CSC207 PROJECT

FINAL Project Report

Created 2023.12.05

https://drive.google.com/file/d/1rq_t97i_LzIzeVpa8pWmO4_Iy8q-aRAL/view?usp=sharing

Colossal Cave Adventure 2.0

Daniel Sequeira
Dema Ismail
Eric Sun
Raiyan Rizwan

SECTION 1: REPORT SUMMARY

The objective of this project is to improve upon the Colossal Cave adventure game. A special consideration is placed on improving the accessibility features of the game, allowing it to be more easily playable by diverse groups who may have different accessibility requirements. A better developed audio adds to the acoustic richness of the playing experience, and also allows people with limited sight to better engage with the game. Other accessibility features in An enhanced graphical user interface will be constructed using JavaFX, where the user will be given more options to change display features like colour and sound features, such as adding auditory cues to make it easier and more entertaining and immersive to play the game. In addition, new interactive features will be added to make the game more challenging and add an additional layer of complexity to it. Design patterns will be employed to enhance the stability and maintainability of the project. The project will be managed using scrum/agile methodologies.

SECTION 2: PROCESS DOCUMENTATION

1.1.1 Sprint Overview:

Our goal for the first sprint was to implement our interactive features before we address the accessibility features. Our reasoning was that the accessibility feature implementations would likely work on top of the underlying ‘core’ mechanics of the game. This would remove unwanted complexity later on when having to implement these interactive features having to take account for fewer affected areas of the project.

The sets of features were to be implemented, the leaderboard and the trolls. The leaderboard feature keeps track of the time taken by the player to complete the game. When the game is over, the leaderboard is shown at the end with the current player’s score(time taken to finish game), along with the scores of other players who played the game. Along with this, three different troll implementations were to be constructed to add extra gameplay elements. These consisted of a Companion Troll, a Snake Troll and a TicTacToe Troll. The Companion Troll would instantiate a game with the player in which they have to solve a riddle. If they win, the Companion Troll joins them on their quest and makes more paths available for them. The Snake Troll forces a player to play a rendition of the classic ‘snake game’ to pass through a room. The TicTacToe is the same, but the player must finish a TicTacToe puzzle to advance.

1.1.2 Stories Selected for this Sprint:

- 1.1 (Leaderboard): Raiyan
- 1.2 (Companion Troll): Daniel
- 1.3 (TicTacToe Troll): Eric
- 1.4 (Snake Troll): Dema

1.1.3 Team Capacity:

We expect to be able to complete the three trolls and the leaderboard by Sunday, November 19.

1.1.4 Participants:

- Raiyan: Creating the leaderboard feature of the game and documenting the first sprint
- Dema: Implementing the snake game troll logic and interface
- Daniel: Implementing the companion troll logic and interface
- Eric: Implementing the tic tac toe troll logic and interface

1.1.5 Tasks Completed:

For the leaderboard, a stopwatch was created to keep track of time taken by the player to complete the game, and at the end the time taken for the player was displayed. Trolls were based on the Troll interface and GameTroll from assignment one. All Trolls had been completed in terms of their logical gameplay mechanics (i.e snake game window, riddle generation, TicTacToe puzzle window) but the functioning with the interface of the game had not been finalized. We realized that this would require another separate user story to resolve in the next sprint.

2.2. SPRINT 1 PRODUCT BACKLOG:

Name	ID	Owner	Description	Implementation Detail	Priority
Leaderboard	1.1	Raiyan	As a competitive adventure game user I want a leaderboard feature so that I can track my progress and compare my scores with friends.	As the game begins, a stopwatch feature will be implemented which will track the amount of time the player takes to complete the game. The time will be recorded on the leaderboard as the high score.	2
Companion Troll	1.2	Danial	As an adventure game user, I want replayable elements in the game so that I have choices that affect the difficulty of my playthrough.	If a player is in a room with a Companion Troll there is a game that they can play where they have to convince the Troll to join them on their quest. If they successfully do that then the Troll follows them on their quest. As a result, they will find that the Companion Troll can allow them to enter more rooms. This will be implemented by instantiating a new AdventureObject in the player's inventory container object.	2
Tic-Tac-Toe Troll	1.3	Eric	As a user who craves for new and interesting challenges in any game, I want a troll who engages me in a game of Tic Tac Toe so that I can hon my ability to strategize.	In the event that the "Tic Tac Toe Troll" is in a room the user enters, the user is presented with a window with the 3x3 tic tac toe board. The user is prompted whether they or the troll should make the first move. The user and the troll would then place marks	3

				on empty spots on the board, in alternating order. For the user, this would be done using keyboard input. The player would win and be allowed to pass if they place three marks in a line or diagonally on the board.	
Snake Game Troll	1.4	Dema	As an arcade enthusiast gaming user I want a troll that challenges me to play the classic snake game so that I can challenge my reflexes and relieve nostalgia of the past.	If the player enters a room with a “Snake Game Troll”, the snake game is displayed using updatescene method. The user can utilize the arrow keys to collect food that when collected increases the length of the snake. This will be handled using an event-handling method. If the user hits the edge of the playing area or the snake wraps around itself the troll sends them back to a previous room. To win, the user must have a snake with a length of thirty. When the user wins the troll allows them to pass through. The Snake Game Troll implements a Troll interface and JavaFx will be utilized to implement the game window, snake, and food.	3

2.3. SPRINT 1 CODE REVIEWS

Story Reviewed	Name of Reviewer	Pull Request Link
[DEV-1.2] Companion Troll	Dema	https://mcsscm.utm.utoronto.ca/csc207_20239/group_8/-/merge_requests/4
[DEV-1.3] Tic Tac Toe Troll	Raiyan	https://mcsscm.utm.utoronto.ca/csc207_20239/group_8/-/merge_requests/12
[DEV-1.4] Snake Game Troll	Daniel	https://mcsscm.utm.utoronto.ca/csc207_20239/group_8/-/merge_requests/11

2.4 SPRINT 1 RETROSPECTIVE:

The meeting attendees included: Daniel, Dema, Eric and Raiyan. The leaderboard feature was done and the logic behind each troll was completed. However, members underestimated how long it would take to make the GUI. Creating the stopwatch feature for the A.GameView was simple, but the duration for the timeline kept speeding up as the player moved to a different room. This had to be fixed through careful placement of the timeline event handler. Members responsible for the trolls were having trouble with the GUI aspect of their trolls, the transitioning and syncing from the troll games to the adventure game as JavaFX was still relatively new to them. The team decided to split each troll user story into two parts: a logical part and an interface part which will be reflected in sprint two.

A tactic that was beneficial during this sprint was the team's frequent communication and discussion of progress made. One thing we will do moving forward is inform one another of any base/important file revisions and discuss their applicability when pushing. A bad practice the team faced which should not be repeated moving forward is not updating each other as regularly as necessary, which led to confusion especially when it came to gitlab. The worst experience during this sprint was working on the interfaces and the transitions from the troll games to the adventure game.

2.1.1 SPRINT 2 OVERVIEW

Our goal for this sprint is to finish the leaderboard feature by adding a feature where each player could include their name and the leaderboard would keep a record of the total time taken for multiple players organized by their respective names, and to finish the GUI and transitioning from the companion, snake game and tic tac toe trolls to the adventure game view.

2.1.2 Stories Selected for this Sprint:

2.5 (Companion Troll interface): Daniel

User story 2.5 was added during the second sprint to deal with the complexities occurring between the troll objects and the adventure game implemented in assignment two. Seeing as Daniel had started working on an implementation for making his Companion Troll work with AdventureGameView and be represented as an element within the game we created another user story for this aspect. Further, to decrease unwanted complexity and to give the code more longevity in terms of future iterations, it was decided that all Trolls would follow the same implementation to work with AdventureGameView. As a result, once Daniel had finished getting his Companion Troll to work with the existing game he reworked the code to be general for different Troll implementations. To achieve this the Troll interface from assignment 1 would have to be redone to satisfy the needs of the current project that had a GUI as opposed to the CLI-based game from the first assignment. In AdventureGameView this new Troll interface was used to implement the Troll mechanic logic. Then all Troll classes would be adjusted to implement this new Troll interface. This would also mean that the functionality of the Snake Game Troll and the TicTacToe Troll were changed to match that of the Companion Troll. This change was decided so that we could prioritize simplicity in the code while retaining the key elements of the features we wished to implement.

2.6 (SnakeGame Troll interface): Dema

User story 2.6 was added (detailed in backlog). This story is a revised version of user story 1.4 and follows user story 2.5. It deals with handling the transitioning and updates between the snake game GUI view stage and the adventure game view stage.

1.3 (Tic Tac Toe troll interface): Eric

1.1 (Leaderboard): Raiyan

2.1.3 Team Capacity: We expect to be able to finish all the trolls and the leaderboard by Saturday, November 25.

2.1.4 Participants:

- Daniel will oversee the completion of the companion troll and will manage the completion of all the troll transitions.
- Dema will oversee the completion of the snake game troll GUI and transitioning
- Eric oversee the completion of the tic tac toe troll GUI and transitioning
- Raiyan oversee the completion of the leaderboard and will be documenting sprint 2

2.1.5 Tasks Completed: All tasks assigned to members were completed.

2.2. SPRINT 1 PRODUCT BACKLOG:

Name	ID	Owner	Description	Implementation Detail	Priority
Companion Troll Interface/Transition	2.5	Daniel	As a developer I wanted there to be a simple streamline integration of different Trolls with AdventureGameView so that the same logic would allow different Troll games to be played.	Trolls will be stored as AdventureObjects in their respective rooms that can be picked up by the player if they win. In AdventureGameView updateItems() will be altered so that if a certain Troll is detected to be in that room a placeholder variable currTroll will be set to the instantiation of that Troll. submitEvent() will be altered so that if currTroll is not null then the player can interact with the Troll appropriately through the text prompt. This is where the logic for the integration with the Troll interface will be.	1
Snake Game Interface/Transition	2.6	Dema	As a game developer, I want to seamlessly integrate the classic	Create a snakeGameView class that that accommodates elements from both games. This class will contain methods related to the snake GUI stage like	1

			Snake Game GUI within the Adventure Game's GUI so that users can navigate and transition between the Snake Game and the adventure game without losing their progress or having one of the games crash.	move (which uses the snake logic implemented in user story 1.4) to start the snake game when the troll approaches that stage and make the adventure game wait for the results before proceeding. Create a backToAdventure and handleGameResultsthat ends the snake game loop when the user wins or loses, closes the snake game stage, and notifies the adventure game so that it continues on with the troll interaction.	
Tic Tac Toe Interface	1.3	Eric	As a user who craves for new and interesting challenges in any game, I want a troll who engages me in a game of Tic Tac Toe so that I can hon my ability to strategize.	In the event that the “Tic Tac Toe Troll” is in a room the user enters, the user is presented with a window with the 3x3 tic tac toe board. The user is prompted whether they or the troll should make the first move. The user and the troll would then place marks on empty spots on the board, in alternating order. For the user, this would be done using keyboard input. The player would win and be allowed to pass if they place three marks in a line or diagonally on the board.	1
Leaderboard	1.1	Raiyan	As a competitive adventure game user I want a leaderboard feature so that I can track my progress and	Before starting the game, a separate window will pop up due to the implementation of AdventureGameStartView.java. As the game begins, the timeline feature will initiate	2

			compare my scores with friends.	the stopwatch feature, which will run till the end of the game. It will track the amount of time the player takes to complete the game. The time will be recorded on the leaderboard as the high score and written into score.txt. Also, whatever score was saved into score.txt beforehand will be shown on the leaderboard along with the score of the current player.	

2.3. SPRINT 2 CODE REVIEWS:

Story Reviewed	Name of Reviewer	Pull Request Link
[DEV-1.1] Leaderboard	Eric	https://mcsscm.utm.utoronto.ca/csc207_20239/group_8/-/merge_requests/5
[DEV-2.5] Companion Troll	Dema	https://mcsscm.utm.utoronto.ca/csc207_20239/group_8/-/merge_requests/4
[DEV-1.3] Tic Tac Toe Troll	Raiyan	https://mcsscm.utm.utoronto.ca/csc207_20239/group_8/-/merge_requests/12
[DEV-2.6] Snake Game Troll	Daniel	https://mcsscm.utm.utoronto.ca/csc207_20239/group_8/-/merge_requests/11

2.4 SPRINT 2 RETROSPECTIVE:

Attendees in this meeting were Daniel, Dema, Eric, and Raiyan. The leaderboard feature was fully completed, and some issues faced while completing the implementation was deciding the design for recording the name of the player so that not too much editing of the AdventureGame.java and related application code was necessary. Also, making it so that the names and scores of each player were properly written into and read from scores.txt took some amount of trial and error. The troll interface was written to accommodate different Troll games and also to work with the logic written into AdventureGameView for when the player encounters a Troll. Taking advantage of polymorphism, CompaionTroll and other Trolls can be integrated easily into the game. Snakegame troll now supports a unified UI that accommodates the adventure game and the snakegame without freezing or crashing. Due to the intense debugging and the fixes and additional work each member had to do to satisfy user stories from sprint 1, user stories intended for sprint 2 were pushed forward into sprint 3.

3.1.1 SPRINT 3 Overview:

Our goal for this sprint is to complete the sound, colour contrast and magnification accessibility features indicated by our product backlog.

3.1.2 Stories Selected for this Sprint:

- 2.2 (Magnification): Raiyan
- 2.3 (Acoustic immersion): Daniel
- 2.1 (Colour Contrast): Dema
- 2.4 (Sound Cue): Eric

3.1.3 Team Capacity: We expect to be able to complete the accessibility features Sunday, December 3

3.1.4 Participants:

- Daniel: implementing acoustic immersion
- Dema: implementing colour contrast and documenting sprint 3 on the final report
- Eric: implementing sound cue and documenting sprint 3 on the final report
- Raiyan: implementing magnification

3.1.5 Tasks Completed:

All accessibility feature stories were completed and documented in the final report. The magnification feature was completed giving the user the ability to change the font of the text. The audio cue feature was also added, allowing the user to listen to an audio description of the three main buttons if they hover the cursor over it. To complete this feature, a verbal reading of the buttons' description was recorded. The acoustic immersion feature was implemented fully. A new RoomSound class was implemented to handle playing additional sounds that were appropriate for the player's current setting. Lastly, the colour contrast feature was completed and added to the game. Now, users with colour blindness are able to adjust the colour scheme of the game so that they can more easily see the game window and its components. Together, those features allowed the game to be more accessible to a wider range of players who may previously have not been able to participate.

3.2: Sprint 3 product Backlog

Name	ID	Owner	Description	Implementation Detail	Priority
Color Contrast	2.1	Dema	As a low vision screen user playing Colossal Cave Adventure 2.0, I want a game interface with high colour contrast so that I can differentiate and navigate the different interface elements like images, buttons, and text.	Add a combo box to represent the “colour options” button that provides a drop-down menu with colour schemes that cater to the different types of colour vision deficiencies so that when the user selects the colour scheme of their interest the background, images, text and buttons in the interface adjust using the updatescene and updateitem methods to the selected low vision friendly contrast scheme.	1
Magnification	2.2	Raiyan	As a low-vision user playing the game, I want to be able to change the font size of the text on the screen so that the text is more legible.	Add a combo box with a drop-down menu that includes varying font sizes that the user can change to. The eventHandler for the ComboBox is located in the AdventureGameView.java file and directly calls on the setFontSize method, which eventually changes the font sizes. On the other hand, the FontHandler.java file handles the mouse event for when the font size from the drop-down menu is clicked.	2
Acoustic Immersion	2.3	Daniel		Room objects will have a container attribute that can contain different keywords that describe the scene. These will be processed if a Game folder has a “sounds.txt” file. Accordingly a roomAudio attribute will be given to the AdventureView class that	1

				holds an instance of a new RoomSounds class that will be updated in the updateScene() method. AdventureLoader will have an extra method added parseSounds() that populates the container attribute of every Room object that has sounds assigned to its room number.	
Button Sound Cue	2.4	Eric	As a visually impaired person interacting with the game, I want to be able to hear a description of which button I click when I hover on a button so that I can navigate the game more easily.	When the user hovers the mouse over a button, an audio recording of the button's description will begin to play.	2

3.3: Sprint 3 Code Review

Story Reviewed	Name of Reviewer	Pull Request Link
ColourContrast	Eric	https://mcsscm.utm.utoronto.ca/csc207_20239/group_8/-/merge_requests/10
Magnification	Daniel	https://mcsscm.utm.utoronto.ca/csc207_20239/group_8/-/merge_requests/7

Acoustic Immersion	Raiyan	https://mcsscm.utm.utoronto.ca/csc207_20239/group_8/-/merge_requests/14#1ebce828c4336af58f2946c26dfcffff8084abe3
Button Sound Cue	Dema	https://mcsscm.utm.utoronto.ca/csc207_20239/group_8/-/merge_requests/13

3.4 SPRINT 3 RETROSPECTIVE:

Attendees in this meeting were Daniel, Dema, Eric, and Raiyan. The magnification feature was successfully implemented, and an issue faced while working on this feature was adjusting the font of certain text labels so that they would still fit the gridpane even when font was increased. The colour contrast and button sound cue features were also finished. In this run we were supposed to implement a power-up feature but since we didn't have enough time we got rid of this feature.

4.1.1 SPRINT 4 Overview:

Our goal for this sprint is to merge what we have with the main branch and prepare required documentation for final submission.

3.4 SPRINT 4 RETROSPECTIVE:

Attendees in this meeting were Daniel, Dema, Eric, and Raiyan. All work was done. Members faced extreme challenges with git in terms of the stacking of everyones code and the push to main. At the end the final product contained some features that did not stack well on top of each other. For example, the colour contrast for the room object images and the functionality of the sound cues.

SECTION 3: SUMMARY

Overall, the project implementation has been very successful. The group worked collaboratively and effectively communicated online. The game has become much more accessible to people who may have visual impairment, with audio and text enlargement aids. Gadgets added, such as the timer, provided useful information which may be needed. With the addition of different trolls and a variety of minigames and challenges, the player's experience is drastically broadened and improved. The diversity of gaming options added allow users to explore the game through their individual preferences. The result is a more fun, accessible, and complex game with a widened appeal.

The biggest deviation from our initial plan was the work on the Trolls. Given the challenge with integrating the Trolls with the GUI it was decided that it would be more efficient and less complex if we had a single interface with which the GUI was integrated with. As a result we had to create new user stories to adapt our process. Along with this we also made changes to the functionality of the different Troll implementations. As each troll would be implementing a common interface which would be hard coded into the logic of the AdventureGameView, some functionality would be shared. This did not affect the other elements of any Troll and made it easier to control how they were used in the game. For example, the process of adding, removing and debugging Trolls is much simpler given how they are all added into the game the same way, being loaded with their information from the objects.txt file. This was an important resolution that was made to advance the project.

In addition, when we finally merged everyone's work together not all the work perfectly stacked like we expected initially. As a result a fair bit of debugging had to be done. Although, it did not turn out as we expected we managed to resolve some of the issues given the time we had.