

## COMP2100/6442 Group Project | Text-based RPG, game engine & tester

**Last modified date:** 29 Mar 2021

**Assessment weight:** This project is **worth 20% of final mark**.

**Due Date:** Friday 11pm, Week 11 (no late submission is allowed)

**Project Demo:** Week 12 (details about demo to be announced later)

### Project Tasks

In the group project, each team will be required to develop a text-based role-playing game (RPG), with a game engine and automatic game tester. A simple example of a text-based RPG will be a treasure hunting game. A player will be given different movement options (e.g., walk forward/left/right) to explore in a maze. A user can control the movement of the player by inputting proper text-based commands. A more advanced example of text-based RPG is Dragon & Dungeons. An online AI-based Dragon & Dungeon can be found at: <https://play.aidungeon.io>

There are three major components in the project

1. *Game engine:* Game designers should be able to customize various game settings and environments, such as game levels, maps, stories, plots, etc, by editing suitable XML/JSON game configuration files. The game engine can generate an RPG game from the game configuration files. For example, in a treasure hunting game, game designers can customize the game maps in the configuration files.
2. *Automatic game tester:* The game tester can generate suitable user input test cases to test the game from the game configuration files in the game design stage. The game tester should also run the game automatically and output the testing results to game designers. For example, in a treasure hunting game, the game tester should generate test cases in terms of player movements in a maze to test if there is an error in the game plot.
3. *Game:* You should also design an exciting playable RPG to showcase the abilities of your game engine and game tester.

Examples of RPGs:

- Treasure hunting
- Dragon & Dungeon
- Zork
- Pokémon

You are encouraged to explore other novel RPGs.

```
Living Room                               Score: 15    Moves: 16
east is a small window which is open.
On the table is an elongated brown sack, smelling of hot peppers.
A bottle is sitting on the table.
The glass bottle contains:
  A quantity of water

>take bottle
Taken.

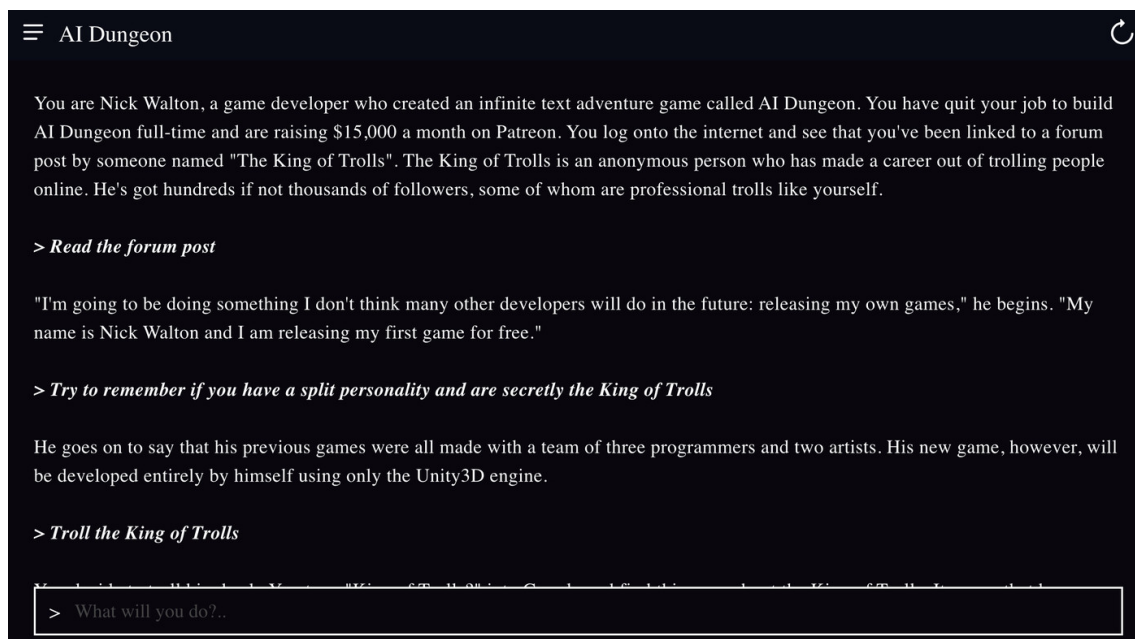
>west
Living Room
You are in the living room. There is a doorway to the east, a wooden door with
strange gothic lettering to the west, which appears to be nailed shut, a
trophy case, and a large oriental rug in the center of the room.
Above the trophy case hangs an elvish sword of great antiquity.
A battery-powered brass lantern is on the trophy case.

>take lantern
Taken.

>take sword
Taken.

>
```

Zork Screenshot



AI Dungeon Screenshot

## Marking Criteria

### Feature Marks (70%)

- Game engine features (35%)
  - Complexity of supported customizable game features in the game engine (15%)
  - Flexibility, expressiveness, readability of configuration files (5%)

- Error checking in configuration files (5%)
- Advanced features: AI features, efficient algorithm designs, GUI, natural language processing, etc. (10%)
- Automatic game tester features (20%)
  - Complexity of automatic test case generation (10%)
  - Completeness of automatic testing process (5%)
  - Usefulness of feedback from automatic testing (5%)
- Game features (15%)
  - Design of game play (5%)
  - Complexity and expressiveness of user inputs (5%)
  - Playability of the game (5%)

#### Overall Design Marks (30%)

- We consider a set of holistic areas, including, but not limited to:
  - Completeness and readability of documentations
  - Development plans: design documents, meeting minutes, GitLab milestones
  - Coverage of testing: Test cases for game engine and game tester
  - Quality and organization of code, applications of design patterns and design by contract
  - Applications of knowledge in this course (e.g., data structures, algorithms, parsing, performance, etc.)
  - Creativity and novelty

## Submission Requirements

- GitLab repo of code (must have at least 2 months of Git commit histories)
- GitLab wiki using markdown for documentation
- Team structure and roles
- Project summary
  - List of features in your game engine with appropriate examples and screenshots
  - List of features in your game tester with appropriate examples and screenshots
  - Description of your game with appropriate screenshots
  - Optional demo videos
- Summary on applications of knowledge in this course (e.g., data structures, algorithms, parsing, performance, design patterns, design by contract, etc.)
- Team meeting minutes (at least 3)
  - Your first meeting should develop a clear plan of how the work will be divided, documented in meeting minutes.
- Individual statement of originality (template to be provided)
- Individual assessment of team member contributions (template to be provided)

## Due Date and Late Submission Policy

The project is due on the Friday of **Week 11 at 11:59 pm**. As the project will be done iteratively over the second half of the semester, every group should have something that will gain a pass mark well before the due date. No late submissions shall be allowed. Whatever your group has done up until the due date will be assessed.

## Originality

**The project must be your own original work.** If you make use of any code that is not your own it must be clearly referenced. This can be done by adding a simple comment next to the code stating where you obtained the code from. You must also add this to the statement of originality document. This is **very** important, as any breach of this needs to be investigated and reported. You are much better off not doing this project then copying a small part of code and risking academic misconduct. Remember, we are assessing your code, not someone else's code.

Every person in the group is responsible for the originality of every part of the project (regardless of who actually wrote or contributed to it). Any significant break of the [academic honesty and plagiarism policy](#) will result in the entire group receiving zero mark for the group project.

## Teamwork

To ensure that each member has made fair contributions to the project, each student needs to provide an individual assessment of each team members' contributions. If a team member is found to have made much smaller contributions than those of other team members, mark reductions will be applied.