Team 9
John Peecook
Andrew Brand
Meg Jones

Team 9 Design Plan

1 - Project Title

Real-time 2D Differential Game Combat Simulation

2 - Refined Problem Statement

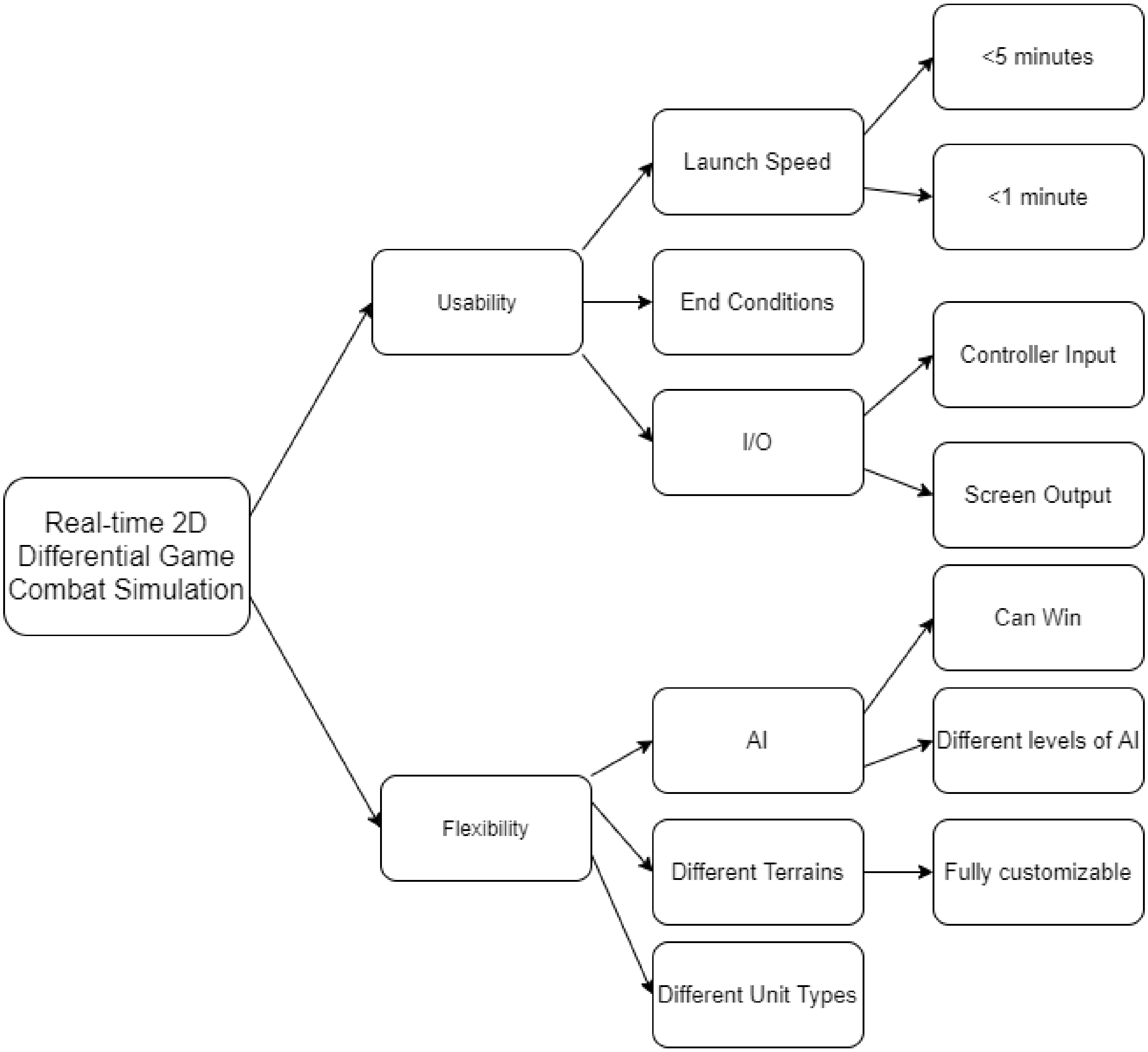
The main goal of this project is to make an aerial combat simulation that has user controlled units. This will allow enjoyment as well as testing hand eye coordination by avoiding incoming enemies and returning fire. The main problem to overcome is designing the game in a way that is challenging but fun as the game market is increasingly filled with games focused so heavily on difficulty that it impacts the enjoyment of the player. While this project has no large-scale impacts on the world many people would enjoy new games to play and things to invest time into so this game is purely for the enjoyment factor and for learning all the units' different styles to better perform in the game.

This is a very saturated field these days as many different groups are releasing video games of various styles and themes so our game would need to have some uniqueness to be successful. Our game would have inspiration from older games like asteroids, galaga, and many more to base some functions off of. We would then need to develop them in our own way to make it stand out in the crowd of other games. Doing this would allow it to be different and offer players a unique enjoyable experience with our game.

3- Objectives/Constraints

Attribute	Objective	Constraint	Function	Means
Launches scenario in less than 5 minutes		**X**	**X**	
Launches scenario in less than a minute	**X**			
Scenarios have an end condition			**X**	
Controller Input			**X**	**X**
Screen Output			**X**	**X**
Enemy AI can win		**X**		

Multiple levels of difficulty	X			
Different Terrain Preset Options		X		
Fully Modular Terrain	X			
Different Unit Types		X		

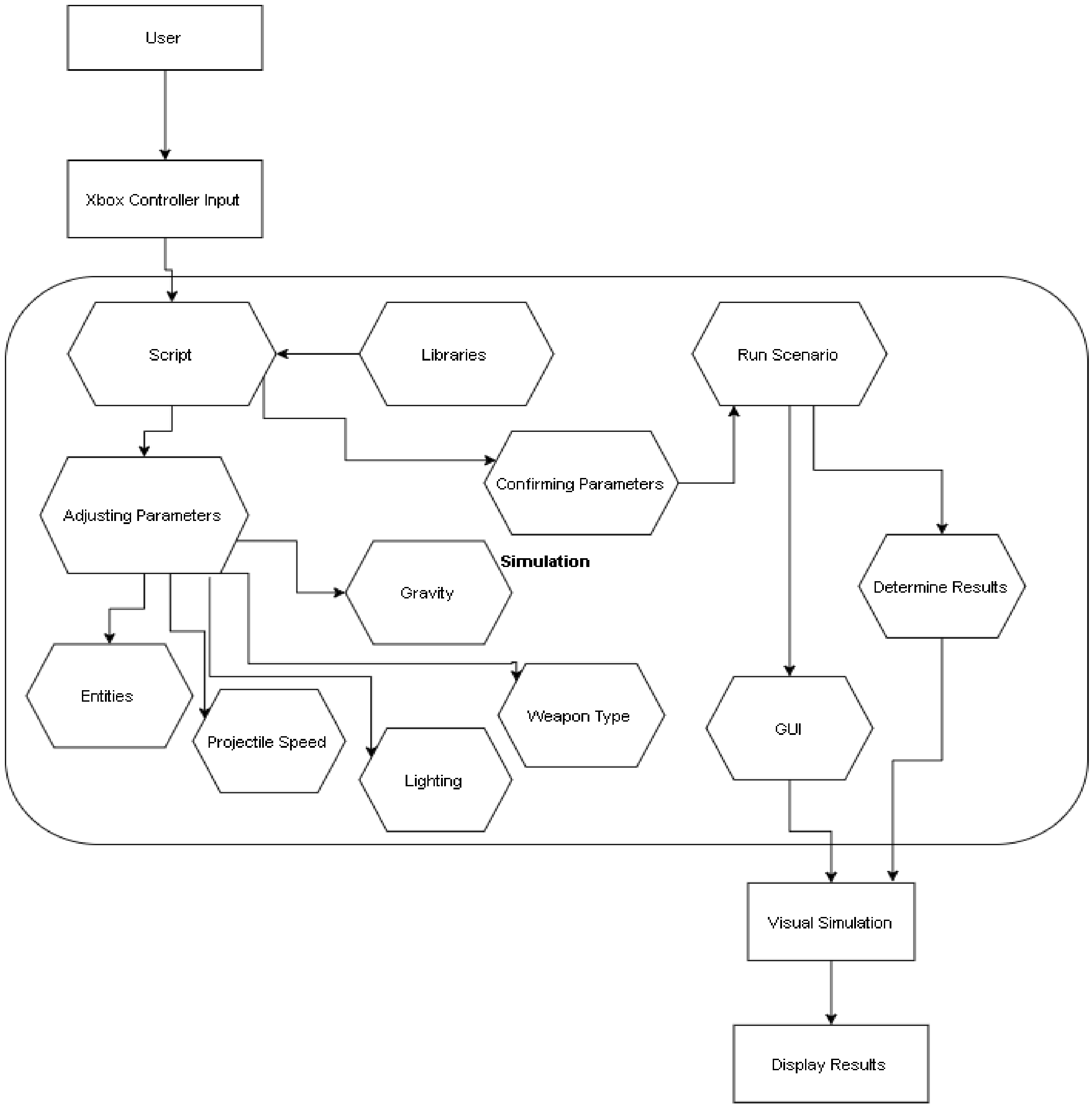


Our design has 2 major attributes: usability, and flexibility. We view the first several attributes as usability since they dictate if the program functions at its core. The later few attributes are to do with flexibility as their purpose is for the user to create different scenarios.

For us, usability is first and foremost dictated by its launch speed, as if someone has to wait a long time for something to work, they are less likely to want to use it. Currently we are not sure what the exact compile time will look like, however we consider this estimate to be fairly conservative yet not unreasonable given modern games. This will be tested with a simple stopwatch. From there we want the scenarios to have clear end scenarios. This is very easily verified and inserted, yet integral and cannot be forgotten. For example if all units on either side are defeated, the side with units remaining would win. The next most important aspect we thought of is input and output as the user needs a way to interact with the simulation, as well as react to the program changing. Perhaps the easiest to verify, simply interacting with the controller, and watching the screen afterwards will confirm this.

As for flexibility, we categorized things under this title by asking if the simulation could function without them existing. First, we wanted to make sure that the AI would have ways to win, as users would not want to return if they did not have a challenge. This test will be not dissimilar to a unit test, just allowing the user to fail the objective. Next we are looking at ways to change the difficulty of the AI, allowing users to get better at using the program. This will be verified by testing the simulation and seeing if the changes are active, changes such as increased movement speed or durability. After that we consider an integral part of the simulation to be different terrains, allowing even more customizability. At first we will have only a handful of presets that users can cycle between, with a long term goal being fully modular terrain, as in the users will be able to take a grid and insert terrain options as they see fit. This is to be verified by us creating a handful of default terrains and alternating between them. Lastly, we want to have various unit types, giving even more variety to scenarios. This is confirmed by entering a simulation and seeing if the units have different characteristics such as durability or movement speed.

4- Functional Description



The user will provide some input from an Xbox controller which is connected to the computer. The program will take this input and interpret it. From there the user controlled unit will react accordingly, such as when the user presses left, the unit will move to the left, From there a series of other factors will be taken into account such as walls or other units to determine how the input is executed, for example if someone attempts to walk through a wall they will not be able to. This information will be displayed to the user via the computer’s screen. Before being deployed into a scenario we will have a screen that allows users to modify the scenario such as changing the layout of the area, or modifying the number of units on the map. This will be in the form of a GUI presented to the user before deploying.

5- Project Requirements

- **Usability Requirements**

  Usability will rely on the base functionality of the game so we will need it to turn on. We will also need the simulation system to connect to the screen and also to read commands from the external controller that the player will use and allow the player to control their character as well as navigate menus. This will also include the interaction with enemies and how combat will be interpreted for both enemy and player characters. We must also implement how the combat will work as well as the visual effects that will occur with the game to let the player know what is happening, whether that be explosions when an enemy unit or the player is hit by attacks or lasers for weapon fire so that attacks can be seen in time to react. We will also need to set exit conditions for the missions and clear goals to be displayed to the player.

- **Flexibility Requirements**

  The flexibility requirements are all about making it so the user can return to the simulation several times without getting bored. By allowing the AI to win it gives the user a reason to pay attention. By giving the AI multiple levels, it allows the user to get better at playing and allows them to learn and adapt. By allowing the user to change the environment it enables the user to change up the game in even more ways. The addition of fully customizable terrain enables the user to build specific scenarios they have envisioned to see how it plays out, additionally adding nearly endless possible scenarios. Finally, by introducing multiple unit types it lets the player not only try to learn how to counter the AI units of the various types, but it allows the player to learn how to play as those various types adding yet another facet for them to master. We view all of these attributes as necessary because in our opinion a game's quality is best quantified by the time spent playing it, and by adding these options it lets the user use the simulator more without getting bored.

6- Team Participants/Qualifications

- Andrew Brand (CompE): Data Structures, OS, Software Engineering, Computer Organization and Architecture, Game Design, Java, AI
- John Peecook (CompE):Data Structures, OS, Software Engineering, Computer Organization and Architecture, Information Retrieval, Software Testing and QA
- Meg Jones (CS):Data Structures, OS, Computer Organization and Architecture, AI

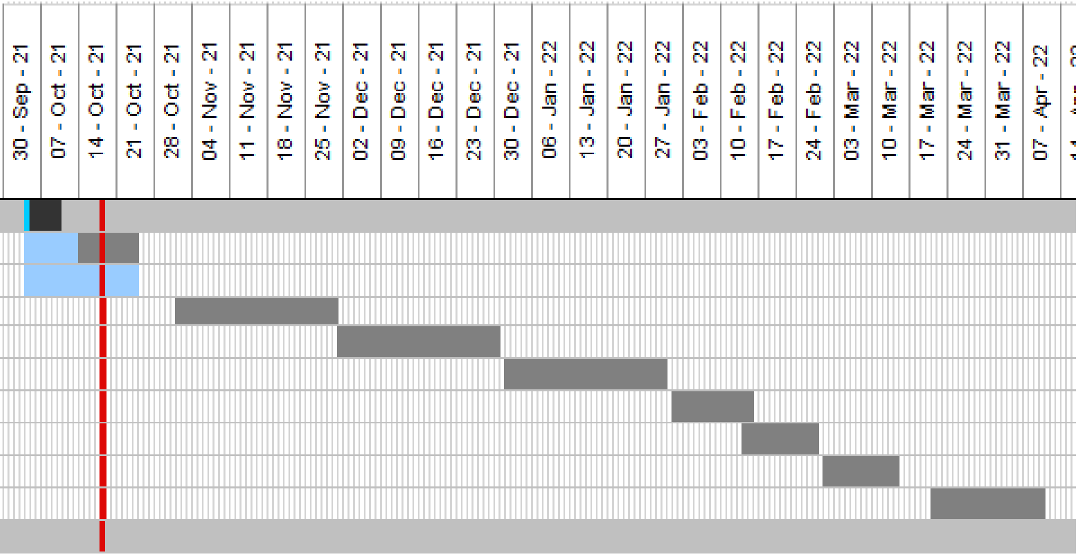
7- Advisor

Professor John Gallagher

- Expertise in computer science and engineering
- Currently researching for USAF so project aligns with his work

8- Project Timeline/Gantt Chart

Tasks	Task Lead	Start	End	Duration (Days)	% Complete	Working Days	Days Complete	Days Remaining
Reasearch	[Name]	10/4/21	10/19/21	7	21%	12	1	6
Decide on Project		10/4/21	10/19/21	21	50%	12	10	11
Find Advisor		10/4/21	10/19/21	21	100%	12	21	0
Resarch		########	#######	30	0%	23	0	30
Simulation Design		11/1/21	12/1/21	30	0%	22	0	30
Simulation Design Bug fixing		12/1/21	12/30/21	30	0%	10	0	30
AI implementation		1/1/22	1/15/22	15	0%	10	0	15
Scenario Creation		2/1/22	2/14/22	14	0%	12	0	14
Scenario Creation Bug fixing		2/14/22	3/1/22	14	0%	15	0	14
Agent Control Design		3/1/22	3/21/22	21	0%	9	0	21
Agent Control Design Bug fixing		3/21/22	3/31/22	10	0%	18	0	22


30 - Sep - 21
07 - Oct - 21
14 - Oct - 21
21 - Oct - 21
28 - Oct - 21
04 - Nov - 21
11 - Nov - 21
18 - Nov - 21
25 - Nov - 21
02 - Dec - 21
09 - Dec - 21
16 - Dec - 21
23 - Dec - 21
30 - Dec - 21
06 - Jan - 22
13 - Jan - 22
20 - Jan - 22
27 - Jan - 22
03 - Feb - 22
10 - Feb - 22
17 - Feb - 22
24 - Feb - 22
03 - Mar - 22
10 - Mar - 22
17 - Mar - 22
24 - Mar - 22
31 - Mar - 22
07 - Apr - 22