

Honey Badger Studios

DI Trying

Request for Proposal

Version 3.0

**Document History**

|  |  |  |  |
| --- | --- | --- | --- |
| Version | Description | Author(s) | Date |
| 0.1 | Setting up and organizing layout | Hunter Casteel | 2-6-2020 |
| 1.0 | Drafting initial proposal | Paul Shannon, Hunter Casteel | 2-7-2020 |
| 2.0 | Finishing initial draft | Paul Shannon, Hunter Casteel | 2-10-2020 |
| 3.0 | Finishing final draft | Jenner Higgins, Will Johnson, Paul Shannon, Zach Benning, Hunter Casteel | 2-11-2020 |

**Table of Contents**

1. [Problem Description](#sep4vtc5ph9y)
2. [Project Objectives](#rxw1p0x14p5j)
3. [Similar Systems](#rgwp55m7v8xz)
4. [Intended Users](#n7pdwxt8ob5j)
5. [Known System Interactions](#7s6s70vooc23)
6. [Known Development Constraints](#76hwfxllrmuf)
7. [Project Schedule](#d6kvillwivh4)
8. [How to Submit Proposals](#nf9gb0hsnvpc)
9. [Glossary of Terms](#kk1v0u9gjuds)

**1.0 Problem Description**

There are already hundreds of platformer games on the market for players to purchase and enjoy, but they all feel mostly the same. Our goal for this project is to create an enjoyable, but challenging, platformer game called *DI Trying* that follows the adventures of the main character, Dianna T., after crashing on an alien planet as she attempts to restore communications with home. We at Honey Badger Studios want to break the mold and provide players with a refreshing take on the classic platformer genre. With procedurally-generated paths that constantly shift as you explore, evolving enemies, and a class-based combat system, we hope to vastly increase the replayability of our game and reward the player for their ability to adapt to an ever-shifting environment.

**2.0 Project Objectives**

Honey Badger Studios is looking for a contractor to build a Metroidvania game, *DI Trying*, that expands and improves on games similar to it such as *Metroid* and *Castlevania*. The game must include the following features:

* Environment
  + Blends between set level design and procedural generation
  + Populates areas in the level with other entities
* Player
  + Spawns in the first room when starting a level
  + Despawns when killed and respawns in the current room
  + Interacts with other entities in the level
  + Moves freely within the level
  + Uses different classes of weapons with upgradable talents
* Enemies
  + Spawns when the player gets within range and despawns when killed
  + Locates the player and attacks them
  + Scales in difficulty as the player progresses through the level
* UI
  + Displays a main menu when the game is launched
  + Allows the user to save/load a game, change settings, and view hints
  + Tracks health and inventory
* Audio
  + Plays sound effects when interacting with the UI
  + Plays sound effects when an entity performs an action
  + Plays background music on the main menu and in-game

**3.0 Similar Systems**

*Metroid* is a classic 2D, action-adventure game developed by Nintendo in 1986. The player plays as the main character, Samus Aran, and tries to reclaim the organisms that were stolen by space pirates to stop them from replicating and weaponizing them. Some key features of the game include:

* Unique mini-boss and boss fights
* Freedom to explore a level and backtrack to previously explored areas
* Upgradeable weapons that unlock new areas in a level
* Enemies that scale in difficulty as the player progresses

**4.0 Intended Users**

This game is designed for avid gamers from the ages 12 to 49 who are ready for an engaging and challenging platformer experience. *DI Trying* is meant to be a short incursion into an unforgiving world, in a metroidvania style. Players will have a plethora of modifications and choices allowing for high replayability.

**5.0 Known System Interactions**

Some known interactions with other systems inside or outside of the client organization include:

* Google Play
* Steam
* Time Server

**6.0 Known Development Constraints**

Some known constraints to developing this application include:

* The deadline and budget for the project
* The exclusive use of Unity for development
* The required objectives for the project

**7.0 Project Schedule**

|  |  |
| --- | --- |
| Description | Date |
| Proposal Submission Deadline | 2-6-2020 |
| Software Analysis Demonstrations | 2-13-2020 |
| Class Diagrams, Sequence Diagrams, Gantt Chart, and Skeleton Code Due | 2-20-2020 |
| Minimum Viable Working Product Due | 2-27-2020 |
| Initial Test Plan Due | 3-5-2020 |
| Second Gantt Chart Due | 3-26-2020 |
| Final Product Due | 4-9-2020 |
| Final Presentation | 5-7-2020 |

**8.0 How to Submit Proposals**

Proposals must be emailed in pdf format to proposals@honeybadgerstudios.com before the deadline. All proposals submitted after the deadline will be ignored. Contact support@honeybadgerstudios.com if you have any questions. All questions and their responses will be distributed to all applicants. Decisions will be made and all applicants will be notified by 5:00pm PST on February 13th, 2020.

**9.0 Glossary of Terms**

**Despawn** The action of disappearing from the environment.

**Entity** A non-static game object in the environment.

**Environment** A dimension which collaborates game rules, objectives, subject,

and theoretical aspects together as a whole to provide an interactive flow of activity.

**Metroidvania** A game genre which generally features a large interconnected

world map the player can explore, though access to parts of the world is often limited by doors or other obstacles that can only be passed once the player has acquired special items, tools, weapons or abilities within the game.

**Platformer** A game genre in which the player controlled character must jump

and climb between suspended platforms while avoiding obstacles. Environments often feature uneven terrain of varying height that must be traversed. The player often has some control over the height and distance of jumps to avoid letting their character fall to their death or miss necessary jumps.

**Procedurally Generated** A method of creating data algorithmically as opposed to manually, typically through a combination of human-generated assets and algorithms coupled with computer-generated randomness and processing power.

**Respawn** The action of reappearing after having been killed.

**Server** A computer or computer program which manages access to a

centralized resource or service in a network.

**Spawn** The action of appearing at a certain point in the game.

**Static** A state of an object that lacks movement, action, or change.

**Time Server** A remote server that the game uses for real-time events.

**UI** The means by which the user and a computer system interact, in

particular the use of input devices and software.

**Unity** Unity is a cross-platform game engine developed by Unity

Technologies, first announced and released in June 2005 at Apple Inc.'s Worldwide Developers Conference as a Mac OS X-exclusive game engine.