Team A Project Proposal

Proposal Statement: Team A will design and implement Drone Wars. It will be an augmented reality player vs. player dogfight. Players will fly actual physical drones controlled by Android devices connecting to a shared server. Players will be able to view their flight via their drone’s front-facing camera. This camera will enable the players to “fire” virtual shots at the opposing drones. All flight and combat actions will be processed by the server.

 Game concept/ Game play: Drone Wars is a live “shooter” type game using physical aerial drones. The drones will be flown by the player using the Android user interface to control every movement. The interface will contain a dedicated kill zone in the middle of the screen. Once the drone is inside this area and the player hits the Fire button, the app will post a message to the server to let the other Android device know that a hit has been registered. The drone which was hit will react accordingly. Each Player starts with 3 lives, and and 10 virtual rounds of ammo to hit their opponent. Every shot fired reduces the rounds the player has available. Every 30 seconds, 1 round will replenish. The reason we are implementing replenishing rounds, is to make the game play a little more challenging, preventing the players from just rapid firing without consequence. After being hit the player has a short invulnerability time period to be determined later to move into safe position, after this period are once again susceptible to enemy fire. Each hit will reduce the players life by one and this will be shown on the players Android device. Once all 3 lives are depleted the drone will start a landing sequence where the user will have a predetermined amount of time to get the drone to a safe location for landing. After this time, the drone, wherever it is, will automatically land. The game will end when either match time runs out or there is only one drone remaining alive in the game. If time allows, there will be a global ranking system that will be implemented to keep track of match and player statistics.

 Graphics and Controls: The game will feature a live video feed from the physical drone. This video will allow the drone’s operator to view the forward view of the drone. The video will be fed to the user’s Android device where a Heads-Up Display (HUD) will overlay the image and provide the user with real time data about the condition of their drone. Included in the HUD will be battery life, health remaining, match clock, target area, a button enabling the drone to fire at the other drones, and enemy drone highlighting when in view. The Android device will be the controller for the drone.

 To control the drone, the user will use two virtual joysticks overlayed on the screen. The left joystick will enable the phone to move in the direction the phone is tilted relative to the forward camera. The right joystick will control the altitude as well as yaw, or the horizontal direction. The user will have the choice between the on-screen firing button as well as an external physical button on the device.

Implementation: Drone Wars will be implemented using a server-client architecture. In one match there will be one server and 2n clients communicating with each other, where n is the number of players in the match. Each client will include a drone and an Android device. The Android device will send a request to the server to perform an action on the drone. The server will then relay that request to the appropriate drone. The drone will send back the response and the server will then relay the response back to the Android device.

The server will be hosted on a local laptop running linux. This laptop will have a direct connection to a wireless router. The server, using the wireless router, will host a wireless LAN connection so that all the clients can connect and communicate.

The codebase will be maintained using Git and deployed on a local machine. If time allows, the user’s information for the leaderboard will be stored in a database which will communicate with the leaderboard server. The database will be maintained using MySQL. Communication with the server will be done using REST API calls. These calls will be reusable in the future if there is ever an iOS version.

 Design Plan: We will create the user interface to control each of the players respective drones. We are going to make this project modular. The first step is to create a server which the Android device can transmit commands to the drones. Concurrently, a very simple UI will be implemented in Android to test commands sent and received. Once this connection has been established, we can use the app that we built to test the connection Then it will be expanded to include multiple clients as well as iteratively adding game logic. The last step is expanding the UI to contain all the HUD information as well as targeting data. If time allows, the global ranking system will be implemented.