**Due** May 17 by 4pm
**Points** 0
**Submitting** a file upload
**File Types** zip
**Available** until May 17 at 4pm

You will need to complete a BASH programming assignment that will be posted to Canvas. **Once you submit your completed source code to Canvas, you will need to schedule an evaluation with the instructor to demonstrate your work.** I will be evaluating your algorithms, your code formatting/commenting/documentation, the way you tested your work, the effectiveness of your implementation decisions, and your understanding of your work.

Do not just find working code from friends, ChatGpt, Stack Overflow and the like. While it is common practice in industry to stitch together code for various problems, we must understand how the code works and what are the consequences of the code in regards to efficiency and robustness. **If all you do is present the instructor with working code but cannot explain it, you will receive a level of Inconclusive.** If you present the instructor with working code and can only give me a basic explanation of the code, the best you can receive is a level of Proficient.

--- Project Description Start ---

## Project Overview:

Create an immersive text-based adventure game using the BASH scripting language on the UWEC Poseidon servers. The game should allow players to navigate through a story, make choices, solve puzzles, and interact with the game world entirely through the command line.

## Game Requirements:

1. **Story-line and Game-play**:
    - Design an engaging story-line with multiple paths and choices that impact the game's outcome. **Hint**: use the free version of ChatGPT to help with a cool story. You should include various locations, items, characters, and challenges within the game world. **Disclaimer**: While I do not care much about your content choices, please keep it appropriate for this academic setting. If your content violates any laws or university policies, it will be given a level of inconclusive and potentially reported as required by law/policy (I

suspect this is a moot point but thank you for reading the disclaimer).

2. **User Interaction**:
    - Implement a user-friendly interface allowing players to input commands and navigate through the game.
    - Use BASH scripting to interpret player input and trigger appropriate game actions/responses.
3. **File Manipulations**:
    - Utilize file I/O to store game progress, player inventory, and game state.
    - Save and load game progress using text files to allow players to continue their adventure.
4. **Game Mechanics**:
    - Implement puzzles, challenges, or riddles that players must solve to progress in the game.  **Hint**: this is another good use of AI.
    - Include combat or decision-making elements that affect the game's outcome.
5. **Game World and Characters**:
    - Create a detailed game world with descriptions for various locations, items, and characters.
    - Implement interactions with characters and objects using Bash scripting.
6. **Graphics and Visuals**:
    - Use ASCII art or text-based visuals to enhance the game's atmosphere and provide visual cues.  **Hint**: what should the screen look like (*e.g.*, colors, shape, scrolling, *etc.*)?
7. **Error Handling and Feedback**:
    - Provide informative and descriptive messages for player actions, errors, or invalid inputs.
    - Ensure robust error handling to prevent crashes and provide a smooth gaming experience.

## Deliverables:

1. **Game Script**:
    - Write a BASH script or scripts that serves as the game engine, handling all game logic, user interactions, and file manipulations.
2. **Game Content**:
    - Develop the game story-line, locations, characters, puzzles, and challenges.

    - Create text files to store game data, including saved progress, locations, items, *etc*.  **Hint**: Can you automate (with BASH) parts of this?  How can the use of regular expressions help you here?

3. **Documentation**:
    - Provide a README file detailing how to play the game, commands available, and a brief overview of the game's mechanics.
    - Include any specific instructions on how to navigate, save progress, or load a saved game.
4. **Testing and Debugging**:
    - Conduct thorough testing of the game to ensure functionality, correctness, and an enjoyable user experience.
    - Debug any issues or errors encountered during testing.

## Additional Notes:

- Incorporate branching narratives and multiple endings based on player choices.
- Prioritize user experience by providing clear instructions and intuitive interactions.
- Utilize BASH scripting techniques such as conditional statements, loops, functions, and file handling to create a robust and engaging text-based adventure game.
- Your script must utilize file operations such as reading from and writing to files, file permissions, file testing (existence, type, permissions), and redirection (`>`, `>>`, `<`, `|`).  Hint: maybe treasure chests can be files and until you get the 'key' the permissions of the file are set so you cannot read it.
- You must figure out a way to use (at least)
    - `grep`
    - `awk`
    - `sed`
    - `find`
    - `sort`
- Your script should follow good standards in regards to being modular and its overall structure.
- Your script should make appropriate use of variables, including environment variables, local vs. global variables, and special variables (`$`, `@`, `*`, `?`, *etc*.).
- Your code should run on the UWEC Poseidon servers and not require root permissions.

Have fun with this.  All I care is that you learn about theses skills so go crazy.  I will try to plan in demonstration times so we can so off our games.  I may even make one.