CSC207 DESIGN DOCUMENTATION

Final Project Report
Created 2023.12.03

Monopoly Express

Sahaj Malavia
Esha Pakalapati
Xi Chen
Robin Zihan Chen

GitLab Repository:
https://mcsscm.utm.utoronto.ca/csc207_20239/group_1

Demo Link: https://youtu.be/RphRA5ikyIQ

SECTION 1: REPORT SUMMARY

The game of *Monopoly Express* is the short version of an adventure game. It is inspired by the idea of the board game Monopoly, where players can buy and sell items, and compete to see who is the "richest" person in the game. Similarly, we also have the buying in the game, and the overall goal for the players is the same, which is to see who is the richest.

Since the game of Monopoly rarely ends, we made our game so much shorter but rich in content at the same time. Instead of using the looped map, we made our game board linear, so that there is an end to the game. Instead of using 2 6-sided dice, we used a 4-sided one, so we don't miss all the fun grids all the time.

On our game board, we have in total of 4 obstacles, with corresponding puzzle trolls standing on each of these grids. When you land on these obstacle grids, you have to play the minigame with the troll, and if you win or it's a draw, you can continue playing, otherwise, you lose the game and will be removed.

When all players still in the game reach the end, the game is officially over. Whoever has the most balance and items worth in price wins the game.

SECTION 2: PROCESS DOCUMENTATION

Sprint 1 Report

1.1 SPRINT OVERVIEW

1.1.1 Sprint Overview

- **Start and End Date**: November 11 - November 16
- **Sprint Goal**: Implement the basics of the game, including the basics of the troll, implementing the queue, setting the board, and placing the objects on their respective squares.
  - Integrate player position attributes and visualization.
  - Implement turn order tracking with Queue class.
  - Develop visual representations for game objects and obstacles.
  - Add and integrate new trolls into gameplay.
  - Provide a troll help method with descriptive rules.

1.1.2 Stories Selected:

- 2.1 Square present player’s position
- 3.1 Order of player movements
- 2.2 Figure of Object and Obstacle
- 3.2 More troll
- 3.3 Troll help

1.1.3 Team Capacity:

Complete Square class, Visualization, Queue class, object/obstacle images, troll integration, and help method within the specified dates.

1.1.4 Participants:

- Xi: Square class method, 4 Troll class implementation.
- Esha: Visualization, Queue class, object/obstacle images, troll help method.
- Zihan & Sahaj: Help with the troll idea and coding. Debugging.

1.1.5 Tasks Completed:

- Square class and visualization method.
- Queue class.
- Images for objects and obstacles.
- Four trolls (rock paper scissors, tic tac toe, riddle).
- Troll help method.
- Debugged.

1.2 SPRINT 1 PRODUCT BACKLOG

- Competed user story:
  - 2.1 Square present player’s position
  - 3.1 Order of player movements
  - 2.2 Figure of Object and Obstacle
  - 3.2 More troll
  - 3.3 Troll help

1.3 SPRINT 1 CODE REVIEWS

- Robin: reviewed Troll help class find the missing text of the introduction of how to play the “Troll“ game as needed.
- Esha: reviewed Square class and found out the Player remove method missing the once player moves to the square it cannot be removed.

- Xi: reviewed Images for objects and obstacles and found one TicTacToe Troll image missing.
- Sahaj: reviewed Riddle Troll and found out one of the answers is not corrected.

1.4 SPRINT 1 RETROSPECTIVE

- The participants in the meeting: All four of us.
- Any unfinished tasks: None
- A summary of practices that went well this sprint and should be continued: When we encountered a problem, we immediately reached our teammates for a discussion and tried to solve it.
- A summary of new or revised practices to include moving forward: We are overall satisfied with our way so no. We do believe more frequent meet-ups would have excelled in communication. However, we always kept each other informed and updated about new changes, ideas, and differences.
- A summary of any bad practices that will not be repeated moving forward: Laziness.
- Your team's best/worst experience during this sprint: We all kept our work until the last day but all got them done in a short time. This really shouldn't be something that we should be proud of, but this also shows that we each have strong abilities to work individually.

Sprint 2 Report

2.1 SPRINT OVERVIEW

2.1.1 Sprint Overview

- **Start and End Date**: November 17 - November 22
- **Sprint Goal**: Implement the skeleton of the game by constructing the outline for the GameController Class. This includes attempting to implement all methods within the class as well as implementing the classes for the attributes of GameController.

2.1.2 Stories Selected:

- 1.6 Print inventory/cash
- 2.5 Square
- 4.2 Game Controller
- 4.3 Update Method

2.1.3 Team Capacity:

The team successfully implemented the show inventory method in the visualizer class, complete with a button to display the player's inventory. The Square class was finalized, and its visualization was created, displaying each square with a name and color indicative of its function. The Game Controller class was created, designed to manage the game's basic attributes and update them as needed. An update method was developed to refresh and apply current settings to the visualizer.

2.1.4 Participants:

- Sahaj: Responsible for the inventory printout method and the update method.
- Xi: Developed the Square class.
- Robin: Created the Game Controller class.
- Sahaj, Esha, Xi, and Robin: Collaborated on various tasks, including placing objects and Troll obstacles on the correct squares.

2.1.5 Tasks Completed:

- Visualization of the inventory and Square class.
- Creation of buttons for inventory display.
- Development of the Game Controller class method and the update method.

2.2 SPRINT 1 PRODUCT BACKLOG

- Competed user story:
  - 1.6 Print inventory/cash
  - 2.5 Square
  - 4.2 Game Controller
  - 4.3 Update Method

2.3 SPRINT 1 CODE REVIEWS

- Esha: reviewed the print inventory method and found that it did not print out objects as required and that it has added each object twice.
- Robin: reviewed square class and found that when it displayed the square it did not print out obstacles, that it did not add the obstacle image in the display array.
- Xi: reviewed the Game Controller class and found it cannot move the player created when a player in square 13 rolls a 4, it will crash because there does not exist square 17, forgot to add a limit to player move.
- Sahaj: reviewed the update method and found that only part of the game updated the image of the player did not get refreshed. Proceeded to correct and debug the code.

2.4 SPRINT 1 RETROSPECTIVE

- The participants in the meeting: All four of us.
- Any unfinished tasks: None
- A summary of practices that went well this sprint and should be continued: We keep discovering unnecessary codes or methods, so we removed them.
- A summary of new or revised practices to include moving forward: We changed some of the naming syntaxes in our methods, e.g. move_player() to movePlayer()
- A summary of any bad practices that will not be repeated moving forward: Still laziness.
- Your team's best/worst experience during this sprint: We made the code much easier by deleting lots of unnecessary features.

Sprint 3 Report

3.1 SPRINT OVERVIEW

3.1.1 Sprint Overview

- **Start and End Date**: November 23 - November 28
- **Sprint Goal**: Completing more basic implementations. Finished up the game Controller so that the methods could work with other classes and be tested. Test on some of the features of the visualizer. Visualizing troll interface. Most of the methods should be able to be tested(print to screen) after this week's sprint.

3.1.2 Stories Selected:

- 1.1:Chose name and color for each player
- 1.2:Roll dice
- 1.3:Buy object
- 1.4: interact with the obstacle
- 1.5: Win the game
- 1.7: player inventory
- 2.2: Figure of object and Obstacle
- 2.3: message
- 4.1: quit player
- 4.2: Game Controller
- 4.3: Update
- 4.4: Bank

3.1.3 Team Capacity:

Successfully created the Game controller class, messages, and player identifications were added to the game. Buying is created, and the player can now buy the items in the squares. When the player lands in the square with obstacles, they should play a minigame with a troll. The game can end now.

3.1.4 Participants:

- Sahaj: Method that lets players choose their color and name, method to roll dice, method to buy objects, method of buying objects, method of letting players interact with obstacles, and method to let players win the game.
- Esha: Image of the obstacle and object and message method.
- Robin: Game controller class, Bank class, and updating current settings and applying them to the visualizer.

3.1.5 Tasks Completed:

- GameController class
- Buy() method in Player class
- Dice class
- Interaction with obstacles
- Win a check in the Controller class
- Bank class
- Quit the player method in the Player class and remove a player from GameQueue.
- Messages
- Update method to update the scene

3.2 SPRINT 1 PRODUCT BACKLOG

- Competed user story:
    - 1.1:Chose name and color for each player
    - 1.2:Roll dice
    - 1.3:Buy object
    - 1.4: interact with the obstacle
    - 1.5: Win the game
    - 1.7: player inventory
    - 2.2: Figure of object and Obstacle
    - 2.3: message
    - 4.1: quit player
    - 4.2: Game Controller
    - 4.3: Update

    - 4.4: Bank

3.3 SPRINT 1 CODE REVIEWS

- Esha: reviewed the quitting player and found that after the player quit, the inventory did not get sold as designed, it just disappeared with the player class. The Player instances were initialized to a position "null" which made it difficult to implement the user story that updates the moves; set it to a Square 0.
- Robin: reviewed the Roll dice method and found that it has a wrong limit our design was D4 but the range of the dice was 2-6.
- Xi: reviewed Bank class and found that after the player buys for sale objects all players will share the cost and gain that it did not just give or take money from one player, instead all players.
- Sahaj: reviewed the Player creation method and found after each player chooses their name and color then at the start of the game each chess piece remains in the default name and color so that it does not save the player input.

3.4 SPRINT 1 RETROSPECTIVE

- The participants in the meeting: All four of us.
- Any unfinished tasks: UpdateItems and UpdateScene
- A summary of practices that went well this sprint and should be continued: We found some of the methods or attributes can be easily implemented in another class, so we have the corresponding person responsible for that class handle it. We handle jobs for each other.
- A summary of new or revised practices to include moving forward: Help each other.
- A summary of any bad practices that will not be repeated moving forward: We worked well this week.
- Your team's best/worst experience during this sprint: Someone is late for 1.5 hours!!!

Sprint 4 Report

4.1 SPRINT OVERVIEW

4.1.1 Sprint Overview

- **Start and End Date**: November 29 - December 5
- **Sprint Goal**: Be able to complete all the methods and basic implementations. Visualizing as much as we could, debugging and updating some of the old versions of the methods. Complete the visualization of the game controller as well as the trolls. Assemble classes and ensure they work well with each other.

4.1.2 Stories Selected:

- 2.3 message
- 2.4 Audible

4.1.3 Team Capacity:

Remade and added more messages to the game, including help and instructions. Record the voice and play the audio when the corresponding message is displayed to meet accessibility criteria—work on debugging the game and complete merging our code. Test for more bugs is possible. Complete the documentation.

4.1.4 Participants:

- All members of the team: Documentation and recording of the demo.
- Sahaj and Esha: Audio
- Esha: Message updates
- Robin: Fixed bugs and merged methods in the GameController class

4.1.5 Tasks Completed:

- Added more messages and voice(making the game audible)
- Debugging
- The game is now basically all done and ready to be played.

4.2 SPRINT 1 PRODUCT BACKLOG

- Competed user story:
    - 2.3 message
    - 2.4 Audible

4.3 SPRINT 1 CODE REVIEWS

- Esha: reviewed interaction with the obstacle method and found that when the troll triggered it did not turn to the terminal and display the troll to let the player play with the troll. The game troll UI and articulation happen after the game passes and not before the player plays the obstacle.
- Robin: reviewed player inventory attributes that find when a player meets an obstacle it mistakenly has been added a Troll to the inventory.
- Xi: reviewed Audio and found that missing audio for the troll help and troll description.

- Sahaj: Realized the colors updating the players' positions are not showing up on the screen. Resorted to a workaround, suggested to and helped Esha with fixing the update scene to keep the buttons as it is and instead create a new button to "see info".

4.4 SPRINT 1 RETROSPECTIVE

- The participants in the meeting: All four of us.
- Any unfinished tasks: We'll see in the code we hand in.
- A summary of practices that went well this sprint and should be continued: This week's task is overall easy so we are just fixing bugs and making our code lighter and better.
- A summary of new or revised practices to include moving forward: We mainly focused on the non-coding part of this assignment, and some merging of our code.
- A summary of any bad practices that will not be repeated moving forward: Wait until the last minute, and we find so many problems at that time. But there's no moving forward.
- Your team's best/worst experience during this sprint: The worst experience is there's so much to do for the documentation and the overall testing. We are so tired!

SECTION 3: SUMMARY

In wrapping up our project, we accomplished key goals like implementing the settings board, player position display, and interactive trolls. Despite successful collaboration, we faced limitations such as bugs and inventory issues. Although we wish we could say that we don't have unfinished user stories, we were unable to

Unexpectedly, we had to tweak the audio and visuals, resolving problems with troll displays and ensuring accurate audio cues. Some coding practices were refined for consistency. While our project didn't unfold exactly as planned, it reflects our adaptability and problem-solving skills, showcasing an iterative and improved game.