Kyle Tranfaglia

Honors 312

Honors Creative Project Proposal

28 Oct. 2024

Libe-Hosted Python PyQt Game Library Project Proposal

Project Overview

This project aims to develop a live-hosted, interactive website featuring a downloadable collection of games created in Python using PyQt. The primary focus is to build a functional and user-friendly interface for users to access, download, and play the games on multiple platforms (Windows, Linux, macOS). Alongside the game development, the project will incorporate a mathematical and computational approach to explore solvability, solution optimization, and efficient representation of solutions for complex puzzle-based games.

Research Questions

1. *Solvability of Games:* How can mathematical methods be applied to determine the solvability of a game, particularly puzzles like the 15-Puzzle or the Platform Runner?
2. *Optimal Solutions and Search Algorithms:* Which algorithmic approaches are best suited to find optimal solutions for each game? How do different algorithms impact solutions?
3. *Representation of Solutions:* What methods can best represent solutions for various types of games, especially puzzle games where multiple solutions might exist?
4. *UI/GUI Considerations in Game Development:* What design principles should be applied when developing game interfaces in PyQt to enhance usability and player experience?

The research will explore efficient algorithms for solving or guiding gameplay in puzzle and strategy games and understand how UI/GUI design principles influence user experience.

Project Scope

The website will host at least four games as a downloadable .zip file collection, with plans for future expansion. The following games are under consideration:

1. *Puzzle Games:* 15-Puzzle, 2048, Tic-Tac-Toe, and Connect Four.
2. *Strategy Games:* Battleship, Memory Game, Hangman, and Card Game (Uno, Poker, Blackjack…).
3. *Arcade Games:* Snake Game and Pong.
4. *Platform Game:* Platform Runner.

These games will be implemented as standalone applications using PyQt, and features such as score tracking, move validation, and hints for solvable games will be included.

Literature Review and Rationale

In AI and game theory research, the importance of algorithms in game design has long been established. Classical works, such as Knuth's *Art of Computer Programming* and Hart, Nilsson, and Raphael's *A Formal Basis for the Heuristic Determination of Minimum Cost Paths*, outline foundational algorithms like Minimax, Dijkstra, and A* search, which are critical in designing puzzles that need solvability checks and optimal solutions. Further, research about GUI design principles, as explored in Plaisant and Shneiderman's *Designing the User Interface*, highlights the impact of interface design on user engagement. Using PyQt as the GUI framework promotes rapid development with high customization, making it suitable for games that require interactive and responsive interfaces. PyQt's native support for event-driven programming aligns well with game development requirements, especially for games where real-time response is essential.

Methodology

1. Game Development:

- *Coding:* Each game will be developed using PyQt in Python
- *Cross-Platform Packaging:* Each game will be packaged as an executable (.zip) for Windows, Linux, and macOS using PyInstaller.
- *Documentation:* Each game will include in-depth documentation covering game rules and user interface elements.

2. Research and Analysis:

- *Mathematical Solvability:* For puzzle games, linear algebra and combinatorial analysis will be applied to determine conditions under which a puzzle is solvable.
- *Optimal Solution Search:* Algorithms such as A* for pathfinding and Minimax for strategy games will be implemented to find optimal moves. For games where an exhaustive search is infeasible, heuristics will be employed to approximate solutions.
- *User Interface Design Evaluation:* Effective GUI design will be applied and evaluated, focusing on usability, intuitive layouts, and responsiveness.

3. Website Development:

- *Front-End Design:* A minimalistic website interface will be created, showcasing game descriptions, download links, and source code links.
- *Back-End Configuration:* A simple backend will manage file hosting, ensuring downloadable files are accessible and securely stored.

Timeline

Phase 1 (Weeks 1-2):

- Test, finalize, and showcase 15-Puzzle and Tic-Tac-Toe (started software)

- Write documentation and construct executables for the completed programs
- Research and identify the following games to create, then brainstorm the design and implementation

Phase 2 (Weeks 3-12):

- Complete, test, and showcase at least two more programs that implement an effective UI design and/or utilize a mathematical method in determining an optimized solution
- Write documentation and construct executables for the completed programs

Phase 3 (Weeks 13-15):

- Design, code, and live-host a basic website for game downloading
- Link all executables as .zip files on the website and provide sufficient documentation for each program on an isolated documentation page

Expected Outcomes and Future Work

This project aims to create a functional, user-friendly website offering a downloadable game collection with executable files and detailed documentation for each game. The research component will provide insights into algorithmic optimization and UI design in game development, contributing to ongoing academic discussions in these areas. Future work may include expanding the game collection, adding multiplayer options, or developing additional games with more complex mathematical requirements, such as AI-based chess or platformer games with procedural generation. Overall, this project will provide a platform for further exploration of game development and mathematical research while offering a practical, accessible collection of games for public use.