

## Manual Testing Doc for Sprint 2

### User Story #1

1. **Acceptance Criteria:** There will be different music playing while the user is in different sections of the map.
  - a. **Description:** This test checks that there is music playing in the background, and when the user moves to a new section of the map, the music will change.
  - b. **TC01**
    - i. **Input:** The user loads the game, and observes that there is music playing in the background. They then move to a new section of the map (up, down, left, or right).
    - ii. **Output:** The user observes that when they move to a new section of the map, whether that is up, down, left, or right, the music will transition to a new track in the background. This is the intended and acceptable output.
2. **Acceptance Criteria:** When the user loads up a game save, the audio playing on open will correspond to the in game location.
  - a. **Description:** This test will check that when the user loads the game, the area that they spawn in will be playing the corresponding music for that area. If the user saves the game in a new area from where they started, and restarts the game, the audio on startup will be the correct music.
  - b. **TC01**
    - i. **Input:** The user loads the game, and observes what music is playing. They then move to a new area of the map, save the game, and exit. Then, the user reloads the game with this new save.
    - ii. **Output:** The user observes that the music being played on startup reflects the correct music for the new area that they moved to before saving and quitting the game previously. This is the intended and acceptable output.
3. **Acceptance Criteria:** Given the game is paused, the music in the background will lower / stop, and will resume when the settings menu is closed again.
  - a. **Description:** This test checks that when the settings menu is opened, the audio volume will decrease by a set amount, and this will persist for the duration of this menu being open, and the audio will be restored to the level indicated by the volume slider once the menu is closed again.
  - b. **TC01**
    - i. **Input:** The user loads the game, and opens the settings menu. After a short period of time, the user closes the settings menu.
    - ii. **Output:** The user observes the level that the audio is at before opening the settings menu, and upon opening the menu, the user observes the audio dropping to a lower level. When the user closes the menu again, the user observes that this level has returned to the indicated volume from the volume slider. If the volume slider panel is opened during this process, the audio reverts to the true indicated volume, to ensure proper volume levels can be set. This is the intended and acceptable output.

## User Story #2

1. **Acceptance Criteria:** Given the battle state has been implemented, if the user enters battle, then a different battle music will play.
  - a. **Description:** This test checks that the music transitions to a new track for the battle scene.
  - b. **TC01**
    - i. **Input:** The user walks up to an enemy and triggers the battle scene.
    - ii. **Output:** The user observes that when they enter the battle scene, the music will transition to a new track that is unique to the battle scene. This music persists through the entirety of the battle scene. This is the intended and acceptable output.
2. **Acceptance Criteria:** Given the battle state has been implemented, if the user exits battle in any way then the battle music will end.
  - a. **Description:** This test checks that when a user exits a battle scene, the battle music that was playing no longer continues to play while in the main area of the game.
  - b. **TC01**
    - i. **Input:** The user walks up to an enemy and triggers the battle scene. They observe that the music playing is the unique battle music. The user then presses the “leave” button in the battle scene, until they successfully escape the battle. The user is then returned to the main game where they were before entering battle.
    - ii. **Output:** The user observes that when they exit the battle, the battle music that was playing is transitioned out and is no longer playing while in the main area of the game. This is the intended and acceptable output.
3. **Acceptance Criteria:** Given the battle state has been implemented, the appropriate music will start playing again.
  - a. **Description:** This test checks that when a user exits a battle scene, the music will return to the music that was playing prior to entering the battle scene.
  - b. **TC01**
    - i. **Input:** The user walks up to an enemy and triggers the battle scene. They observe that the music playing is the unique battle music. The user then presses the “leave” button in the battle scene, until they successfully escape the battle. The user is then returned to the main game where they were before entering battle.
    - ii. **Output:** The user observes that when they exit the battle, the battle music that was playing is transitioned to the corresponding music track for that area of the map, which was playing prior to entering the battle. This is the intended and acceptable output.

## User Story #3

1. **Acceptance Criteria:** The user will be able to hear different SFX depending on what UI/menu interaction occurs, i.e. opening the inventory vs clicking the menu button.
  - a. **Description:** This test will verify that different UI interactions will produce

different sound effects.

b. TC01

- i. **Input:** The user will open and close the settings menu. And will then open the inventory. While in the inventory, the user will use the “Add item” button to spawn items in the inventory. The user will move some items to the item bar, and some potions to the potion bar.
- ii. **Output:** The user will observe that opening and closing the settings menu will produce a page-turning sound effect. Opening the inventory produces a bag-opening sound effect. Adding items and potions, as well as moving them around produces a clanging metal sound effect and a bubbling potion sound effect, respectively. This is the intended and acceptable output.

2. **Acceptance Criteria:** The user will be able to hear different SFX for player manipulation, such as walking sounds and battle sounds.

a. **Description:** This test will verify that moving the character and certain aspects of battling produce unique SFX.

b. TC01

- i. **Input:** The user will move the character around, and then will move towards an enemy to initiate a battle sequence. In the battle, the user will attack with an item, and will take damage from the enemy. The user will use a healing potion to heal the damage they took. The user will then attempt to flee the battle, but will fail. After doing so, the user will successfully flee the battle.
- ii. **Output:** The user will observe that moving the character around produces a walking sound effect. When the battle sequence is started, the user will observe that attacking with an item produces a losing health sound effect. Taking damage from the enemy produces the same sound effect. After using the healing potion, we can observe that there is a gaining health sound effect. When the user fails an attempt to flee, there will be a failure sound effect played. When the user successfully exits the battle, there will be a success sound effect played. This is the intended and acceptable output.

3. **Acceptance Criteria:** The user will be able to adjust the level of SFX using the volume slider in the settings.

a. **Description:** This test will ensure that using the volume slider in the settings will adjust the volume level of all SFX accordingly.

b. TC01

- i. **Input:** The user will be on a new save. The user will open the inventory and add some items. They will then go into the settings menu, open the preferences menu, and lower the volume to just above 0. They will then leave the settings menu, go back out into inventory, and repeat the original step of adding items.
- ii. **Output:** The user will observe the volume level of the item/potion SFX in the inventory before adjusting the volume setting, and then will observe it

again after adjusting the volume setting. The observed effect will be that the volume level of the SFX after adjusting them in the settings will be noticeably lower, almost 0. This is the intended and acceptable output.

## User Story #4

1. **Acceptance Criteria:** The user will be able to see different enemies located around the map.
  - a. **Description:** This test ensures that multiple enemies are located in various locations around the map.
  - b. **TC01**
    - i. **Input:** The user will create a new save. Once the game is running, they will walk upwards towards Wetherill. After doing so, they will walk even further upwards to Ross-Ade Stadium.
    - ii. **Output:** The user will be able to observe a blue enemy positioned in front of Wetherill. After doing so, they will be able to observe a green enemy in front of Ross-Ade stadium. This is the intended and acceptable output.
2. **Acceptance Criteria:** The user will be able to interact with the random enemies to initiate a battle sequence.
  - a. **Description:** This test will verify that the various enemies located around the map are interactable, not just static sprites.
  - b. **TC01**
    - i. **Input:** The user will create a new save. Once the game is running, the user will walk upwards towards Wetherill, and approach the blue enemy. Once inside the battle sequence, they will use one item to attack and do damage to the enemy. They will then use healing potion after the enemy attacks and deals damage to them. Then, they will leave the battle sequence. The user will then walk upwards towards Ross-Ade stadium, and will repeat the above steps with the green enemy.
    - ii. **Output:** The user will observe that they can approach the blue enemy, and go through a full battle sequence with it. The item attack will do damage to the enemy, and they will take damage from the enemy. They will also be able to heal this damage with a healing potion, and will be able to leave the battle (given they beat the chances of failing to leave). The user will then observe the same behavior with the next enemy, the green enemy in Ross-Ade. This is the intended and acceptable output.
3. **Acceptance Criteria:** Given an enemy has been defeated, the enemy will not be able to be battled again on the current save.
  - a. **Description:** This test will verify that defeating an enemy will have them disappear from the game, unable to be battled again unless you go back to a previous save or make a new game.
  - b. **TC01**
    - i. **Input:** To speed up the process of this test, we will adjust the health level of our test blue enemy to 1 hp, ensuring a win on the first successful

attack from the user. The user will create a new game. They will then walk upwards towards Wetherill, where they will approach the blue enemy. During the battle sequence, they will attack the enemy with the pickaxe item. Once the battle has been one, the user will save the game, and exit the game. They will then reload the game from this save.

- ii. **Output:** Once the game has been loaded, we will be able to observe the blue enemy positioned in front of Wetherill. The battle sequence will start when the user approaches the enemy. The user will attack the blue enemy, and we will see the enemy's health reach 0. We will see a "You Won!" message on the screen, and the battle sequence will end. When the player is returned to the main map, we can observe that the blue enemy is no longer there, and approaching the position they were in elicits no action for the battle sequence. After the user saves and reloads the game, we can observe that the enemy does not respawn, as the enemy has already been defeated on this save. This is the intended and acceptable output.

## User Story #5

1. **Acceptance Criteria:** Given that the player is exploring the world, when the player moves close to a building and clicks it, a pop up should appear asking if the player wishes to enter the building
  - a. **Description:** This test verifies that only buildings that the player can see on their main playing screen can be entered by clicking on the desired building and answering the pop up prompt
  - b. **TC01**
    - i. **Input:** The player walks up to a part of the map where their desired building is located. Then, the player uses their mouse to hover over the building and click on it.
    - ii. **Output:** A pop up prompt appears asking the player if they wish to enter the selected building, labeled by its abbreviation such as ARMS. The player can then choose between two buttons: Yes or No. Yes takes the player to the building interior map and closes the prompt, while No simply closes the prompt.
2. **Acceptance Criteria:** Given that the player has entered the building, they should be taken to separate, simpler "map" with a few classrooms
  - a. **Description:** This test verifies that the player is able to enter buildings properly without anything breaking with UI or other functionality.
  - b. **TC01**
    - i. **Input:** The player clicks on Yes to a building enter prompt.
    - ii. **Output:** The player sees themselves in a simpler interior building map that they can freely move around in. Walking around, they should see three classrooms each with desks inside that they can enter.
3. **Acceptance Criteria:** Given that the player is exploring the building, there should be proper collision interactions in place.
  - a. **Description:** This test verifies that proper collision physics are implemented inside buildings.

- b. **TC01**
  - i. **Input:** Once the player is inside a building, they will walk to each of the four main walls. They will also enter classrooms and walk to each of those walls, as well as desks placed inside.
  - ii. **Output:** Upon walking to those walls and objects, the player is stopped in place and does not phase through anything. The camera also follows the player the entire time and does not obscure anything from view.

### User Story #6

- 1. **Acceptance Criteria:** Given that the player has entered a building, they should see a simple building layout map that they can explore and walk around in.
  - a. **Description:** This test verifies that there is a building/classroom design visible to the players.
  - b. **TC01**
    - i. **Input:** After entering the building, the player visually scans the playing screen and walks around.
    - ii. **Output:** The player should see three classrooms in each building, each equipped with the same amount of desks in the same layout.
- 2. **Acceptance Criteria:** Given that the player has entered a building, they should be able to enter different classrooms that are clearly represented by the building layout.
  - a. **Description:** This test verifies that classrooms are clearly designed and are able to be entered.
  - b. **TC01**
    - i. **Input:** After entering the building, the player walks up to any of the three classroom doorways.
    - ii. **Output:** The player sees clearly that there are classrooms that can be entered, and that they are able to freely enter and leave each one while exploring.
- 3. **Acceptance Criteria:** Given that the player has entered a building, they should see a tilemap/color scheme that is consistent with the rest of the game.
  - a. **Description:** This test verifies that the artstyle of the building interiors is consistent with the rest of the game.
  - b. **TC01**
    - i. **Input:** After the player enters the building, the player visually scans the tilemap and design of the map.

### User Story #7

- 1. **Acceptance Criteria:** Given that the player is exploring the main campus, there should be sparkle icons on objects and NPCs that provide trivia facts, which can be clicked when close.
  - a. **Description:** This test verifies that there are visual distinctions for places to earn trivia facts, and that they are able to be interacted with when the player sees them

on the playing screen.

b. **TC01**

- i. **Input:** The player walks around the map until they see a yellow sparkle icon on their player screen. Then, the player clicks on the sparkle icon and any associated sprites if any.
- ii. **Output:** A pop up appears letting the player know they've learned a trivia fact. A line of text at the bottom lets the player know if this is the first time they've learned this trivia fact and the number of the fact. The player is able to close the pop up via an x button in the upper left of the pop up.

**2. Acceptance Criteria:** Given that the player is learning new trivia, they should be added to a notebook.

- a. **Description:** This test verifies that a trivia book is being properly updated as the player learns more trivia facts.

b. **TC01**

- i. **Input:** The player first opens the trivia book at the very beginning of the game. Then, the player explores and clicks on an interactable item. After doing so and closing the pop up, the player opens the trivia book again.
- ii. **Output:** After the initial opening of the trivia book, each trivia fact should be numbered with “????” next to each one. There are up to four trivia facts on each page, and the player is able to flip freely in the book. The player cannot flip to less than the first page or greater than the last page. Page numbers are properly updated as the player flips. After the player earns trivia facts, upon opening the trivia book again, the corresponding trivia fact is filled in and no longer has question marks. These unlocked facts stay unlocked even if the player closes and opens the book again.

**3. Acceptance Criteria:** Given that the player wishes to look back at trivia that they have collected, there should be a book icon somewhere on the main HUD or pause menu that the player can press to access the notebook

- a. **Description:** This test verifies that there is a functioning book button that the player can press to access the trivia book.

- i. **Input:** The player goes to the upper left of the screen and clicks on the book icon. Then, the player clicks it again.
- ii. **Output:** After clicking the button the first time, the trivia book is visible in the middle of the screen. After the second time, the trivia book is no longer visible. The state of the book's contents remains constant with every open and close as long as the player isn't unlocking more trivia. If the player unlocks more trivia, it is properly updated as detailed by the last acceptance criteria.

**4. Acceptance Criteria:** Given that the player is viewing the notebook, there should be an indicator telling the player how many trivia facts they have and haven't learned

- a. **Description:** This test verifies that there is a functioning counter on the trivia book that keeps count of how many trivia facts the player has unlocked.

b. **TC01**

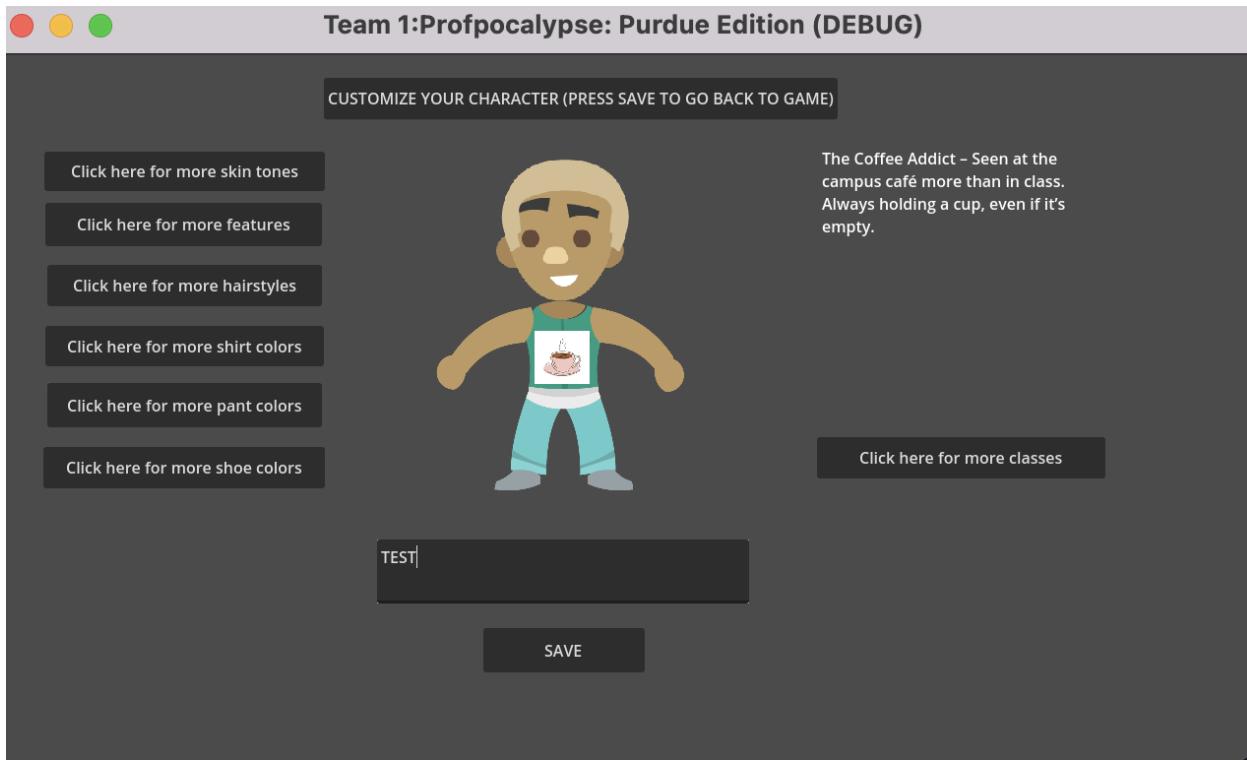
- i. **Input:** The player opens the trivia book and looks at the bottom of the left page. Then, the player closes the book and goes to look for an

undiscovered trivia fact. After clicking on an interactable trivia icon, the player opens the trivia book again and checks the bottom of the left page again.

- ii. **Output:** Upon opening the book, a counter is clearly visible on the bottom of the left page in the format <Unlocked: # unlocked / total #>. After the player closes the book, collects a new fact, and opens the book back up, the # unlocked part of the counter has increased by 1.

## User Story #8

1. **Acceptance Criteria:** Given a player selects a class for their character, then a corresponding class icon should appear on the character's body.
  - a. **Description:** This test verifies that selecting a class from the character customization menu applies the correct class icon on the character.
  - b. **TC01**
    - i. **Input:** The player opens the customization screen and selects the "Coffee Addict" class.
    - ii. **Output:** A coffee cup icon appears on the character's body.
2. **Acceptance Criteria:** Given a player saves their character, then the class selection and icon should be stored along with other customizations.
  - c. **Description:** This test checks that a selected class persists after saving the character.
  - d. **TC01**
    - i. **Input:** The player selects the "Coffee Addict" class and clicks "Save".
    - ii. **Output:** Saved data includes "Coffee Addict" class and its corresponding icon.
3. **Acceptance Criteria:** Given a player reloads their saved character, then the class and icon should persist.
  - e. **Description:** This test checks the reloading process for class persistence.
  - f. **TC01**
    - i. **Input:** The player reloads a previously saved customized character.
    - ii. **Output:** The selected class and its icon are visible on the character.



## User Story #9

1. **Acceptance Criteria:** Given a player selects a course, then the difficulty rating should be displayed.
  - g. **Description:** This test ensures difficulty is shown for each course.
  - h. **TC01**
    - i. **Input:** The player selects "CS 240" from the course list.
    - ii. **Output:** A difficulty rating (e.g., ★★) is shown. Yes
2. **Acceptance Criteria:** Given a player exits and reopens the course menu, then the difficulty ratings should persist.
  - i. **Description:** This test checks that difficulty ratings are saved and reloaded properly.
  - j. **TC01**
    - i. **Input:** The player closes and reopens the phone menu, then checks "CS 240".
    - ii. **Output:** "CS 240" still shows its difficulty rating (e.g., ★★).

**3. Acceptance Criteria:** Given a player switches between different courses, the difficulty rating should update accordingly.

**k. Description:** This test ensures difficulty dynamically updates when browsing.

**l. TC01**

**i. Input:** The player clicks "CS 250", then "CS 354".

**ii. Output:** Each course updates the visible difficulty display correctly.



## User Story #10

**1. Acceptance Criteria:** Given a player selects a professor, then their associated courses and office location should be displayed.

**m. Description:** This test confirms clicking a professor shows correct course and office info.

**n. TC01**

**i. Input:** The player hovers or clicks on "Prof. Gust-Stack".

**ii. Output:** "LWSN 3154G" is shown as the office, and related course info appears.

**2. Acceptance Criteria:** Given a player exits and reopens the professor menu, then the

information should persist.

- o. **Description:** This test checks if professor data is retained between menu loads.
- p. **TC01**

i. **Input:** The player opens the "Professors" tab, views "Prof. Turkey", closes, then reopens it.

ii. **Output:** Prof. Turkey" still shows the correct office and courses.

3. **Acceptance Criteria:** Given the player switches between different professors, their associated courses and office locations should update accordingly.

- q. **Description:** This test ensures professor info dynamically updates.
- r. **TC01**

i. **Input:** The player hovers over "Prof. Doomsmore", then "Prof. BugSquasher".

ii. **Output:** The office and course info change accordingly for each.



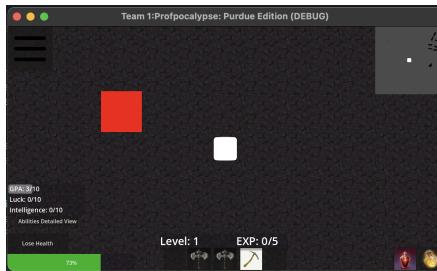
## User Story #11

1. **Acceptance Criteria:** Given the player walks up to a specific object, then the Battle UI will open.
- a. **Description:** This test verifies that if the player walks near the enemy, they will open up the battle UI.

b. **TC02**

- i. **Input:** Use the movement keys (W, A, S, D) to move the user to the enemy(red box)

1. Before walking next to the red box or into the red box



- ii. **Output:** After the user walks into the red box, as you can see the battle UI opens and the battle starts as well, the terminal also prints “Entered Body”.



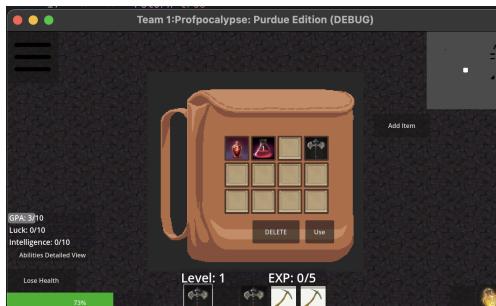
2. **Acceptance Criteria:** Given the player wants to use a weapon, then they can see the weapons they can use.

- a. **Description:** This test verifies that the weapons they have obtained through the game are still visible in the correct potions when in the game.

b. **TC01**

- i. **Input:** Move an item from the inventory to the main HUD and move items around in the main HUD.

1. Before walking next to the red box or into the red box



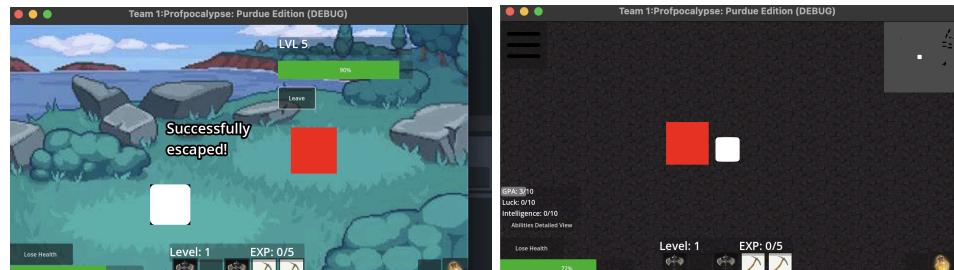
- ii. **Output:** After the user walks into the red box, as you can see, the items are in the same position when they leave or enter the fight, as well as if they lose health or lose an item in the fight, this will save for after the game too.



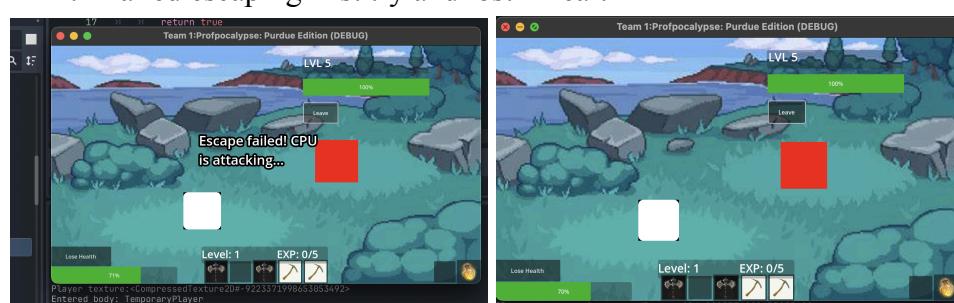
3. **Acceptance Criteria:** Given the player runs tries to run away, then a chance will allow the user to leave the fight.

- a. **Description:** This test verifies That if the user want to leave a fight can with a chance, if the user fails, then the user cannot attack and the enemy attacks again, but if the user escapes, then no more damage will be inflicted on the user and the user can leave the fight.
- b. **TC01**
  - i. **Input:** Click the leave button with your mouse or mousepad.
  - ii. **Output:** After clicking, you will have a chance where you will either be successful and can leave or will take damage and will still be in the fight.

1. Escaping first try



2. Failed escaping first try and lost 1 health



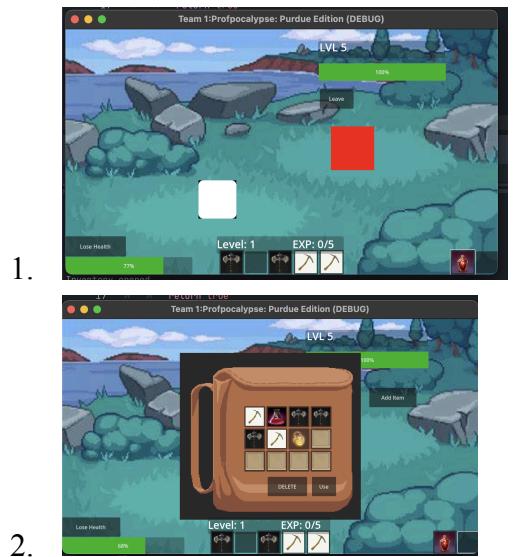
## User Story #12

1. **Acceptance Criteria:** Given the player clicks a potion in the Battle UI, then the potion will be used correctly.

- a. **Description:** This test verifies that if a player uses a potion it will work correctly, then will remove the potion, then will use the user's turn so the enemy can attack next.

b. **TC01**

- i. **Input:** Have a potion in your potion bar in your HUD or inventory, then click it if it is on the HUD or click use potion on the inventory.



- ii. **Output:** After clicking, the heal potion in the HUD will heal you 10, and then will remove it, and then the enemy will attack. As well, if you use the use button on the inventory, it will remove a potion, close the inventory, and then the enemy will attack.

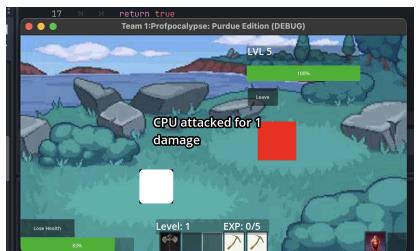


2. **Acceptance Criteria:** Given the player switches weapons, then ui will update accordingly.
  - a. **Description:** This test verifies that if the player switches weapons with the inventory and the weapon hot bar, the screen will update correctly and the stats are still correct for specific objects.
  - b. **TC01**
    - i. **Input:** Move the item between the hotbar and the inventory
    - ii. **Output:** After moving the item like normal, it will save and work normally, as well, because they are objects, they will have the same data as before.
3. **Acceptance Criteria:** Given the player switches the items out from the inventory, then their turn will be skipped.
  - a. **Description:** This test verifies that if the user uses switches out a weapon it will use their turn and the enemy will attack next.
  - b. **TC01**
    - i. **Input:** Move an item from hot bar to inventory



1.

- ii. **Output:** After moving the item, the inventory will close, the screen will lock so the user can't do anything else, and the enemy will attack



1.

### User Story #13

1. **Acceptance Criteria:** Given the player clicks a weapon, then the weapon will do that specific damage to an enemy.
  - a. **Description:** This test verifies that if the user uses a weapon, then it will attack the enemy.
  - b. **TC01**
    - i. **Input:** Click the weapon you want to use when your inventory is closed and in your hotbar

- ii. **Output:** After clicking the weapon the item will inflict damage on the enemy and then the enemy will hurt you.



2. **Acceptance Criteria:** Given the player switches, then weapons, then the weapon damage will change.

- a. **Description:** This test verifies that if the user uses different types of weapons they will have different stats like damage.
- b. **TC01**
- i. **Input:** Click the axe then click the pickaxe
  - ii. **Output:** After clicking, they have different amounts of damage they can do.



3. **Acceptance Criteria:** Given the player uses a special weapon, then it will have a specific chance to do something to the enemy(it may stun them or critical them).

- a. **Description:** This test verifies that if the weapon has special abilities like critical chance, these can be used correctly, including breakability and can lose the weapon as well.
- b. **TC01**
- i. **Input:** Click on the axe

- ii. **Output:** After clicking, there are chances for it to break, stun, and do critical damage or just do normal damage.



**User Story #14:** As a player, I would like to be able to view my character's abilities.

1. **Acceptance Criteria:** Given the player is in gameplay, they can see a preview of simplified main abilities.

- a. **Description:** This test verifies that the first three player's abilities and their current level are being displayed on the main gameplay screen.

b. **TC01**

i. **Input:**

1. Run the game.

ii. **Output:**

1. View the abilities preview in the bottom left corner.



2. **Acceptance Criteria:** Given the player presses the button or key command for detailed view, a window appears with bars to show player abilities.

- a. **Description:** This test verifies that the player is able to open a window to see all of their abilities and what level each ability is at.

b. **TC01**

i. **Input:**

1. Run the game.
2. Hit the “Abilities Detailed View” button



3.

ii. **Output:**

1. View the abilities Detailed view



2.

3. **Acceptance Criteria:** Given the player is viewing the bars for each ability, the bars have a max number that they can be increased to.

a. **Description:** This test verifies that the player is able to increase the progress bars for each ability and that they cannot increase the ability more than a set number.

b. **TC01**

i. **Input:**

1. Run the game.
2. Hit the “Abilities Detailed View” button.
3. Hit add Tokens or Level Up until 12 study tokens are available.



4.

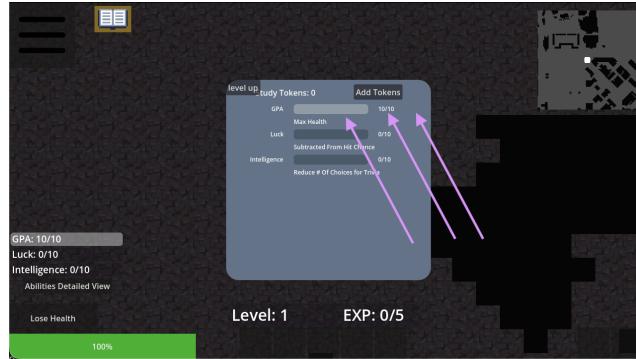
5. Hit add next to one of the abilities 10 times.



6.

ii. **Output:**

1. View the #/10 that each ability bar displays.
2. The progress bar increases till it's full.
3. The add button disappears after 10 are added.



4.

**User Story #15:** As a player, I would like to get stronger or more abilities based on my level.

1. **Acceptance Criteria:** Given the player levels up, the player is given a number of credits they can apply to increase one or more abilities.

- a. **Description:** This test verifies that the player is able to receive a reward for leveling up in the form of credits to purchase more stats.
- b. **TC01**

i. **Input:**

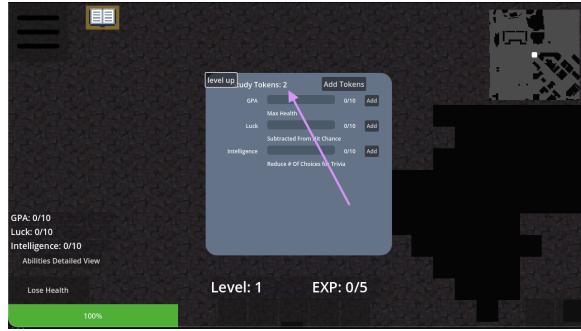
1. Run the game.
2. Hit the “Abilities Detailed View” button.
3. Hit level up



4.

ii. **Output:**

- View the study token count rise by two.



2.

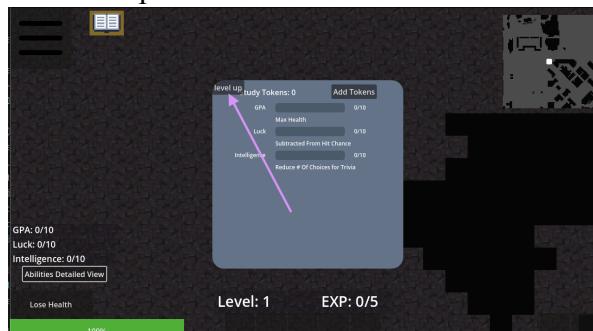
- Acceptance Criteria:** Given the player levels up to level three, the player gains access to a new ability.

a. **Description:** This test verifies that the player is able to view more abilities based on their level.

b. **TC01**

i. **Input:**

- Run the game.
- Hit the "Abilities Detailed View" button.
- Hit level up 2 times



4.

ii. **Output:**

- View an extra ability appears upon the second level up.



2.

- Acceptance Criteria:** Given the player has increased their health ability 1 time, there is space for 2 more health in the bar.

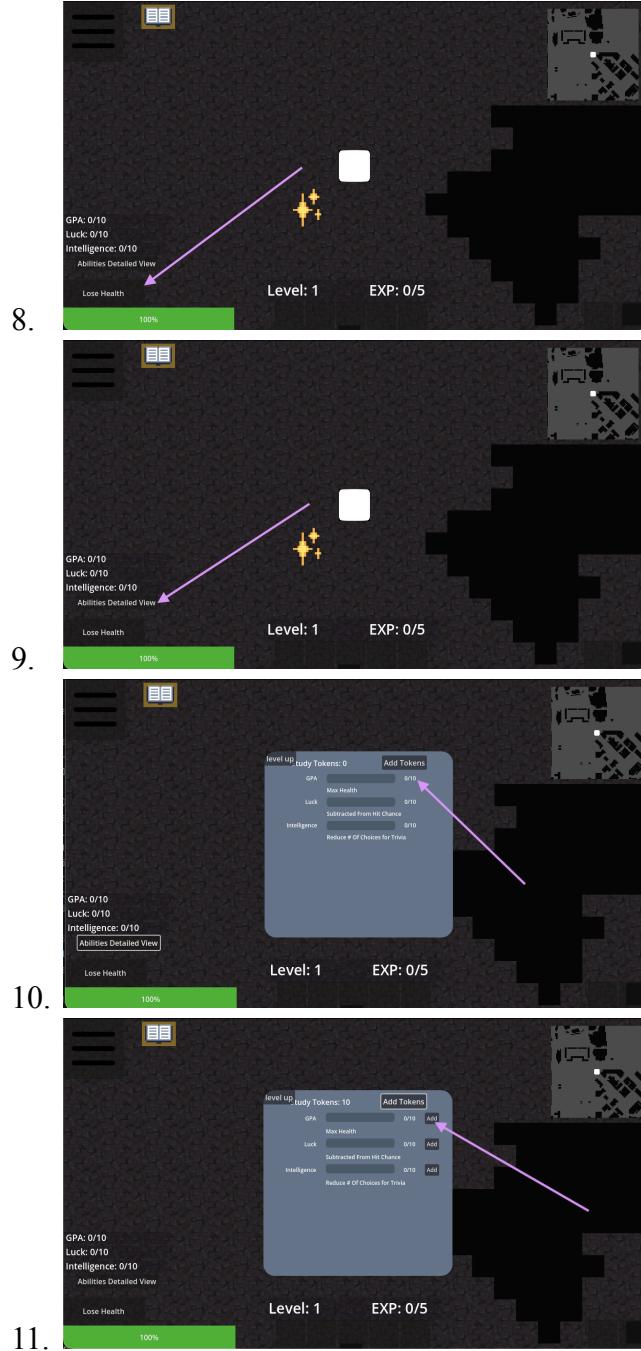
a. **Description:** This test verifies that the health bar is affected by the GPA ability.

b. **TC01**

i. **Input:**

- Run the game.
- Hit Lose Health 10 times.

3. Re-run the game
4. Hit the “Abilities Detailed View” button.
5. Hit the add tokens button 2 times.
6. Hit add next to the GPA ability 2 times
7. Hit Lose Health 14 times



## ii. Output:

1. The first time the health bar needed 10 hits to completely empty.
2. The second time depleting the health bar took 14 hits to completely empty.

**User Story #16:** As a player, I would like to talk to my academic advisor when I need new classes.

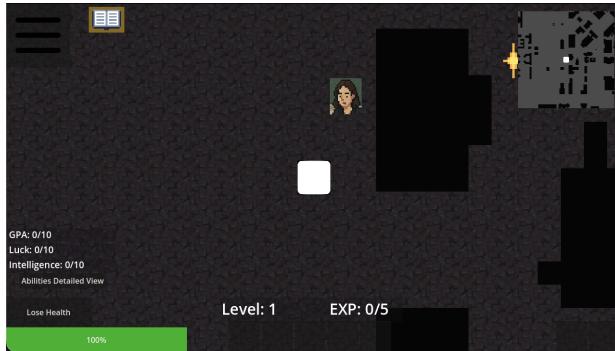
1. **Acceptance Criteria:** Given the player finds the advisors building on the map, an option to visit with their advisor appears.

- a. **Description:** This test verifies that the player is able to find their advisor on the map and that they can choose to meet with them.

- b. **TC01**

- i. **Input:**

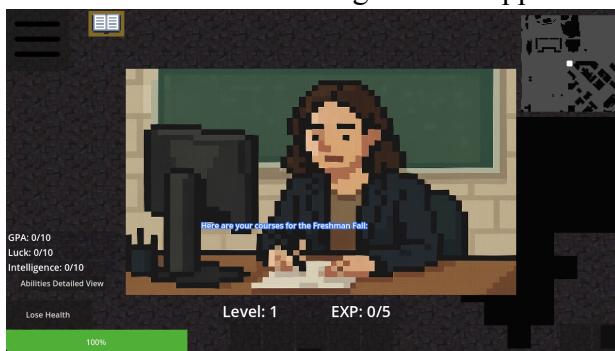
1. Run the game.
    2. Move the mouse using ASDW to the building HAAS.
    3. Find the advisor icon and click it.



4.

- ii. **Output:**

1. Observe the advisor meeting window appear.



2.

2. **Acceptance Criteria:** Given the player chooses to visit with their advisor, a window with an image of the advisor and dialog instructing them what to do appears.

- a. **Description:** This test verifies that the player is able to meet with their academic advisor and that the meeting is interactive.

- b. **TC01**

- i. **Input:**

1. Make your way to the advisor opener using the methods described criteria 1.
    2. Hit the button to display the advisor meeting.
    3. Wait for dialog

- ii. **Output:**

1. The window displaying the advisor appears.
    2. The advisor has a dialog congratulating the player on completing the semester and informing them of their next semester courses.

3. The advisor asks if the user would like to start the next semester.



- 4.



3. **Acceptance Criteria:** Given the player chooses to enroll in courses for the next semester with the advisor, the current course information on the phone updates.

- a. **Description:** This test verifies that the player is able to update their course information during their advisor meeting.

b. **TC01**

i. **Input:**

1. Go to the phone course information page. (Hit c on the keyboard)
2. Make your way to the advisor opener using the methods described criteria 1.
3. Hit the button to display the advisor meeting.
4. Wait for dialog.
5. Hit yes in answer to the question “would you like to start the next semester?”

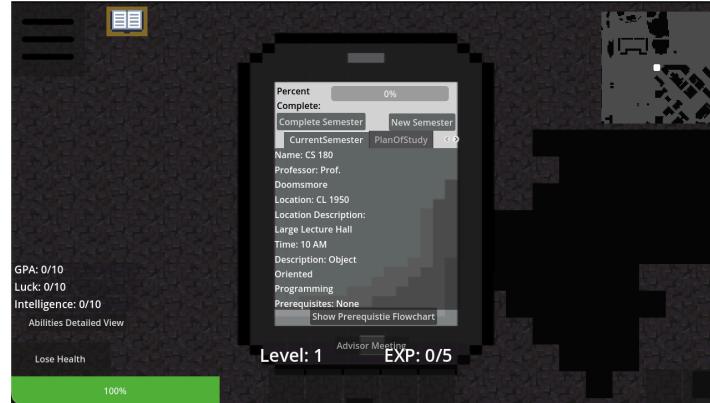


- 6.

7. Go to the phone course information page. (Hit c on the keyboard)

ii. **Output:**

1. View the starting current course information on the phone page.
2. Once completing the advisor meeting and going back to the current course information observe that it has moved on to the next semester.



3.



## User Story #17

4. **Acceptance Criteria:** Given that I am able to find major events at Purdue, I will compile them into a list with specifications about each of them
  - Description:** This test checks that the user will be able to view the events at Purdue within the calendar UI
  - TC01**
    - Input:** The user loads the game, then presses the button on the main screen to access the calendar UI
    - Output:** The calendar is populated with a list of events that occur on Purdue's campus and they are positioned at the correct date within the calendar
5. **Acceptance Criteria:** Given that the user is within the game, there will be a method to access the calendar
  - Description:** This test will check that when a player opens the game to the main menu, there will be a button to open the calendar and when pressed the calendar will display on the user's screen
  - TC01**
    - Input:** The user loads the game, views the UI of the game, and presses

- the calendar button within the game
  - ii. **Output:** The calendar will display on the user's screen and contain the relevant information that is kept within it
6. **Acceptance Criteria:** Given that the user access the calendar, there will be a UI so that the user is able to view all of the events that occur on the calendar
- a. **Description:** This test checks that there is a UI for the calendar and it is functioning so that the user is able to view all of the events contained within the calendar
  - b. **TC01**
    - i. **Input:** The user loads the game, and presses the button to open the calendar menu
    - ii. **Output:** The user observes the calendar UI and is able to press buttons to scroll through the different months that are listed with events that occur at Purdue

## User Story #18

7. **Acceptance Criteria:** Given that I am within our game file on Godot, there will be an object class that adds functionality to items so they are able to be picked up
- a. **Description:** This test checks that the Object class is within the game file on Godot
  - b. **TC01**
    - i. **Input:** The grader loads github and navigates to out game file and clicks on the folder labeled object
    - ii. **Output:** The folder opens and displays the code within that corresponds to the object class
8. **Acceptance Criteria:** Given that there is already items implemented into the game, they will be added to the new item class so that they have the ability to be picked up
- a. **Description:** This test checks that the items within the game have been added to the new item class and are able to be picked up
  - b. **TC01**
    - i. **Input:** The user loads the game and navigates to the main screen when playing the game, then they look for and item on the ground and press the key they have binded to picking up an item
    - ii. **Output:** The item will then appear in the user's inventory like any other item and be useable
9. **Acceptance Criteria:** Given that the user is within game, they will have the ability to pick up items they find within the map
- a. **Description:** This test checks if an item within the game is able to be picked up by a user and added to their inventory
  - b. **TC01**
    - i. **Input:** The user loads the game, and navigates to the main game screen where they are currently playing the game. They then find an item by navigating across the map using the keys they have binded to character

movement. They press the key they have binded to picking up an item on or near the item

- ii. **Output:** The user is able to view the item in their inventory and sees that the item is no longer on the ground and able to be picked up again

## User Story #19

**10. Acceptance Criteria:** Given that I create new items and old items exist, they will be added to a list of all the possible items that can be collected in the game

- a. **Description:** This test checks whether or not the list of all the items has been created by navigating into the inventory and viewing the menu for the item list
- b. **TC01**
  - i. **Input:** The user loads into the game and navigates to the inventory by pressing the button they have binded to that function, then they find the button labeled items and press it
  - ii. **Output:** The item menu will display and be populated with a list of items that the user will be able see while within the menu

**11. Acceptance Criteria:** Given that the user is in the game, they will be able to access a UI where they can view the list of all the possible items that can be collected

- a. **Description:** This test checks whether or not the UI for viewing the items within the game is accessible
- b. **TC01**
  - i. **Input:** The user loads into the game and navigates to the inventory by pressing the button they have binded to that function, then they find the button labeled items and press it
  - ii. **Output:** The UI for items will appear on the user's screen and they can verify that it is functioning by being able to close it and reopen at will

**12. Acceptance Criteria:** Given that the item list UI is successfully created, it will have the functionality to display the items that the user has already collected

- a. **Description:** This test checks if the item list UI has the ability to mark off certain items that are already in the user's inventory denoting the fact they have at least one copy of the item
- b. **TC01**
  - i. **Input:** The user loads into the game and navigates to the inventory by pressing the button they have binded to that function, then they find the button labeled items and press it
  - ii. **Output:** The user is able to view the list of items and can see that by each item that they have collected there will be a check mark meaning that they have the item the checkmark is associated with in their inventory while the items that do not have a checkmark next to them are not within their inventory and therefore not owned by the user.