CSIS 4495 - Section 071

**Applied Research Project**

Group - Sep 08, 2025

Instructor

#### Bambang Sarif

Douglas College

World Rebalance

**Project Proposal**

A multiplayer game project with backend integration and real-time analytics





#### Group

Hao SUO - 300392463

Yaolong Liu - 300380547

Muye Li - 300388453

#### Lead Contact

Yaolong Liu

liuy122@student.douglascollege.ca

+1 263-978-5806

**Introduction**

We are developing a multiplayer game for our CSIS 4495 Applied Research Project. We have divided the project into three major parts: the game itself, the game backend, and the website portal.

We are creating *World Rebalance*, an online first-person shooter multiplayer game inspired by the worldview of *Command and Conquer 3*. Players can choose between two camps (GDI and Nod). In the game, we allow players to buy new weapons and vehicles to defeat enemies and achieve victory. Each camp will include three types of vehicles: assault vehicles, scout vehicles, combat tanks, and special high-end vehicles. For now, we will focus on developing one map, but if all core mechanisms are completed, we plan to add more selectable maps. After each battle, we will generate gameplay data such as kills, wins, time spent, vehicle usage, and payments through the backend.

We have observed that most games only provide in-game statistics, lacking detailed post-game analysis for players and stakeholders. Although some popular online games, such as *World of Warcraft* and *League of Legends*, provide websites where players can view statistics, these platforms are often limited in scope. For example, Blizzard provides an online portal for game statistics, but it only shows limited data and updates are often delayed. Similarly, Riot Games offers match history for *League of Legends*, but players cannot customize or compare data outside the built-in system. These limitations highlight a research gap: the lack of a real-time external analytics portal that allows players to access and compare gameplay data more flexibly.

The purpose of our project is to design and implement such a portal, which will provide players and stakeholders with clear and interactive access to game statistics, performance indicators, and comparative data. We assume that our platform will be able to perform real-time analysis of match data and provide live player rankings.

# Objectives

Our objectives are to enable players to access game statistics such as win rates, kill/death ratio, and game time without logging into the game itself. We will provide data analysis and visualization on the website to help users better understand performance trends. In addition, we will allow users to search and filter information related to other players, matches, or equipment data. We will also implement a basic user login system with secure authentication to provide controlled access to personalized features.

# Scope

* We will connect the website to the backend API developed by our teammates.
* We will present data in the form of tables and charts.
* We will implement interactive visualization of game statistics using libraries such as Chart.js.
* We will build the website with PHP and a MySQL database.
* We will develop the game using Unreal Engine, with C++ and Blueprints.
* We will create game asset models using Maya.
* We will use Python to analyze player performance and vehicle performance.

# Methodology

# We are building our online multiplayer first-person shooter in Unreal Engine 5 using C++ and Blueprints, and we are integrating it with Steam services. The game will support both server hosting and joining. Once all players have joined a skirmish, they will be able to choose a faction before entering the battle. When the game ends, we will read the log file from the host and asynchronously POST the data to the server through FHttpModule.

# For each game, we will store players’ performance data in a database, and we will use this data to train models that evaluate both overall player performance and vehicle-specific performance. We also plan to train models to determine whether certain vehicles are too powerful or too weak. These models will be retrained once a day, and performance will be evaluated when the backend program receives requests from the website.

# We will connect the website to the backend API that we are developing. We will use PHP to request and process the data, and we will set up a MySQL database to store it. Our main focus will be presenting the information in tables and charts, and we plan to use a JavaScript library such as Chart.js for visualization. Finally, we will deploy the site on a simple cloud platform so that it can be accessed during testing and presentation.

# Justification of Methodology

# We chose PHP and MySQL because we already learned them in our previous courses, and our team is more familiar with them. PHP is simple to use for building a website, and MySQL is a database for storing game data. Using these two together also makes the project easier to finish within the time limit.

# For data visualization, we plan to use D3.js. This library is good for making interactive charts and can update data in real time. Since our project is about showing match results and player rankings, D3.js fits with our goals well.

# For data analysis, we choose python. We will use models in scikit-learn library. This library is an open-source Python library that provides simple and efficient tools for data mining and machine learning.

# We will also deploy the system on a simple cloud platform. This makes the website easier to test and show to others, because it can be accessed from anywhere, not only one computer. We also plan to use Firebase authentication, so that users can log in safely and access their own data.

# Expected Results

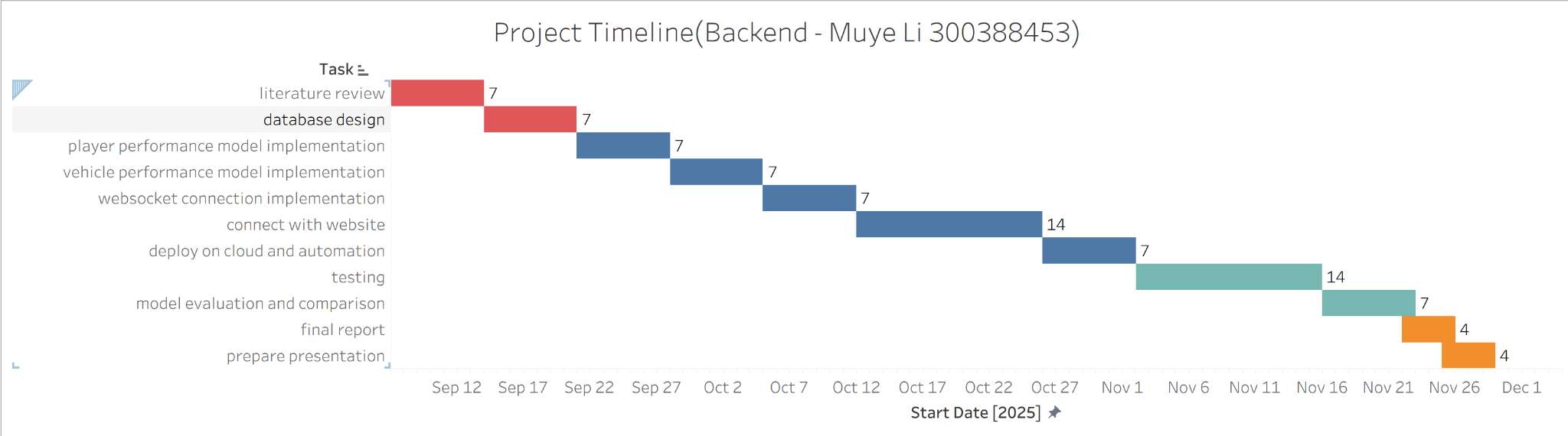
# A stable running multiplayer game. Minimum 10 players could run at the same time. Tracking and output data by players’ actions. Combat functions and purchasing systems are reliable.

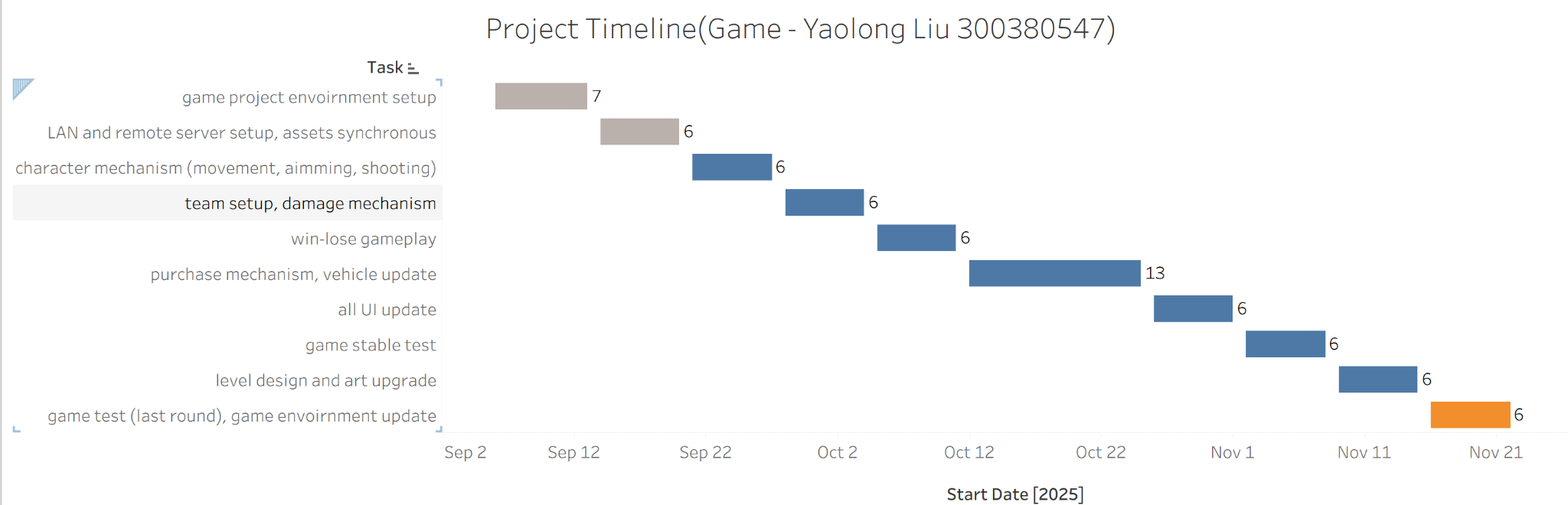
# The expected result of this project is a web portal that can connect to the backend and show real-time gameplay statistics. It will also provide interactive charts and live player rankings, so players can easily access and compare their game performance.

# Research Planning and Timeline

# The project timeline is illustrated in the Gantt chart below. It shows the planned start and duration of each major task in the website portal development.

# A graph on a computer screen AI-generated content may be incorrect.





# Deliverables

Muye Li

| 9/21 | Create a local database and generate data tables. |
| --- | --- |
| 10/5 | Finish two models’ code, ability to evaluate user and vehicle performance through game recording training. |
| 10/26 | Finish connect to website and game, put the game records into MySQL database, receive web page requests and return data, product API documents. Create midterm report and midterm video. |
| 11/2 | Deploy programs on cloud. |
| 11/16 | Linkage test documentation files. Create test documentation files. |
| 11/22 | Complete the model evaluation document. |
| 11/29 | Complete final report. |

Hao SUO

| Sep 18 | System Design | System architecture diagram/use case diagram with descriptions |
| --- | --- | --- |
| Sep 25 | Environment Setup | Configured PHP + MySQL environment/environment setup report with screenshots |
| Oct 3 | Basic Frontend Development | Prototype report |
| Oct 12 | User Authentication | User registration and login functions, session management, password hashing and input validation |
| Oct 19 | Database & API | Database connection, basic CRUD API endpoints, test screenshots showing data retrieval |
| Oct 29 | Data Visualization | 2–3 interactive charts (win rate, K/D ratio, playtime) implemented with Chart.js; integrated frontend display pages |
| Nov 10 | Unit Testing | Test cases for modules, unit testing report (pass/fail results), repository includes test files |
| Nov 15 | Integration Testing | End-to-end testing (login → upload data → visualization); bug report and fixes; integration testing summary document |
| Nov 20 | Cloud Deployment | Website deployed on cloud platform (AWS/Heroku/Azure); deployment guide with screenshots |

# Project Contract

# Weekly Meeting: Thursday during the class break and Sunday afternoon via Zoom

# Meeting Format: Review completed tasks, discuss blockers, and revise tasks as needed

# Communication: Whatsapp for daily updates, email for formal discussions

# Workflow: Tasks are assigned and tracked through the schedule tab in the group’s Google Doc

# Contributions: spend at least 10 hours/week to finish assigned tasks and reports

# We, the undersigned, agree to the above terms and commit to our responsibilities.

# Hao SUO Yaolong Liu Muye Li

# 

# 

# \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

# Work Logs

**Hao SUO**

| **Date** | **#Hours** | **Description of Work Done** |
| --- | --- | --- |
| Sep 07, 2025 | 1 | Reviewed the document *General Guidelines and Deliverables* provided by the instructor |
| Sep 08, 2025 | 3 | Reading the *A Guide to Technical Report Writing* to be familiar with the rules |
| Sep 08, 2025 | 2 | Drafted the proposal for my part of the project, which focuses on the development of the website portal for displaying and analyzing game data |
| Sep 09, 2025 | 3 | Worked on project planning and created the Gantt chart for the website portal tasks using Tableau. Updated the proposal with the timeline section. |
| Sep 11, 2025 | 6 | Compared different data visualization libraries and finalized Chart.js as the choice. |
| Sep 12, 2025 | 2 | Revised the project proposal based on instructor’s feedback, including unifying the narrative perspective, adding detailed deliverables, and correcting tense and grammar issues in the report. |
| Sep 13, 2025 | 3 | Discussed data structure issues with teammates, finalized basic tables and related columns, and aligned on game mechanics and design approach. |
| Sep 15, 2025 | 4 | Researched server options for handling game data uploads; compared multiple online server providers but have not finalized the choice yet. |
| Sep 18, 2025 | 4 | Set up the basic environment with PHP and MySQL database; used local resources temporarily since the server has not been finalized. |
| Sep 20, 2025 | 1 | Revised the project proposal again, reorganized the Deliverables section for clarity and detail. |
| Sep 22, 2025 | 3 | Discussed data structure with teammates, confirmed the columns of the match table, and analyzed how the portal should retrieve data and the basic API design. |

**Yaolong Liu**

| **Date** | **#Hours** | **Description of Work Done** |
| --- | --- | --- |
| Sep 07, 2025 | 5 | Created a test unreal project to learn ClientListener, ReplicatedAsset, Information synchronization. |
| Sep 08, 2025 | 2 | Learnt RPCs(Remote Procedure Calls) |
| Sep 09, 2025 | 4 | Redefine the game's worldview; Low-meshes in maya (GDI: 4 vehicles, Nod: 4 vehicles) |
| Sep 12, 2025 | 6 | Gamelog output; debug crush with build in unreal 5.6.1; change coding part from visual studio to Rider; Unsolveable crush in unreal 5.6.1, then switch to unreal 5.5.4. |
| Sep 13, 2025 | 4 | Setup new project, test LAN multiplayer connect. |
| Sep 14, 2025 | 8 | Learned using steam as the agent to hold a remote server. Add session create, join.Unsuccessed with a multiplayer test in a remote environment. |

**Muye Li**

| **Date** | **#Hours** | **Description of Work Done** |
| --- | --- | --- |
| Sep 7, 2025 | 2 | Discuss about the project proposal. |
| Sep 12, 2025 | 2 | Drafted the proposal of the project. |
| Sep 13, 2025 | 2 | Made detail plan about timeline and the function of the part I am responsible for. |

# Game Concept

| **Camp-Nod** Raider Buggy |  |
| --- | --- |
| ArtStation - NOD Raider Buggy Concept - (COMMAND & CONQUER 3) | Health: 100 |
| Speed: |
| Damage: |
| Rate of Fire: |
| Cost: |
|  |

| **Camp-Nod** Attack Bike |  |
| --- | --- |
| A black and white drawing of a motorcycle  AI-generated content may be incorrect. | Health: 70 |
| Speed: |
| Damage: |
| Rate of Fire: |
| Cost: |
|  |

| **Camp-Nod** Scorpion Tank |  |
| --- | --- |
|  | Health: 300 |
| Speed: |
| Damage: |
| Rate of Fire: |
| Cost: |
|  |

| **Camp-Nod** Flame weaponry |  |
| --- | --- |
| Flame weaponry - Command & Conquer Wiki - covering Tiberium, Red Alert and  Generals universes | Health: 500 |
| Speed: |
| Damage: |
| Rate of Fire: |
| Cost: |
|  |

| **Camp - GDI** APC |  |
| --- | --- |
| APC (Tiberium Wars) - Command & Conquer Wiki - covering Tiberium, Red Alert  and Generals universes | Health: 150 |
| Speed: |
| Damage: |
| Rate of Fire: |
| Cost: |
|  |

| **Camp - GDI** Pitbull |  |
| --- | --- |
| Pitbull (Tiberium Wars) - Command & Conquer Wiki - covering Tiberium, Red  Alert and Generals universes | Health: 100 |
| Speed: |
| Damage: |
| Rate of Fire: |
| Cost: |
|  |

| **Camp - GDI** Predator Tank |  |
| --- | --- |
| Predator tank (Tiberium Wars) - Command & Conquer Wiki - covering ... | Health: 700 |
| Speed: |
| Damage: |
| Rate of Fire: |
| Cost: |
|  |

| **Camp - GDI** Mammoth Tank |  |
| --- | --- |
| Рассказ об игре Command and Conquer 3: Tiberium Wars + История C&C. | Пикабу | Health: 1500 |
| Speed: |
| Damage: |
| Rate of Fire: |
| Cost: |
|  |

Our Mod:



# Game Mechanism

| Character mechanism | Health, Walk, Run, Jump, Aim, Shoot, Damage, Purchase Vehicle. |
| --- | --- |
| Light Vehicle mechanism | Drive, Accelerate, Gunshoot, Zoom lens rate, Damage, Steering speed. |
| Bike/Pitbull mechanism | Drive, Accelerate, Rocket, Zoom lens rate, Damage, Steering speed. |
| Tank mechanism | Drive, Accelerate, Rocket, Zoom lens rate, Damage, Steering speed. |

# Reference

Unreal Engine, (2025). Unreal Engine 5.6 Documentation. <https://dev.epicgames.com/documentation/en-us/unreal-engine/unreal-engine-5-6-documentation>

Command & Conquer Wiki. Command & Conquer 3: Tiberium Wars. <https://cnc.fandom.com/wiki/Command_%26_Conquer_3:_Tiberium_Wars>