

I had used Copilot for my AI use. I mainly used AI for error checking. I also used it to keep track of my progress by using the rubric. While I wish my AI usage was not as frequent, I am learning more about the topics we have covered in class and the usage, and limitations of AI. My main issue is my lack of practice. Throughout the course, I have often found myself feeling overwhelmed. I can understand what the different topics are, but trying to implement them has been challenging. My focus over break will be to fix this issue. I plan to go back and look over topics that have challenged me and practice what I have learned. This will allow me to become confident in my coding abilities and less reliant on AI.

Input:

Check the rubric I have given you to see where I am at so far.

Output:

Looking at your code against the rubric, here's your current status:

#### Grading Rubric Assessment (25 Points Total):

##### Core OOP Design (5 Points) - COMPLETE

-  Abstract class `GameCharacter` with abstract methods
-  3+ subclasses (Knight, Wizard, Thief) using inheritance
-  Polymorphism with `GameCharacter` references
- **Score: 5/5**

##### Multithreading Implementation (5 Points) - PARTIAL

-  3+ threads running concurrently in Phase 1
-  Proper thread management with `start()` and `join()`
-  **Missing Phase 4 execution** - you set choices but don't continue adventures
- **Current Score: 3/5** (Need to add Phase 4 continuation)

##### Concurrency Safety (3 Points) - NOT STARTED

-  No shared resources (dragon health, treasure, etc.)
-  No synchronization mechanisms
-  No race condition examples
- **Score: 0/3**

##### Functional Features (5 Points) - NOT STARTED



##### Story & Creativity (3 Points) - PARTIAL



**✗ Documentation & UML (2 Points) - NOT STARTED**



**✗ Reflection & Code Quality (2 Points) - NOT STARTED**



**Current Total: 11/25 Points**

**Immediate Priority (to get to 15+ points):**

continueAdventure()

This would get you to **17/25 points** - a solid foundation to build from!

Claude Sonnet 4 • 1x