COMP 303

Winter 2025

Milestone 4

Due: Tuesday, April 15^{th} by 11:59 p.m.

Introduction

You must make your final Git commit by this date. Your repository will be cloned soon after midnight for final evaluation. This milestone will be worth 12% of your grade.

Grading criteria

One grade will be assigned per group, though individual penalties may apply. The following criteria will be used to calculate the grade for this milestone.

Front-end

- How interactive is it? Can the player walk up to objects and press the space bar (or other keys) to interact with them? Are pressure plates, menus, or other such techniques employed? We don't want a lot of chat commands.
- Does it explain to the user what they can/should do in the space? Or would the user be confused upon entering?
- Are the graphics representative of the actual objects? Has effort been put into choosing or making proper graphics?
- Was the presence of multiple players in the space handled in some way? E.g., is there any support for a collaborative or competitive experience?

Back-end

- Are one or more `Map` objects correctly defined?
- Are `MapObjects` created as and when appropriate? MapObjects must be created to appear on the grid.
- Are the `player_entered` and `player_interacted` methods (or similar such methods) used as and when appropriate?
- Is database state set and retrieved appropriately, using `get_state`/`set_state` on `Player` and/or `Map` objects? We don't want state to be lost when a player leaves your space or when the server restarts.
- Was the existing `Player` class used? We don't want any duplicate classes.
- Was `NPC` subclassed as and when appropriate?
- Were `Message` objects correctly created and returned appropriately from the various methods?

Class diagram

- Does it all contain all classes defined in the code?
- Are there special annotations for the design patterns?

Code design

- Good use case and implementation for your four design patterns
- Absence of any anti-patterns
- Proper encapsulation and information hiding
- Consistent use of design by contract (preconditions and asserts)
- Minimal state space for objects
- Consistent use of type annotations and docstrings
- Unit tests for important pieces of behaviour

Besides the above factors, penalties will apply to a person's grade in the following situations:

- if features mentioned in the proposal were not implemented, leading to a lower effort/complexity project.
- if the requested changes listed in the TA's review of the proposal were not incorporated.
- if a team member's Git contributions during the project were minimal or none, or very low compared to other team members.