

**Project Proposal**

CS 370 - Software Engineering

Kamin Fay, Celine Fucci, Kevin Ho, Ethan McGowan, Jalen Pestillo

1. **Introduction**

For CS 370 - Software Engineering, our project is composed of building a game using the MiniAT Architecture. According to [miniat.org](http://www.miniat.org/), “the MiniAT is an open-source embedded microcontroller packaged as an easy to use C library, designed for and with computer science students.” Using this architecture, we were assigned the task of implementing a project titled Jump Rope City.

1. **Project Description**

Jump Rope City is a game where the goal is to navigate through a map of randomly placed jump ropes. The game will be programmed by the user (You) and implemented on a randomly generated or user generated course developed by our team of developers.

1. **Project Goals and Objectives**
2. To successfully accomplish the goals laid out by our client (the course instructor)
3. To learn and use the MiniAT Architecture
4. To create a visually appealing game that is intuitive for the user to play and fun for the user to program
5. To explore various software customs/habits throughout each development cycle, thus improving our skills in the field
6. By the end of the 3rd development cycle, have a working product that depicts our accomplishments throughout the semester in CS 370
7. **Features**
   1. Player controlled via assembly code
   2. Random generated mapping
   3. User generated mapping
   4. Graphical User Interface (GUI) map editor
   5. SDL animations and sounds
   6. Head to head local multiplayer
8. **Development Cycles**
9. 1st Cycle: During the first cycle, the goal will be to implement a visually appealing map structure.
10. 2nd Cycle: During the second cycle, we will be working with implementing the MiniAT architecture.

1. 3rd Cycle: During the third cycle, we will be finalizing the project and enhancing the game. To elaborate, we will be working on any bug fixes and improvements that are requested by the client.
2. **Tools/Resources**
   1. Programming
      1. SDL2-2.0.5 (<https://www.libsdl.org/download-2.0.php>)
      2. SDL2\_image-2.0.1 (<https://www.libsdl.org/projects/SDL_image/>)
      3. SDL2\_mixer-2.0.1 (<https://www.libsdl.org/projects/SDL_mixer/>)
      4. SDL2\_ttf-2.0.14 (<https://www.libsdl.org/projects/SDL_ttf/>)
      5. C++ (<https://isocpp.org/>)
      6. Python (<https://www.python.org/>)
      7. SCons (<http://scons.org/>)
   2. Communication
      1. Slack (<https://slack.com/>)
      2. Github (<https://github.com/>)
      3. Asana (<https://asana.com/>)
      4. Trello (<https://trello.com/>)
3. **Project Roadmap**
   1. Important Milestones And Deliverables

* Development Cycle #1
* Development Cycle #2
* Development Cycle #3
* Installation/Build  
  1. Timeline Indicating Our Schedule

For each development cycle, we will meet each class period with the course instructor to report on project status, present implementation or design concepts, seek feedback, etc (outlined in the course syllabus). Additionally, our team will have detailed discussions and demonstrations for the client to ensure our product is meeting the Cycle Intent. Furthermore, once the development cycles begin, we will post a formal electronic copy of a team status report to Blackboard no later than 5:00 PM on Sundays. At the end of each cycle, our team will:

* Deliver the product
* Have written documentation
* Perform an oral presentation
* Fill out one peer evaluation from each team member
* Have a product that can be built, installed, and contain an up-to- date user manual
* Continue to maintain good engineering discipline
  1. Possible Risks And Dependencies
* Completing each task outlined in the development cycles
* Maintaining frequent communication with team members
* Ensuring our code is error-free and able to be compiled
* Time needed for certain tasks could be longer than anticipated

1. **Challenges/Obstacles**
   1. Mapping system: importing a file. For the architectural spike, we did this using a simple 2D array of characters. The issue was that the basic SDL tiling system uses a standard array or vector for the tiling of images. This will cause a challenge when implementing the map system with graphics.
   2. Creating a map: implementing ropes. The ropes were limited to four directions. In addition to being limited in directions, they can’t overlap. This will cause issues with randomly generating ropes, as they will occasionally overlap.
   3. Learning new tools, such as the Sample DirectMedia Layer (SDL), the MiniAT Architecture, and SCons.
   4. Following the requirements of our client. This will be difficult because of the experience that our client has in the field of software engineering.

* 1. At the end of our project, everything will need to work properly and be able to run with the tools established by our client. This will be difficult depending on the familiarity of these tools.
  2. Producing code that meets the graphical requirements of the team. In other words, producing a product that is visually appealing and functional.