JUTSU

PROJECT MANAGEMENT PLAN

Final Version

2/19/2015

VERSION HISTORY

Version #	Implemented By	Revision Date	Approved By	Approval Date	Reason
1.0	Anne Zhou	11/20/2014	Michelle Zhou	12/01/2014	Initial copy
2.0	Sachin Mehta	12/22/2014	Diane Wilson	01/11/2015	Updated diagrams
3.0	Michelle Zhou	01/12/2015	Sachin Mehta	01/13/2015	Improved information
4.0	Anne Zhou	02/11/2015	Daniel Ho	02/19/2015	Final draft with all updates

UP Template Version: 11/30/06

TABLE OF CONTENTS

1 INTRODUCTION

1.1 PURPOSE OF PROJECT MANAGEMENT PLAN

The intended audience of the JUTSU Project Management Plan (PMP) is the project sponsor, senior leadership and the project team. The purpose of the PMP is to provide said parties with a complete and comprehensive overview of the project and its components, so as to increase efficiency, prevent confusion, and facilitate organization of materials. As the PMP is updated, it will also serve to notify said parties about and track updates, revisions and other changes.

1.2 PURPOSE OF GAME

The purpose of the project itself is to capitalize on the recent surge of popularity in 8 bit style games. This game is sponsored by Cognitive Though Media, which hopes to create a retro-style puzzle game reminiscent of Tetris and Dr. Mario. The game, set in feudal Japan, allows the player to help defend a small village against attacking Samurai.

2 EXECUTIVE SUMMARY OF PROJECT ABSTRACT

The project charter, written by the project sponsor, Cognitive Though Media, covers the system architecture requirements along with requirements for the actual game components, such as gameplay, functional, interface, performance, and database.

2.1 ASSUMPTIONS/CONSTRAINTS

System Architecture:

- Must run on Windows, Mac or Linux personal computers
- Minimum hardware specifications of minimum hardware specifications of a 1GHz CPU, 1 GiB RAM, and several hundred MiB of hard disk storage
- Coded in the following language:
    - Oracle Java 7

Common:

- Publishing and development credits followed by title screen
- Title screen: begin playing, adjust options (if any), view instructions, or quit the application
- Scrolling tiled sprites and unique level designs
- Original sounds for actions and background soundtrack
- Menu that can be accessed during gameplay with options to save the game (for loading and continuation of play later), exit to the title screen, and exit the application

Functional

- Multiple levels of difficulty
- Creative modes and designs

User Interface

- Data such as the upcoming items, high score, and current score
- Prevent information overload

Performance

- At no time during runtime should frame- rate be below 30 fps (frames per second)

3 PROJECT OVERVIEW

3.1 FUNCTION

The objective of the project is to create a puzzle style game similar to classics such as Tetris and Dr. Mario. The game is set in feudal Japan; the player will take the place of farmers trained in the art of ninjitsu and defend against hordes of incoming samurai. The samurai will march from the top of the map in various combinations of formation and armor color. The player's purpose is to efficiently last through increasingly difficult samurai raids by strategizing where to best place the invaders against the base and appropriately arranging and eliminating the enemies. He or she will have the option of changing the position and organization of the samurai groups and deciding whether or not the groups can "charge", or slide quickly toward the base. As the samurai pile up against the base of the map, an algorithm will calculate, depending on the number of rows of samurai present, damage to be applied to the ninjas. Another mechanic regarding the color of the samurai will allow the player to eliminate same-colored samurai from the board if three or more are adjacent to each other.

3.2 OVERVIEW

The application will open with credit to Cognitive Thought Media, followed by a reference to the development team. The meu will then appear, containing options to start a new game, view tutorials, access high scores, see credits, and exit the application.

Upon starting a new game, brief text will introduce the storyline behind the game. The player is nearby a small village in feudal Japan that is being attacked by Samurai. In Tetris-style gameplay, groups of Samurai will march up the screen toward the village gate. Each group will consist of four Samurai, each of any four different colors: purple, green, red, and yellow. The player will use the left and right arrow keys to move the incoming Samurai and the up arrow key to "charge", or quickly slide the current group in place against the wall.

In order to eliminate Samurai from the game screen, the player must slash with the mouse across any group of three or more adjacent Samurai of the same color. The Samurai below will then shift upwards to fill in the empty spaces. Power-ups, awarding additional time or eliminating more Samurai, will be conferred on the player for the removal of groups of three or more Samurai.

There are five different levels for the player to progress through, in order of increasing difficulty. All five levels are timed, and an optimized scoring algorithm balances the number of blocks removed, current health, and time elapsed to calculate total scores. The player may continue the game until health reaches 0, the point at which the village gate breaks, or until time runs out.

Power-ups will be awarded for the removal of groups of four or more Samurai and activate immediately. There are four different power-ups included in the game. The first kind slows time, so that the countdown proceeds less rapidly but the pieces move at the same speed as before, allowing the player to complete more moves in one game. The second power-up increases health, which in turn lengthens the game and increases the players score. The third power up provides the player with an overlay grid for assisted placement. And finally, a fourth power-up provides a score multiplier for nine seconds.

Pressing the pause key will allow the user to resume, return to the menu, or exit. All scores are saved to a text file upon closing a session.

4 GAMEPLAY FEATURES

All of the game tiles in Jutsu are animated, without slowing down framerate or inducing excessive strain on the machine. This makes the Samurai look like they are really marching toward the village gate. Additionally, Jutsu's graphics are not rendered with a grid system as they are in Tetris and its many variants; rather, pieces move pixel by pixel down the screen. This brings greater clarity to the "walking" animation of the Samurai tiles, making gameplay smooth and aesthetically-pleasing.

Two-way rotation allows pieces to both move downward and rotate at the same time, opening up new possibilities for strategic maneuvering. The graphics algorithm is also highly efficient, converting image files into image arrays and using the raster feature to render them on the game screen.

A novel scoring system along with power-ups and mouse functionality bring a new level of strategy and engagement to the game. The player starts with a certain amount of health points that are diminished as Samurai build up on the screen, but also receives points for removing Samurai from the screen. The total score takes into account both health points remaining and Samurai removed, which allows multiple avenues for constructing strategies.

Power-ups, used to multiply score, create gridlines for visual assistance, slow time, or increase health, are activated when five horizontally or vertically adjacent Samurais are clicked consecutively. Power-ups may be activated once per minute of gameplay, and present additional intermediate goals for the player while trying to achieve the highest possible score. The player's role in removing pieces by using the mouse adds another layer of involvement to the game, as the player must now consider carefully the pieces to be removed based on the effects on health points as well as the possibility of a future power-up.

5 SCOPE MANAGEMENT

The initial framework of the game will include the login screen along with a tiled map and moving sprites. The next stages include designing levels, calculating and applying statistics, and setting up the scoring algorithm, menu, and database. Each stage must be approved by all four team members before the next stage is programmed. Stages can be revised even after moving on to the next stage.

5.1 WORK BREAKDOWN STRUCTURE

The outline and pseudo-code is to be done by the programmer. Most of the actual coding will be done in conjunction by the programmer and manager, with occasional contributions by the artist and sound engineer. The manager is also in charge of the final testing for bugs and other problems before approval. The artist is in charge of the code that integrates JPEGs, PNGs, and other graphic files into the game.

5.2 CHANGE CONTROL MANAGEMENT

All changes and revisions to the project must be approved by at minimum three members of the development team. The changes, after being implemented, will be subject to examination by at least two members and careful scrutiny for bugs and other problems by the manager. If the changes in question decrease efficiency, functionality, or cause some other kind of error, they will be removed. However, if the changes are approved, they will be integrated into the project and the changes summary and date will be logged.

6 SCHEDULE/TIME MANAGEMENT

A hypothetical schedule will be created at the beginning of the project. It is the responsibility of the task manager to update the schedule with the latest information.

6.1 MILESTONES

Game Milestone 1 (October 23, 2014)
Sachin- opening window started, music coded
Daniel- sprite music composed and coded, sound effects started
Anne- UML Class Diagram done, game function finalized, keep up game error logging form
Michelle- 1 view of major sprites complete, ideas for other sprites drafted and finalized.

Game Milestone 2 (November 5, 2014)
Sachin- opening window complete with music and buttons, credit buttons working and finalized in portfolio
Daniel- opening music composed and coded
Anne- timeline adjusted, keep up game testing form, Software Development Lifecycle finalized
Michelle- all major sprites finished, buttons complete

Game Milestone 3 (November 11, 2014)
Sachin- sprites tested and working
Daniel- sound effects complete
Anne- UPLOAD game function and timeline, keep up game testing form and error logging form
Michelle- sprites tested and working, other sprites started

Game Milestone 4 (November 20, 2014)
Sachin- work on object bounds and gridlines
Daniel- work on scoring algorithm
Anne- Data Flow Diagram finalized in portfolio
Michelle- other sprites complete, start game backgrounds

Game Milestone 5 (January 7, 2015)
Sachin- final game state completed
Daniel- all music and sound effects complete and coded

Anne- all documentation UPLOADED, powerpoint presentation planned, project plan and timeline finalized
Michelle- game backgrounds complete

6.2 PROJECT SCHEDULE

Date	Deadline
Sept 12	Project description Google document
Sept 16	**Upload** and register project with BPA
Sept 18	Game portfolio cover sheet
Sept 23	Tentative game timeline
Sept 26	Game timeline with Gantt chart responsibilities finalized
Oct 1	All team and individual paperwork
Oct 9	Finalized game storyboard
Oct 16	Game complete function finalized
Oct 21	Game error log form started UML Class Diagram
Oct 22	Major hero/villain sprites (one view each) Sprite music composed and coded
Oct 23	Game Milestones 1
Oct 24	Opening window with dummy buttons working
Oct 27	Game testing form started
Oct 28	Software Development Lifecycle finalized
Oct 29	Opening music composed and working with opening window
Oct 31	Credit buttons working and finalized in portfolio
Nov 4	Sprites of major heroes/villains (all views)

	Adjustments to timeline
Nov 5	Game Milestones 2 Sprite movement tested and working
Nov 6	**Upload** game function and timeline
Nov 11	Game Milestones 3
Nov 12	Data Flow Diagram finalized in portfolio
Nov 18	Non-hero/villain sprites
Nov 20	Game Milestones 4
Dec 9	Tentative PPT presentation blocked out
Dec 11	Complete project plan and timeline finalized
Dec 16	**Upload** all documentation
Jan 7	Game Milestones 5 Game backgrounds
Jan 9	Readme.txt complete and tested
Jan 13	All game features finalized
Jan 20	All game testing finalized
Jan 27	All game documentation finalized
Jan 29	1st run through the PPT (equalize speaking parts)
Feb 3	2nd run through the PPT (exact script due)
Feb 4	All game-related material on Ms. Wilson’s flash drive
Feb 5	**Upload** complete project with documentation
Feb 20	3rd run through the PPT (external audience with comment sheet)

Feb 26	4th run through the PPT (verified by comment sheet)
March 3	5th run through the PPT (in class)
March 12	Skype presentation

7 QUALITY MANAGEMENT

The development cycle of the project will be conducted in a waterfall manner. The component will be designed based on the requirements by the programmer. The basic code will be implemented and verified by each member of the team based on their specific roles. Once the segment has been verified by every member, it will be entirely integrated into the software and will be subject to routine maintenance as the other components of the project are completed.

8 COMMUNICATIONS MANAGEMENT

The team will use BitBucket with Mercurial, a distributed version control system, to share code and track issues and bugs. Other information will be sent through email and IM.

8.1 COMMUNICATION MATRIX

Name	Role	Email
Diane Wilson	Sponsor	dwilson1@ccisd.net
Anne Zhou	Manager	azhou997@gmail.com
Sachin Mehta	Programmer	smehta1215@gmail.com
Michelle Zhou	Artist	michellezhou_66@sbcglobal.net
Daniel Ho	Audio Engineer	danielho133@gmail.com

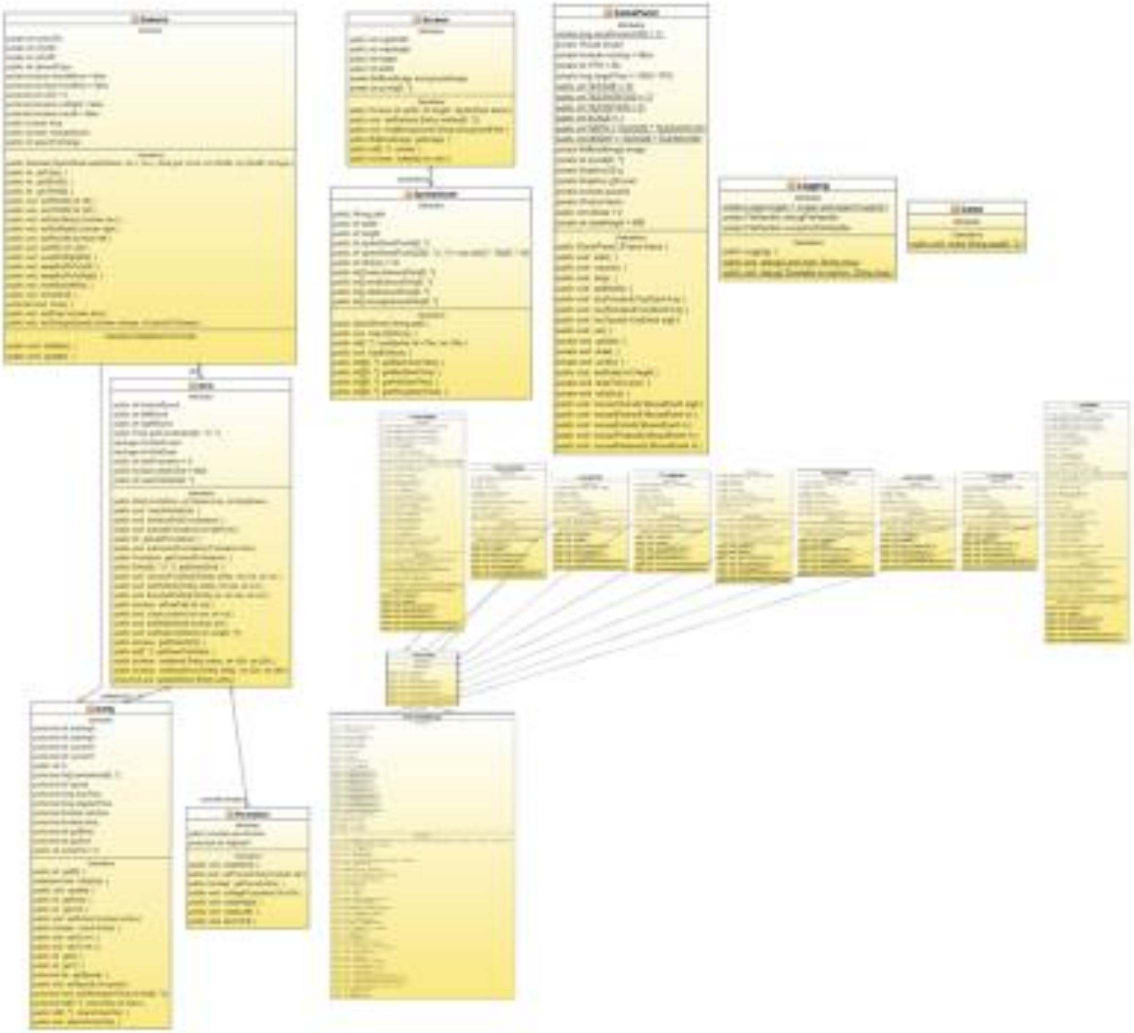
9 ISSUE MANAGEMENT

All problems that arise, such as performance issues and other bugs, will be logged to an external text file. The log and information contained in it will then be utilized to debug the project. See Appendix B.

10 DATA AND DIAGRAMS

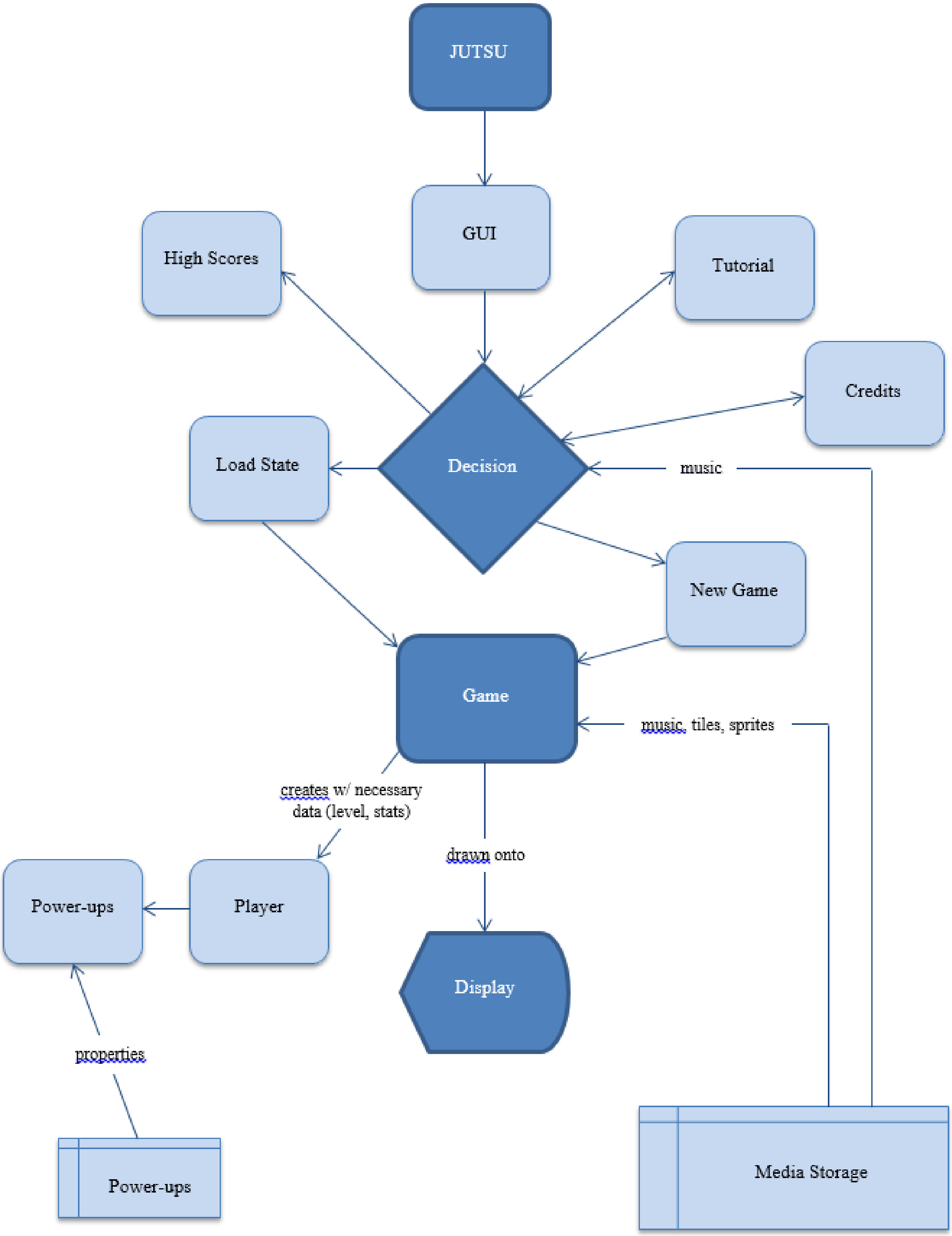
Data and diagrams will be outlined prior to game development to ensure successful integration of the projects components.

10.1 UML CLASS DIAGRAM



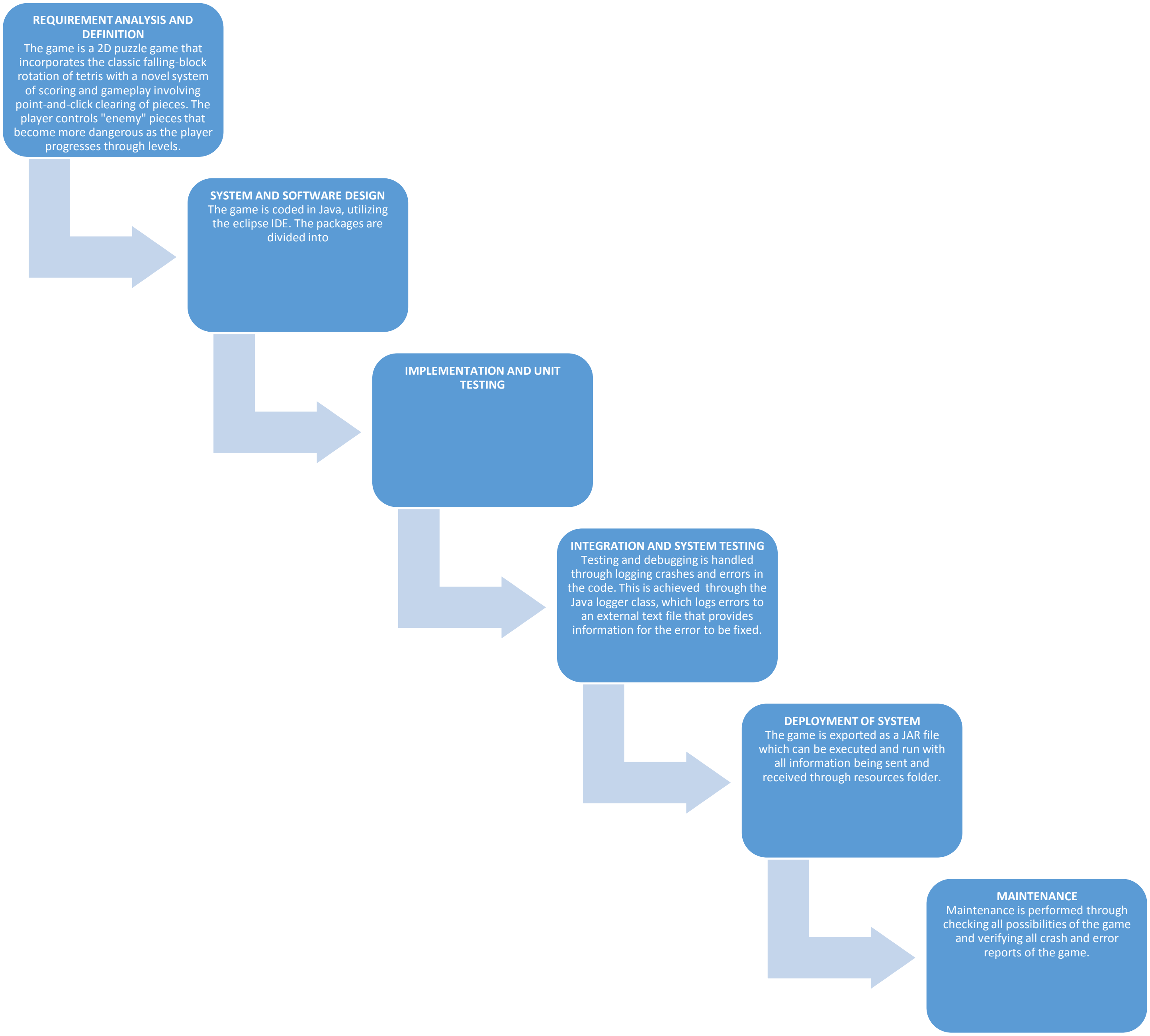
See accompanied UML for clarity.

10.2 DATA FLOW DIAGRAM



10.3 WATERFALL DIAGRAM

The Waterfall Diagram (Life Cycle) shows the project development phase, from understanding the requirements, to planning the creation process, implementing the process, and confirming the process.



REQUIREMENT ANALYSIS AND DEFINITION
The game is a 2D puzzle game that incorporates the classic falling-block rotation of tetris with a novel system of scoring and gameplay involving point-and-click clearing of pieces. The player controls "enemy" pieces that become more dangerous as the player progresses through levels.

SYSTEM AND SOFTWARE DESIGN
The game is coded in Java, utilizing the eclipse IDE. The packages are divided into

IMPLEMENTATION AND UNIT TESTING

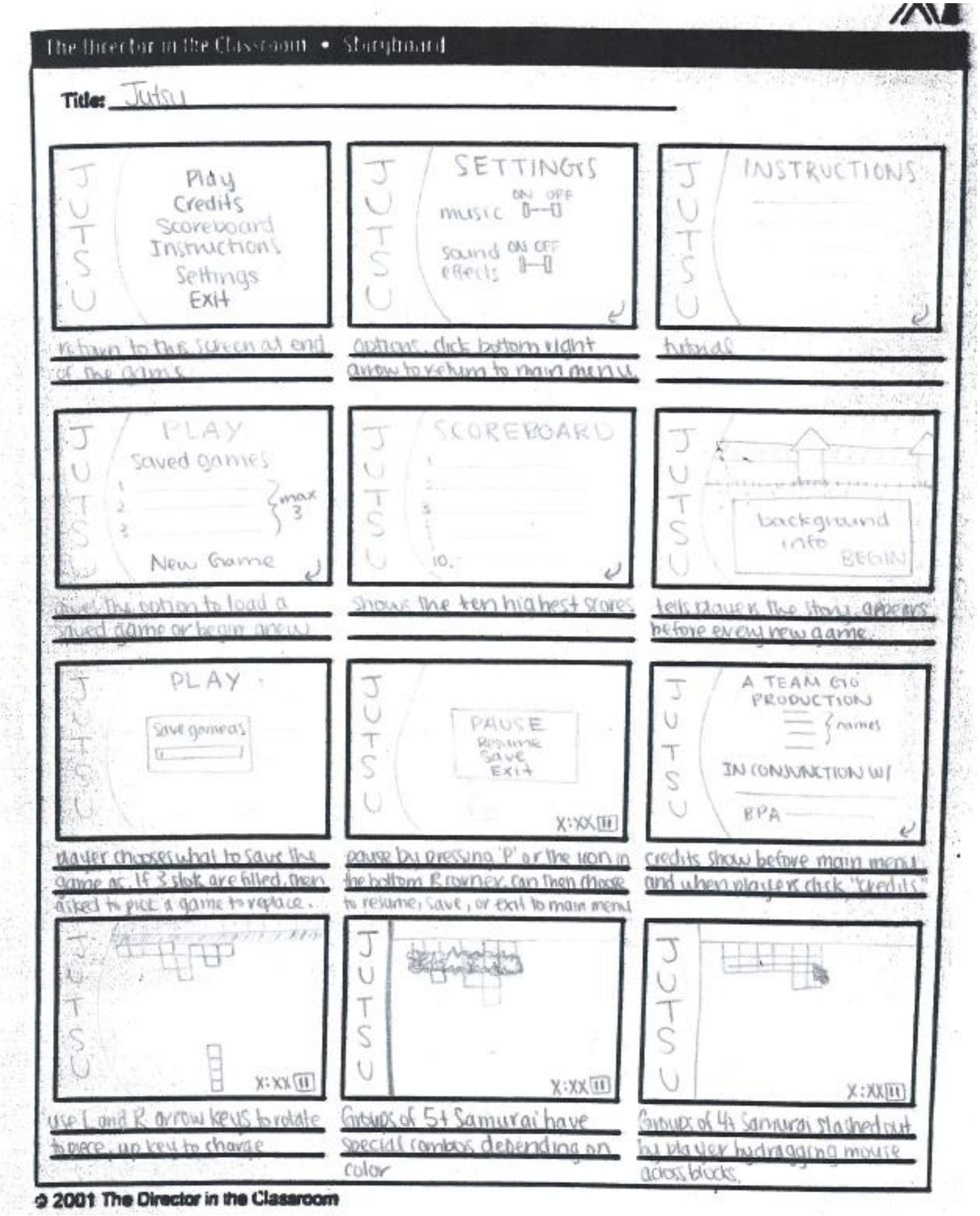
INTEGRATION AND SYSTEM TESTING
Testing and debugging is handled through logging crashes and errors in the code. This is achieved through the Java logger class, which logs errors to an external text file that provides information for the error to be fixed.

DEPLOYMENT OF SYSTEM
The game is exported as a JAR file which can be executed and run with all information being sent and received through resources folder.

MAINTENANCE
Maintenance is performed through checking all possibilities of the game and verifying all crash and error reports of the game.

11 STORY BOARD

The story board outlines the features in the game and correlates them to the planned graphics. The following presents the various scenarios for the final project, and what the projected output should look like.



The Director in the Classroom • Storyboard

Title: Jutsu

return to this screen at end of the game.

options. click bottom right arrow to return to main menu.

tutorial

gives the option to load a saved game or begin anew.

shows the ten highest scores.

tells player the story. appears before every new game.

player chooses what to save the game as. If 3 slots are filled, then asked to pick a game to replace.

pause by pressing 'P' or the icon in the bottom R corner. can then choose to resume, save, or exit to main menu

credits show before main menu and when players click "credits"

use L and R arrow keys to rotate the piece, up key to change

Groups of 5+ Samurai have special combos depending on color

Groups of 4+ Samurai slashed out by player by dragging mouse across blocks.

12 APPENDIX A: PROJECT MANAGEMENT PLAN APPROVAL

The undersigned acknowledge they have reviewed the JUTSU **Project Management Plan** and agree with the approach it presents. Changes to this **Project Management Plan** will be coordinated with and approved by the undersigned or their designated representatives.

Signature: ______________________ Date: __________

Print Name: ______________________

Title: ______________________

Role: ______________________

Signature: ______________________ Date: __________

Print Name: ______________________

Title: ______________________

Role: ______________________

Signature: ______________________ Date: __________

Print Name: ______________________

Title: ______________________

Role: ______________________

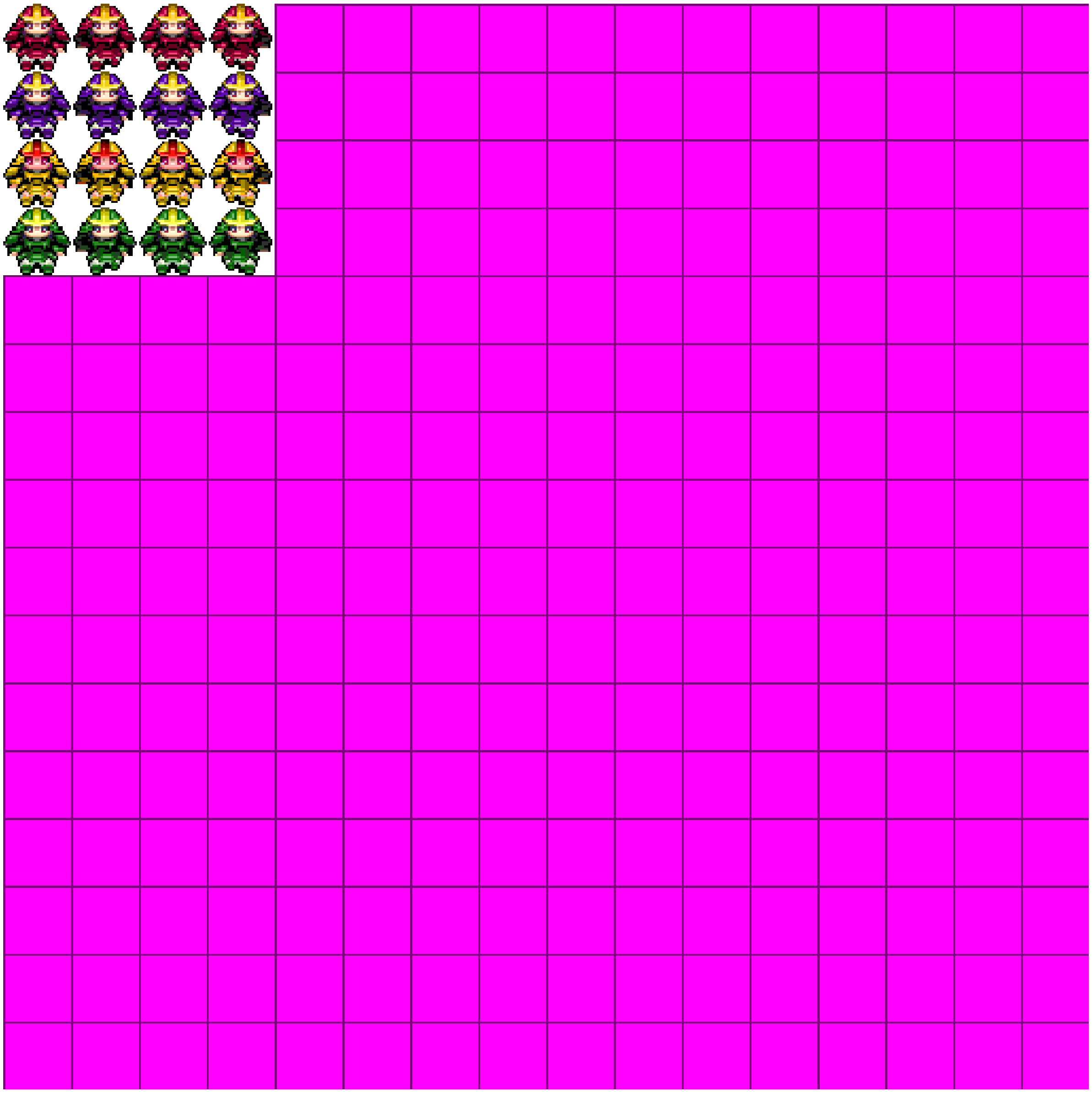
13 APPENDIX B: REFERENCES

The following table summarizes the documents referenced in this document.

Document Name and Version	Description	Location
Issue Log [1.0]	External text file log of issues encountered during program creation and editing	*/jutsu*
Error Log [1.0]	Online shared log of errors encountered during testing	*GitHub*
Game Testing Form [1.0]	Online shared log for testing various parts of the project	*Google Drive*

14 APPENDIX C: GRAPHICS

(RE-COLORED SPRITESHEET)



(BACKGROUND ARTWORK)