Project Plan: Mountain Climbing Game

Team: Draco

Frank Brasington
Barry Hall
Edwin Grove

Introduction:

This project is a game that will be built using python 3.4 and pygame. The game will have the basic objective to climb as high as possible before either reaching the top of the mountain or failing. The game will teach the members of the team the python language and some of the fundamentals of game design. The team will design the game for user interaction.

Project Description and User Experience:

From the user's point of view the project should be a simple, but entertaining game. The user will be able to use typical input devices such as a keyboard or a controller. When the player starts the game the player will attempt to find the path that will move them upward thru a procedurally generated world in order to climb to the top of a mountain. On their way to the top, the user will encounter enemies, items and traps that will have to be avoided if they want to make it to the pinnacle. The player will have successfully completed the game on the completion of 20 levels.

Structure of the Software:

The development team decided that the basic structure of the game should be fairly simple. Since the game is standalone and currently there are no plans for any databases or outside support. The team decided to take an object-oriented architecture coupled with the MVC (Model-View-Controller) design pattern to organize the code. The MVC design pattern is a way of organizing code in a game by separating and abstracting the controller, or the way the user interacts with the application, the model, which serves as a computational approximation or abstraction of the world being created by the game, and the view, or what is displayed on the screen. We will use this design pattern as a model for making and organizing libraries that will act as the game engine.

Libraries, Languages and Tools:

- Python 3.4
- Pygame
- Pynum
- Pytime
- IDLE
- Random
- Time
- Github - https://github.com/flbrasington/CS467-Draco-python-game

Team Member Responsibilities:

Each team member will be responsible for completing tasks. Team members will sign out tasks and record the number of hours required to complete the task. The team members will report back to the group in order to properly share the work. All hours worked and what the team member worked on will be logged into a shared excel document. This excel document will be used to simulate billing the client for hours worked on the project. Each team Member will be assigned to different tasks based on weekly group

meetings. One group member will be responsible for ensuring that internal standards of commenting is met to avoid any confusion in the programming.

Project Timeline and Milestones:

At the end of each milestone each member of the dev team will spend 1 hour testing gameplay and make any notes about the game in order to make changes.

Minimum Viable Product:

This is the bare minimal that is required for the game. This is the product before any of the graphics, items or enemies. For this project the minimum Viable Product would be the player's movements, collision detection, platforms and the procedural generated levels.

Cells refer to a section of the level that the player is required to cross. Each level will consist of 5 by 5 grid of cells. Each cell is to be 8 by 8 blocks. Each block is currently a 120 pixel square block See appendix A. Each of these variables are subject to change based off of playtesting.

- Player movement
    - Walking
    - Running
    - Jumping
    - Climbing
- Collision detection
- Level
    - Generate path to top of mountain
        - Entrance to level at bottom
        - Random path leading to exit at top of level
    - Create cell templates
        - Exits on left and right
        - Exits on top and/or bottom
        - Exits on left and right, top and/or bottom
    - Procedurally generated levels
        - Movement Cells (cells that have a path for the player)
        - Other Cells (Difficult or possibly impossible for the player to cross)

Once the Minimum Viable Product is done the team will expand to include other aspects

Phase 1:

- Simple Traps
    - Traps such as spikes that don't move
    - Traps such as a rolling rock that has a basic moving pattern
- Simple graphics
    - Animations
        - Player walking, running and jumping animation
        - Platform graphics
- Advance player movement/features
    - Player health
    - Fall damage
- Items
    - Climbing Gear for the player to use

    - Ropes, ladders, and others

Phase 2:

- Basic menu options
    - New game
    - Saved game
    - Window Size
    - Konami code
    - Other
- Advance Traps
    - Floor/walls that break
    - Enemies with basic AI
- More Graphics

Phase 3:

Phase 3 is to add more to the experience to the player. At the End of Phase 2 the group should have a fully functional game. The group having never worked on a game from start to finish before are having to use their best estimates for the required amount of time to complete the work. Phase 3 Will be adding and changing the program based on user input from individuals outside the dev team. I.E. Play testers.

- Play Testers
    - Requires 3 different groups to look over the project and make changes based on their experiences
- Add money & shop for the player to buy items and store items.
    - Add Shopkeeper.
    - Shopkeeper AI if the store is robbed or the the Shop Keeper is killed
- Add a boss
- Add a boss room or a challenge room for the player to get past.
- Expand on traps and enemies
    - This will allow the group to add enemies and add advance enemies for the player to avoid

Work schedule and time estimates:

Tasks	Time Estimate(hrs)
Week 3 - Minimum Viable Product	42
Set Up environment	4
Player movement	4
Collision Detection	4
Level Design	12
Level design Algorithm	8
Level size & pixel requirement Testing	4
Video Update	2
Group Meeting	1
Playtesting	3
Week 4 -	44
Playtesting	3
Finalize Level & Cell size	4
Revisiting Old Work	4

Identify/make Art Assets Simple Graphics Items - rope, ladder, etc Video Update Advance Character Movement/Health Simple Traps Group Meeting	4 8 10 2 8 4 1
Week 5 - Finish Phase 1 Finish simple Graphics Playtesting Video Update Expand on Level design Finish Items Finish Simple Traps Group Meeting Mid-project Check	34 4 3 2 12 4 4 1 6
Week 6 - Basic Menu Playtesting Video Update Revisit old work Advance Traps - Static Advance Traps - enemies Group Meeting Algorithm for enemy Placement	40 12 3 2 8 8 8 1 5
Week 7 - Finnish Phase 2 Video Update Group Meeting Play testing Oversee group 1 of playtesters Make changes based on playtesters Revisiting Old Work Challenge room/level Boss room/level Expand on Enemy placement Algorithm	48 2 1 3 2 10 4 8 8 10
Week 8 - Expand on Level design Video Update Group Meeting Play testing Oversee group 2 of playtesters Make changes based on playtesters Add gold/money to level and Algorithm Add shopKeeper & Shopkeeper AI	40 8 2 1 3 2 10 6 8
Week 9 -Finish phase 3 Finish Graphics	45 12

Video Update Group Meeting Play testing Oversee group 3 of playtesters Make changes based on playtesters Expand on Enemies & traps Finish Challenge Rooms/Bosses	2 3 3 2 8 10 10
Final Report Demonstration	6 3
Total Hours	302

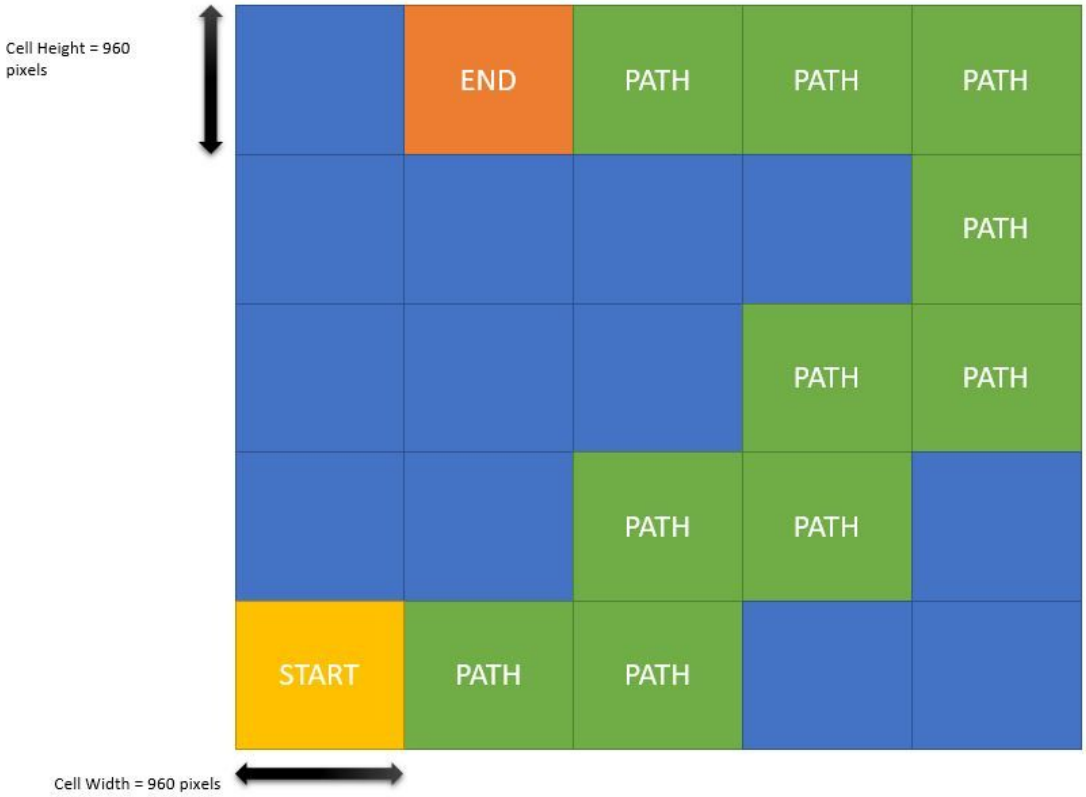
If the group finds that the hours do not correctly reflect the required work then updates in the weekly meetings will update the Client. If the work appears to be too easy then the group will ask the client for work to help expand the project.

Conclusion:

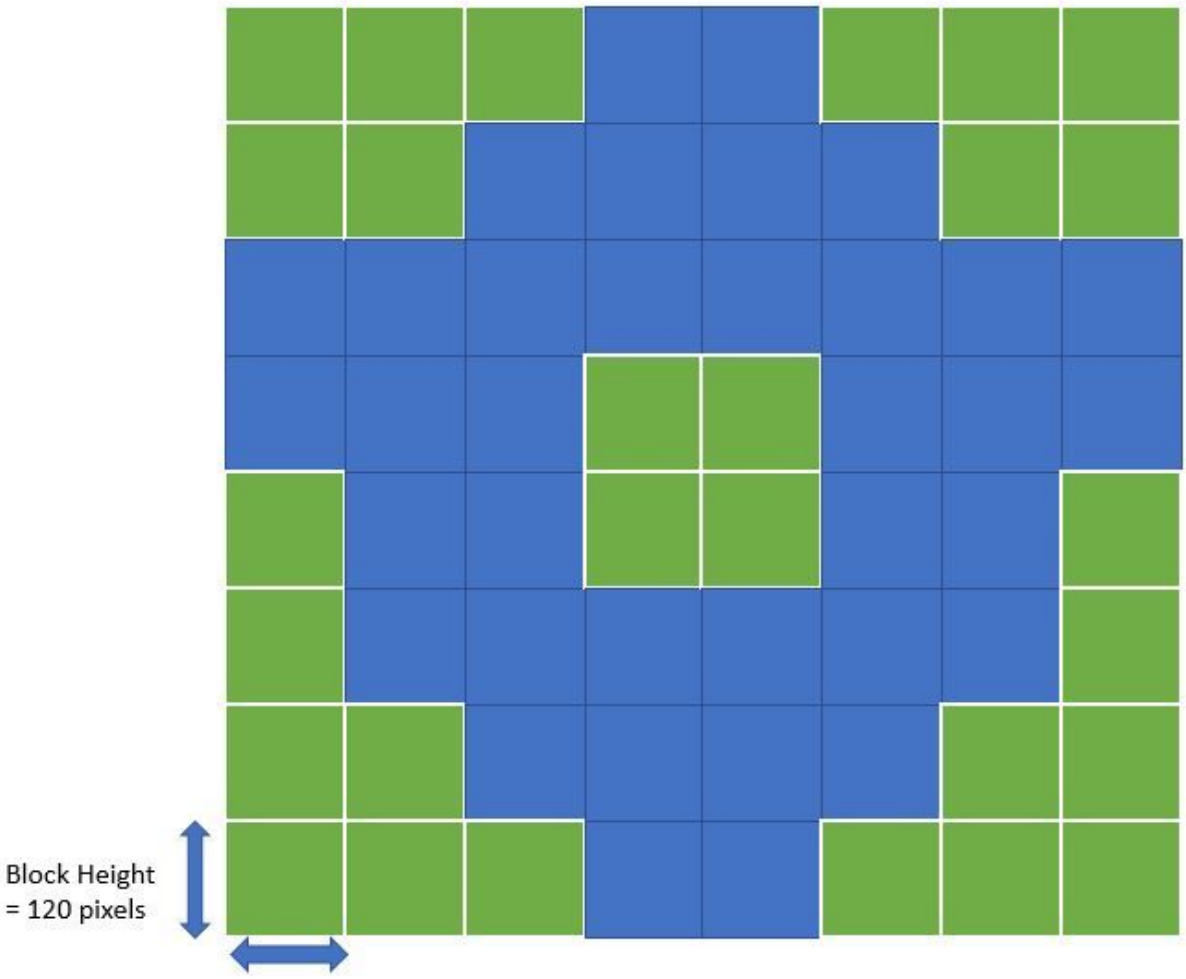
The development team is excited to work on this project. There are new challenges for the team to try and overcome. One of the biggest challenges the team will face is amount of hours worked and staying on task. The development team plans to mitigate this by parsing out the work while the other team member is asleep or busy. The team plans on designing an exciting and fulfilling game that all ages can enjoy.

Appendix A:

Example Level design:



Example Cell Design


Block Height
= 120 pixels
Block Width
= 120 pixels