

BPA Software Engineering Project Preview

Game Design Document

Title: Jungle Dash: The Quest for Gas

Team: Gajus Juodvalkis, Jayson Cauble, Michael Joubert, Myles Newton

Thematic design: As can be told by the title, the game is set in the jungle. The mood of this setting is immediately expressed both visually by the main menu appearance and accompanying theme music. We include different musical pieces in each level, and they all mesh well with the setting and overall theme of the game. The game's intro cutscene shows the player, who is a trailblazing jungle trekker, have his jeep run out of gas. After running out of fuel in his vehicle, the player looks around, and in the far distance sees the arch of a gas station; his goal is now to reach the gas station, get a jug of gas, and make it back to the jeep. In the first level, we aimed to portray the overall jungle setting in the most typical way; presenting the setting in its most generic fashion at the beginning of the game seemed to be the most effective way of establishing a starting-point in terms of visual aesthetic from which we later build off of in the subsequent levels. To initially illustrate the jungle setting in this familiar and inviting way, we used a swampy forest background, mostly grassy forest-like tiles for the floor (along with a decrepit bridge and some stone temple flooring), spike booby traps, trees with vines to climb, a monkey mini-boss battle near the end of the level, and many other small visual details including fitting foliage and NPC interaction. Level one ends with the player entering an ancient cavernous temple, which is foreshadowing the visual theme of the second level. We knew that though level two's appearance was to be in stark

contrast to the first level, it still had to be relatable to the games overall setting. One of the things we did to accomplish this was reuse the spike traps from the first level, just with a new color to match the grey stone of the cave which the player explores in the second level. Additionally, we added some colorful crystal decor and plantlife in the cave to retain an overarching pallet at somewhat reminiscent of the other levels. However, the level is also slightly given its own color scheme by the deadly hot orange lava the player encounters throughout in the level. Level two concludes with the player triumphantly exiting the underground temple and stepping onto walkable clouds in the sky. The third level's unexpectedly elevated location serves as a way to surprise the player after they've completed the second level, and also conveniently advance the games plot, as the third level ends with the player touching down back on the Earth's surface at the gas station. The overall jungle setting is maintained in this level by the background decor of Toucan birds and the recurrence of mini-boss fight with the monkey enemy from the first level. After finishing level three, the player must replay slightly edited (increased difficulty, different transition areas between levels, new content, etc.) versions of each level to make it back to his Jeep, at which point the outro cutscene plays, wherein the player drives off in his no-longer-defunct Jeep.

Mechanical design: The game mechanics take inspiration from old-school arcade games along with a rogue-lite style game. In which, you have either limited numbers of lives before the game restarts or you have limited health before you have to retry. We went in favor of limited health before you have to restart from the beginning. This meaning, if on level 2 your health hits 0, you still have to restart from level 1 along with

the losing of all weapons and items gained in the run. Punishing but planned. Also, inside of the levels are shops. The shops instead of taking money, take health for payment. This allows the player to have to focus more on avoiding the spikes, traps, and enemies if they really want that upgraded weapon on level 2 or a torch on level 1. This also enables what we call high risk but high rewards items. In the level 1 shop, there is a item called "Demon's Special Brew". This item cost 99% of the player health making them a one shot to all enemies and traps, but it gives them double movement speed and double damage. This allows multiple replayability to test different items and see if you can win doing some of these more risky runs.

Workflow / team roles: Our team began work on the game in early October 2017, at which point the concept brainstorming began and lasted until the end of that month. After that, an initial collection of sprites, music, and other assets were curated for the game. Next, level one was sketched out in MS Paint and was discussed for until a precise vision for thematic and mechanical direction was reached, while the first iterations of the menu, world map, pause, and death screens were made as well. Level one was then developed in-engine, while the other screens were touched up and finalized. Levels two and three were developed by the same process as level one. Lastly, the characteristics of the re-play versions of each level were jotted down in a collaborative document and are planned to be added in after the initial versions of each level got finished. After all of that, the intro and outro cutscenes will be quickly developed while the rest of the game is given general polish, which will continue until the competition.

Though the listed team roles below are accurate, note that each member was not specifically given an assigned role; we each collaborated fairly equally on all parts of the game from a holistic point of view.

- Gajus Juodvalkis - level designer / developer / conceptualizer, programmer, asset curator
- Jayson Cauble - developer / conceptualizer for levels and menus, main programmer
- Michael Joubert - initial concept creator, designer / conceptualizer for levels and enemies, curator for music and other assets, programmer
- Myles Newton - developer / conceptualizer for levels, menus, and other features, main programmer

Code structure:

Our team decided to use the Unity Game Engine, mainly due to its abilities and efficiently in the making of our game. Most functions in our game have their own C# class file, which adds organization to our code, and allows us to drag and drop the code we need onto specific GameObjects. We also took advantage of prefabs in Unity because they allow us to make duplicate items, such as block, and edit a version of all of them at once across all levels. Finally, we are also able to animate objects in Unity, such as the player's walking animation, because of the built in animation tools that Unity allows us to use.