Final Project Report

Created 2023.12.05

Halloween Harvest: A Spooktacular Adventure!

Authors:
Garry Galsinh
Keith Tran
Mahmood Saifi
Sabrina Hirani

Link to project demo
Link to project repository

Contents

1 Report Summary

Halloween Harvest: A Spooktacular Adventure is a Halloween-themed role-playing game where players hunt for candy in an alternate Halloween dimension. Players navigate a series of neighbourhoods, collecting candy as much candy as possible, before eventually encountering (and hopefully defeating) the candy monster. Our motivation for building this game is to provide an entertaining experience and to spread the spooky Halloween spirit.

The project sets clear objectives with the intention of providing an engaging game that is completed within the given timeframe of 3 sprints. The project aimed to create a Halloween-themed role-playing adventure—Halloween Harvest: A Spooktacular Adventure. The gameplay features the player navigating various neighbourhoods, finding different costumes they can wear, forging friendships, and confronting the formidable candy monster for an epic showdown. The game's theme was further expressed in the meticulously crafted and visually striking art of the costumes, neighbourhoods and the NPCs. Meeting these objectives required paying close attention to the design document's game outline and the project specifications' time and technical requirements.

2 Process Documentation

This section documents each sprint conducted by the team. Due to unexpected issues with git and coordinating the repository initially, we were only able to complete three sprints. The issues we faced at the start of the project, and later in each sprint, as well as what we did to adapt and overcome those issues in an agile manner, are all further documented below.

2.1 Sprint 1 Documentation

Dates: 2023.11.08 – 2023.11.18

2.1.1 Overview

Our goal for this sprint is to implement rudimentary foundational features. This includes loading game data and implementing core classes such as the `Neighbourhood`, `Player`, `Costume` and `NPC` classes. We also aim to begin working on the base layout for the graphical user interface, and utility classes or methods for loading and handling data (such as images, scenes, JSON, etc.), which will make development smoother in the future.

2.1.2 Selected User Stories

Selected User Story	Associated Tasks	Owner
1.0	Loading game data (from JSON files)	Sabrina
2.0, 2.1	Implementing the Neighbourhood class and instantiating map locations	Sabrina
3.0, 3.1, 3.2, 3.3	Implementing Character classes (base class, NPC, Player)	Garry
4.0	Costume class	Garry
5.1, 5.2	Implementing Combat functionality (the GUI scene and turn-based attack logic)	Keith
6.0	Filling NPC data (names, descriptions, stats, etc.)	Keith
8.0, 8.2	Base GUI setup/layout	Mahmood

2.1.3 Team Capacity

In our first sprint, we laid out the following expectations for our team:

- Set up the git workflow for each participant. We were all new to `git flow`, and expected that it would take some time getting used to it. We should try to get this done within the first few days of the sprint so that we can later focus on implementing user stories.

- Fill out the most important game data in the `data.json` file. This data includes NPCs, items, costumes, neighbourhoods, and other "static" data for the game. Our goal was to complete the important data early, as this would then help us design and structure the rest of the game. This should be done early as well, preferably by 2023.11.12 as the rest of the code and design depends on this.

- Implement base classes representing the entities in the `data.json` file – again, this includes NPCs, the Player themselves, neighbourhoods, and costumes. This should be done by 2023.11.18, the end of the sprint.

- Begin implementing GUI-related functionality – we expected that the GUI will take time, as implementing it is not only time-consuming but also requires other classes to be implemented as well since it depends on them for communicating game state. As such, our goal was mainly to have a basic layout and utility methods for dealing with GUI implemented in the first sprint. This should also be done by 2023.11.18, the end of the sprint.

2.1.4 Participants

Participant	Assigned User Stories	Tasks & Additional Responsibilities
Sabrina	1.0, 2.0, 2.1	• Add rudimentary data to the `data.json` file; • Implement a rudimentary `NeighbourhoodBuilder` class and interface; • Implement a rudimentary `Neighbourhood` class; • Implement the `Driver` class; • Project coordinator; • Maintain sprint process documentation.
Garry	3.0, 3.1, 3.2, 3.3, 4.0	• Implement a rudimentary `Player` class; • Implement a rudimentary `Costume` class.
Keith	5.1, 5.2, 6.0	• Implement a rudimentary `Attack` class; • Complete the `NPC` class; • Creating graphics and sprites.
Mahmood	8.0, 8.2	• Create a basic GUI layout and utilities; • Ensuring best practices for the team's git workflow.

2.1.5 Tasks Completed

Green represents completed user stories.

User Stories	Completion Status
1.0	In progress
2.0, 2.1	Completed
3.0, 3.1, 3.2, 3.3	In progress
4.0	Completed
5.1, 5.2	In progress
6.0	Completed
8.0, 8.2	In progress

2.1.6 Product Backlog

Name	ID	Owner	Amendments
Load Game	1.0	Sabrina	Use a `paths` mapping to update `Player` location (instead of adjacency list)
View NPC in neighbourhood	2.0	Sabrina	–
View Costume in neighbourhood	2.1	Sabrina	–
Player Statistics	3.0	Garry	–
Player Interaction with Costumes	3.1	Garry	–
Player Inventory Remove	3.2	Garry	–
Player Inventory Add	3.3	Garry	–
Switch Costumes	4.0	Garry	–
Attack NPC	5.1	Keith	–
Hit By NPC	5.2	Keith	–
GUI Theme Change	8.0	Mahmood	–
GUI Sidebar Layout	8.2	Mahmood	–

2.1.7 Code Reviews

Note: initially, in the first sprint we conducted code reviews in a separate branch, where the reviewer checked the code and let the author know of any suggested changes or issues. These were then resolved in the same branch or the appropriate feature branch. We soon realized this was a bad practice and revised it; see the retrospective section for more details. Suggested changes were discussed in-person, and have been documented here directly (this practice was also later revised at the end of sprint 2).

User Story	Reviewer	Suggested Changes & Issues	Pull Request Link
2.0, 2.1	Mahmood	• Incorrect return type for `getCostumes` and `getBackground`; • Unhandled `NullPointerException` in `takePath`; • `costume` and `npc` should be private.	None; the reviews were conducted in the Code Review branch (Link to branch)
3.0, 3.1, 3.2, 3.3	Sabrina	• Use consistent `camelCase` naming convention; • `equipped` should point to the index of the equipped costume in inventory.	See above
4.0	Keith	• Make class attributes private where appropriate, and implement getters; • Implement the `equals` and `hashCode` methods.	See above
6.0	Garry	• Revisions to JavaDoc comments; • Implement the `equals` and `hashCode` methods.	See above

2.1.8 Retrospective

Participants	Sabrina, Garry, Keith, Mahmood
Unfinished Tasks	• We will need to continue working on and refining the `Player`, `Costume`, and `NPC` classes. • The GUI layout is still unfinished. • Some game data is still to be completed (though the remaining data is not very high priority at the moment).
Best Experience	Keith drew very impressive sprites for the game – it motivated us to finish it faster so we could see the graphics in action as well!
Worst Experience	We accidentally deleted the remote `develop` branch when handling a merge request!
Bad Practices	• We are not used to working with feature branches, so we sometimes would merge into develop without creating a merge request; • We weren't used to git flow, and sometimes accidentally created feature branches manually instead of through git flow.
Revised Practices	• Do not make commits directly to develop; • Before starting on a new feature, use `git flow feature start <name>` instead of manually creating branches.

2.2 Sprint 2 Documentation

Dates: 2023.11.18 – 2023.12.01

2.2.1 Overview

Our goal for this sprint is to finalize our logical classes and to complete the layout for the graphical user interface. We also want to start integrating the game logic into the graphical user interface by allowing the user to navigate around the map.

2.2.2 Selected User Stories

Selected User Story	Associated Tasks	Owner
1.0, 1.2	Loading game data (from JSON files) & handling player movement	Sabrina
3.0, 3.1	Implementing Character classes (base class, NPC, Player)	Garry
4.0	Costume class	Garry
5.0, 5.1, 5.2	Implementing Combat functionality (the GUI scene and turn-based attack logic)	Keith
8.0, 8.2, 8.3	Base GUI setup/layout	Mahmood

2.2.3 Team Capacity

In our second sprint, we laid out the following expectations for our team:

- Finish the `Player` and `Costume` classes, since they were left unfinished at the end of the last sprint, and are important for developing the rest of the application. Our goal is to finish this by 2023.11.21.

- Make progress on the GUI layout, as not much progress was made last sprint due to issues with file paths (for loading images) and setting up JavaFX. We aim to have a basic layout ready by 2023.11.25.

- Begin integrating game logic into the GUI – in particular, we want to complete navigation functionality, allowing the user to navigate around neighbourhoods by pressing arrow keys. This is obviously an important functionality of the game, but will also help lay the foundation for integrating other logic with the GUI, and is the main goal of this sprint. As such, we aim to complete it by the end of the sprint, 2023.12.01.

2.2.4 Participants

Participant	Assigned User Stories	Tasks & Additional Responsibilities
Sabrina	1.0, 1.2	• Complete the `data.json` file; • Complete the `NeighbourhoodBuilder` class and interface; • Implement player movement in the `Neighbourhood` class; • Maintain sprint process documentation.
Garry	3.0, 3.1, 3.2, 3.3, 4.0	• Complete the `Player` class; • Complete the `Costume` class;
Keith	5.1, 5.2, 6.0	• Implement a rudimentary `Item` class; • Project coordinator. • Complete the `Attack` class; • Begin work on combat functionality; • Creating graphics and sprites.
Mahmood	8.0, 8.2	• Finish the basic GUI layout and complete utilities; • Ensuring best practices for the team's git workflow.

2.2.5 Tasks Completed

Green represents completed user stories.

User Stories	Completion Status
1.0, 1.2	In progress
3.0, 3.1	Completed
4.0	Completed
5.0, 5.1, 5.2	In progress
8.0, 8.2, 8.3	Completed

2.2.6 Product Backlog
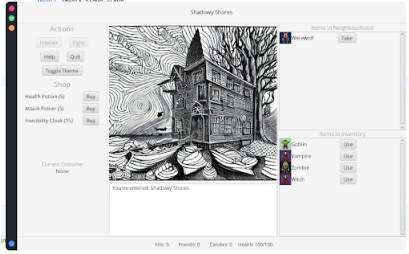
Name	ID	Owner	Amendments
Load Game	1.0	Sabrina	–
Handle Mouse Input	1.2	Sabrina	–
Player Statistics	3.0	Garry	–
Player Interaction with Costumes	3.1	Garry	–
Switch Costumes	4.0	Garry	Updated implementation details for how costumes are equipped (now point to the index of equipped costume in the inventory)
Combat Scene	5.0	Keith	–
Attack NPC	5.1	Keith	–
Hit By NPC	5.2	Keith	–
GUI Theme Change	8.0	Mahmood	–
GUI Sidebar Layout	8.2	Mahmood	–
GUI Responsive	8.3	Mahmood	–

2.2.7 Code Reviews

Note: some code reviews were conducted in-person, with the author of the merge request and the reviewer looking over the code in person. Changes were suggested in-person as well, before the request was approved and merged. As such, there are comments missing on the GitLab interface; this was later revised for the last sprint (comments were added on GitLab as well). The suggested changes are documented below.

User Story	Reviewer	Suggested Changes & Issues	Pull Request Link
6.0	Sabrina	• Add consistent JavaDocs.	Link to PR (!5)
4.0	Mahmood	–	Link to PR (!9)
8.2	Garry	–	Link to PR (!11)
8.0, 8.3	Keith	–	Link to PR (!10)

2.2.8 Retrospective

Participants	Sabrina, Garry, Keith, Mahmood
Unfinished Tasks	• Player movement within the GUI is unfinished. • We are still working on adding items to the game. • The logical aspect of Costumes is finished, but they still need to be integrated into the GUI. • We are still working on implementing combat functionality. The GUI is close to finished, but the underlying logic still needs more work. • We are still working on building the graphical user interface. Now that we have the basic layout, we are focusing on integration with our game logic.
Best Experience	We have a pretty graphical interface layout.
Worst Experience	One of us couldn't figure out why a JavaFX method wasn't working and tried everything they could. They even swapped IDE. On the verge of tears, it turned out the JavaFX method simply wasn't in a method in the class they were working on and just free floating in the class block.
Bad Practices	• We did not properly conduct code reviews during merge requests. While we fixed our last bad practice of conducting the review in a separate "code review" branch, we still did not leave comments on GitLab itself, the reviewer simply communicated with the author via our group chat or in-person to notify them of issues or suggested changes. This was effective in communicating the issues, but was ineffective in terms of documenting our changes and processes and why certain design decisions were made.
Revised Practices	• Use the comment and code review features on GitLab, *in addition* to communicating directly.

2.3 Sprint 3 Documentation

Dates: 2023.12.01 – 2023.12.05

2.3.1 Overview

Our goal for this sprint is to finalize the game by integrating the game logic with the graphical interface. We also want to complete combat functionality.

2.3.2 Selected User Stories

Selected User Story	Associated Tasks	Owner
1.2, 1.3	Handle user input and complete movement	Sabrina
3.0	Finish player statistics implementation	Garry
4.2	Implement Costume attacks and abilities that are added to the player	Garry
5.0, 5.1, 5.2	Finish implementing combat functionality (finalize the GUI scene and finish turn-based attack logic)	Keith
5.3	Create a "Game Over" scene for when the player loses	Mahmood
8.1	Add accessibility texts to each element	Mahmood
9.0	Implement the GUI for the in-game shop	Mahmood

2.3.3 Team Capacity

In our last sprint, we acknowledged we would be unable to complete all user stories to the level we initially wanted to, and decided to focus on completing the highest-priority features:

- Fix the remaining bugs with player movement and user input, by 2023.12.02

- Finish player statistics, and integrating them into the GUI by 2023.12.02

- Implement Costumes attacks and abilities, which will be integrated into the combat functionality. Since the combat functionality will likely take longer, the aim is to complete this by the end of the sprint on 2023.12.05.

- Finish the GUI theme toggle and sidebar item lists which show costumes as well. We expect to finish this by 2023.12.03.

- Implement combat functionality, including the GUI elements and the logical aspect, by 2023.12.04.

- Finally, integrate the combat functionality into the main GUI, bringing everything together. We expect to finish this by 2023.12.05, the end of the sprint.

2.3.4 Participants

Participant	Assigned User Stories	Tasks & Additional Responsibilities
Sabrina	1.2, 1.3	• Add key input listener with relevant functionality; • Maintain sprint process documentation.
Garry	3.0, 4.2	• Update player statistics; • Update player class with player attacks and costume abilities; • Ensuring best practices for the team's git workflow.
Keith	5.0, 5.1, 5.2	• Complete the combat functionality; • Finish graphics and sprites.
Mahmood	5.3, 8.1, 9.0	• Complete shop GUI; • Create an end game scene for when the player loses; • Add accessibility text to elements; • Project coordinator.

2.3.5 Tasks Completed

Green represents completed user stories.

User Stories	Completion Status
1.2, 1.3	Completed
3.0	Completed
4.2	Completed
5.0, 5.1, 5.2, 5.3	Completed
8.1	Incomplete
9.0	Completed

2.3.6   Product Backlog

Name	ID	Owner	Amendments
Handle Mouse Input	1.2	Sabrina	–
Game Ending	1.3	Sabrina	–
Player Statistics	3.0	Garry	–
Costume Abilities	4.2	Garry	Updated implementation details for how costumes are equipped (now point to the index of equipped costume in the inventory)
Combat Scene	5.0	Keith	–
Attack NPC	5.1	Keith	–
Hit By NPC	5.2	Keith	–
Combat Result	5.3	Mahmood	–
GUI Alt Text	8.1	Mahmood	–
Game Shop GUI	9.0	Mahmood	–

2.3.7   Code Reviews

User Story	Reviewer	Pull Request Link
8.0	Sabrina	Link to PR (!16)
5.0	Mahmood	Link to PR (!20)
1.2	Garry	Link to PR (!17)
5.3	Keith	Link to PR (!24)

2.3.8 Retrospective

Participants	Sabrina, Garry, Keith, Mahmood
Unfinished Tasks	• While we implemented a shop section in the GUI, the game logic itself was not fully implemented such that the user could buy items (into their inventory) and then use them during combat. • The base logic for items was completed, but specific items (like the attack potion and invisibility cloak) were not implemented. • Besides fighting the NPC (which functions as expected), due to time constraints, the other interaction options with the NPC are not implemented fully. • A proper "ending" for the game does not currently exist (besides the "Game Over"). Essentially, there is no way to "beat" the game.
Best Experience	When everything came together, and all the code worked, we could see and play the game we worked so hard on making. When GitLab performed any operation successfully.
Worst Experience	GitLab became bogged by a lot of people using it which led to it being very, very slow. A merge request cycle would take 30 minutes to process fully.
Bad Practices	• A while back, we merged a branch into master directly (without merging to develop first) which went under the radar until this sprint, which led to a lot of headache and file shenanigans.
Revised Practices	• Do NOT merge a branch directly to master. Merge to develop first.

3 Final Summary

As we conclude our project, we are delighted with the achievements and overall progress made during Phase 2. Our game has incorporated many of the features outlined in our user stories. Features include the ability to toggle between a dark and light theme and provide alt text for any graphical elements to make the game accessible for people with vision impairments. We also added different costumes that you find spread out through the map, which adds a feature that allows the player to gain abilities that help when interacting with specific types of NPCs while being somewhat detrimental to other NPCs; the costumes also give the player access to different attacks as a way to keep the player engaged and allow the player to change the way they approach the game. This costume variation also allows the player to have various playthroughs that differ every time they play, meaning they stay engaged and interested for longer. The player can utilize the different costumes along with them to change the gameplay. Lastly, there is a player statistics bar at the bottom of the GUI so that the player can keep track of their progress and see how they are doing. All these features were seamlessly combined in a GUI that enhances the game's functionality and creates an enjoyable user experience.

Although most of the project went seamlessly, discussing the limitations and hurdles we faced is still essential. One of the biggest hurdles in this project was getting used to GitFlow and GitLab. While some of our members were more accustomed to Git than others, all of us were new to GitFlow and GitLab. Initially, figuring out how to begin GitFlow and create branches was confusing. And the problems did not stop there. When we performed our first merge request, we deleted the develop branch, which was problematic. Furthermore, we sometimes merged without requests or to master directly by accident, or we would delete the branch after merging. But, with each sprint, we began making fewer and fewer Git mistakes, and whenever there was one, we worked together to sort the problem as fast as possible. Another problem we had was with loading data files and their related paths. To address this issue, we restructured the project directories, created a central resource folder and loaded the files. Despite these hurdles, we adapted and changed parts of our design to ensure the game stayed close to the original proposal and followed the same user stories and UML diagrams.

In the ever-changing scope of game development, flexibility is essential. Although all the user stories may have yet to be fulfilled, our focus on the game's core features helped ensure we could provide a viable, working and fun game. Any changes made were to keep the team on track and make the development process seamless. During the revision, some user stories were changed as they were too similar to others, and some were changed as they did not fit the revised scope and idea of the game. For instance, the ability to upgrade costumes was not implemented mainly to keep the game simple enough —without having to go through the process of learning to upgrade the costumes and having to grind to upgrade all the available costumes, which is slow, tiresome and boring — so that most people reasonably enjoy the game while still being challenged. Another feature not implemented was the NPCs having natural language dialogue; this feature was not implemented due to the time constraint, as our priorities were to make a functioning game and work on the final report. If there were enough time, this feature would have been implemented using ChatGPT; the dialogue would have been accessed using the ChatGPT API. An example of how some user stories were also adapted and combined is seen with the case of user stories 3.1 and 1.2 or user stories 3.6 conflicting with 5.0 and 5.1, as they essentially encompassed the same idea, which meant combining the two would eliminate redundant user stories

already implemented.

While certain parts of the game may differ from the original plan, these adjustments were made to optimize performance, improve the game's functionality, and address our unforeseen challenges. The journey from start to finish involved some unforeseen challenges, but the final product proves our ability to adapt, improvise and overcome. It also showcases our ability to communicate and collaborate, as well as apply the concepts of SOLID, agile, and git from class during the game development.