

SE2250b – Software Construction

Project: Role Playing Game

Objective

This project provides hands-on experience on various aspects of software construction including practical experience with software implementation and testing. It will give you an opportunity to improve your software development practices and to try the iterative model of the software development life cycle through three phases of the project.

Phase 2:

Deadlines:

Section 002: 10:30 am, Tuesday, March 17th, 2020

Section 003: 10:30 am, Monday, March 16th, 2020

This phase will continue the game you described and started in the Phase 1. From the Phase 1, you should have the complete look of your level 1 virtual world. After the Phase 2, you should have working Level 1.

NOTE: Proper coding practices must be used. Examples include but are not limited to the following: naming conventions, project organization, proper code formatting, meaningful comments, clean code (no old commented out code or unnecessary code), and any other topics discussed in lectures or labs. The object-oriented design must be used throughout the project.

Requirements:

Features to develop in this phase correspond to your Level 1 and include:

- **Player Movement:** You player must move in your virtual environment and needs to be able to explore that environment. Player movement must be adequate for your environment: for example, player should not go through obstacles such as trees or walls. In addition to a simple movement such as walking, there must be at least one other type of movement: examples include, but are not limited to jumping, leaping, and teleporting. The edges of your world should be meaningfully handled. For example, if your environment is a forest, you need to meaningfully handle the player getting to the end of the forest. This does not mean that you must keep generating new spaces, but the player cannot just exit your virtual world.

WESTERN UNIVERSITY
FACULTY OF ENGINEERING
DEPARTMENT OF ELECTRICAL AND COMPUTER ENGINEERING

- Combat. You can have fighting or shooting style combat, but you can also have other types of combat such as skill/knowledge-based ones. Your game should include:
 - At least two different types of combat. If you are doing a fighting game, maybe one way of fighting enemies is shooting and another one is fighting with a sword. However, stabbing with a knife and stabbing with a sword are the same type of combat as they represent the same movement.
 - Enemies/non-player characters. If you are doing a fighting game, your combat will include enemies. If you are doing a non-fighting game, this includes non-player characters. Some of your enemies/non-player characters must move in your virtual world.
 - The player and enemies taking damage. There must be a way for the player to defeat enemies. At this project Phase, this can be as simple as the enemy getting destroyed after a number of stabs. You will implement player progression in the Phase 3.
- Character customization. There must be a way to customize the player. This cannot be only the player's look but needs to include other elements such as a different skill, or a different fighting style. The same fighting style cannot count as a character customization and as one of the two required types of combat (described above). The difference should include more than just a simple change: for example, a faster player is just a simple change. This customization needs to affect your level 1 and you need to be able to demonstrate difference during gameplay.

Elements your submission must include:

- At least one git submissions from each team member. The submissions must demonstrate the collaboration and contributions of different team members. (Difference between submissions cannot be just comments or small code changes)
- Object-oriented principles must be followed. Your implementation must include at least one use of inheritance (or interfaces), singleton, and C# properties. Other object-oriented techniques should be used as required by your specific game.

Deliverables for Stage 2:

- Functionality demonstrated to a TA during the lab. All group members must be present for the demo.
- All game files uploaded to OWL. This should contain all the files needed to run your game from UNITY.
- Only one group member should submit the project. Names of all group members should be included with the OWL submission.
- Code submitted to GIT. Your GIT repository name and structure must follow instructions from the Git tutorial.
- In OWL submission provide a reference to the GIT submission corresponding to Phase 2: Project, Phase 2: git submission xyz

For no demo, 50% of the available mark will be deducted. Each group member must be present during the demo. Not being present for the demo without justifiable reason (providing documentation to Undergraduate Office or self-reported absence) will result in a deduction of 50% of the available mark for that student.