# Student Information

|  |  |
| --- | --- |
| Student Name: | Matthew Oliver |
| Degree Program: | BSGD |
| Team Role: | Producer/Programmer |
| Team Name: | Wholehearted Games |
| Game Name: | Shroom and Doom |
| Game High Concept: | Fungal Based Tower Defense Game |
| Team Source Control Link: | https://svn.digipen.edu/projects/wholeheartedgames |

# Weekly Work Log

|  |  |
| --- | --- |
| Week Number | 4 |
| Total Estimated Hours Contributed this Week: | 7.5hr |
| Estimated Time Spent  On Best Practices: | 1hr |
| Did you check in code or assets to the team repository this week (Yes / No)? | Yes |
| What was your overall goal for this week? | Try to get ImGui started |

## Work Tasks

|  |  |  |  |
| --- | --- | --- | --- |
| **Date** | **Task Description** | **Time Spent** | **Was this a Best Practice?** |
| 1/23 | Spent time trying to fix a bug in lua | 15min |  |
| 1/24 | Changed when transform matrices are calculated and fixed snowshroom bug | 15min |  |
| 1/24 | Weekly meeting with Caleb and Ben to discuss work | 20min |  |
| 1/25 | Weekly programmer meeting to discuss systems and changes | 15min |  |
| 1/25 | Attempting to integrate ImGui into an editor system, ran into bug | 1.5hr |  |
| 1/26 | Spent some time trying to fix issue with ImGui | 30min |  |
| 1/26 | **ALL – Pairs Programming:** Strike force meeting with Caleb and Rishabh, added player health with their help | 1hr | Yes |
| 1/27 | Team lab, fixed ImGui issue and started implementation of roller enemy | 3.5hr |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

## Personal Comments (Optional)

Are there any other comments you would like to include? If so, please enter them here:

|  |
| --- |
|  |

# Instructions

## Work Tasks

Identify all tasks that you, personally, performed as a member of a team in sophomore game. Make sure to include tasks directly contributing to your project, tasks representing Best Practices, and meetings (including team/faculty/TA meetings).

Each task must be identified using a date, one or two sentences of description that includes sufficient information for the reader to determine what you worked on, with whom you performed the work, and other relevant details, as appropriate. Providing insufficient information, such as “*implemented physics”*, will result in your report being rejected and a penalty applied to your grade until the report is submitted with enough information.

## Best Practices

Best Practices are activities, beyond the basic requirements of the course, that directly contribute to the success of your team, or other teams and individuals within the course. To receive credit for a Best Practice, the category of the Best Practice must be clearly communicated in the task description (see the list of categories below).

Example:

|  |  |  |  |
| --- | --- | --- | --- |
| **Date** | **Task Description** | **Time Spent** | **Was this a Best Practice?** |
| 10/1 | Implemented new AABB collision detection | 2 hours |  |
| 10/5 | **ALL – Participation / Conducting Playtest Session:**  Ran a playtest session over MS Teams video chat with student **Jane Doe** to test the new wall jump mechanic | 30 minutes | Yes |

## Recommended Best Practices:

**ALL – Asking for Help:**

* Clearly identify the issue addressed and the person(s) providing help (teammate, classmate, TA, and/or instructor).

**ALL – Offering Help:**

* Clearly identify the issue addressed and the person(s) receiving the help.

**ALL – Team-on-One Participation:**

* Special, 4-hour team tune-up meetings between the team and Professor Rutherford.

**ALL – Pairs Programming:**

* Work sessions during which two or more individuals actively work together to implement a specific feature.

**ALL – Code Reviews:**

* A formal process in which a specific class or feature is reviewed for code correctness and proper documentation.

**ALL – Participation in a Playtest Session for Another Team:**

* Clearly identify the team for which you participated in the playtest.

**TECH – Build Automation:**

* Creating an automated build process that allows any member of the team to create a build using the latest files from the repository.

**TECH – Build Verification Testing:**

* Creating an automated process for verifying that a change does not break the build.

**TECH – Test Automation:**

* Creating an automated process for testing the project, such as having the game play itself.

**TECH – Unit Testing:**

* Creating an automated process for testing code for correctness.

**TECH – Implementing Data Analytics**

* Data that is collected automatically by the executable, written either locally or to a server, for later analysis.

**DES – ASF Updating:**

* Clearly identify what aspects of the ASF list were updated.

**DES – A-B Testing:**

* Comparative testing of two features (which may be in separate executables).

**DES – Analysis of Data Analytics or Data Tracking:**

* Analysis of data that has been collected automatically by the executable.

**DES – UI Wire-Framing:**

* Clearly communicate which elements of the UI have been wire-framed.