INFO 2000, Fall 2024

Term Project
Due: December 9, 2024, 11:00 PM

Project Description

You are tasked with designing a computer game. This game does not need to focus on advanced visuals but should showcase your programming skills and creativity. For example, a game like the 1A2B number guessing game discussed in class would suffice.

Your submission should include:

1. **A one-page Word document** describing your game, its rules, and its unique features.
2. **Your Python code** submitted as a .py file (see submission details below).

Requirements

a. You can design your own game or modify a game we wrote in class.

b. Avoid overly simple games like Rock-Paper-Scissors or Tic-Tac-Toe unless heavily enhanced with unique features or AI. Simpler games will not earn a score higher than 90%.

c. Bonus points are available for incorporating advanced features such as an AI that allows the computer to play against the user. See grading criteria for details.

Grading Criteria

1. Creativity (30%)

- Your game should reflect original thought and innovation. For instance, a creative twist on a classic game or a concept that challenges conventional gameplay could earn high marks.
- Example: Adding a story-based progression or dynamic puzzles.

2. Fun (20%)

- The game should be engaging and enjoyable for players. Features like adaptive difficulty, humor, or surprise elements can increase the fun factor.

- Example: Levels that become progressively harder or secret Easter eggs.

3. Programming Technical Details (50%)

- Demonstrate the skills learned in INFO 2000 by using advanced programming techniques. For full points, your project should incorporate at least **4** of the following:
    - **Modules**: create and import your own module(s).
    - **Command-line arguments**: Enable game configuration or inputs through the command line.
    - **Object-Oriented Programming**: Use classes to represent game elements such as players, levels, or mechanics.
    - **File Handling**: Save and load game progress, or use external files for configuration.
    - **Exception Handling**: Manage errors gracefully, such as invalid inputs or file issues.
    - **Advanced Logic**: Implement AI that reacts intelligently to the player or introduces strategic challenges.

Bonus Points

- **AI Implementation (up to 10 points)**: Design an AI that competes against the player or assists them in gameplay. The AI should adapt its behavior based on user actions or game state.

Submission Instructions

- Submit your Python file as a .py file.
    - **Preferred Option:** Push your code to GitHub.
    - **Alternate Option:** Upload the .py file directly to ELC.
- Include a brief README.md (if using GitHub) or notes in your Word document to explain how to run your game.