**Quest: Escape Room**

[Unit 1 - Introduction to Python](https://my.ironhack.com/cohorts/64f9d729a7ab01002ae1abce/lms/courses/course-v1:IRONHACK+DAFT4+202409_RMT_ES/modules/ironhack-course-chapter_1)

**Quest: Escape Room**

**MINI PROJECT**

Generate a quiz about this unit

Summarize this unit

Have any doubt about this content? Ask!

**Overview**

Welcome to the Python Analytics Quest! Your mission: code a text-based game, ‘Escape Room’, utilizing Python skills learned in the bootcamp. By doing this, you’ll reinforce Python theory, enhance teamwork, and problem-solving skills. Ready? Let’s go!

**Setup**

**Team Roles**

You’ll work on the quest in a team of four, split into two pairs: the **Dev Team** and the **Research Team**.

**Dev Team:** Responsible for the game’s coding such as constructing the game’s structure and functions, ensuring smooth game flow and that the project requirements are met. Pair programming is strongly encouraged to allow collaborative problem-solving, exchange of ideas, and ensure quality in your code.

**Research Team:** This team is responsible for researching topics that are relevant to the project and then sharing that information with the Dev team. They will also apply the findings to the project, so each team has hands-on experience each day.

Note: Roles will **rotate daily** for skill diversification.

**Typical Day**

Each day the Dev team and Research team have different responsibilities in order to meet the requirements of the quest. With an exception of day 1, where you will have some time allocated to read the project brief in full (this page) each day should be structured like this:

**Collaborative Development & Research Hour (pairs) | 14:00 - 15:00**

* **Dev Team:** Implements daily topic-related code into the game.
* **Research Team:** Carries out topic-specific research.

**Interactive Knowledge Exchange (full group) | 15:00 - 15:30**

* **Dev Team:** Shares coding progress and challenges with Research team.
* **Research Team:** Shares research findings and key takeaways with Dev team.

**Application & Testing Phase (full group) | 15:30 - 16:30**

* **Research Team:** Applies research findings to the game.
* **Dev Team:** Supports Research team, tests changes and ensures seamless code integration.

**Topics by Day**

Each day teams focus on different topics. These topics dictate the Dev Team’s coding and Research Team’s research activities. Detailed instructions for each day are provided. Remember, these are suggested paths; you can manage your team as long as the requirements are met.

**Dev Team Key Topics:**

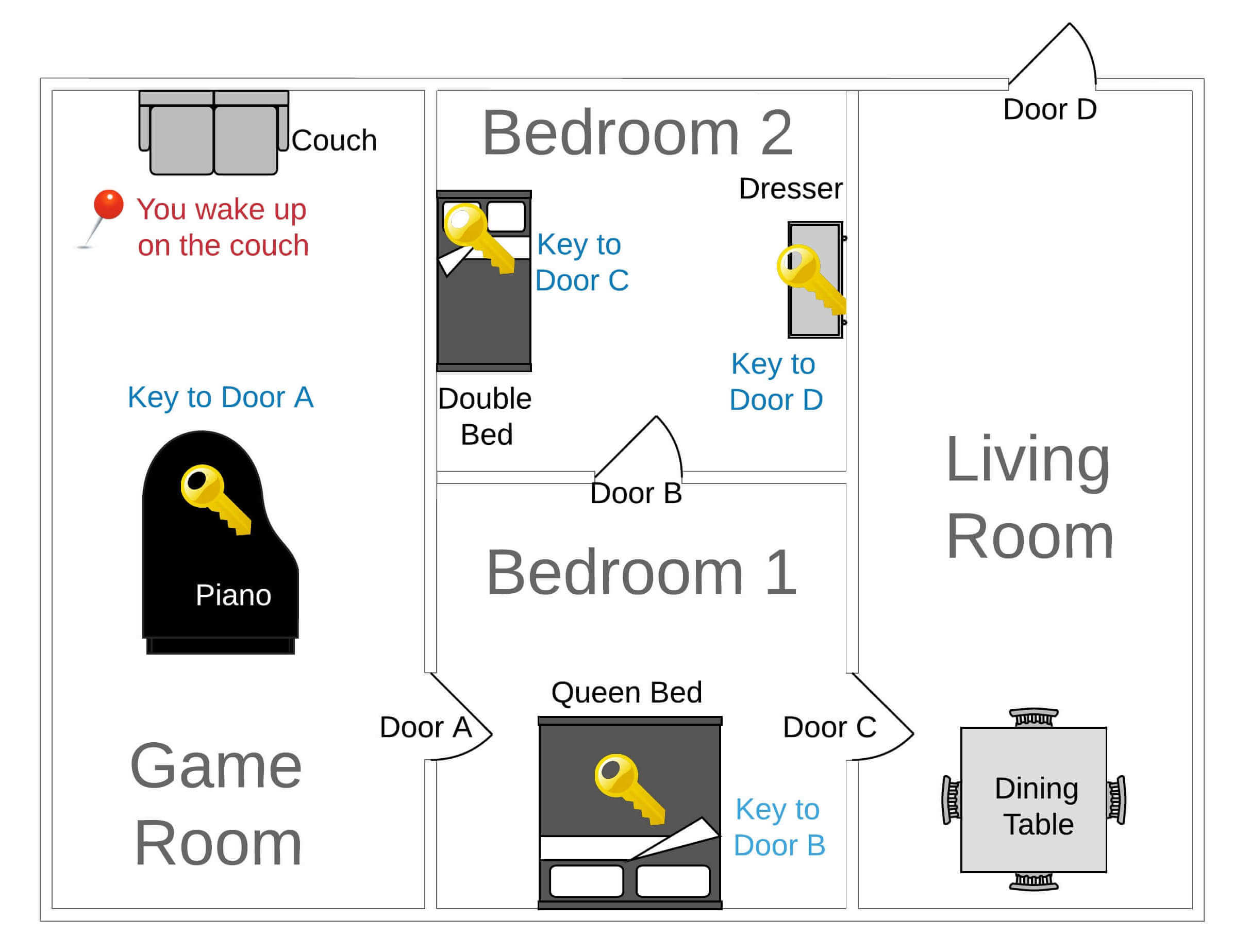
* **Day 1**: Create dictionaries and lists serving as game data structures.
* **Day 2**: Define functions and introduce flow control logic.
* **Day 3**: Refine functions and flow control, adding interactivity.
* **Day 4**: Finish implementing all of the game’s code.

**Research Team Key Topics:**

* **Day 1**: Learning how to read Python error messages, function documentation and reinforce understanding of Python data structures and flow control.
* **Day 2**: Learn about Python modules and packages, structuring code.
* **Day 3**: Learn about good coding practices, including clean, readable code and proper documentation.
* **Day 4**: Review list and dictionary comprehensions, read about Python testing.

**Quest Brief**

Your team will develop the “Escape Room” game. Implement Python code, manage errors, and test your game to ensure robustness. Each room presents unique challenges. The goal is to escape from the starting room (Game Room) to the end (Outside). Look at the image below and read the game narrative to understand the game logic fully:



**Game Narrative**

Suddenly, you find yourself waking up on an unfamiliar couch, in an eerie house devoid of windows. Your memory fails to provide any explanation about how you ended up here, or what transpired previously. You can sense an imminent threat lurking somewhere - your gut tells you to escape the house immediately!

You look around and realize you’re in a game room.

“What should I do?” you ponder.

You decide to explore the room, which uncovers a couch, a piano, and a locked Door A. The couch appears mundane and unexciting. However, examining the piano rewards you with a key for Door A. With the newfound key, you manage to unlock Door A.

Will you step into the unknown through Door A?

**Bedroom 1**

You step into Bedroom 1.

Again, you find yourself contemplating - to explore or examine?

Your exploration reveals a queen bed, Doors A, B, and C. Underneath the queen bed, you find a key for Door B. Door C remains an impenetrable barrier as it’s locked and you lack the key. But with the key you found under the bed, you manage to unlock Door B.

Is it time to venture through Door B?

**Bedroom 2**

Your journey leads you to Bedroom 2.

“What now?” you think.

A thorough exploration uncovers a double bed, a dresser, and Door B. Hidden in the double bed, you discover a key for Door C, and within the dresser, a key for Door D awaits. Equipped with the key for Door B, you decide to unlock it.

Shall you retrace your steps through Door B?

**Bedroom 1**

Once again, you find yourself in Bedroom 1.

“What’s the next move?” you wonder.

With the key for Door C in your possession, you manage to unlock it.

Are you ready to brave the unknown behind Door C?

**Living Room**

Stepping through Door C, you find yourself in the living room.

“What secrets does this room hold?” you muse.

Exploration reveals a dining table, Door C, and Door D. With the key for Door D in your grasp, you unlock it.

Is it time to step through Door D and see what lies beyond?

**Outside**

Freedom! You’ve managed to escape the room! Congrats on your successful adventure! Now, can you recount the thrilling journey and the clever strategy that got you here?

**Team Collaboration**

For this project, we’re using Google Colab for collaborative work. It’s user-friendly, perfect for team projects, and lets you focus on Python concepts. As you progress through the bootcamp, you’ll transition to Jupyter Notebooks and GitHub, which is standard in the industry. Some Google Colab tips to get started are:

* **Communication:** Keep your team informed about changes.
* **Working on the Same Cell:** Avoid simultaneous cell editing.
* **Saving Your Work:** Save your work manually frequently.
* **Refreshing:** Refresh your browser if changes aren’t showing.
* **Version History:** Access through File > Revision history.
* **Sharing:** Click Share to give teammates access.

**Deliverables**

**The Game**

Deliver a main.ipynb file with your Python game solution and the .py file(s) containing your functions. It should reflect the Python concepts learned each day and be pushed to GitHub. You may download the files from the Google Colab and push these. The end product should be fully functional but also well-structured, efficient, robust, and entertaining!

Here is a checklist of deliverables for the quest that you can refer to throughout the week to make sure you are on track:

* The game incorporates various data structures, specifically dictionaries and lists, to define game rooms, paths, and items.
* The game uses a Python dictionary to store the game state which is updated when progress is made, such as a key being collected.
* Flow control (if, elif, else statements) has been utilized to create game logic, for example, to determine what happens when a player makes certain choices or interacts with game items.
* The code appropriately uses loops where needed, such as for repeating game actions until the player reaches a certain outcome.
* The game uses functions to play the game and compartmentalize tasks such as navigating between rooms or interacting with items.
* Arguments are used in functions where appropriate to allow for reusability.
* The game uses return statements in functions to output certain results, which are then used elsewhere in your code.
* After completing each game action, functions are called to continue playing until the winning condition is reached.
* The game utilizes comprehension for creating more efficient and readable code where applicable.
* Any user input is validated and the game incorporates error handling techniques to handle any possible errors or exceptions, providing a smoother user experience.
* The code is well-documented with comments explaining what different sections of the code are doing. Functions are properly documented as well following *docstring* good practices.
* The game has a clear end goal or “win” condition that the player can achieve (making it “Outside”).

**Additional Game Features (Optional)**

If you want to go above and beyond the basics, as optional extra features,here are some ideas of improvements for your project. You can also find some recommended reading to get started:

* Apply proper error handling try-except-finally blocks to handle user input or other errors.
* Args and kwargs: Update the functions to accept **\*args** and **\*\*kwargs** parameters. Modify the function calls to pass the required arguments using **\*args** and **\*\*kwargs**.
  + [**Python args and kwargs: Demystified**](https://realpython.com/python-kwargs-and-args/)
* Zip and unzip: Implement the use of **zip** to iterate over two or more iterables simultaneously, such as iterating over items and their corresponding keys.
  + [**Zipping and Unzipping Iterables in Python**](https://towardsdatascience.com/zip-function-in-python-da91c248385d)
* Regex: Incorporate regex patterns to validate user input for actions.
  + [**Regular Expressions: Regexes in Python**](https://realpython.com/regex-python/)
* Apply OOP: Refactor the code into classes to encapsulate the game logic, map, and state.

**The Presentation**

You will present your game on Friday morning to the class showcasing how your game works as well as a short presentation in the form of some slides. The format of the presentation will be as follows:

* Talking with Slides: **3 minutes**
* Demo: **2 minutes**
* Total: **5 minutes**

All presentations will be done **from a staff member’s computer**, so **your slides need to be online**. PowerPoint files, Keynote files or files of any kind **will not be accepted**.

* Suggested online slide applications:
  + [**Google Slides**](https://www.google.com/slides/about/)
  + [**Slides**](https://slides.com/)
  + [**Prezi**](https://prezi.com/)

**Presentation Structure**

Feel free to present the game in as you feel best represents your work, but below you will find a suggested format for the presentation as a guide.

1. **Title Slide** (1 slide): your project’s name & your names
2. **Project Overview** (1-2 slides):
   * How is the game’s structure organized?
   * Which functions do you have and how are they imported?
   * Which features did you add to the game?
3. **Technical Challenge** (1-2 slides):
   * What was **the most important** technical challenge you faced?
   * How did you overcome that challenge?
4. **Big Mistake** (1-2 slides):
   * What was **the biggest** mistake you made during this project?
   * What did you learn from it?
5. **Demo Slide** (1 slide): literally says “DEMO” with **a link** to your project so you can open it easily
6. **Closing Slide** (1 slide): your project’s name, your name & a *“Thank You”*
7. **Total**: 6-8 slides

**Quest - day 1: Kickstarting the Python Analytics Quest**

[Unit 1 - Introduction to Python](https://my.ironhack.com/cohorts/64f9d729a7ab01002ae1abce/lms/courses/course-v1:IRONHACK+DAFT4+202409_RMT_ES/modules/ironhack-course-chapter_1)

**Quest - day 1: Kickstarting the Python Analytics Quest**

**MINI PROJECT**

Generate a quiz about this unit

Summarize this unit

Have any doubt about this content? Ask!

**Introduction**

Kickstart your “Python Analytics Quest” today! Begin developing the intriguing text-based game, “Escape Room,” using Python. The skills gained during the morning will fuel your project work which entails designing rooms, filling them with interesting items, and plotting an escape route.

**Daily Overview & Schedule**

**Part 1: Reading the brief & Project Setup | 15:50 - 16:20**

Set the groundwork for your project. Understand the project brief together, assign roles, and initiate your team’s Google Colab notebook.

* Review project brief thoroughly
* Assign roles (Dev team and Research team)
* Start a [**Google Colab notebook**](https://colab.google/) (Select “New Notebook” and share it with your teammates)

**Part 2: Collaborative Development & Research Hour | 16:30 - 17:20**

Split into two pairs: Dev team and Research/Innovation team.

**Dev Team:**

* Review **sample-code.ipynb** thoroughly.
  + [**Click here to see sample code**](https://github.com/data-bootcamp-v4/lessons/blob/main/1_intro_to_python/quest/d1/sample-code.ipynb)
  + Note: Do not worry if you don’t understand all of the code today, the sample-code is provided for you to refer to throughout the week.
* Understand how Rooms and Items are defined and linked.
* Start constructing your own game structures.

**Hints for getting started**

Overwhelmed and don’t know where to start? This is a tough challenge we know. But don’t worry. We have included a working example for you to reference in which only 1 room (game room) is included. Read the example and make sure you understand it. Then you can expand on top of the example to code the rest of the rooms.

The provided example is just a Minimal Viable Product (MVP). It is fully functional but not bullet proof. You should be aware of its limitations while you are working and try to make your final product as robust as possible.

You can take the following steps throughout the week:

* Launch sample-code.ipynb and read through the file. Execute the code and play the game and try to make sure you understand what each line of the codes does.
* Create main.ipynb and copy the code from sample-code.ipynb. Expand the code following the example to create Bedroom 1.
* Test the game with Game Room and Bedroom 1. Make sure everything works and then work on Bedroom 2, and then the Living Room.
* Test the whole game. Try to make all kinds of inputs to make sure your game is rock solid and will not break.

**Research Team:**

* Read lesson in student portal: **Interpreting Python errors**
* Review the day’s lectures to reinforce your understanding of Python data structures and flow control.
* You may create a document with the key points of research carried out to present to the Dev Team if you wish.

**Part 3: Interactive Knowledge Exchange | 17:20 - 17:50**

Now, both teams present their progress and understandings (15 mins each pair). This phase is crucial for effective collaboration and progress for the quest.

**Dev Team:**

* Explain understanding of the code and game structures (from sample-code.ipynb)
* Present progress of the data structures designed so far. You may run the game in its current state as a demonstration.
* Share challenges encountered and solutions implemented.
* Answer queries from Research team.

**Research Team:**

* Share learnings about Python error messages. Explain how these could help when facing errors while coding.
* Recap the day’s lectures with focus on game development.
* Propose improvements and fixes for the game code.
* Address queries from Dev team, ensuring everyone understands the information shared.

**Part 4: Application & Testing Phase | Homework**

In this final phase, the Research/Innovation team will apply what they’ve learned about the project, adjusting the code accordingly. Meanwhile, the Dev Team will review and test these changes.

**Research Team:**

* Apply your knowledge to the quest to identify and rectify potential errors in the code.
* Adjust code based on your review of the morning’s lectures and self-guided reading.

**Dev Team:**

* Review the changes made by the Research/Innovation Team and challenge them as necessary.
* Test the game with the applied changes and ensure they fit seamlessly with the game’s overall code.

Remember to communicate with your team members, make frequent saves, and refresh your work on Google Colab to ensure everyone is updated. Have a productive Day 1!