
- Change `drawFlight` function to colour-code by owner (in multiplayer) instead of by speed
- Potentially add fuel, or similar, functionality to force players to land planes