Deep Purple

Galactic Gladiator
Request for Proposal
Version 2.0

Document History

Version	When	Who	What
1.0	01/28/19	Kyle Hild	Initial Drafting
1.1	01/30/19	Zack	Section 1, 2
1.2	02/03/19	Gabriel	Section 3, 4
1.3	02/04/19	Oshan Karki	Section 5, 6
1.4	02/05/19	Kyle Hild	Section 7,8,9,10
2.0	02/16/19	Kyle Hild	Final Drafting

Table of Contents

1.0 Problem description
2.0 Project Objectives
3.0 Current System(s) – if any or similar systems
4.0 Intended users and their interaction with the system
5.0 Known interactions with other systems inside/outside the client organization
6.0 Known constraints to development
7.0 Project Schedule
8.0 How to Submit Proposals
9.0 Dates
10.0 Glossary of terms

1.0 Problem description / opportunity / expression of need

The market for video games is currently saturated with new games being released constantly. Many of the games being produced are very similar to previous games and have little or no replay value. For those reasons, players frequently seek out games that have an endless array of possibilities. We can see this at play in many of the most popular games that are out right now. Minecraft has procedurally generated terrain and a deeply modifiable environment. Fortnite and PUBG have random loot drops in various locations. Even very competitive games like League of Legends, Counterstrike, and Overwatch have a variety of characters and/or character builds to choose from. The main idea tying these all together is variety.

Deep Purple is looking for contractors to design a game that capitalizes on a player's need for variety and replay value. We plan to do this by introducing a fairly unexplored idea which is an alien themed, procedurally generated dungeon crawler. Currently, there are no major games that fit into this niche. Sci-fi dungeon crawlers are already rare, but we want to take it to the next level with randomness, difficulty scaling, and variety woven into every layer of the game. This game would be a fresh addition to the dungeon crawler genre and would give players a unique experience every time they played.

2.0 Project Objectives

Our objective is to make a fun and entertaining game that also provides a challenge to the player. Our broad goal is to create a third person, procedurally generated, dungeon crawler. The player has been abducted from earth by aliens and will be placed into a random level with a variety of challenges or “experiments” that they must beat in order to win their freedom. The experiments start off fairly easy, but you quickly are placed into progressively more difficult experiments until you are able to beat the final boss. Here is a list of the main elements that this game must have:

1. Procedural Level Generation
    a. Every level consists of a series of connected rooms.
    b. There must be a variety of room types such as combat, puzzle, shops, treasure, bosses, etc.
    c. Every room type shall have some randomization built into it such as randomly spawned enemies, traps, and/or loot.
    d. Every level will end in a boss fight that you need to beat to go onto the next level.
    e. There should be a variable indicating the current level which is used to calculate difficulty when generating levels.

2. Procedural Item/Loot Generation
    a. There should be a variety of items and item types which have randomly generated stats based on your current level.
    b. There should be a variety of item classes such as melee weapons, ranged weapons, consumables, tools, and specials.
    c. The weapons shall have a subtype as well based on it’s primary type. A melee weapon could be metal sword, laser sword, shield, or hammer and each subtype changes how the weapon interacts with enemies and the world.
    d. The weapons subtypes shall have unique effects on enemies. An example could be a metal sword has a random chance to cause a “bleed” effect.

3. Player
    a. The player is at the center of the screen viewed with a tilted bird's eye perspective. Camera should be in perspective mode.
    b. The player is controlled by keyboard and mouse to move about the map.
    c. The player has an inventory system which includes currency, currently equipped items, and a miscellaneous inventory

   d. The player has stats associated with them such as current health, status effects, and movement speed.
   e. If the player dies, you are sent back to the main menu.

4. Enemies
   a. The enemies should be randomly spawned during level generation.
   b. Enemies should have stats associated with them, like the player, such as health, status effects, and movement speed.
   c. Enemies should drop randomly generated loot when killed.
   d. Enemies should have a basic inventory system for equipped items. No miscellaneous storage is necessary.
   e. Enemies should automatically attack the player character when they are within a certain range
   f. There should be a variety of different enemy types. Some types of enemies might be light and fast enemies that don't do much damage, but can attack and move very quickly, tank enemies that have a lot of health but move slow, etc.

5. User Interface
   a. There should be a start screen which lets you enter the game, change options, or view help menu.
   b. In game, there should be elements that display basic info such as the level you're currently on, your current health, your currently equipped weapon, the amount of currency you have, and current status effects.
   c. In game, you should be able to pause the game and modify settings or exit the game.
   d. You should be able to open your inventory to display all the items that you currently have and change your currently equipped items.
   e. There should be some visual indicator when you're detected by an enemy or when you take damage.
   f. Whenever you hit an enemy, a health bar will appear above the enemy showing their new health.
   g. There should be a crosshair wherever the player currently has their mouse so that they can target a specific enemy or interact with other entities.
   h. Whenever a player dies or beats the game, a menu will show up with score and stats associated with that playthrough.
   i. There should be an in-game text based console that allows you to set options or modify the game while playing. Example commands could make you invincible or teleport back to the start of a level. Used for cheats or debugging.

6. Audio
    a. There should be sounds associated with clicking menu options
    b. There should be sounds associated with attacking an enemy or being attacked such as a grunt, shout, or gunshot.

3.0 Current system(s) – if any / similar systems

While we have not implemented any programming for systems in our game currently, we have a very firm grasp on the systems we would like to be implementing, as we begin to develop our game further. The basics of these systems can be seen in other games, and we would expand upon these ideas further. These games and systems are:

Binding of Isaac - PC Game

The 'Binding of Isaac' has many similar systems in which we would like to use when we implement our own game project. These systems consist of:

- Random room generation
- Random items
- Permanent death
- Scaling difficulty

Portal - PC Game

The systems that we would like to use from Portal are based more on story elements, rather than gameplay mechanics. These elements we would like to elaborate more on in our own game project are:

- The player is a test subject in which they have to go through different trials the game presents to them.
- The games main antagonist would be, the person that is conducting these experiments on the player.

4.0 Intended users and their basic interaction with the system

Users

- Gamers that are looking for a more challenging game.

Uses

- Clicking on enemies for the player character to attack them.
- Movement of the player character from the ‘w’,’a’,’s’,’d’ keys.
- Exploration of randomly generated levels each playthrough.
- Variety of different items that can be collected each playthrough.
- Fighting a boss at the end of each randomly generated level.

5.0 Known interactions with other systems within or outside of the client organization.

- Players Computer.
- DeepPurple development computers.
- Various assets package for development.

6.0 Known constraints to development

- Lack of access to 3D modelling assets and/or digital designers.
- Overall/individual complexity of algorithms used in the development process.
- Total time required for the development cycle.

7.0 Project Schedule

Date	Description
February 7, 2019	SA Demos
February 14, 2019	Code repository chosen and invitation sent, Diagrams Due
February 21, 2019	Makefile complete, and initial code should compile and run.
February 28, 2019	Have two compile paths one for testing and one for running. Then run them both
March 21, 2019	Code should look like a video game with features missing
April 25, 2019	Project Completion
May 2, 2019	Final Presentation

8.0 How To Submit Proposals

Please send all proposals using email in either a doc or pdf format. Proposals should be sent to Kyle Hild, Project Manager for Deep Purple at hild2801@vandals.uidaho.edu. Any questions should be sent and addressed to Kyle Hild at the same email.

Full Disclosure: All questions received and answered will be sent to all applicants.

9.0 Dates

Deadline for proposal submission with be February 5, 2019 at 5:00 pm (PST). The winner will be chosen on February 7, 2019. All applicants will be notified of our decision by 6:00 pm (PST) on February 8, 2019.

10.0 Glossary of terms

System: Product, service and/or system we would like to see built

Procedurally Generated: Creating an environment, enemy, or thing based on a set of rules instead of being statically generated.

Loot: Item or reward for finding secrets, defeating enemies, or solving a puzzle.

Dungeon: A closed virtually created environment containing a sequence of rooms.

Dungeon Crawler: A type of action role-playing game that contains various enemies, puzzles, and secrets while rewarding players for completing such tasks. Usually takes place in a Dungeon type scenario. See Dungeon above.

Randomness: Unpredictable occurring events.

Boss: A powerful enemy that usually requires strategy to defeat.

Tank: An enemy or player that is usually slow and can take a bit of damage to defeat.

Debugging: A way of discovering and fixing unexpected events in the game.

Ramping: Changing the scale of something gradually over time.

Several game examples were used in our proposal. I will provide a brief explanation and a source to the developers website for more detailed information

Minecraft: A lego style adventure game that one can collect resources, build, and survive.

Resource: www.minecraft.net

Fortnite: A Battle Royale type game where players are dropped onto a big map and must collect weapons and gear and be the last one to survive.

Resouce: www.epicgames.com/fortnite

PUBG:A Battle Royale type game where players are dropped onto a big map and must collect weapons and gear and be the last one to survive.

Resource: www.pubg.com/

League Of Legends: A multiplayer online battle arena which involves players defending their base from other players and computer generated characters.

Resource: https://na.leagueoflegends.com/

Counter Strike: A first person shooter game that involves two teams in opposition. One team to complete a goal, and the other team to prevent that goal from being completed.

Resource: https://blog.counter-strike.net/

Overwatch: A first person shooter and multiplayer online battle arena hybrid game involving two teams competing to complete an objective.

Resource: https://playoverwatch.com/en-us/

Binding of Isaac: A dungeon crawler utilizing procedurally generated levels with turn based combat.

Resource: http://bindingofisaac.com/ (Use the wiki link on the right for more information

Portal: A first person shooter puzzle game that utilizes a set of portals and objects to solve problems.
Resource: http://www.thinkwithportals.com/