Design Document

Version 1.1 – 2023.11.05

Created 2023.10.24

Project Name

Andrew Ding
Alan Wang
Wenjun He (William)
Yingfan Hu (Daniel)

GitLab Repository:

https://mcsscm.utm.utoronto.ca/csc207_20239/group_59

SECTION 1: PROJECT IDENTIFICATION

- The motivation behind this project is to further utilizing design concepts taught in this course to perfect our version of the adventure game introduced. We are able to work on this project as a group to follow the design methods such as waterfall and scrums.
- This will enhance and add functionality in the game by enhancing existing features such as objects and images to make the game more immersive. It will add functionality by adding trolls, map, start/ setting menu to add difficulty, more immersiveness and accessibility.

SECTION 2: USER STORIES

User Stories

Name	ID	Owner	Description	Implementation Details	Priority	Effort
Map	1.1	Alan	As an adventure gamer, I want to be able to move around the map without losing track of where I am so that I can backtrack.	There will be two layers, one of which is the map and the other which is completely black, with the black part on top. Parts of the black frame will be toggled transparent to show the map underneath as the player traverses through rooms.	1	2
Fancy GUI	1.2	Alan	As a developer, I want to make the background images full screen and interactable objects pop up when you hover over them so that my users can feel more immersed in the game.	When the user hovers over objects they will be replaced with a bigger version of the object to give the illusion of a pop up. As for the images, the frame containing the image will be expanded to the entire screen.	2	1
Menu	1.3	Andrew	As a developer, I want to add a menu to the start of the game so that the user isn't pushed directly into the game and can change	A setting class will consist of all the options for users to change the game to their comfort. A several helper class will	1	2

			Difficulty, settings etc before playing.	be created to handle the specific implementation.		
Shop	1.4	Daniel	As a user, I want to use the coins I collected from games/defeating trolls to purchase gear for my character such as armor to increase defence, or power ups to increase health/attack, so that I can counteract the difficult challenges I face.	GUI that can be universally accessed where users can buy/sell items to obtain objects that provide user status effects. There will be a hashmap that maps the objects to their quantity in the shop.	1	2
Ending Select	1.5	Daniel	As a user who has completed the games >= 1 times, I wish to view the endings such that I can experience the glory of defeating the final boss/ save my race from extinction again.	GUI that can be universally accessed where users can view their ending sequences that are unlocked. Ending sequences should be a sequence of forced rooms with text giving narration. The interface design is similar to load game page	3	2
Home-page	1.6	William	As a user who plays game for leisure, I don't want the start the game immediately, so that I can choose to continue game, start a new one, or change difficulty and settings to enhance my satisfaction when playing games.	Initialise UI with a homepage with start new game with chosen difficulty, continue game, settings menu for change settings	1	1

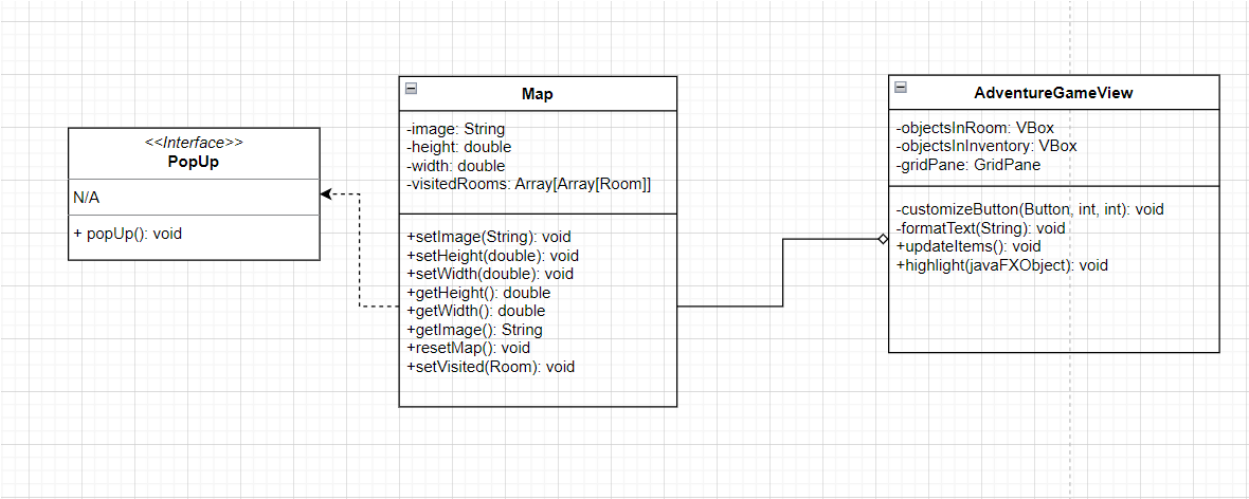
Acceptance Criterias

Name	ID	Description
Map	1.1	The location of the player should be displayed accurately. The player will always be positioned in the center of a room. The range of the map should encompass every room. The map should be oriented with north facing up. The map should have layers if there exist layers.
Fancy GUI	1.2	The object that the user's mouse is hovering over will appear to “popUp” and after hovering over it for a couple seconds will show a description of the object. The room itself will cover the entire screen with the objects scattered around the entire room.
Setting	1.3	Given that I clicked the setting, I can change the volume, set the contract display, and turn on the audio description for the game. Given that I am with low hearing, i can turn on subtitle and descriptions in the text, also, I could see the description of music in test
Shop	1.4	Given that I click the SHOP button or shop command, I can access the shop. I am able to sell an object player has in their inventory for funds. I am able to buy an object if and only if player has enough funds and there is stock available for that object. If the shop is out of stock, there will be an error message that notifies the player.
Ending Select	1.5	Given that I click the Ending select button, I am able to access the ending sequences that I have unlocked and view them. I am able to see the locked endings that exist behind access. I am able to access the endings I have unlocked only in this game file. When I start a new game, all settings should be locked in the ending select interface.
Homepage	1.6	Given I run the program, the GUI first starts with a homepage, Then I can choose to start new game/continue game/change settings/shop, etc. If start new game is selected, I am able to choose difficulty of the game, in which the game would have difficulty that aligns the one I chosen

SECTION 3: SOFTWARE DESIGN

Design Pattern #1: Strategy Pattern

Map Overview: This pattern will be used to implement the map feature.



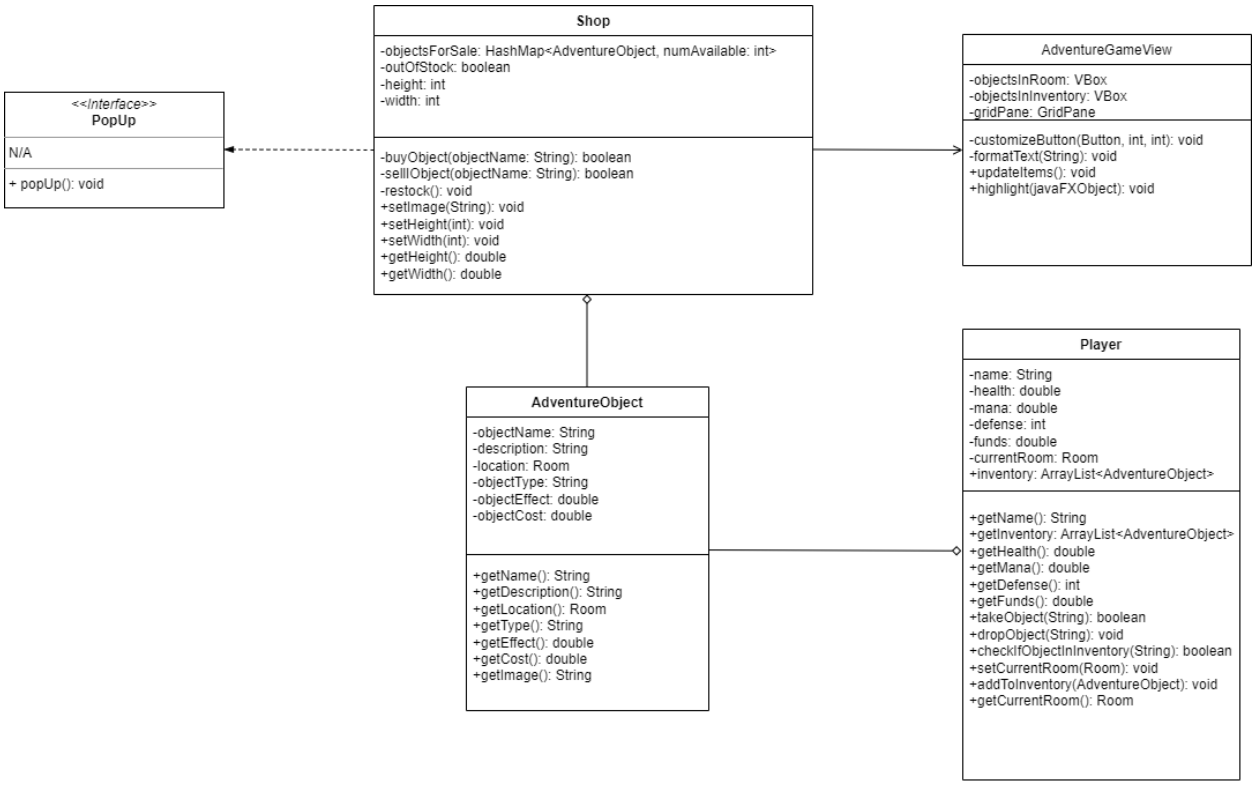
Implementation Details: The UML diagram outlines these main components:

- The *PopUp* interface, which includes one method: popUp.
- The *Map* class, which will implement PopUp along with generating the map itself using the parameters given by having two frames on top of each other with the top one being a black frame and the bottom being the map itself.
- The *setVisited(Room)* method will make the *Room* visible by making that part of the black frame transparent.
- The *AdventureGameView*, which will use the map created and add it into the game in the top right corner

The *setVisited* method which will be called by AdventureGameView will help *Map* get information on which rooms are being visited so that the rooms will be shown on the map. The *setHeight, setWidth, setImage* methods will aid in initializing the map image itself to a size that the game creator sees fit. The *popUp* method will aid in adding the map to the screen.

Design Pattern #2: Visitor Pattern

Overview: This pattern will be used to implement a shop, where the player is able to sell and buy objects



Implementation Details: The UML diagram outlines these main components:

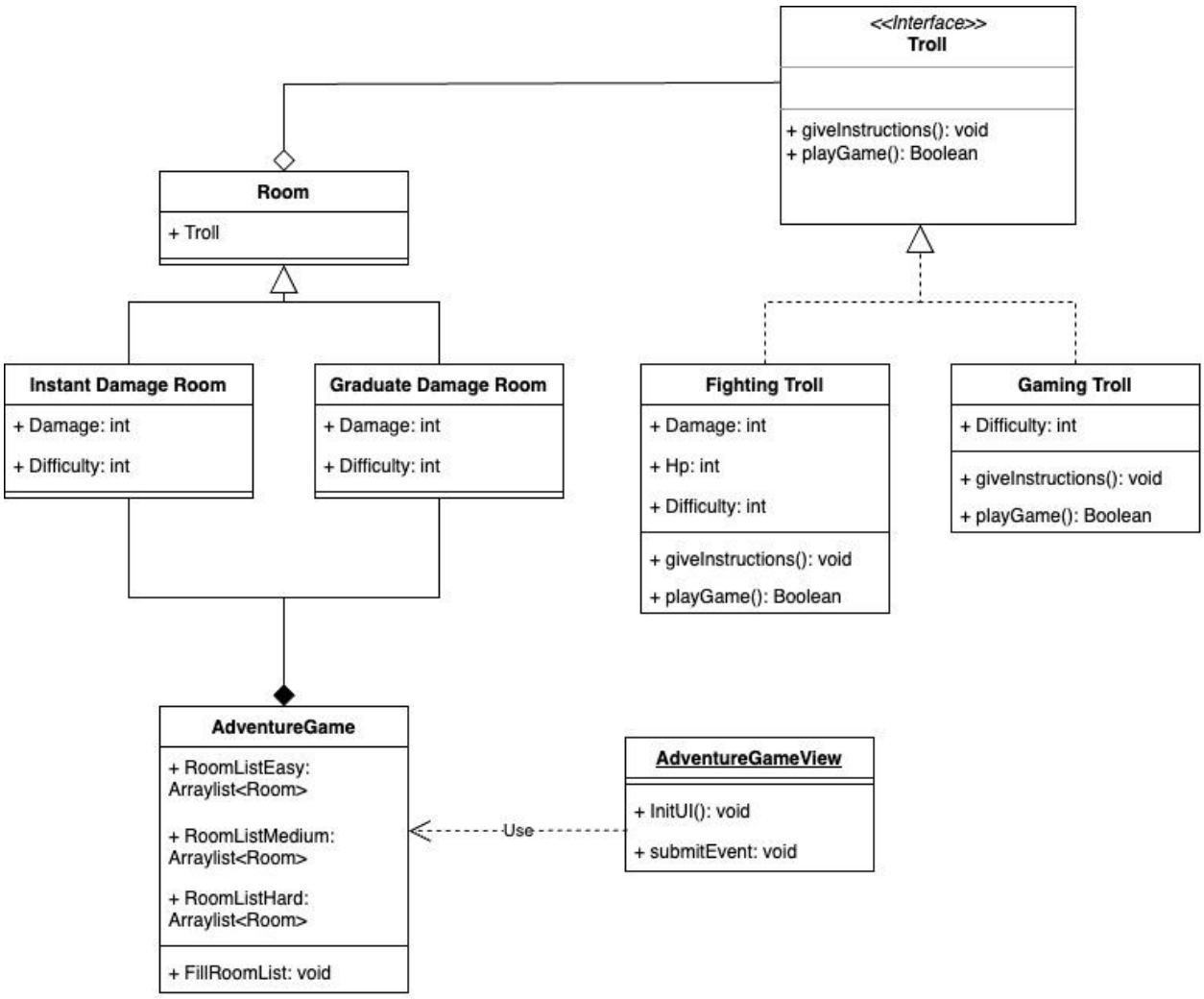
- Shop class: the shop class contains the AdventureObjects that are for sale and their quantity available. Method outOfStock notifies players if all objects for sale are depleted. Height and width is to set the shop interface's dimensions.
- When there is a successful purchase of an object, that object will enter the player's inventory and the respective value in the hashmap will decrease by one. The cost is to be subtracted from the player's funds. However, when the player sells an object, they receive the funds and that item will be removed from the inventory without appearing in the shop.
- Because we add or take away the objects from player inventory based on transaction, we need to include the Player class, as we need access to the inventory. The player will have a new attribute funds that describes what

player can afford

- Shop class implements *PopUp* interface, which satisfies Single Responsibility Principle (there will be many popups such as pause menu, map, etc.)

Design Pattern #3: Behaviour Pattern

Overview: This pattern will be used to implement ways to create different difficulty of games from different kinds/difficulties of trolls and rooms



Implementation Details: The UML diagram outlines three main components:

- Room class (Models a normal room as we did in assignment. It has two inheritance classes, Instant Damage Room and Graduate Damage Room. The first one represents once a player goes into that room, it results in a one-time

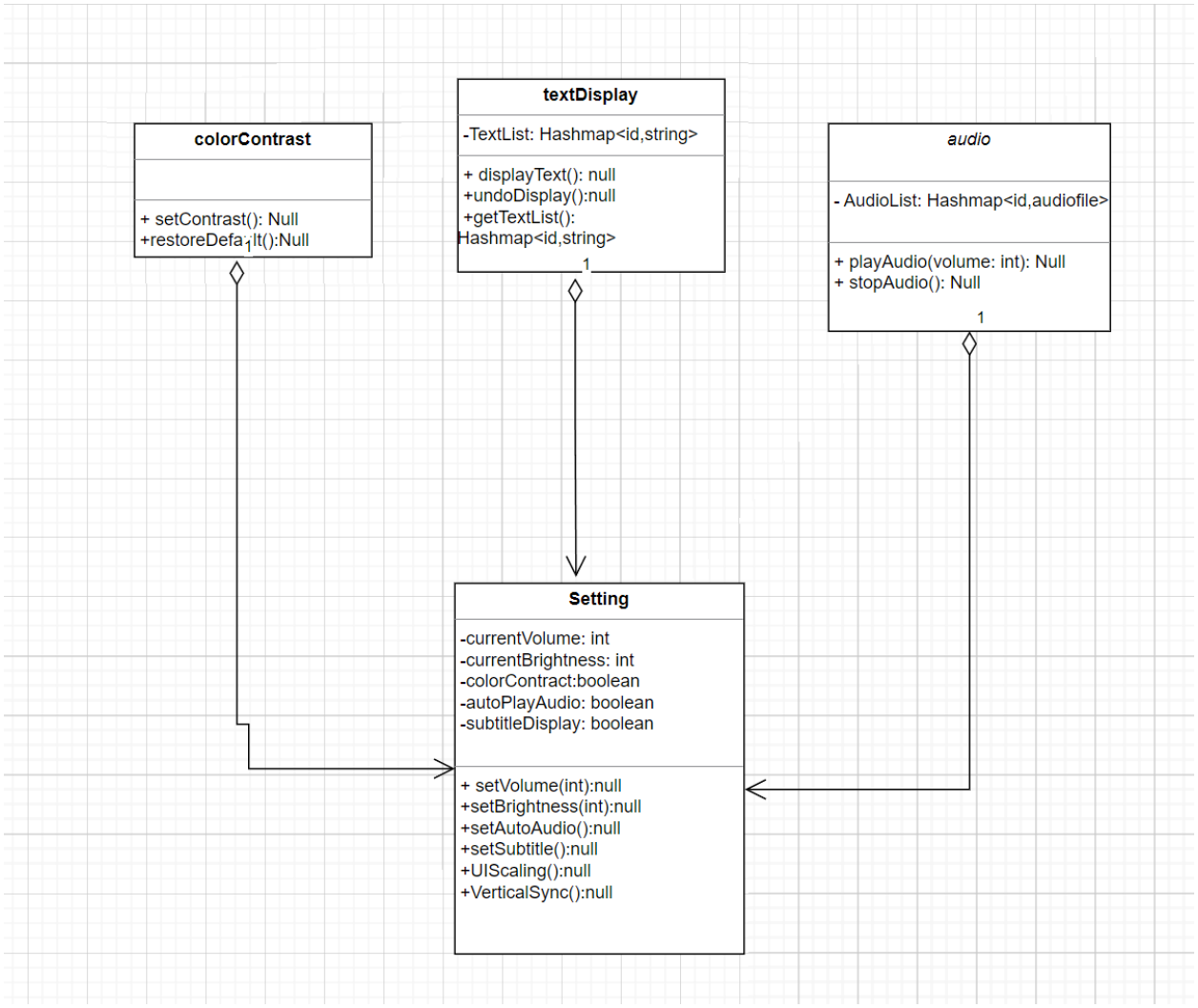
instant damage of some value. The latter would result in continuous damage per unit time (second). Damage can be negative (healling).

- Troll interface: We changed the troll such that the troll is now inside the room, so a player must be the troll in order to proceed to ANY other room. There are two kinds of trolls, fighting troll and gaming troll. Fighting troll means player must beat the troll by force. Gaming troll will initialise a game just like assignment 1.
- AdventureGameView class will have initUI method change so that a homepage would be shown, submitevent will also be added so that user can choose so that user can choose to continue/start new game/go to settings. AdventureGame would fill three roomlist with three different difficulties. If one difficulty is chosen, AdventureGameView will call AdventureGame so that when create a game model, it will use the room with the same difficulty.

For trolls: giveInstructions and playGame method are the same as usual. One difference is that when a player meets a fighting troll, the fight automatically begins within 3 seconds (with a count down), meaning the user does not have to input anything.

Design Pattern #4: Accessibility

OverView:This pattern will be used to set Audio/Display/Accessibility



Implementation Details: The UML diagram outlines four main components:

- Setting Class: Allows users to change volume, brightness, and set accessibility features. Uses Optional colorContract, text display, and Audio class. If the Game does not provide these classes, related features would be unable to be used.
- colorContrast Class: Sets the game to color contrast mode for users that have low vision
- textDisplay Class: Allow users with low hearing to be able to see the description and audio in text format

- audio class: Allow users with low vision to be able to play the game with information in audio format