COMP4920: Management and Ethics

Project Plan

Tetris Fight

Jiazhou Liu (Joe) - z3351904
Mathew Yeap - z3290568
Nelson Wee - z3352078
Xuanru Sun (Sean) - z3398364
Natalia Vaudagnotto - z3377468

Contents

1. Project Overview

Tetris is a popular block puzzle game created by Alexey Pajitnov in 1984. Tetris had the goal of matching seven specific geometric shapes (I, J, L, O, S, T, Z) consisting of four smaller blocks, Tetriminos, into a row of ten blocks called lines. This is achieved by controlling individual falling blocks, moving them sideways or by rotating the block by 90 degree movements. When a row is created, it is deleted and all blocks above it fall by the number of rows cleared. After a certain number of rows cleared or after a preset interval, the Tetriminos begin to fall faster incrementally. The game is concluded when the player is unable to handle the speed in which the Tetriminos are falling, and inevitably passes the upper limit of the available playing field. The playing field is 10 blocks wide and 20 blocks tall.

The aim of our project, Tetris Fight, is to create a Tetris-like game on the PC with speed and competitive action in mind. This will be done by improving upon the traditional gameplay of Tetris to create a more fluid and customizable gameplay experience, and creating numerous additional game modes. The game will feature multiple single player modes, and additional two-player modes on the same computer. It will also feature an AI opponent for two player-modes.

Our target audience are people who enjoy the traditional game of tetris, who are also interested in an increased competitive and fast gameplay along with new functions to add new variety. The main platform we have chosen is the computer, as our goal for Tetris Fight enable quick gameplay and reaction time. Although from a business perspective, porting the game on mobile devices like smartphones, would allow for a wider market and audience, we feel this is ineffective as many ported games have shown that accuracy, speed, and control all suffer greatly with the controls available on the devices.

To potentially produce revenue for the game, our marketing strategy is a Free-to-Play model, with additional non-intrusive advertisements and micro-transactions for customizable music, icons and effects. None of the micotranscations can drastically change the player's ability to play the game.

Our total development and testing time is estimated to take roughly 4 weeks.

2. Goals and Scope

2.1 Project Goals

Project Goal	Priority	Comment
Functional Goals	5	
Basic Ruleset implementation	1	Rotation, Hard Drop, Soft Drop, Line Clear, Left/Right
Additional Gameplay	2	Storage, Bombs, Reject, Combos, Customizable settings
Solo Game Modes	3	Timed, Level, Race, Puzzle
Two Player Modes / AI opponent	4	Fight, Timed, Race
Tutorials / Documentation	5	Rulebook, Instruction Manuals
Business Goals	2	
Development Time	1	Releasing the game within the timeframe
Marketability	2	Potential for creating revenue
Quality Goals	3	
Bug Fixing	1	
Alpha Testing	2	Basic gameplay testing
Beta Testing	3	“Fight” Mode testing

2.2. Project Scope

Our main deliverable features of Tetris Fight include:

Gameplay additions include:

- Customizable Tetrimino movement speed and icons
- A "*Storage*" function, giving players the option to store and swap blocks anytime with their current block.
- Cumulative "*Combo*" points, where successive cleared lines generate greater points.
- "*Bomb*" lines, which appear as lines generated by their opponent or the game. By touching the bomb in the line with their Tetromino blocks, players can extend their combos and clear the lines. (*Competitive Mode only*)
- "*Reject*" function, where a player can throw away whatever Tetromino they get, but receive a penalty for doing so. (*Competitive Mode only*)

Game Modes:

- Classic "*Timed*" mode, where players clear as many lines as they can within a time limit. Scores are based on the number of Lines cleared..
- Classic "*Level*" mode, where players pass levels of increasingly fast drop speed. Scores are based on the number of Levels and Lines cleared.
- "*Race*" mode, where the player tries to clear a set number of lines as quickly as possible. Scores are based on the best timing.
- "*Fight*" mode,
    - Direct Competitive Mode where the number of lines, and bonus points, generated by combos or bombs, cause generated lines on their opponents field
    - A "*knockout*" or KO occurs when a player exceeds the upper limit of their playing field due to the other player's lines. The KOed player's field is then cleared of all their opponent's generated lines, their opponent gets a KO point, and the game continues.
    - Scores are based on the highest number of KO points or number of lines if KO points are equal. If a player manages to get a 4 KO points first, that player automatically wins.
- "*Puzzle*" mode, similar to "Fight" mode with the addition that both players start the match with a generated map of blocks.

All game modes can be played individually, against another player, or against an AI.

2.3 Excluded

All two player modes will be available on the locally on the same computer only, and will not feature an online network mode. All advertising and microtransactions are beyond the project's design scope and will not be added.

3. Project Organisation

3.1 Project Manager

Role	Name
Project Manager	Natalia Vaudagnotto
Technical Project Mgr.	Mathew Yeap

3.2 Project-internal Functions

Function	Name
Quality Assurance Risk Management	Natalia Vaudagnotto
System Test Lead	Mathew Yeap
Game Database Lead	Nelson Wee
AI Lead	Jiazhou Liu (Joe)
Game Mode Lead	Xuanru Sun (Sean)

3.3 Project Team

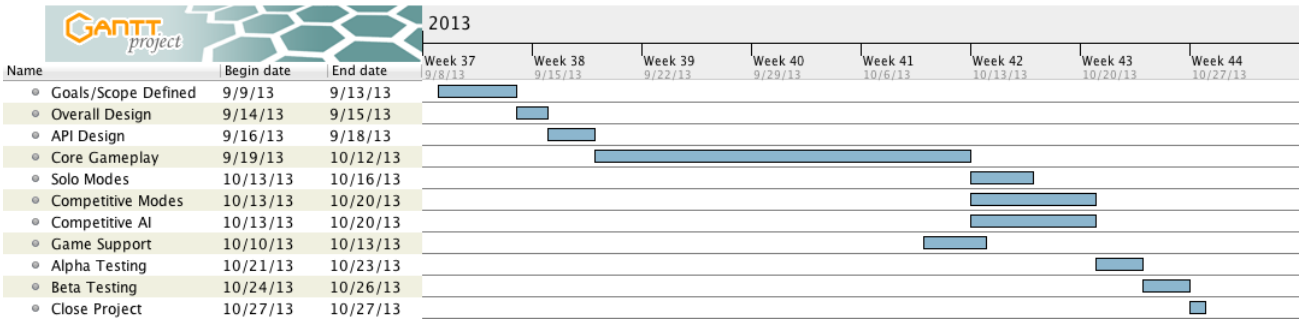
Name	Role	Responsibility
Natalia Vaudagnotto	Project Manager, Assistant Project Programmer	Group planning and time management.
Nelson Wee	Assistant Project Manager, System Support Programmer	Assist in group planning and time management. Player support systems, i.e Tutorials, Knowledge Base
Mathew Yeap	Technical Project Mgr.	Lead Programmer in charge of API, GUI, and system design
Jiazhou Liu (Joe)	Project Programmer	Advanced Gameplay Mechanics, AI Development
Xuanru Sun (Sean)	Project Programmer	Individual and Competitive Game Modes

4. Schedule

4.1 Milestones

Milestones	Description	Members	Milestone Criteria	Estimated Days Req	Planned Date
M0	Start Project				09/09/13 - 13/09/13
	Project goals and scope defined	All	Deliverable features and modes defined	3	13/9/13
M1	Design Phase				14/09/13 - 18/09/13
	Overall Design	All	Overall architecture design reviewed	2	16/9/13
	API Design	Mathew Yeap	Basic API reference completed	3	18/9/13
M2	Development Phase				19/09/13 - 20/10/13
	Core Gameplay	Mathew Yeap, Jiazhou Liu (Joe), Xuanru Sun (Sean)	Playing Field, Tetrimino Movement, Line Clear, Storage, Combos, Reject, bombs, GUI	15	12/10/13
	Solo Modes	Xuanru Sun (Sean), Nelson Wee	Timed, Level, Race, Puzzle	3	16/10/13 27 Total
	Competitive Modes	Xuanru Sun (Sean), Natalia Vaudagnotto	Fight, Race, Puzzle	7	20/10/13
	Competitive AI	Jiazhou Liu (Joe), Mathew Yeap	AI can successfully compete in competitive modes	7	20/10/13
	Game Support	Nelson Wee, Natalia Vaudagnotto	Tutorials and guides available for players	5	15/10/13
M3	Testing Phase				21/10/13 - 26/10/13
	Alpha	All	Core Gameplay and solo modes stable	2	23/10/13
	Beta	All	Two Player modes, and AI system stable	3	26/10/13
M4	Close Project				27/10/13

4.2 Gantt Chart



The Tetris Fight Gantt Chart file is documented in [1] and a screenshot is shown above.

4.3 Development Environment

Development of the project will be carried out on each team members' individual computers. The operating systems that will be used for development are Mac OS X 10.7 and Windows 7. Java has been chosen as the programming language. There should not be any problems with developing on different operating systems as Java is platform independent. All programmers will be using the Eclipse IDE.

The decision to use Java was mainly influenced by the following criteria:

- Familiarity with the language from all team members
- It is independent of the operating system running the computer
- Object-oriented because it matches the modular nature of this game
- Robustness: Java is robust which allows reliability in our game
- Java is efficient enough to make our game playable.

5. Risk Management

Risk Management will be identified, prioritized and handled by the Project Manager, and the Asst. Project Mgr.

Type	Priority	Control / Mitigation
Progress Delay	1	Weekly Meetings to identify, and prioritize the areas of delay and focus different members on prioritized tasks.
Development Problems	2	Special Development Meetings will be held to identify and solve issues
Online Data Loss / Corruption	3	Local copies of all Project Data will be stored before sharing them online.

Our main strategy for handling risks, follows the A-CAT Strategy.

- **A**void Risk - Modify plans to circumvent the problem
- **C**ontrol / Mitigate Risk - Reduces the impact of the risk by taking countermeasures
- **A**ccept Risk - Take the chance of the negative impact
- **T**ransfer Risk - Outsource the risk, specifically via redistribution to other members. Third parties will not be used.

This framework for software risk management is supported by two groups of practices:

5.1 Software Risk Evaluation (SRE)

We have set up two main meetings to address and mitigate all project risks. This ensures coverage of all areas of potential software technical risks. Both the Weekly Project Meetings and Development Meetings will involve all project team members to evaluate and resolve potential risks and resolve them quickly.

We have decided to use the GanttProject software to help manage our project schedule and identify immediate risks in our planning. The biggest risk at this stage is the deliverable dates of all the development milestones as development times may take longer than originally estimated. However, to prevent the overlap we have scheduled longer periods to the more intensive development milestones due date than estimated allowing for some leeway.

6. Communication and Reporting

A weekly meeting is scheduled where all the progress will be traced and problems will be reported to the project manager. This will help us to meet the deadlines and tackle any problem that might arise.

Type	Method	Frequency	Comment	Participants
Project Meetings	In-Person	Weekly and When required.	Project Progress, Problems and changes	Project Manager Project Team Members
Sharing Project Code	Shared Server - Bitbucket	Always Available	Project Code and updates	Project Team Members
Sharing Project Documents	Shared Server - Google Drive	Always Available	Project Documentation and Reports	Project Team Members
Development Meetings	Online	When required during Development Phase.	Discuss Development progress and issues	Project Team Members
Final Project Meeting	In-Person	M4	Debrief	Project Team Members

7. Configuration and Change Management

During the Weekly and Development Meetings we will identify and document the need for any changes. This is done through analysis and evaluation of the change required and subsequently, submitting a formal change request and producing a change proposal. If the change is approved by the responsible parties, we will verify and implement the release of change.

8. Quality Assurance

Software testing will be performed upon completion of every function and feature to ensure that immediate problems are found and fixed before new code is implemented. This will be done via unit testing, and by additional black and white box testing at each developmental milestone of our game to ensure completeness.

After the competitive modes of our software been completed, we will conduct performance and stress tests to ensure that the game is consistent in speed and responsiveness.

Once we reach the Final Testing, M3, stage, we will have extensive Alpha and Beta testing to verify that all parts of the game are stable before the final product demonstration and subsequent release.

9. References

[1] Gantt Chart for Tetris Fight <TetrisFight.gan>
[2] Tetris Rules
[3] Project API