CP3 Project Proposals



|  |  |
| --- | --- |
| **Name** | William Tarrant |
| **T-Number** | T00019474 |
| **Module** | Computing Project |
| **Course** | Computing with Games Development |
| **Date Submitted** | 26 November 2012 |

**Student Name:** William Tarrant **T-Number:** T00019474

**Course:** Games Development

**Project Title:** Sky Wars (for Android)



Figure 1 Modelled aircraft that will be used in the game

**Project Description:**

This will be a **3D** game for the android platform where the user will take control of a fighter craft and partake in missions where they must shoot down invading enemy planes. It will be programmed using Unity android pro and C# as my primary coding language.

The player will take a third person view and patrol the sky’s, looking for AI controlled enemy craft, with the planes manoeuvres being controlled from the phones in built tilt axis sensors. The player must first align themselves with the enemy craft before they obtain missile lock; this applies also for the enemy craft. Once the missile is launched, it will fall under its own decision making controls and use captured data on the targets heat signature in order track, outsmart and calculate an intercept on its target (detailed below). The missile must obey the laws of physics to a certain extent in that its flight must be realistic (no sharp turns). The enemy craft will have the ability to drop flairs as a decoy the missile to avoid hits. The above rules will also apply for the players craft and will also have access to a number of flairs.

As I am aiming for a military type simulation for this project the main algorithm here will involve the controlling of the enemy craft and also of the missile guidance system. As mentioned, getting missile lock means line of sight (or falling within a cone of sight); this must be done to obtain a preliminary heat signature of the target. The AI will operate under an algorithm that will track the player’s movements using tools like radar (or Infrared) then analyse this data being obtained and try and make a best decision in obtaining its goal. Because we are dealing with two objects moving at high speeds and while also taking into account possible erratic manoeuvres the missile guidance system must continuously track the heat signature of the plane, store and analysis the results and calculate a best path to intercept its target as illustrated in the diagram below. This I am hoping should make gameplay more interesting as it is simply not a case of one object chasing the targets path but comes down to a game of psychology as here is an object making its own decision on how it is going to intercept you as a player base on your actions.

The main challenge I foresee will probably be the algorithm, as I am aiming at practical military application, all the characters are operating independently and will not have access to the information of other characters. The AI must perform correctly and obtain this information by its self (the position of its target).

Predicted intercept based on analysis of captured data

Capture and analysis data

Capture and analysis data

Capture and analysis data

Missile alters intercept path

**Student Name:** William Tarrant **T-Number:** T00019474

**Course:** Games Development

**Project Title:** 3D flight simulator (with Kinect controls)



Figure Concept gameplay view from the cockpit showing flight controls

**Project Description:**

The idea for the project I am pitching is to have the ability to control an aircraft using either Unity or the XNA library and integrating Kinect SDK and its skeletal tracking camera to track hand movements to control the craft.

The scope for the actual game would be having the user placed in the cockpit with access to the ‘usually’ aircraft controls so that they can simulate the flying of an aircraft. They will have the ability to take off, fly and land safely by using hand gestures to manipulate the cockpit controls i.e. pushing forward with both hands will make the cockpit joystick physically move forward and the craft to descend and vice versa. A hand gesture would also control the throttle.

An algorithm I was thinking about implementing was an auto pilot system. The user should be able to pass control of the craft over to the computer to fly it to its destination. The craft must be able to account for wind directions and fuel in order to make the most efficient flight path possible. Also it will need some collision avoidance system for other air traffic.

Some other scope for algorithms would be generating the height terrain from an image or a map and having model buildings placed as they come into view as they will be flying in effect in an open world.

Also using the Kinect I believe it is possible to have facial recognition so it opens the possibility to have facial recognition login to restore game states form a database.

The main challenges I foresee here would be learning about the Kinect and its SDK to be able to capture the player’s movements and associating them with an actual game control. When capturing the movements there is also the issue of ‘smoothing’ the operation and not having the controls react to every minute reaction movement.