Sprint 1 Retrospective: Profpocalypse: Purdue Edition

Team 1: Trey Rosenfeldt, Jonah Irons, Helen Fivecoate, Austin Rogers, Jennifer Sheng, Mohana Barve

What went well?

In general, our team successfully completed all planned user stories and met all acceptance criteria, establishing a strong foundation for the game.

We implemented game state saving and loading, a save deletion system, and a pause menu with volume, brightness, and contrast controls. A custom key binding system was also integrated, ensuring flexible player controls. We fully developed character customization, allowing players to modify their head, hairstyles, facial features, and clothing, with selections properly saved and reloaded. Additionally, custom and pre-made character management was implemented. The HUD and inventory system were completed, enabling players to track health, experience, actions, and items. For navigation, we built a dynamic mini-map of Purdue University and a quest system to track objectives. The game's movement system was also finalized, ensuring smooth keyboard-based navigation. To enhance the experience, we implemented a graduation progress system, allowing players to track coursework, complete classes, and view prerequisite information. A tutorial system was also developed to introduce players to the game's controls and mechanics.

Overall, this sprint was highly productive, with everyone completing their assigned user stories and successfully meeting all acceptance criteria. The team efficiently adapted to Godot, collaborated effectively, and laid the groundwork for expanding gameplay features in future sprints.

User Story #1
#2: As a player, I would like to save my game and be able to log into the same location, and point of the story I was at when I exited.

#	Description	Estimated Time	Owner
1	Devise game state saving implementation	1 Hr	Jonah
2	Create a save function and a corresponding save button	3 Hrs	Jonah
3	Debug and play test	2 Hrs	Jonah
4	Connect to the menu UI	2 Hr	Jonah

Completed:
The game saving was fully implemented for our current game state, and works in all edge cases of game disruption (force quit, save and exit, etc). This functionality is up to date for all of the components we currently need to save, but will need to be consistently updated as we add more and more features to the game, as this means we must save more things. This is connected to the settings menu UI through buttons, and the game can be saved from the explicit save button, or when quitting the game through the menu, where it automatically saves the game for the user.

User Story #2
#3: As a player, I would like to delete my save from the game.

#	Description	Estimated Time	Owner
1	Create a delete function and a corresponding delete button	3 Hrs	Jonah
2	Debug and play test	2 Hrs	Jonah
3	Connect to the menu UI	2 Hr	Jonah

Completed:
Similar to the previous user story (#1), the delete function works for our current game state. It simply deletes the save file from the local storage, and so the game cannot be loaded back to its previous game state. It is reset, and the player must start again from the beginning. This is also connected to the settings menu UI via a button, and when the player selects the “delete save”

button, a pop-up confirmation prompts the user to confirm they would like to delete the save, where the game is then closed upon confirmation.

User Story #3
#4: As a player, I would like to interact with a pause screen to adjust the volume of sound effects and music.

#	Description	Estimated Time	Owner
1	Create a pause button on the display	2 Hrs	Jonah
2	Create an interactable pause menu UI	5 Hrs	Jonah
3	Create a volume control for music + SFX	2 Hrs	Jonah
4	Connect to the menu UI	1 Hr	Jonah
5	Debug and play test	1 Hr	Jonah

Completed:
This story was fully completed, and all acceptance criteria were met. This means that we have an interactable settings menu UI that is opened using the “escape” key. Here you can access all settings for the game, such as saving, deleting, mapping keybindings, and manipulating preferences such as volume. This menu can also be opened by the menu button on the HUD, which gives the user ways to access the menu via keyboard or mouse. This preference persists after closing the game, assuming the game state has been saved. If the game is deleted, then the volume defaults to the original max volume.

User Story #4
#5: As a player, I would like to interact with a pause screen to adjust the brightness of the screen and contrast.

#	Description	Estimated Time	Owner
1	Create a brightness/screen contrast control	2 Hrs	Jonah
2	Connect to the menu UI	1 Hr	Jonah
3	Debug and play test	1 Hr	Jonah

Completed:
This story was fully implemented. The menu UI (which has already been created at this point), contains a preferences tab, where this setting resides. There is a slider for adjusting game brightness, that the user can manipulate to lower or raise the brightness of the game. This preference persists after closing the game, assuming the game state has been saved. If the game is deleted, then the brightness defaults to the original max brightness.

User Story #5
#6: As a player, I would like to be able to set custom key bindings for controls in the game.

#	Description	Estimated Time	Owner
1	Creation of a menu where default key bindings can be inspected	2 Hrs	Austin
2	Allow for custom key bindings settings	2 Hrs	Austin
3	Create a method for saving the new key bindings once entered	2 Hrs	Austin
4	Connect key binding menu to the pause menu	2 Hrs	Austin

Acceptance Criteria:

- Given that the player is within the game, they will be able to access a menu where they can view the default key bindings and their functionality
- Given that the player is within the key binding menu, they will be able to change the key binds to their liking
- Given that a new key binding has been entered, the player will be able to save their bindings so they stay until changed again

Completed: This story was fully completed, and all acceptance criteria were met. This means that when the player is within the game they can access a menu with the default key bindings and their functionality. They can also change the keybindings to their liking and it will save until the player decides to change them again.

User Story #6
#7: As a player, I would like to be able to move through the map using keyboard controls (up, down, left, right).

#	Description	Estimated Time	Owner
1	Learning + Understanding the basics of Godot	5 Hrs	Austin
2	Determining the ideal base key mappings for the player	2 Hrs	Austin
3	Creating functionality and response when the player presses a key	5 Hrs	Austin

Acceptance Criteria:

- Given that the player is using a keyboard, the default key bindings will be in comfortable positions that are intuitive and fluid to use
- Given that the actions available to a player are planned out, there will be a keyboard control for each action
- Given that the user presses a key, that keys specific action will perform within the game
- Given that I understand the fundamentals of Godot I will be able to create a framework for basic movement before I implement one into our project

Completed: This story was fully completed, and all acceptance criteria were met. This means that if the player is using a keyboard the default keybindings are well placed. Also there are keyboard controls for each action within the game and when the key is pressed the corresponding action will be performed within the game. I also created a tutorial for my own practice for the movement fundamentals in Godot.

User Story #7

#8: As a player, I would like to be able to view a mini-map of Purdue University for easier navigation during exploration and quests.

#	Description	Estimated Time	Owner
1	Track player's location and scale it down	5 Hr	Jennifer
2	Design mini-map (colors, layout)	2 Hr	Jennifer
3	Playtest and debug	3 Hr	Jennifer

Completed:

This story was fully completed, and all acceptance criteria were met. This means that when the

player is playing the game and exploring, the mini-map displays a scaled-down version of Purdue University's campus. Additionally, the mini-map accurately updates its location with the player's without any lag, and does not overlap with other UI elements.

User Story #8
#9: As a player and Purdue student, I would like to explore where buildings are on the main campus and their names.

#	Description	Estimated Time	Owner
1	Brainstorm map color scheme	1 Hr	Jennifer
2	Create the basic main campus map layout plan	5 Hr	Jennifer
3	Create tilemap assets	4 Hr	Jennifer
4	Add icons on the main playing screen that players click to pause, access inventory, view characters, view achievements, and view quests	4 Hr	Jennifer
5	Make a pop up with the name of the building that the player hovers their mouse over	3 Hr	Jennifer
6	Playtest and debug	3 Hr	Jennifer

Completed:
This story was fully completed, and all acceptance criteria were met. This means that when the player explores the map, the same location and buildings and displayed in the mini-map. When the player hovers their mouse over a building, a clear label with its name appears. When the player clicks UI buttons such as the pause or save buttons, the map retains its functionality and all other button functions are also unaffected.

User Story #9
#24: As a player, I would like to save the characters that I create in a database and also access a database of pre-made characters throughout the game

#	Description	Estimated Time	Owner

1	Learn how to use Godot and set up a plan on how the character database should be implemented.	5 Hr	Mohana
2	Create a system for saving custom characters into a database.	2 Hr	Mohana
3	Implement a system to load and display pre-made characters from the database.	1 Hr	Mohana
4	Implement UI for selecting and switching between custom and pre-made characters.	1 Hr	Mohana
5	Debug and test to ensure the character selection and saving systems work properly.	1 Hr	Mohana

Completed:

A full character saving and loading system was implemented, allowing players to save their custom characters locally and retrieve them when needed. Pre-made characters were also made accessible through a selection interface, ensuring a smooth transition between saved and default options. The system successfully retained the last selected character upon exiting and reloading the game, meeting all acceptance criteria. Debugging and testing confirmed that characters loaded correctly and all data persisted as expected.

User Story #10
#25: As a player, I would like to customize my character's head, including hairstyles, facial hair, mouth structure, and eyes.

#	Description	Estimated Time	Owner
1	Create a design for the character customization UI focused on head features.	3 Hr	Mohana
2	Implement functionality for selecting different hairstyles, facial hair, and face details.	3 Hr	Mohana
3	Implement a system to apply selected head customizations to the character model.	3 Hr	Mohana
4	Save selected head customization settings to local storage.	3 Hr	Mohana
5	Debug and test to ensure head customization options	3 Hr	Mohana

	function correctly.		

Completed:

The character customization system was fully developed, enabling players to modify their character's head attributes, including different hairstyles, facial structures, and other key facial features. The customization interface updates in real-time, and selections are saved and reloaded accurately. All acceptance criteria were met, ensuring a seamless and responsive customization experience.

User Story #11
#26: As a player, I would like to customize my character's clothing and colors, including outfits, ties, bowties, and overall color schemes.

#	Description	Estimated Time	Owner
1	Create a UI for customizing the character's outfit and colors.	1 Hr	Mohana
2	Implement outfit selection, including different clothing types (shirts, jackets, accessories, etc.).	1 Hr	Mohana
3	Implement color customization for different clothing parts.	1 Hr	Mohana
4	Save selected outfit and color choices in local storage.	1 Hr	Mohana
5	Debug and test the customization system to ensure proper application of outfits and colors.	1 Hr	Mohana

Completed:

The customization system now allows players to select different outfits and modify colors for various clothing items. The system ensures that selections are applied immediately, and all changes are correctly saved and reloaded when the game is restarted. Debugging and testing verified that clothing selections persist and display correctly upon reloading, fully meeting all acceptance criteria.

User Story #12
#32: As a player, I would like to be able to view my character's item inventory outside of battle.

#	Description	Estimated Time	Owner
1	Create a design for the inventory.	1 Hrs	Trey
2	Implement the design for the inventory and allow the item bar to connect with the inventory UI	3 Hr	Trey
3	Allow to have new items into the inventory and delete items in the inventory	2 Hrs	Trey
4	Implement a use option for usable healing potions.	3 Hrs	Trey
4	Debug and test the program to make sure all UIs work together	3 Hrs	Trey

Completed:

The inventory UI is complete with the ability to do everything you would want. First You are able to open and close the inventory with one click. The user will be able to click and unclick on items which allows you to move the items, use the items, and delete the items. This means that a user can organize their inventory how they want and move the items around, as well as being able to remove any items they want if they are not useful anymore, and you are able to use potions or other usables that will give benefits like heal the character, or give experience to level up. Finally the user will be able to get new items in their inventory, as of now it is a button that randomizes the buttons but is built for testing until we have battles.

User Story #13

#38: As a Player, I would like to have a HUD with things like being able to see how many attacks I have left and being able to see how much more experience I need to level up and a health bar.

#	Description	Estimated Time	Owner
1	Create a design for the HUD and how it will interact with others	1 Hr	Trey
2	Implement a health bar, item bar, and any other needed for the layout	3 Hrs	Trey

3	Implement experience and a level bar	2 Hrs	Trey
4	Debug and test response time with the HUD	3 Hrs	Trey

Completed:
The skeleton of the main HUD (Heads-Up Display) is finished. Most of the UI for the display is complete but may add more for the future so as of now it has all of the needed components for the final game. The user is able to see their current health, experience, have a weapon bar and potion bar, and have a way to see the current tasks they are working on. First the weapon bar and potion bar work with the inventory and you can put the items into their respective bar to then use them without going into the inventory. Second with the ability of losing health you can use health potions to gain health either in the inventory or in the potion bar which will update how much health the user has as well as having an experience bar where you can increase experience and increase your level. Finally with the quest bar you are able to pin wanted available quests and then can see them while you are not in the quest menu. This all works together with having low latency making the game load in relatively quickly.

User Story #14
#47: As a player, I would like to have a quest menu so I can see what I need to complete, as well as potential rewards for completing these quests.

#	Description	Estimated Time	Owner
1	Create a design for the quest screen and how it will interact with the main HUD	1 Hr	Trey
2	Implement a UI panel to display quest information as well as the reward you will receive for completing the quest.	4 Hrs	Trey
3	Implement an addition to the main HUD to see the quests you are on now.	2 Hrs	Trey
4	Debug and test the program	2 Hrs	Trey

Completed:
The Quest UI is complete with the ability to complete quests, have prerequisites for quests, ability to pin quests, have a description for quests, and give the reward to the users inventory. The user will have a lot of quests in the future so with that I was able to create an easy way to make new quests, with that they can have prerequisites with any quest already in the game, any reward in the game, and have a description specific for the quest. Along with this the UI has a separate location for locked quests and available quests and when you pin a quest they will

appear in the main HUD and will appear on the top of the available quests. When a player finishes a quest the quest disappears, the reward gets added to the inventory, and the quests that required the quest to become available become available.

User Story #15
#50: As a player, I would like to go onto a computer UI to see my graduation status and when I need to do my classes.

#	Description	Estimated Time	Owner
1	Create a phone icon that players can click to access information about status and course information.	1 Hr	Helen
2	Create a screen with sections for graduation status, current courses, and a link to information about further classes.	2 Hrs	Helen
3	Create a progress bar to indicate graduation status.	3 Hrs	Helen
4	Develop section that details current course information including information about the lecture hall/classroom, professor, time, and course description.	3 Hrs	Helen
5	Test and debug.	1 Hr	Helen

Completed:
The phone screen that displays the graduation status and courses is complete. There are two tabs on the main phone screen that allow the player to see the current courses and the total courses. In addition there is a progress bar that updates when a player competes a semester. The progress bar and current courses are temporarily updated through complete semester and new semester buttons. These attributes will help players to keep on track of their current courses and how far into the game they are. The game is now set up to accommodate this user story.

User Story #16
#52: As a player, I would like to be able to have a tutorial experience to ease me into the controls of the game.

#	Description	Estimated Time	Owner
1	Creating a button on the menu to access the tutorial	1 Hrs	Austin

2	Developing the layout of where the character will move through in the tutorial	4 Hr	Austin
3	Designing the tutorial	5 Hrs	Austin

Acceptance Criteria:

- Given that the menu is functional, the player will be able to access the tutorial by pressing a button
- Given that the tutorial is able to be accessed, the layout will be a miniature section of the Purdue campus that will give the user a good idea of what they will be playing in the full game
- Given that the tutorial is functional, the player will be able to pass it if they follow along with the provided instructions

Completed: There is now a button in the setting menu that will bring up the tutorial for the player. The tutorial takes place in the same minimap of Purdue that the game takes place in. The tutorial provides the player with a map of all the keybindings in the game and their functionality. It also tells the to use all of the buttons and is able to be successfully completed once the user does so.

User Story #17

#54: As a player and Purdue student, I would like to see the classes I will take each semester and which classes I need for prerequisites.

#	Description	Estimated Time	Owner
1	Set up an outline for and create a screen that displays from the button clicked on the current information page.	2 Hrs	Helen
2	Create a display for the player's current degree plan.	4 Hrs	Helen
3	Create more information buttons for each course.	1 Hrs	Helen
4	Create a button that displays the course prerequisite flowchart.	2 Hrs	Helen
5	Test and debug.	2 Hr	Helen

Completed:
There is now a section on the phone screen that displays the players plan of study and the prerequisites each course needs. On the phone screen there are two tabs and the second displays the plan of study. There are 8 sections, one for each semester, that display the courses the player will need to complete to graduate. Their current semester is highlighted yellow, completed semesters are green, and incomplete semesters are red. Each course on the plan of study is a button that displays more information including the course's prerequisites. Those prerequisites are buttons that take you to their information page so the player can learn a little more about them. In addition to this information page, there is now also a button that will display the course prerequisite flowchart so players can see in a more visual manner.

User Story #18
#56: As a player and Purdue student, I would like to learn where the classes are, whether that is large lecture halls or small classes.

#	Description	Estimated Time	Owner
1	Import list of courses, information about them, and their prerequisites.	5 Hrs	Helen
2	Create popups for each course to display more information including a description of the course, professor, location with a description, and prerequisites.	3 Hrs	Helen
3	Test and debug	1 Hr	Helen

Completed:
The game now helps students to learn about the locations of the classes they are taking. On the phone screen in the current course information and the more information page for each course it displays the location of that course. In addition, in those same places there is a place for a short description of the location, for now they mostly say that it is a large lecture hall. Players will be able to use this to learn a little more about the classes in a clear cut manner.

What did not go well?

We were able to complete all of our user stories. This meant we were able to complete all the features we planned to as well as create strong manual tests to be sure about each user story. A few things we did not think about when we started but may need to be reworked later include:

how we call the objects for items, how we create the characters, and how to complete quests. First, we call objects inside inventory, and looking at how we create quests we decided this would be a better way to create items because it will allow for more items to be made and a more efficient way of updating and organizing items. Second, we built a front view of the player model but in a sky-view game, we now have to figure out how to create a player model for that aspect of the game. Lastly, the implementation of quests is well done but to be able to auto-complete quests and achievements has not yet been implemented and will be a team decision on how we get that done. Overall our team chose the right way to start by first learning how to use Godot and then implementing our user stories with strong brainstorming so we could optimize our time and do it correctly first.

How should you improve?

Overall, this was a very productive first sprint, and the team worked very well together to make sure we achieved what we set out to do at the beginning of the sprint. Additionally, we communicated very well within our group, and held weekly meetings that were very productive in discussing the next steps for our work in the upcoming weeks.

If there was something we needed to improve on, I would say we need to plan ahead a little more in depth for our next sprint. We would often plan ahead in general with what stories we were going to do, but past this, we didn't really know what we were getting into before we were already starting it. If we started planning ahead at an even finer level (such as certain days within the week we want to have certain parts done), then we can ensure we will be perfectly on time with a spaced out work schedule, instead of cramming our work in on the last day. Cramming in work on the last day can make it difficult for other team members if the work you did affects other work, or if other parts depend on these last minute updates. Being more on top in this area will increase overall productivity as a group, and will make for a much smoother workflow.

Additionally, being more communicative of our work specifically is an area we need to improve in. In this past sprint, we didn't really discuss too much the specifics of our work - it was generally up to yourself to pull the changes and figure out what has changed. This can cause

problems, as all of our parts are generally intertwined, so understanding of most aspects of the project is essential. If we take the time to communicate the changes we have made as we push them, it will help us all to understand the structure of the project more, which will smooth out integration of our parts and help ensure we are consistent in our code and design approaches throughout the project. Thus, being generally more communicative of our work and what we have been doing will help us work together easier and more effectively, which is ideal.