|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Task** | **Allocated completion time** | **Actual completion time** | **Allocated to (the name of the group member)** | **Comments** |
| Documentation. | 3 hours | 6 hours | Jack Matters  Cordell Smith | All documentation required (cover sheet, group declaration, design and software documentation, etc). |
| Implement Bullet Physics API into project. | 30 minutes | 1 hour | Jack Matters | Had trouble with CMake, but got it working eventually. |
| Create physics world to handle all collision detection. | 1 hour | 1 hour | Jack Matters | Fairly simple, just followed a guide. |
| Create static and dynamic rigid bodies. | 1 hour | 30 minutes | Jack Matters | Fairly simple, just followed a guide. |
| Test that collision detection works. | 30 minutes | 1 hour | Jack Matters | Took longer than expected as I had trouble drawing shapes to the screen so I could visually see collision was working. |
| Add collision box to the camera. | 1 hour | 3 hours | Jack Matters | This took a lot longer than expected. Adding a force to the camera rigid body, then moving the camera to the resulting position was difficult. Got it ‘mostly’ working. |
| Refactor code to read all game object data from data structure, to create all rigid bodies. | 30 minutes | 1 hour | Jack Matters | Not hard, just long time to write code. Got it all set up, but without having the data structure implemented, the code can’t be implemented. (Ended up not implementing, but code is still there) |
| Create heightfield terrain shape from heightfield file. | 1 hour | 2 hours | Jack Matters | Rather difficult. Eventually got the data being read in and creating some kind of terrain rigid body, but without having a visible terrain, it is hard to see if it is working properly. (Ended up not implementing, but code is still there) |
| Splitting up components to delegate | 30 min | 30 min | Cordell Smith | Planning of which components needed to be completed and who was going to do which part. |
| Create class UML | 1 hour | 2 hours | Cordell Smith | Made a few changes multiple times before the UML was correct. |
| Research shaders | n/a | n/a | Cordell Smith | Research about shaders to be able to implement them in our 3D engine. |
| Implement shaders | n/a |  | Cordell Smith | Began implementing basic shader useage into our current working project so that our program didn’t need to make hard coded glDraw() calls and rather used the more efficient method of shaders vertex array objects and vertex buffer objects. |
| Change file structure | 1 hour | 1 hour | Cordell Smith | Changing the file structure of the program to maintain a consistent system. This included changing class names to help understand the purpose of the class. |
| Code refactoring | 3 hours | 3 hours | Cordell Smith | Changing parts of the code to maintain a consistent coding syntax. |
| Create a simple floor to display that the shaders work | 3 hours | 3 hours | Cordell Smith | Using the skeleton class setup I was able to rotate the simple coloured panel to make a floor, then just looped through to change their position and rendered again. |
| Research into possible file importers |  |  | Cordell Smith | I researched possible libraries that could be used to import many file types. I decided on Assimp Import Library and continued to research about how it handles loading scenes and .obj files which we intended to use in our game engine. |
| Using assimp to load model data |  |  | Cordell Smith | After implementing assimp import library, I followed tutorials to import the .obj file type by taking a file path. The relevant information was then stored in a vector data structure for further use. |
| Created the GameAssetFactory class | 1 hour | 2 hours | Cordell Smith | Created this class to fulfill the software design pattern requirements to separate the creation of different game assets without knowing the type. |
| Created the IGameObject interface class | 1 hour | 2 hours | Cordell Smith |  |
| Created object classes to inherit from the IGameObject interface class | 2 hours | 3 hours | Cordell Smith |  |
| Plan 3D engine |  |  | Cordell Smith |  |
| Plan resource loading |  |  | Cordell Smith |  |
| Doxygen Commenting | 4 hours | 5 hours | Cordell Smith | Going through each header file and adding correctly formatted doxygen comments to each class function. |
| Code cleanup | 3 hours | 2 hours | Cordell Smith | Removing any unneeded lines of code as well as smaller things such as code syntax and line indenting. |
| Debugging | n/a |  | Cordell Smith | I have spent many unknown hours debugging the program during the tasks listed above. |
| Source control code reviewing | n/a |  | Cordell Smith | Reviewing other group members code before merging with the master branch. |
| Organise meeting times | 5 minutes | 5 minutes | Cordell Smith | Remind team about meeting times and organising where to meet. |
| Assignment 2 Specific Tasks Below |  |  |  |  |
| Planned and implemented software design patterns | N/A | N/A | Cordell Smith | This was an ongoing task for the unit so no time could really be placed on it. Involved planning out how to best utilise good software design patterns, then using them in the various game engine components. |
| Refactored IGameObject class to IGameAsset class | 1.5 hours | 1.5 hours | Cordell Smith | Just to help with understanding that an object can be an asset, but not the other way around. This created some confusion at the start so had to be changed. There were also some changes to the code which involved the loading asset functionality. |
| Created terrain base class | 2 hours | 2 hours | Cordell Smith | This class was used as an interface class for other terrain generation classes, incorporating the factory method pattern which utilises virtual methods and functions. |
| Added bruteforce class | 10 hours | 20 hours | Cordell Smith | This class was used to generate the terrain procedurally using the brute force method. The class inherited from the base class terrain but had other helper functions of its own. The most tricky part of this was understanding how the height field data was read into a char array. Then using this data to create a triangle mesh with different heights according to that byte data. The bruteforce class needed its own way to create a mesh and then pass it to the 3D renderer for its VAO to be created and used later on. |
| Created custom height maps using Gimp | 1 hour | 2 hours | Cordell Smith | The concept to creating a custom height map was not the issue in this case and probably shouldn't have taken so long however, there were a few specific setting that needed to be set in order for the .raw image to be properly read into a char array. This was that it needed to be a grayscale image when defining the settings of the image and when saving as a .data file type (which is .raw in Gimp) you needed to select the Planar option that read it in as RRR GGG BBB. Once you have the .data file, you can change the extension to .raw manually in the file structure by renaming it. |
| Implemented correct texturing of models and terrains | 4 hours | 6 hours | Cordell Smith | This was not worked on in one chunk but rather come back to here and there in order to try fix. The VAO data was not correctly read in which meant the UV mapping was using Normals or Colour data to map textures. This was due to the stride being off and defining the size of the buffer. |
| Implemented 3D engine | 4 hours | 4 hours | Cordell Smith | The 3D engine component of the engine was successfully abstracted into its own file structure and classes. Initially i was going to use the factory method to make a base 3D engine class for the creation and use of different 3D facade classes such as openGL and Direct3D, however due to time constraints this was not possible. |
| Created openGL renderer class | 2 hours | 3 hours | Cordell Smith | The openGL renderer will/can inherit from the base class IRenderer and takes in a Model variable to either Prepare() or Render(). The prepare function creates VAOs and loads the data into VBOs for further use. Then the render function accesses the specific Model VAO and VBO, does any further uniform calculations and changes, then renders it to the screen. This uses modern openGL shaders. |
| Refactored camera class | 4 hours | 4 hours | Cordell Smith | This also involved researching how to create a 3rd person camera. Added more functionality to be able to do so. |
| Created 3rd person camera | 4 hours | 4 hours | Cordell Smith | Putting what I had researched into action wasn't as easy as intended. Applying basic trigonometric functions to the cameras view matrix was difficult but I got there in the end. |
| Refactored input engine | 3 hours | 5 hours | Cordell Smith | Changing the functionality of what specific keyboard or mouse inputs would do. This had to be reworked with the 3rd person camera. |
| Started .md2 model animation | N/A | 2 hours | Cordell Smith | Did research on different model types that store animation data and how to read this in. Started on a Md2 class to do so however never completed this component due to time. |
| Added initial collision detection between objects and terrain | 1 hour | 30 min | Cordell Smith | This only required finding the height of the terrain at any specific point and passing an adjusted value of that to the object to be above the terrain. |
| Code refactoring | N/A | N/A | Cordell Smith | This involved removing any unnecessary code from the engine and was completed throughout the assignment. |
| Code optimizations | N/A | N/A | Cordell Smith | Optimizing some pieces of code to work better. This varied from loading model data to rendering. |
| Doxy Commented files | 2 hours | 2 hours | Cordell Smith | Applied doxy commenting to header files that were missing them. |
| Debugging | N/A | N/A | Cordell Smith | I have spent many unknown hours debugging the program during the tasks listed above. |
| Fixed issue to run on all systems | N/A | 3 hours | Cordell Smith | As the compiled program did not run on all systems, I spent a lot of time at uni researching on how to fix this, applying these methods and repeatedly compiling and moving to multiple computers to test if it worked. In the end I got it to work on all systems. |
| Source control code reviewing | N/A | N/A | Cordell Smith | Reviewing other group members code before merging with the master branch. |
| Organise meeting times | 5 minutes | 5 minutes | Cordell Smith | Remind team about meeting times and organising where to meet. |
| Organised final game engine presentation | 1 hour | 1 hour | Cordell Smith | Involved creating powerpoint slides detailing what was going to be talked about and delegating speaking roles. Also made the presentation look nice with images and styles. |
| Presented the game engine to faculty and students | 20 min | 20 min | Cordell Smith  Jack Matters | Presented to the rest of the unit and faculty. Also answered questions from audience. |
| Write texture manager class | 2 hours | 4 hours | Jack Matters | Found an online example and followed that. Altered to fit the needs of the project. |
| Write lua class | 1 hour | 4 hours | Jack Matters | Took far longer than expected. Tried to find a good way to implement, but ended up going with something that works so could move on. Would like to rewrite this class given the time. |
| Write lua script for textures | 30 minutes | 30 minutes | Jack Matters | Fairly simple, no issues faced. |
| Write lua script for window initialization | 30 minutes | 30 minutes | Jack Matters | Fairly simple, no issues faced. |
| Write lua script for model initialization | 30 minutes | 30 minutes | Jack Matters | Fairly simple, no issues faced. |
| Write lua script for height map initialization | 30 minutes | 30 minutes | Jack Matters | Fairly simple, no issues faced. |
| Write lua script for camera initialization | 30 minutes | 30 minutes | Jack Matters | Fairly simple, no issues faced. |
| Handle reading of all lua scripts, and storing of data | 1 hour | 5 hours | Jack Matters | Trying to make data structs to hold all the data, and then access the data, took more time than expected. |
| Write structs for holding of lua data | 30 minutes | 1 hour | Jack Matters | After working out how to store data, the implementation of the structs was fairly simple. |
| Write FSM | 2 hours | 2 hours | Jack Matters | Followed lecture slides. No issues faced. This involved the writing of several classes. |
| Write AI class that uses FSM | 2 hours | 2 hours | Jack Matters | Followed lecture slides. No issues faced. |
| Create different AI states | 1 hour | 1 hours | Jack Matters | Fairly simple, no issues faced. Not too in-depth. Got some simple AI to demonstrate it, but couldn’t make it better due to time constraints. |
| Create AI in game | 1 hour | 3 hours | Jack Matters | Issues with AI not acting independently. All move to same waypoint. Spent a while trying to fix, but was unable to fix in time for final submission. |
| Add collision to models and player | 1 hour | 1 hour | Jack Matters | Fairly simple, just had to alter previous code from assignment 1. |
| Created a structs class for useful structs | 30 minutes | 30 minutes | Jack Matters | Fairly simple, no issues faced. |
| Doxygen commented classes I wrote | N/A | N/A | Jack Matters | Did this while writing the classes. |
| Debugged code | N/A | N/A | Jack Matters | Debugged code whenever it wouldn’t compile. |