**Deliverable 4 – Final Release**

Submit as a compressed (.zip) file on MUOnline containing your (minimized!) game project. Only 1 needs to be submitted per group (except a brief writeup for part 3).

See page 2 for final project requirements & rules.

|  |  |  |
| --- | --- | --- |
| 1. | Complete your proposed game project and all of the requirements listed on the next page. Also attempt to implement appropriate and feasible feedback received from playtesters or if I provide other specific feedback in class. | 60% |
|  |  |  |
| 2. | As a group, write a Post Mortem about your project and the process of making it. Include a description of who did what, what problems you had, what went right and wrong, what you would do differently, and what you would continue to work on if you could. | 10% |
|  |  |  |
| 3. | Individually, submit a ½-1 page summary of your group’s performance and if you have any praise or comments or concerns about your groupmates. |  |
|  |  |  |
| 4. | Package the project as an .exe, at least for Windows. During the scheduled final ‘exam’ time (Tuesday May 7, 10:15AM) we will have a Gallery Walk, which just means you will walk around and play/comment on each other’s games. Note that means you need to attend for the final, unless you let me know ahead of time if you can’t make it for some reason. | 10% |

**The minimal requirements of the final game are as follows:**

|  |  |  |
| --- | --- | --- |
| 1. | 3 Levels, 1 Main Menu and 2 game levels. Some feature should persist between the game levels, like an item you pick up in level 1 will still be in your inventory in level 2. | 10% |
|  |  |  |
| 2. | A way to win and a way to lose. Indicate when the payer wins or loses, with widgets or other levels. Players should be able to play again without re-running the game. | 10% |
|  |  |  |
| 3. | Instructions, credits, and some lore or a brief story provided in game such-as through widgets with text. | 10% |
|  |  |  |
| 4. | 1 mechanic that uses an interactive in-game widget, such as a shop menu, inventory screen, unit recruitment window, etc. | 10% |
|  |  |  |
| 5. | 1 static particle system that just plays on its own, and 1 dynamic particle system that is activated through gameplay events. | 10% |
|  |  |  |
| 6. | A Dynamic material that changes based on gameplay events. | 10% |
|  |  |  |
| 7. | Background music and multiple sound effects. At least 1 3D sound effect. | 10% |
|  |  |  |
| 8. | At least 2 different types of NPC’s (Pawn or Character as the base class). At least one should use pathfinding via a Nav Mesh and movement component. | 10% |
|  |  |  |
| 9. | A mechanic that uses the mouse to click and activate something. This can be a hit-scan weapon, a building that can be clicked on to make a widget appear such as your widget from requirement 4, etc. Buttons in widgets do not count since they have this built-in. | 10% |
|  |  |  |
| 10. | At least one blueprint derived from a class written in C++. The C++ class should contain at least one UFUNCTION and one UPROPERTY that can be called and used in the child blueprint. | 10% |

**The following rules will incur a penalty if broken:**

|  |  |  |
| --- | --- | --- |
| 1. | Minimize the project upon submission (including use of the Migrate tool). | -5% |
|  |  |  |
| 2. | No errors reported in the message log after playing (warnings are ok). | -10% |
|  |  |  |
| 3. | Runs at reasonable (>30) framerate. | -5% |
|  |  |  |
| 4. | Comments in code/blueprints/materials indicating who worked on it. | -5% |
|  |  |  |
| 5. | Credits and attributions for any assets you use but didn’t make. | -10% |