In this assignment, you will build on top of the design patterns to expand on Unity3D's functionality by developing tools and plugins.

Purpose: In this assignment, you will implement plugins and use DLLs to optimize and customize your game (your course project game or GDW).

This assignment is graded out of 100 points. The assignment is worth 10% of your final grade.

This assignment must be done in the course project groups. Please note that the larger the team, the more plugins are required.

**Mandatory for grading. Without these, the assignment will not be graded:**
- Base build with singleton and observer patterns
- Proof of repository usage
- Proof of project management tool usage
- Statement of contributions as per Appendix A
- Working repository
- Working build

**PART 1:  Design Patterns /45**
o   Leverage the roles of the team members and use the following design patterns.
o   The purpose for team members to contribute by implementing various design patterns is to increase the breadth and functionality of your game.
o   Game projects with one person can implement one design pattern.
o   Game projects with two people can implement two design patterns.
o   Larger groups are required to implement more design patterns according to the number of team members (a design pattern for each member). This could be done by implementing variations of command, factory, or other design patterns that you believe will enhance your game. For example, you can have two command design patterns, two factory, and any two others of your choice.

- **Command Design Pattern**
- Use the Command design pattern to undo/redo options.
- Other examples:
  o   Character customization
  o   World editing
  o   Custom button inputs
  o   Undo movements in strategy games
- **Factory Design Pattern**
- Use the Factory design pattern to spawn objects in-game (and/or create a "enemy spawner")
- Other examples:
  o   Creating variations of game assets such as enemies, character variations, or objects with various different properties
- **Implement an additional Design Pattern of your choosing.**

- **PART 2: Pseudocode/UML /15**
  o   Please choose one of the patterns developed in Part 1, and present it in Pseudocode, UML, or flowchart format

- o The option you choose should clearly communicate how the design pattern was implemented and how it works

- **PART 3: DLL Customization / 30**
  - o Use two DLLs of your choice.
  - o Game projects with one or two members can implement one DLL.
  - o Examples:
    - ▪ Saving / Loading of your Game Level with the use of a C++ DLL
      - ▪ Include a simple user interface to enable this feature (this could be simple buttons and labels or a full-blown menu)
    - ▪ Language translation
    - ▪ Modding: characters, environment, objects

- **PART 4: Demo Video /10**
  - ○ You will be creating a Demo Video for each Assignment part
  - ○ The first (and only) Slide of your Slide Deck must include a current image of you and your team members (no avatars allowed) that is displayed appropriately on the page. You must also include your Full Name(s), Student ID(s), the Course Code, Course Name, and your Assignment information
  - ○ You will demonstrate your app's functionality. Your UI must be clearly visible.
  - ○ You will describe the code, algorithms, math, and logic behind the implementation
  - ○ The sound for your Video must be at an appropriate level so that your voice may be clearly heard. Your screen should be clearly visible
  - ○ Your Short Video should run no more than 5 minutes
- **TO SUBMIT:**
  - o Submit a link to your GitHub repository (cleaned project, no garbage files) with executable
  - o Video link to a 5-minute presentation

- **NOTE**:
  - o Acknowledge any external, internet, or any code other than yours. Failure to do so will result in no marks for the assignment. If using code from others, please provide a justification for this and discuss your contributions to it
  - o If Unity assets or any sort of third-party plugin or code are used, the grade will depend on your contributions. For example, using existing assets or third-party code as is, without contributions, will render the item non-existent. Grades on items with contributions will depend on the explanation and quality of the statements.
  - o If we have any questions regarding the assignment, we will reach out to you for clarifications
  - o Deadlines are final; please consider repository committing times

  - o Note: All members must contribute equally. If this is not the case, the contributions will be used to calculate the final grade. For example, a group of four people should have

equal contributions of 25%, and those with less than that will get a grade proportional to it.

## Rubric /100

| TOTAL: /100 | Non-existent | Exists (25%) | Basic (50%) | Decent (70%) | Well Thought (85%) | Above and Beyond (100%) |
|---|---|---|---|---|---|---|
| **Required for grading** | Not present | | | | | Present |
| **Command Design Pattern /15** | The Command Pattern does not exist, or the feature does not work.<br><br>Explanations require extensive clarification. | Implementation does not adhere to Command Pattern and/or throws errors or has poor performance<br><br>Explanations require some clarification. | Implementation somewhat adheres to Command Pattern.<br><br>Explanations require some clarification. | Implementation adheres to Command Pattern. The contribution requires some clarifications. | Implementation adheres to Command Pattern and is well organized.<br><br>Contributions require minimum clarification. | Implementation adheres to command patterns, is well organized, and is extensible.<br><br>Contributions do not require clarifications. |
| **Factory Design Pattern /15** | The Factory Pattern does not exist, or the feature does not work.<br><br>Explanations require extensive clarification. | Implementation does not adhere to Factory Pattern, throws errors, or has poor performance<br><br>Explanations require some clarification. | Implementation somewhat adheres to Factory Pattern.<br><br>Explanations require some clarification. | Implementation adheres to Factory Pattern.<br><br>The contribution requires some clarifications. | Implementation adheres to Factory Pattern and is well organized.<br><br>Contributions require minimum clarification. | Implementation adheres to Factory Pattern, is well organized, and is extensible.<br><br>Contributions do not require clarifications. |
| **Optional Design Pattern /15** | The design pattern does not exist, or the feature does not work.<br><br>Explanations require extensive clarification. | Implementation does not really adhere to the chosen design pattern, throws errors, or has poor performance.<br><br>Explanations require some clarifications | Implementation somewhat adheres to the design pattern but has limited functionality.<br><br>Explanations require some clarification. | Implementation adheres to the design pattern.<br><br>The contribution requires some clarifications. | Implementation adheres to design patterns and is well organized.<br><br>Contributions require minimum clarification. | Implementation adheres to design patterns, is well organized, and is extensible.<br><br>Contributions do not require clarifications |
| **Pseudocode /15** | Pseudocode, UML Diagram, or flowchart does not exist. | Pseudocode, UML Diagram, or flowchart does not describe the pattern, is difficult to understand, or is missing various elements. | Pseudocode, UML Diagram, or flowchart somewhat describes the pattern but could be organized better. | Pseudocode, UML Diagram, or flowchart describes the pattern and is clear and understandable. | Pseudocode, UML Diagram, or flowchart could be given to another developer and would quickly teach them how to use the system. | Pseudocode, UML Diagram, or flowchart could be given to another developer and would quickly teach them how to use the system. The diagram is pleasing to look at. |

| C++ DLL /15 | C++ DLL does not exist, or the feature does not work. | The saving/loading feature throws errors or has poor performance. | The saving/loading feature works but is missing controls and usability. | The saving/loading feature works well. The related UI elements enable feature use. | The saving/loading feature works well. The related UI elements are organized and well thought out. | The saving/loading feature works flawlessly. The related UI elements are organized, well thought out, and aesthetically pleasing. |
|---|---|---|---|---|---|---|
| C++ DLL /15 | C++ DLL does not exist, or the feature does not work. | The saving/loading feature throws errors or has poor performance. | The saving/loading feature works but is missing controls and usability. | The saving/loading feature works well. The related UI elements enable feature use. | The saving/loading feature works well. The related UI elements are organized and well thought out. | The saving/loading feature works flawlessly. The related UI elements are organized, well thought out, and aesthetically pleasing. |
| Demo Video /10 | Demo Video is missing or includes sound and visuals that make it impossible for the viewer to hear the presenter(s) and/or see the screen. | Demo Video is incomplete and does not describe the implementation of design patterns. Demo Video includes sound and visuals that make it difficult for the viewer to hear the presenter(s) and/or see the screen. | Demo Video is partially complete, is missing a section, or does not fully describe the implementation of each design pattern. Demo Video includes sound and visuals that make it sufficient for the viewer to hear the presenter(s) and/or see the screen. | Demo Video is complete and includes an appropriate Title Page and describes the implementation of each design pattern. Demo Video includes sound and visuals that are good. | Demo Video is complete and includes an appropriate Title Page, and clearly describes the implementation of each design pattern. Demo Video includes sound and visuals that are very good and includes an organized structure. | Demo Video is complete and includes an appropriate Title Page, and clearly describes the implementation of each design pattern. Demo Video is also pleasing to look at and is well structured. |

APPENDIX A

Statement of Contributions

We. Vincent Lau, Matthew North, and Kyle Disante, certify the following:

Contributions

| # | Last name | Role | Contribution % | Contribution description | Signature |
|---|-----------|------|----------------|--------------------------|-----------|
| 1 | Lau | Game Designer | 33% | Enemy Spawner (Factor Pattern) | VL |
| 2 | North | Programmer | 33% | Singleton Pattern and Observer Pattern | MN |
| 3 | Disante | Programmer | 33% | Command Pattern, Save/ Load DLL, and UML Diagram | KD |

We additionally certify that the following are original (original means that it was developed with the purpose of solving a problem without fully relying on code found or provided in the labs) contributions of our work:

|  |  |  |  |
|--|--|--|--|
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

We also certify that that the following third-party assets were used:

| Name of asset | Source and licensing | Reason for use |
|---|---|---|
| Spikeball | quaternius ultimateplatformer \| CC0 | Easy to grab asset |
| | | |
| | | |
| | | |
| | | |
| | | |
| | | |

Finally, we certify that the following third-party code was used:

| Name of asset | Source and licensing | Reason for use | Your contribution |
|---|---|---|---|
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |