

MYTHICAL MAZE

PROJECT PLAN

Contracted By Cognitive Thought Media



Table of Contents:

Description	Page Number
1.o Project Overview	3
2.o Project Life Cycle	6
3.o Detailed Design	8
4.o Quality Management	11
5.o Resource Management	12
6.o Communications Management	13
7.o Project Change Management	14
8.o Risk Register	15
9.o Schedule Management	18

1.0 Project Overview

1.1 Project Overview - Description

Mythical Maze is a puzzle game which combines the falling tiles of a style reminiscent of Tetris (1984) and Dr. Mario (1990) with path making mechanics from games like Flow Free (2012) and platforms from games such as Fez (2012). The user plays as a small child who is lost on a family vacation to see the wonders of the world. The child must create paths using falling hollow blocks in order to navigate his way through the numerous obstacles across a variety of unique environments. In the child's travels around the world, he will journey to many famous sights such as Great Wall of China, the White House, and the Pyramids and find amazing treasures on a quest to make his way back to his family.

1.2 Detailed Project Description

When the user first opens the *Mythical Maze* game, he or she will be presented with an interactive game menu. Many options will be located on this menu. The first will be a player login, which allows users to access previously saved profiles. The second option will be a system to create a new profile. The third will be a tutorial level, which features the game rules, objectives, and controls. Finally, an exit button is provided in order for users to leave the game. The player may access the game via player login or new profile creation. By starting the game, information is extracted from the central database in order to create the specifics of the game. The database includes player credentials and game statistic information such as scores and level. When the user plays the game, certain statistics such as player level and score will change, therefore, when the game is saved, these parts of the database will be overwritten by the new information. The game can be saved at any point when playing the game and is automatically saved when the user completes a new level. The information will then be available to be accessed later when the user restarts the game. Once the game is saved, the user can then either exit the game or restart with a new profile.

Mythical Maze will be divided into a series of levels which progress in difficulty from simple levels that allow the player to get the feel for the basic game mechanics to more complex ones meant to challenge the player. Levels will switch between a number of different themes, each of which centers around a different iconic location and possesses unique challenges.

The player plays as a small child who controls falling blocks which are connected to form various shapes. Some walls on the sides of blocks will be solid while others will not be. The pieces of the puzzle will resemble pipes, with openings on the ends. These

pipes will be arranged into patterns in order to create a path for the user to travel through. Obstacles presented in the game, such as rocks, walls, and other miscellaneous debris, can be navigated with the careful positioning of the pipes. After the player is finished dropping blocks he must take control of the boy and make his way through the blocks to get to the end of the level. The pipe designs will be inspired by the blocks used in the Tetris game but will feature a variable number of blocks which will allow for more variety in the shapes of pieces. Openings will be available on the ends of the shapes to allow for navigation through the pipes. Once all the pipes are connected into a successful path, the user will traverse the area and move to the next level.

Besides the pipe aspect of the game, *Mythical Maze* will also feature elements of a platformer role playing game. As the areas of the pipes are traversed, new obstacles will be presented in which the user must jump between platforms as part of the individual missions. The order that the blocks appear will be random, providing the user with a different experience each time. As the game progresses, the user will face harder levels. These levels will have faster moving blocks, forcing the user to think quicker in order to solve the levels. Also, the user must place the blocks into spaces with more obstacles, which will also force the user to think more quickly.

1.3 Team Roles

Name	Job Title	Job Description
Justin Hill	Project Manager	In charge of managing the progress of the overall project and delegating jobs to other group members. Additionally, responsible for writing the code for the game along with the Lead Developer.
Richard Dong	Lead Developer	Responsible of writing the code for the project and assisting the Documentation and Organisational Manager with documentation.

Richard Wu	Graphics Manager	Responsible for developing graphics and music for the game. Works closely with the Project Manager and the Lead Developer to implement these assets into the final project.
Abhijeet Venkataraman	Documentation and Organizational Manager	Responsible for working closely with all other team members to manage the design and overall completion of the project. Responsible of overseeing the production and implementation of the project plan and Javadocs.

Fig. 1.3.1

1.4 Project Overview - Deadlines

15 September 2014, 12:01AM PST	Team registration opens
3 October 2014, 11:59PM PST	Team registration closes
7 November 2014, 11:59PM PST	First project plan submission deadline
19 December 2014, 11:59PM PST	Amended project plan submission deadline
23 February 2015, 11:59PM PST	Final project materials submission deadline
3 March 2015, 5:00PM PST	Presentation scheduling closes

Fig. 1.4.1

2.0 Project Life Cycle - Expanded Waterfall Design Methodology

The project will be developed in accordance with an expanded waterfall design methodology. The project will be divided into phases which will flow from one to another. However, in each phase previous phases will be assessed and reevaluated. The development phases are as follows:

- Requirements - The guidelines for the requested game are given and will be fully understood.
- Analysis - The team completes the task of determining needs or conditions of the requested

product. The requirements given by stakeholders analyzing, validating, and managing software or system requirements are all taken into account.

- Design - The game will be designed so that it is an action adventure Role Playing Game in an

8bit style. The player will control and manage his character as they progress through the game. The player will battle hostile non playable characters as they play in real time and may defeat them and collect other items in order to gain points and level up.

- Implementation - Code will be written to meet design specifications a neat and efficient manner by the team of programmers. Documentation will occur; its purpose is to make the job of future maintenance and enhancement easier as code is easier to read.

- Testing and Maintenance - Testing will be done on the game to ensure that the created product runs smoothly in the way described in the description. Different tests will be run to find software defects. The first tests will be aimed at giving feedback for all new code. At an alpha stage, a test plan is written and bugs are reported. During a beta stage, clear assignments will be given to the testing team on a daily basis.

- Operation - The game will be available for use after extensive testing for any bugs that inhibit and decrease the appeal of game play. Requirements will have been met thoroughly.

Lifecycle Diagram - Expanded Waterfall Design Methodology

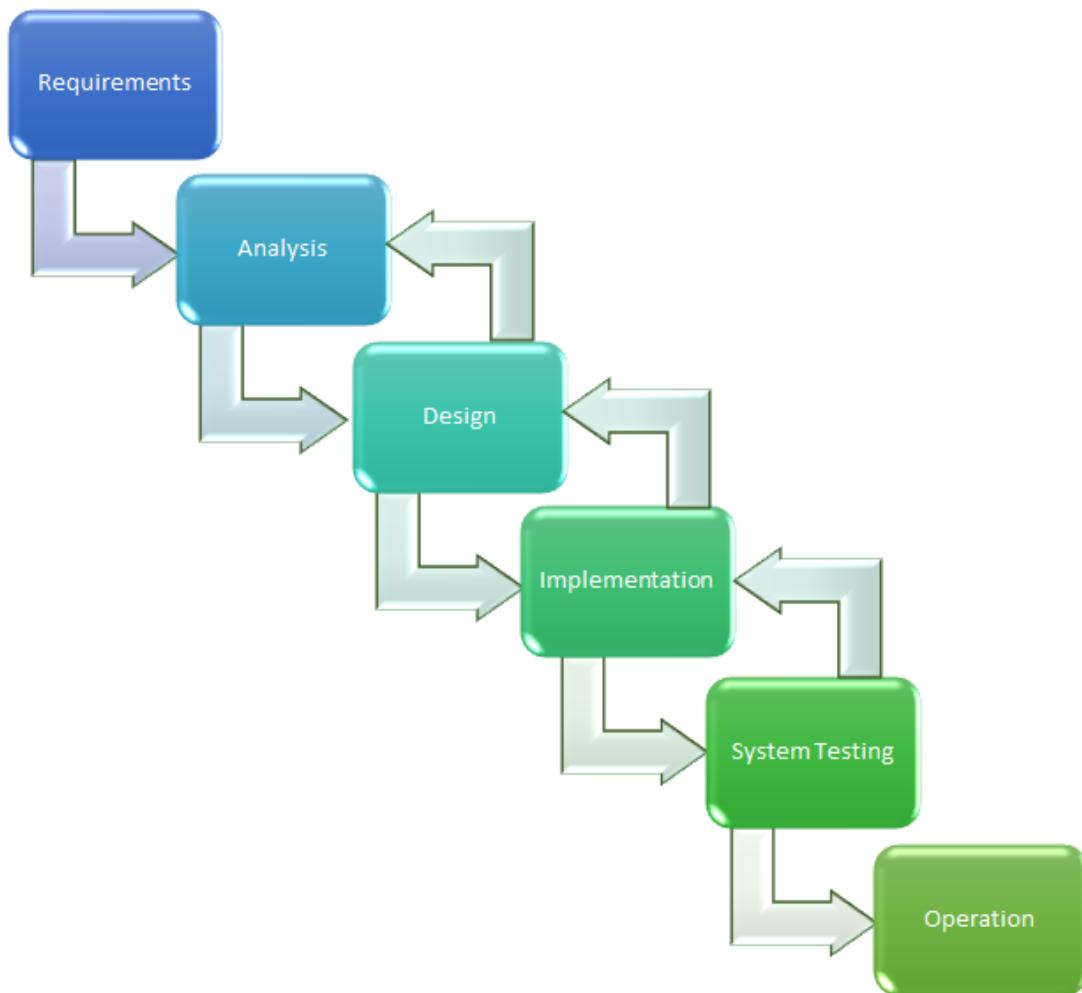


Fig. 2.0.1

3.0 Detailed Design

3.1 Data Flow Diagram:

The data flow diagram shows the flow of data in the *Mythical Maze* game. When the user first opens the game, they are presented with a menu. During a login, information entered will be checked against previous information stored in the database. New profile information can also be stored into the database. When the game starts, data stored in the database will be used to generate the game environment. This includes the game map, sprites, and character statistics. When levels are complete, new information can be stored into the database. The information in the database can be edited and rewritten.

Data Flow Diagram

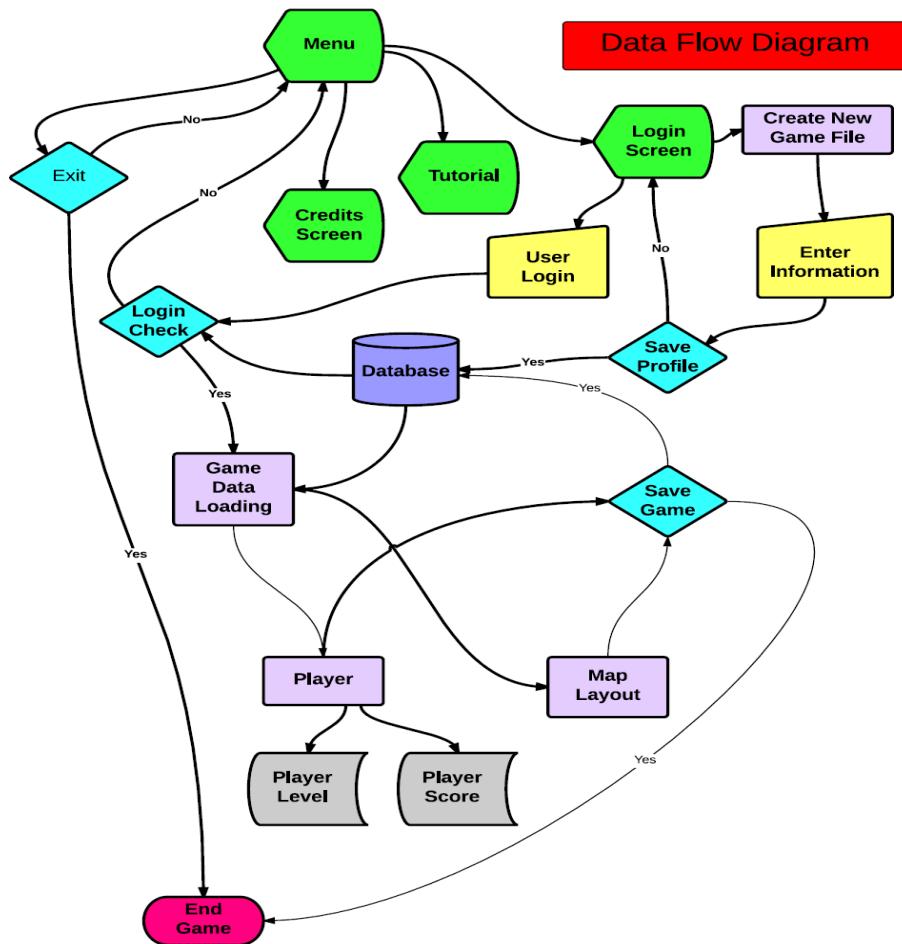


Fig 3.1.1

3.2 UML Diagram:

The UML diagram shows the plan for the class hierarchy of the *Mythical Maze*. The UML diagram shows the relationships between the classes of the game. A more detailed version can be found in the project materials folder.

UML Class Diagram

Fig. 3.2.1

Please refer to detailed version titled [MythicalMazeUML.pdf](#)

3.3 Storyboard:

The storyboard shows a plot summary of the *Mythical Maze* game. Each box in the storyboard shows a step the user can encounter during the playing of the game. We developed this storyboard to help plan our project design and the look of our art assets.

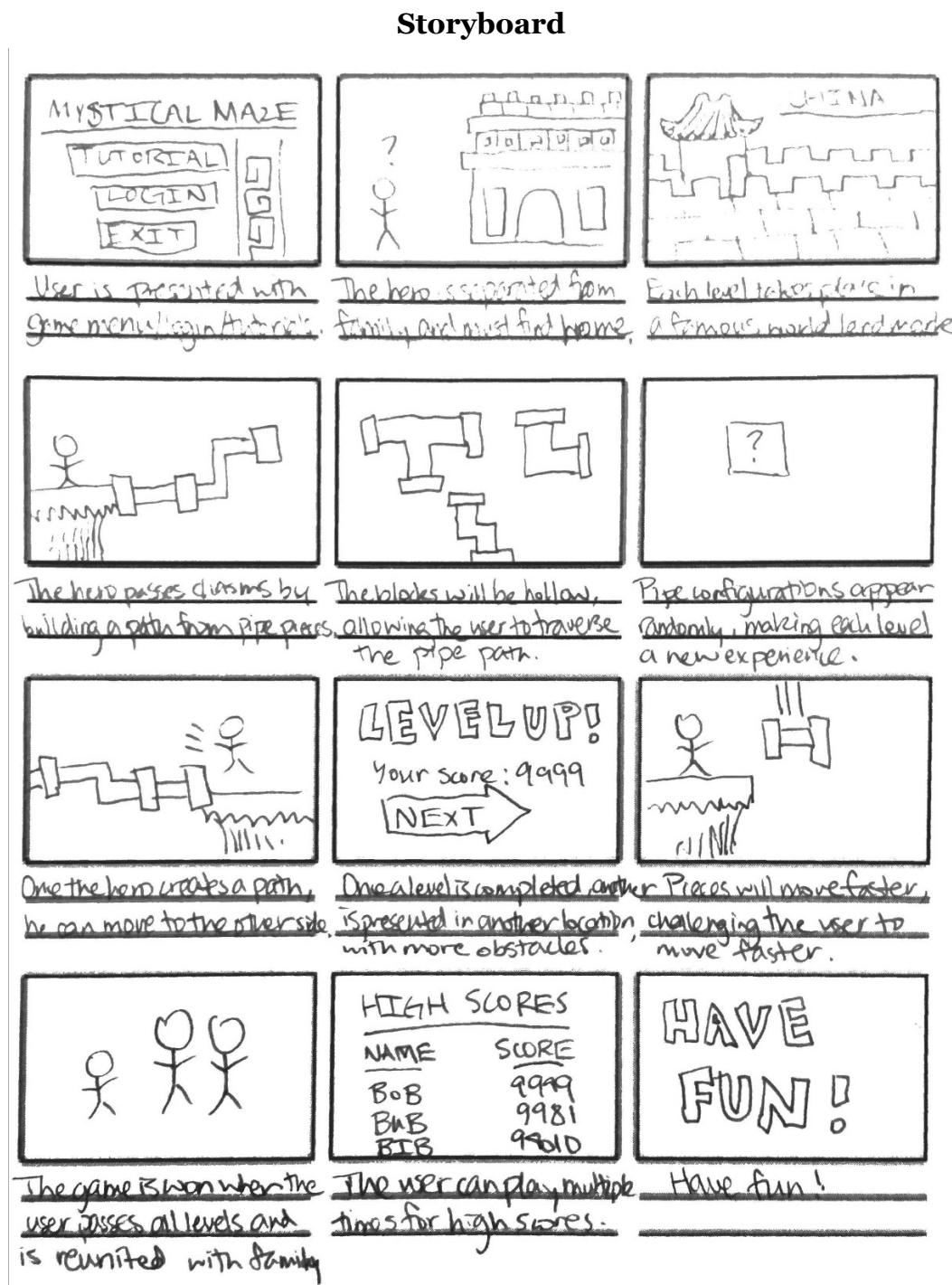


Fig. 3.3.1

4.0 Quality Management:

Quality will be ensured with periodic quality checks to make certain that the program runs as expected without glitches or errors. These checks include gameplay testing, the use of multiple programmers to address coding errors, multiple backups of important files, and compatibility checks on multiple systems. After the game is initially implemented, repetitive tests and modifications of the code will occur to insure a more succinct end product. Internally bugs and development suggestions will be logged in BitBucket's™ internal error tracking system. This will allow group members to easily share any bugs or problems with the game to assign a group member to fix or implement the problem. Externally a google form will be used for testers to log their findings and suggestions regarding the game. These could be anything from features they would like implemented to glitches they find to difficulty and balance issues.

Testing Form For External Testers

The screenshot shows a Google Form titled "Mythical Maze Testing Form". The form includes the following fields:

- A large text area labeled "What did you test?".
- A date selection field labeled "When did you test?" with dropdown menus for "Month" and "Day".
- A text area labeled "What bugs or errors did you find?".
- A text area labeled "What improvements do you think could be made?".
- A blue "Submit" button at the bottom.

Fig. 4.0.1

Internal Testing and Issue Tracking

Issues

+ Create issue

Filters: [All](#) [Open](#) [My issues](#) [Watching](#) Advanced search

Issues (1–5 of 5)

Title	T	P	Status	Votes	Assignee	Created	Updated
#5: We should have different sizes of block combinations			NEW		richarddong2016	just now	just now
#2: Main Menu Looks Bad			RESOLVED		Justin Hill	2014-09-24	a minute ago
#3: Buttons Don't Re-size With Menu			NEW			8 minutes ago	2 minutes ago
#4: Credit Screen Needed			NEW		Justin Hill	8 minutes ago	8 minutes ago
#1: There is nothing in this repository			RESOLVED	1	Justin Hill	2014-09-17	2014-09-17

Fig. 4.0.2

Internal Testing and Issue Tracking Subpage

Issue #5 NEW

We should have different sizes of block combinations

[Justin Hill](#) created an issue 48 seconds ago
We should have 1, 2, and 3 block combination in addition to 4 block ones.

Comments (0)

What do you want to say?

Resolve **Workflow** ▾ **More** ▾ **Edit**

Assignee richarddong2016
 Type proposal
 Priority major
 Status new
 Votes 0 [Vote for this issue](#)
 Watchers 2 [Stop watching](#)

Fig. 4.0.3

5.0 Resource Management:

The school system will provide all resources, specifically from the computer science lab, including computers, all necessary software, and internet onsite. When team members are not at school, all resources will be handled by the team members themselves. All team members should easily be able to access any documentation, code, or issue tracking anywhere as it will all be stored on Google Drive™ and BitBucket™.

6.0 Communications Management:

The primary means of communication between team members will be email. Every week the Project Manager will send out an email to each of the group members that details the progress on each element of the project that is in development. The email will also list upcoming due dates on the timeline and will detail what each group members should be working on. Additional emails will be sent before any date that requires a submission such as a project upload that details what all need to be completed by that date and by which member of the development team. Google Drive™ will be used for collaboration and sharing of documentation and resources between group members. BitBucket™ will be used for integrating GIT™ functionality into the development process. It will be used to allow multiple developers to collaborate on the code simultaneously and to share any issues, suggestions, and progress made with the rest of the team through branching, revision history, issue tracking, and committing.

7.0 Project Change Management:

If any changes to the project must be made from the initial plan then they will be addressed by the entire team and a decision will be made in order to ensure that the project still meets all requirements and that the proposed changes will be consistent with the rest of the project's implementation.

8.0 Risk Register:

Risks are inevitable in any large project. We have attempted to mitigate several potential risks documented in this risk register. For other risks and unforeseen challenges we have included a degree of cushion on each date in the timeline from when the material is actually needed and allotted more time than is necessary for completing the task.

Risk Description	Consequence	Mitigation and Recovery
Hardware failure or loss of data	A loss of files could potentially delay the project as they would need to be recreated. Time for this is not accounted for on the timeline.	All files will be uploaded to the cloud. All documentation and project assets will be placed on Google Drive™ and all code will be managed through BitBucket's™ online servers. These steps should minimize the consequences of hardware failure or data loss as nothing will be stored exclusively on a local machine. In the event that data is lost, the project timeline will be amended to make up for the additional time needed to recreate or rebuild what was lost.
Program is discovered to not be compatible with some platforms	This could potentially jeopardize the project as the product would no longer meet the requirements	The program will be tested on multiple different machines with different capabilities and operating systems throughout the development process. Using multiple tests will ensure that each system tested will be compatible. If certain operating systems do not support the program, different revised versions will be offered or the project will be revised so it will run on all platforms. Testing throughout the development process will ensure that any incompatibilities will be caught early and that minimal redesign and backtracking on development will need to be done.
Gameplay will not be engaging	The game would not be fun to play or	Projected Solution: Each team member will assist in the creative process of the

or will not be consistent with the intended design	would not meet the requirements	game's creation and will test the game to ensure that each of its aspects function as intended. Testing will occur in multiple phases and everyone will participate in these tests. Developers will submit anything they find when testing through BitBucket™ issue tracking. Outsider tester will also be used throughout the development process and will submit feedback and bug reports through Google forms. Getting a variety of testers throughout the development process will help ensure that the game lives up to both specification and remains engaging to new players.
--	---------------------------------	--

Fig. 8.1.1

9.0 Schedule Management

9.1 Integrated Master Plan:

We have established the following key milestones delineated in table 9.1.1.

Date	Description
10/23/14	The basic function and design of the game will be complete by the team. Some preliminary artwork and sound will have been created by graphics designer. The Project Manager and Lead developer will be in the process of structuring the code and producing a preliminary UML diagram.
11/5/14	All the basic graphics, music, and sound effects will be created by the Graphics designer. The Project Manager and Documentation Manager will have the project plan for the game finalized. The Lead Developer and Project manager will have implemented graphics, music, and sound into the game. The main menu will be complete and work will have basic on the basic tetris functionality.
11/11/14	More graphics and sound will be produced and implemented into the game. The basic tetris functionality will be finalized and some preliminary work will have been done on the character movement. Enough code will be in place for the Documentation Manager to begin producing Javadocs.
11/20/14	A basic playable game will have been developed by this point. The Documentation Manager will have produced Javadocs for some classes. All of the basic graphics and sound will have been created by the Graphics Manager and will be in the project.
1/7/14	The game and most of the documentation will be finalized. The team will begin working on final documentation, corrections, and begin developing a presentation.

Fig. 9.1.1

9.2 Integrated Master Schedule:

Date	Task	Task Leader
October 9th	Initial Storyboarding	Graphics Manager

	Completed	
October 16th	Game Function Finalized	Project Manager
October 21st	First Revision of UML Class Diagram	Lead Developer
October 22nd	First Sprites Produced	Graphics Manager
October 23rd	First Milestone Completed	Project Manager
October 24th	Basic Main Menu Working	Project Manager
October 27th	Internal Issue Tracking and External Testing Forms Created	Project Manager
October 28th	Life Cycle Finished	Documentation and Organisational Manager
October 30th	Initial Music and Sound Produced and Implemented	Graphics Manager
October 31st	Project Plan Finalized	Project Manager
October 3rd	All Materials Ready to Upload	Project Manager
November 4th	Timeline Analyzed and Adjusted	Documentation and Organisational Manager
November 5th	2nd Milestone Complete	Project Manager
November 6th	Basic Sprite Movement Implemented	Lead Developer
November 11th	3ed Milestone Complete	Project Manager
November 18th	Major Sprites Complete	Graphics Manager
November 20th	4th Milestone Complete	Project Manager
December 9th	Outline of presentation produced	Documentation and Organisational Manager
December 11th	Project Plan Revision Complete	Documentation and Organisational Manager

December 14th	All Documentation Finalized for upload	Project Manager
January 7th	5th Milestone Complete	Project Manager
January 9th	Readme Completed and tested with user who have never tried the game	Documentation and Organisational Manager
January 20th	Game testing and development complete	Project Manager
January 27th	All documentation revised and finalized	Documentation and Organisational Manager
February 2nd	All materials ready to upload	Project Manager

Fig. 9.2.1

9.3 Gantt Chart

We developed a gantt chart to help develop the project timeline and track what each team member is working on. The gant list each week of the project development horizontally and each task vertically. Each vertical line represents a group member working on a task. We used this as the basis for our project timeline and in developing our milestone objectives.

Gantt Chart

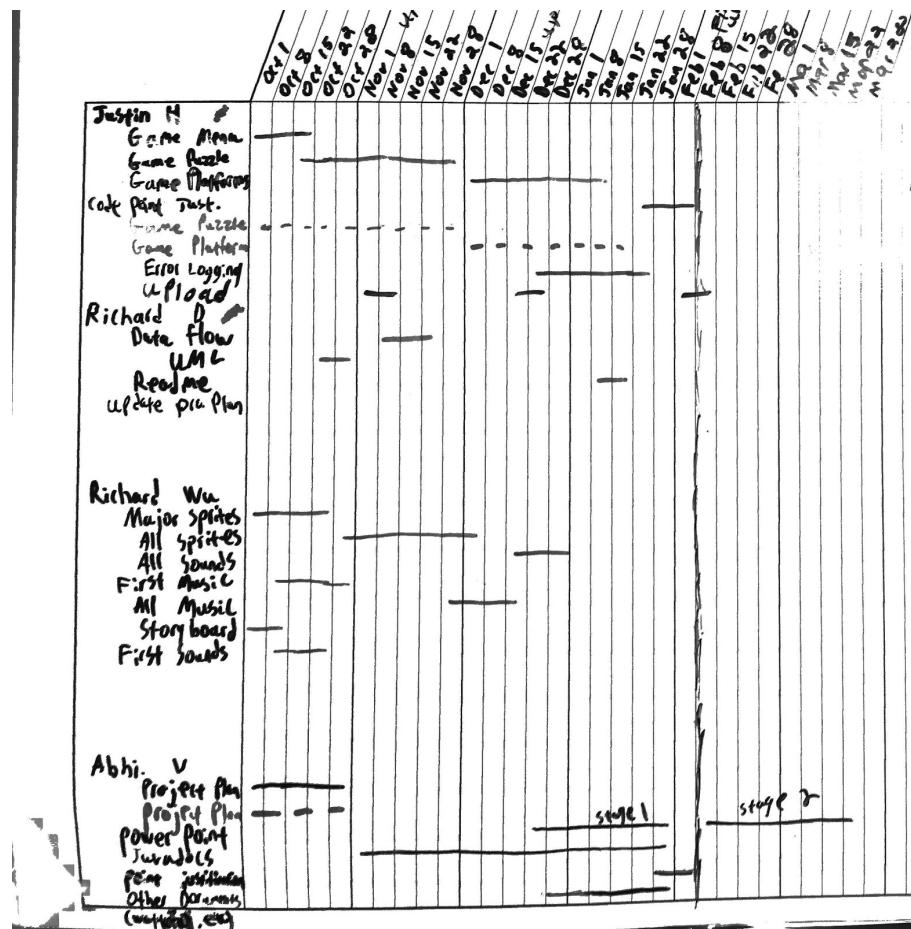


Fig 9.3.1