Game Overview

Game Concept

Collaboration – Players must work together controlling one vehicle to complete the game. Unless the players are able to communicate what they are doing effectively and work towards the same goal, they will be unable to defeat the enemies.

Game Genre

Action Adventure – Sequences of puzzle styled combat, through various landscapes, with story interspersed.

Target Audience

Teenagers 13+ - High school age students to young adults, to help develop social skills at an important stage of development. The way the levels are designed – linear, with lots of action and clear goals, caters best to the killer and achiever player types, but the social aspect will likely also appeal to the socializer.

Game mode

Co-operative multiplayer – The game must be completed with another player, and the entity the players control requires 2 people to use. Both players must drive this entity (likely a tank or vehicle of some sort) to defeat a series of bosses.

Goal

The goal of this game is to promote collaboration and teamwork in order to achieve a common task. The tasks are presented to the players in each level and they have to work together to control a unit that will help them complete them.

This is achieved by allowing two players to control one vehicle. They act as each character, controlling different parts of the vehicle in order to defeat level bosses. This promotes collaboration as the operations to be carried out will be designed in a way such that it requires two players.

Storyline

This game revolves around 2 characters who are both top engineers but rivals in the industry. They have been assigned by NASA to work the same project which requires them to create a defence machine that can be used by one pilot.

Both are stubborn and their poor communication skills and lack of teamwork results in their final design becoming inoperable by a single person. Each engineer only knows how to operate the components they've made and that is when disaster strikes!

An alien invasion has occurred and as it turns out, their machine is one of the best the world has and is needed to stop this! They are the only ones that can operate it and so they are sent out on the mission… together.

Central Characters

The central characters in this game are two engineers, one female and one male. They have the skills and expertise for great design, however, their attitude and personal differences stop them from effectively working together. This results in lack of communication and teamwork between the two.

They are then forced to learn to work together as they are assigned on the same mission which means they will have to set aside their differences to succeed and save the world.

Game World
The game world starts on Earth where the players work to defend, defeat the boss and stop the alien invasion. If the completion of this level is successful, different missions are unlocked which leads them to different parts of the universe.

Scoring
Rather than a high-scores system, the achievement in the game will come through level completion. There will be a currently undetermined number of levels to play through. Additionally, to add a competitive aspect, and replayability, levels will be timed. The fastest times through levels are recorded on a leader board, encouraging players to try and beat these times.

Lives
These are represented by a health bar for both the shared character and the boss. This allows for attacks that do different amounts of damage, and is a well-known interface component.

Level Generation
Each level will be pre-built and have a boss with different mechanics which the players have to outwit. Some bosses will incorporate randomness to increase the challenge.

Design Features
- Highscore screen [5%]
- Monetisation [5%]
- Sound [10%]
- Local Multiplayer with Leaderboard - Turn based [10%]
- Fixed level design [5%]
- Implement game in 2.5D [10%]

Advanced Feature
Online multiplayer [10%]

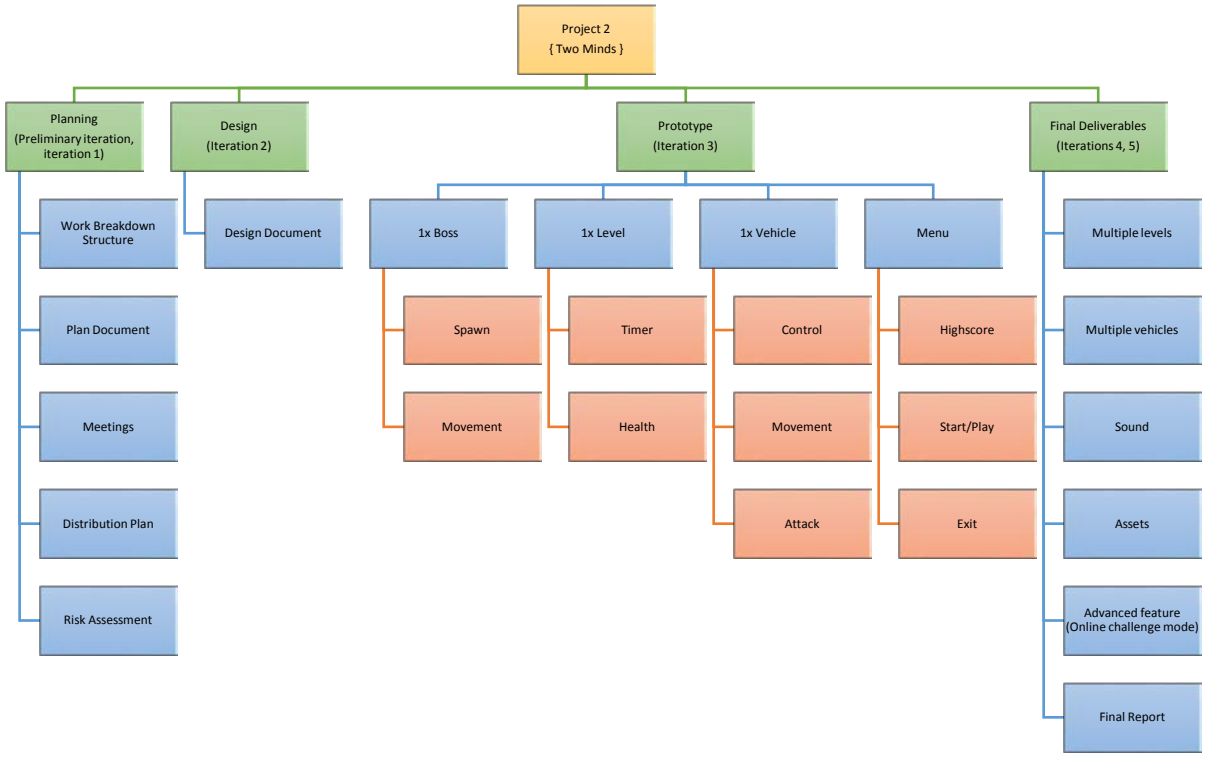
A team of two people enter the lobby on one computer and queue up until another team of two enter the lobby on a different computer and queue up. The server pairs the two teams together and they both enter the game with their own instances of the same map. The team to finish first is declared the winner.

If time permits, a matchmaking rating (MMR) would be added for pairing up the teams instead of getting the first two teams from the queue.

This is technically challenging because a server and interface would need to be created. An asynchronous server system is needed for the winner to be determined as the server must be notified when a player finishes. In addition to the challenges mentioned above, there would need to be a system for assigning and calculating MMR scores.

Tools and Technologies
- Github https://github.com/Jordanwmk/InfernoInteractive
- Github wiki https://github.com/Jordanwmk/InfernoInteractive/wiki (Documentation)
- Unity
- Google Docs https://drive.google.com/drive/folders/0Bz6IinbxfoQ3VzZiREpkdUdzTnc (Meetings)

Work Breakdown Structure



Description

Plan Document - Create a plan document with an overview of the project, including game concepts, time management, and risk management plans.
Meetings - Create a schedule for regular team meetings and record minutes in those meetings.
Distribution plan - Determine how the work is going to be distributed throughout the project, taking into account each member's strengths.
Risk assessment - Determine potential risks which may impact the project and devise methods to avoid or mitigate them.
Design document - Document elaborating on the details of the game concept and design.
1 x Boss - Create a proof-of-concept boss enemy which spawns minions to attack the player.
1 x Level - Create a proof-of-concept level environment with a timer and health bar for the player.
1 x Vehicle - Create a proof-of-concept vehicle for players to drive and fight with.
Menu - A simple menu with a high score page, start, and exit button.
Multiple levels - Create multiple levels each with different boss monsters and mechanics.
Multiple vehicles - Create multiple vehicles with unique mechanics for players to pilot.
Sound - Add sound effects and background music to the game.
Assets - Replace placeholder objects used in the prototype with actual art and model assets.
Advanced feature - Create an online challenge mode where pairs of players are matched up against other pairs of players to compete for who can finish a level in the shortest time.
Final report - Create the final report for submission.

Risk Management

Risk	Probability	Impact	Exposure	Actions	Warning Signs
Burnout	5	8	40	Making sure everyone has enough of a break every week. Avoid too much overtime work. No all-nighters.	Fatigue and absence. Low productivity and motivation.
Scope creep	1	3	3	Clearly define the required scope and minimum viable product. Document this on GitHub and only allow change if the required scope is not met.	Constant addition of new features into the project.
Project delays	3	10	30	Set deadlines with free float. Start work as early as possible. Ensure the minimum viable product for each iteration is completed before additional features are added. Ask for help when required.	Deadlines not met. Last minute sloppy work.
Getting suitable assets	3	5	15	During the design phase, perform a rough search of the freely available assets. Use top-down design. Design levels and characters around available assets.	No free assets found.
Extensive learning curve of new technology	2	6	12	Start learning as early as possible. The members who already understand the technology should help the rest. Use online support.	Slow learning and bad comprehension of technology used.
Unavailability of members due to other deadlines	4	5	20	Mention new deadlines as soon as they emerge. Reschedule project plan.	Assignments from other courses. Job interviews.
Inadequate advanced feature	3	6	18	Have a backup plan that is more complicated but also feasible.	Technological implementation seems too simple.
Integration conflicts	2	7	14	Regularly review code. Test that components fit together as soon as they have been completed. Design software architecture thoroughly.	Code doesn't seem to fit in together as a whole.
Game concept doesn't meet specifications	3	9	27	Have a thorough discussion of the concept in the early stages of the project. Ensure that new features match the requirements.	Ideas don't sufficiently satisfy requirements. Members start to doubt the game concept.
Miscommunication between members	2	7	14	Have weekly meetings about where the project currently is. Bring up any misunderstandings or doubts as early as possible. Support and keep an eye out for each other.	Members don't seem to know what everyone else is doing. Members don't have a strong sense of the current project situation.