GAME OVERVIEW

Choice of serious game concept – combination of collaboration and empathy.

Choice of game genre – puzzle and adventure with role playing elements.

Target audience – intermediate and high school students (with a stronger focus on siblings). Our target player consists of a combination of an explorer, achiever and puzzle solver.

Game mode – Dual player; both are on the same computer so as to enforce greater teamwork and empathy.

Overall goal and storyline – The storyline is based on 2 siblings, one younger than the other. The younger sibling wanders into a cave while spending time with their older brother/sister. Because of this the older sibling follows and the cave entrance collapses behind them. The siblings, armed with only a torch must delve deep into the cave to find a way to escape. Puzzles and traps must be overcome by the siblings in order for them to reach the next level and finally the exit of the cave. The aspect of the game which enforces collaboration is the torch; players are not able to view areas outside the radius of light produced and thus they must share their resources and skills effectively. There also exists another game mode which is effectively an endless runner and allows users to compete, however the collaboration aspects are still intact.

Central Character – This game will have 2 main characters as described above. Their attributes will differ, for example the older sibling will be stronger and thus does more damage to enemies and can move heavy objects. They will also have higher HP and defence attributes whereas the younger sibling will be faster, smaller (can fit into tight places) but will have less defence and HP comparatively. Along with these, each sibling will have a range of special abilities which users may select from at the start. For the older sibling these include Shout (repels monsters), Stun, Charge (attack skill) and Taunt (baits enemies). For the younger one users can choose from Shine (makes a temporary light source), Heal, Boost (short burst of speed) and Awareness (used to tell when monsters are nearby).

Game World, Mechanics, Dynamics – The environment is a cave belonging to an old civilization that contains treasure and mystery. The cave has an entrance and exit, with the objective being to unlock and advance through the exit to the next level. The cave contains traps, puzzles and monsters (large spiders, bats) that protect the treasure and block the main character's path to escape.

When entering the cave, the siblings find a torch (open flame). The torch can be shared but only one person can use the torch at a time and it creates an area of vision that both siblings share. Outside the light range of the torch is the fog of war. If the other sibling enters the fog of war, they have extremely limited vision and can be attacked by invisible monsters in the dark. This ensures that there needs to be collaboration amongst the two siblings as without the light, nothing can be explored. The person holding the torch is also vulnerable as they are unable to use any weapons or attack. This means that they are dependent on the other sibling. This interdependence is the essence of collaboration for the game which is further built up by the puzzles and challenges explored through the cave.

To advance to the next level, a final puzzle will need to be solved. Each level will have a different level-wide puzzle, which may include problems such as finding a key or collection of items, or pulling a series of levers in the correct order. Generally, these puzzles will require both players to solve which often will relate back to the player's unique abilities.

Scoring and Lives Mechanism – Siblings can each get treasure along the cave and use this as currency (for merchants). When the sibling that falls in combat, they lose a portion of their treasure that they find inside the cave. The objective is to win the game together, but also create a rivalry between both siblings on who has the most treasure at the end of the game. Lives are shared amongst the siblings. When one dies the game doesn't end due to a shared life-system. Both will respawn at a safe location at the start of the level, and decrease the total lives left.

Level Generation Plan - For both game modes, the assets such as walls, traps, rooms and monsters will be randomly placed in the map for a level. However, for the endless runner there will be no set end level, thus allowing users to play till they lose all their lives. This differs in the story mode as users must reach level 15 (last level) in order to complete the game. The puzzles will be predefined however their placement in the levels and in the game will be random. Each puzzle will belong to a difficulty and be placed appropriately in the game.

DESIGN FEAURES

- A high score screen **[worth 5%]**
- Adding sound to the game **[worth 10%]**
- Random level generation with increasing difficulty and infinite number of levels [**worth 10%**]
- Monetisation - [**worth 5%**]
- Implement a 2.5D version of the game **[worth 10%**]
- Local Multiplayer with Leaderboard **[worth 10%]**

ADVANCED FEATURES

Seed: Each time a new run of the game is started, a randomly generated string of 8 alphanumeric characters is assigned. All RNG elements in the game (level generation, monster generation etc.) will be generated based on the characters in the string. This allows each individual run to be randomly generated, but players can also replay certain levels by submitting a key that will be used for that run. Players will also be able to challenge each other on the same levels and compare completion times and scores. This will require the game to be built initially based on the seed for RNG. Everything in the game that is randomly generated will need to be integrated with the seed concept.

Daily run system: Due to the randomly generated nature of this game, a single seed could be used to determine all randomly generated elements of this game (level generation, potential random events etc.). A "daily run" system could be implemented, where each day there will be a single, completely new random seed that determines the level layout for every player. Players can compete between each other to see who gets the highest score (shown through an online leader board) in this daily run. This will require online integration of the game with a leader board, and distribution of the identical seed.

Level/puzzle editor: This will allow players to create their own levels and puzzles. They will interact with a drag and drop UI that will give them access to all assets used in the game. Players will be able to create puzzles and entire levels with this UI, as well as challenge their friends on their custom levels. This feature will require the development of a whole new UI and extend the original game by a significant amount to integrate this extra functionality.

TOOLS AND TECHNOLOGY

- Github link: https://github.com/Wombotastic/BigByte.git
- Documentation: We will use GitHub wiki for documentation
- Issue tracking: We will use GitHub issues for tracking bugs and tasks
- Game engine: The game will be a web-based game built using the Unity framework.

CHANGES TO TEAM PORTFOLIO

Our game idea and design has changed since our initial team portfolio and as we felt that it did not fulfil the necessary requirements with regards to game features/mechanics nor did we have a clear vision as a team of the direction the game would take. Rather than using the concept of empathy we now focus on the concept of social collaboration.

WORK BREAKDOWN STRUCTURE

Week	Task	Members	Description
8	Planning	Everyone	Risk assessment, game overview, features, tools and technologies.
	Initial design concepts	Everyone	Consists of game mechanics, type of levels and story in the game.
9	Design documents	NS, AB, JX, KY	Develop documents that are to be delivered showing the concept of the game. It details the mechanics in detail and the methodology of implementing certain game artifacts.
	Research on assets	AN, JB, JJ, CF	consists of finding potential objects which can be used in-game and are freely available.
	Set up technologies	Everyone	Sets up the required technologies to enable product development.
	Start prototype development	AB, NS, JX	Develop initial gameplay with single player and one level to achieve. Placeholder assets can be used.
10	Local networking	JB, KY, CF	Allow for multiplayer via LAN technologies.
	Level generation	AB, AN	The possibility of randomly generate levels. The levels consist of pre-set puzzle types and potential mini battles and shops. Level generation also should include how the puzzles should be placed so that it is fair and not frustrating to play.
	Sound engineering	JX, CF	Creating and searching for sounds to be used in the game. Incorporating the sounds at certain levels and interactions with game artifacts.
	Basic	JJ, NS	Development of playable characters and their attributes in terms of both functionality and

	characters + objects		art style
11	Multiple levels	Everyone	Further develop the prototype to enable different levels and increased variety in the game.
	Monetisation	JX, KY	Implement monetisation to allow users to buy in game currency. Design pricing schemes.
	Implement local multiplayer	JB, AN	Using the already completed local networking implementation, allow for multiple players to be in the same game.
	Scoring system + leader board	NS, JJ	A local scoring board for offline gameplay and a global leader board for each daily level comparing with other players
	Advanced features	Everyone	Total completion of advanced features which are incorporated into the game.
12	Polish	Everyone	Further visual enhancements and smoother gameplay. Also includes adding additional features which are feasible.
	Integration testing	Everyone	Tests that everything can function together and ensures that all functionality constraint is upheld.
	Deployment to web	JB, AB	Deploy the game to a webserver to allow any user to play from their browser.
	Report	Everyone	Written report about our final product which consists of the development phase, decisions made etc.

Risk	P	I	E	Actions	Warning Signs
Inability to learn new tools and platforms.	2	10	20	Pair Programming/ Learning sessions. Use of tutorials freely available online and help forums on Unity.	Slow progress, and not meeting deadlines
Scope Creep	5	7	35	Create release plan collaboratively; clarify requirements; refine often	Frequent changes to the plan
Burnout	3	7	21	Take a break. Have people rotate around roles so that they have something new to do. Don't assign people on same task whole time.	Decreased morale and productivity
Bad time management	5	8	40	Develop a plan which everyone follows. Have meetings and to check on progress and allocate help as necessary.	Work completed in last minute
Infeasibility	5	10	50	Prepare back up and mitigation measures for all course of actions. Derive simpler interactions in game to prevent hard to implement mechanics.	Too much time spent on research
Unavailability of team members.	4	1	4	Prepare work which can be completed individually. Ensure meetings are arranged such that all members are available. Communicate through different mediums to ensure that every member is aware of the current situation.	Members not showing up for meetings, asking to reschedule meetings
Lack of resources	4	8	32	Develop own resources or refine the specification to reduce burden for resources such as assets.	Too much time spent searching for resource
Bad team communication	4	8	32	Document meetings and have them in greater frequencies. Divide into groups so that people have more opportunity to speak their ideas and collaborate all the ideas when regrouping.	Quiet team members and repeating information to team members.
Lack of planning	2	10	20	Ensure plans are established as part of the development process. Plans can be developed in weekly meetings and further refined so everyone will be on the same page.	Members asking what to do
Version control failure	1	10	10	Make sure people have good understanding of GIT. Ask team members when in doubt.	Broken builds.