SE 3XA3: Software Requirements Specification
Tetris Tussle
Group 2

Nicholas Lobo
lobon3
400179304

Matthew Paulin
paulinm
400187147

David Carrie
carriedd
000661652

February 13, 2021

Contents

List of Tables

List of Figures

Table 1: **Revision History**

Date	Version	Notes
February 7 2021	1.0	Section 1, Non-Functional Requirements
February 10 2021	1.1	Section 2
February 11 2021	1.2	Section 4, Functional Requirements
February 12 2021	1.3	Corrected grammar and added detail

1 Project Drivers

1.1 The Purpose of the Project

With the recent emergence of large scale triple A games with advanced features flooding the market there has been an absence of puzzle games that require quick decision making and careful planning. It was decided to put a spin on a simple but beloved classic to fill that void. Tetris took the world by storm in the 1980's becoming the first hit game ever created by selling over 202 million copies. The aim of this project is to take a bare bones version of this game and update it to better suit the modern standards of gaming through updated visuals and player versus player (PvP) interaction.

1.2 The Stakeholders

1.2.1 The Client

The clients for this system are Dr. Asghar Bokhari, the professor for 3XA3 as well as the teaching assistants for this course. These clients are responsible for monitoring the development of the product and ensuring it meets their given criteria. They can also offer clarification for any ambiguities about the project. Lastly, they will determine the degree to which the requirements in this document are met by the final product.

1.2.2 The Customers

The customers include anyone who will play Tetris Tussle. This includes people of all ages who own or have access to a computer. They must also have a stable internet connection to access the game as well as a modern web browser. The customers will use the software as a form of entertainment and to compete with people around the globe.

1.2.3 Other Stakeholders

Any current or future developers of this project are also stakeholders.

1.3 Mandated Constraints

- The product must be completed by the week of April 12$^{\text{th}}$.

- The product must be playable on Windows 10, Mac OSX, and Linux.

- The product shall support automated testing.

- The product must be free to play.

- The product must remain open source.

- Development costs shall not exceed $0.

- The project must be a redevelopment of an open source project.

- The project shall contain some graphical or gameplay improvements compared to the original.

1.4   Naming Conventions and Terminology

Tetromino: A Tetromino is the geometric shape used in Tetris. It is composed of four squares connected at their edges to form 7 different shapes.

- **Active Tetromino:** A Tetromino that can be moved by the player.

- **Held Tetromino:** A Tetromino that is saved and can be used at another time by the player

- **Dead Tetromino:** Is when the bottom of an Active Tetromino hits the top of a Dead Tetromino or the end of the board.

1.5   Relevant Facts and Assumptions

- Users can read English at an intermediate level.

- Users can operate a computer.

- Users can navigate the internet with a modern web browser.

- Users will have a system with at least 2GB of RAM and a processor clock speed of at least 1GHz.

2 Functional Requirements

2.1 The Scope of the Work and the Product

The scope of work is the redevelopment of the open source project located at `https://github.com/dionyziz/canvas-tetris`. The project scope contains additional elements as well as redeveloping and implementing the core game of Tetris. These additional elements are: a user interface including a game menu, a scoring and leader board system, and the ability to play a 2 player multiplayer game. We will only be providing the software and system to enable users to access the features above through modern web browsers with JavaScript enabled and over an internet connection.

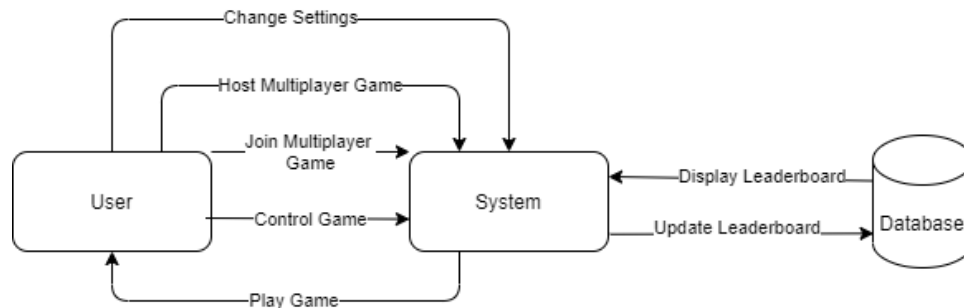### 2.1.1 The Context of the Work

Figure 1: Work Context Diagram

Table 2: **Work Partitioning**

Event Number	Event Name	Input	Output
1	View Leader board	Mouse	Leader Board
2	Create Game	Mouse	Multiplayer Game Screen
3	Join Game	Mouse/Keyboard	Multiplayer Game Screen
4	Hold Tetromino	Keyboard	Game State
5	Move Tetromino	Keyboard	Game State & Score
6	Start Game	Mouse	Game Screen
7	Exit Game	Mouse/Keyboard	Home Screen

Table 3: **Work Partitioning Summary**

Event Number	Summary
1	The user must be able to view the highest scores.
2	The user shall be able to create a game lobby that other users can join for multiplayer gameplay.
3	The user must be able to join a game hosted by another player.
4	The user shall be able to hold the active Tetromino or retrieve the held Tetromino.
5	The user shall be able to rotate or move the active Tetromino within the bounds of the game.
6	The user must be able to go to the game screen from the home screen
7	The user shall be able to exit the game screen and return to the home screen

2.1.2 Work Partitioning

2.1.3 Individual Product Use Cases
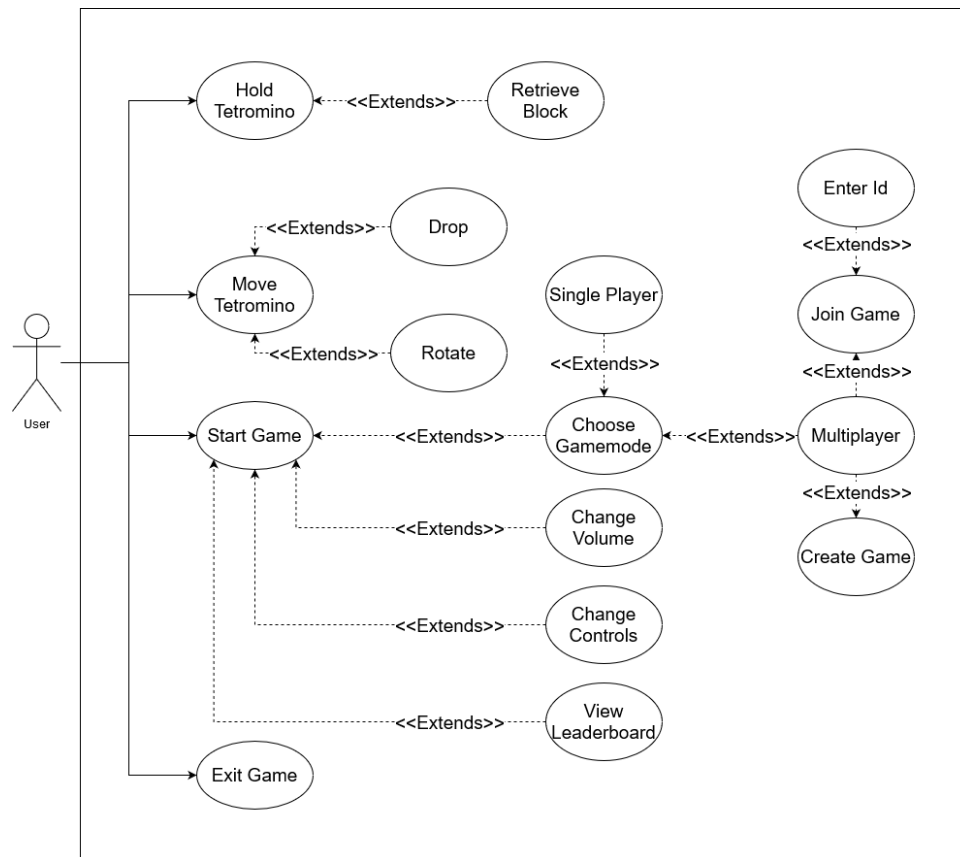
Figure 2: System Use Case Diagram

Table 4: **Use Case 1: Hold Tetromino**

Use Case UC-1:	Hold Tetromino
Related Requirements	BE1: FR1 - FR3
Initiating Actor:	The user
Actor's Goal:	To hold an active Tetromino for later use
Participating Actors:	The user
Preconditions:	The current active block was not held before
Post conditions:	The current active block will be held
Flow of Events for Main Success Scenario:	
→	**1. User** has a current active block and presses the key to hold the block
	2. extend::*Retrieve Block*
←	3. **System** signals to the **user** the Tetromino has been held and switched the current active Tetromino

Table 5: **Use Case 4: View Leader Board**

Use Case UC-4:	View Leader Board
Related Requirements	BE4: FR1 - FR4
Initiating Actor:	The user
Actor's Goal:	To view the players with the highest scores
Participating Actors:	The user
Preconditions:	The user is on the home or game screens
Post conditions:	The leader board is shown to the user
Flow of Events for Main Success Scenario:	
→	**1. User** is on the home page or has completed a game and selects the view leader board option
←	2.**System** signals to the **user** and displays the leader board

2.2 Functional Requirements

BE1. The user wants to hold a Tetromino.

FR1. The system shall provide the user with the ability to hold an active Tetromino.

FR2. If there is a held Tetromino, the system must switch it with the current Tetromino.

FR3. The user shall not be able to hold more than one Tetromino until the current Tetromino is placed.

BE2. The user wants to move a Tetromino.

FR1. The user shall be able to move an active Tetromino down, left, and right.

FR2. The user must have the ability to rotate an active Tetromino 90° at a time.

FR3. The user shall be able to rotate an active Tetromino clockwise and counterclockwise.

FR4. The system shall provide an option to drop an active Tetromino to the bottom of the screen.

FR5. If an active Tetromino reaches the bottom of the board or a previously placed Tetromino, it becomes a dead Tetromino and must stop moving.

FR6. Once a Tetromino is placed, the next Tetrimino shall become active and appear at the top of the board.

FR7. After a Tetromino is placed, it must be unable to be moved by the player.

FR8. If a line of Tetrominos is formed, that line will disappear and the remaining Tetrominos will move to the bottom of the board.

FR9. If a line of Tetrominos is formed, the score must be updated.

FR10. The scoring shall follow standard Tetris rules.[1]

BE3. The user wants to start the game.

FR1. The system shall provide the user with multiplayer and single player options.

FR2. The user shall be able to change the game's volume.

FR3. There must be an option to change the controls of the game.

FR4. The system shall provide an option to close the application.

FR5. The system must generate Tetrominos randomly.

FR6. The current Tetromino shall descend from the top of the board.

FR7. The descending rate must increase as the game progresses.

FR8. The next Tetromino shall be shown on screen.

FR9. The game must end if there is a dead Tetromino with a portion above the board.

BE4. The user wants to view the leader board.

FR1. The system shall provide the user with the ability to view the leader board from the home screen.

FR2. The user shall be prompted with the ability to go to the leader board after completion of a game.

FR3. The user must be able to scroll through the leader board rankings.

FR4. The user shall be able to switch between the multiplayer and single player leader boards.

BE5. The user wants to host a multiplayer game of Tetris.

FR1. The system shall provide the user with the ability to create a multiplayer lobby.

FR2. The system must provide a unique room code that another user can input to join the multiplayer lobby.

FR3. The system must allow at most 2 people in a multiplayer lobby at a time.

BE6. The user wants to join a multiplayer game of Tetris.

FR1. The system shall provide the user with the ability to join a multiplayer lobby.

FR2. The system will provide the user the ability to enter in a code to join the corresponding multiplayer lobby.

FR3. The user must only be able to join one multiplayer lobby at a time.

3 Non-functional Requirements

3.1 Look and Feel Requirements

LFR1. The games should look smooth with a consistent frame rate.
FIT CRITERION: The game should run at a consistent 30 fps on the minimum recommended hardware.

LFR2. The game should include modern design elements and styling that reflect the visual trends in modern gaming.
FIT CRITERION: A sample of 30 users that fit our game's demographic will complete a survey to indicate if they think the game looks modern.

LFR3. The text and game entities must be legible at a standard computer monitor viewing distance of 20-40 inches.[2]
FIT CRITERION: A random sample of users will be tested to ensure the legibility of the game's visual elements.

LFR4. The typeface, iconography, and colouring must remain consistent throughout the system.
FIT CRITERION: A sample of 30 users that fit our game's demographic will complete a survey to indicate if they think the game looks consistent.

3.2 Usability and Humanity Requirements

UHR1. Mechanisms will be in place to instruct the user of what to do in specific areas of the game.
FIT CRITERION: 90% of users must be able to play the main game after reading the instructions.

UHR2. System will follow the correct colour contrast ratio(s) for text and backgrounds to make the visual elements of the game legible to users.
FIT CRITERION: All colour contrast ratio's will be within the range dictated by the Web Content Accessibility Guidelines [3]

UHR3. All buttons necessary for user interaction must be easily identifiable on screen.

FIT CRITERION: Sample users must be able to locate and press selected buttons on the screen within 5 seconds.

UHR4. The language should be easily understood by users.
FIT CRITERION: Sample users with at least a fifth grade reading comprehension level can comprehend all the text in the game.

3.3 Performance Requirements

PR1. The game must be responsive to valid users inputs.
FIT CRITERION: The average response time between user input and game response should be less than 0.25 seconds.

PR2. The system should support at least 40 users concurrently.
FIT CRITERION: The system is still able to run without crashing while 40 instances of the game are running concurrently on different machines.

PR3. The system shall quickly respond to network requests with the necessary information.
FIT CRITERION: The system shall respond in less than 3 seconds 99% of the time.

PR4. The system shall have an availability of at least 99.99%.
FIT CRITERION: The system's downtime will be measured to ensure high availability.

PR5. The system shall require internet access for multiplayer functionality.
FIT CRITERION: The system will be tested on machines on separate networks.

3.4 Operational and Environmental Requirements

OER1. The game will be able to run on browsers installed on systems with an operating system of Windows 10, Mac OSX or Linux.
FIT CRITERION: The game will be run on all three operating systems without crashing.

OER2. The game will be able to run on different modern browsers with HTML5 and JavaScript enabled.

FIT CRITERION: The game will be run on a set of browsers with HTML5 and JavaScript compatibility without crashing.

3.5   Maintainability and Support Requirements

MSR1.   System updates shall not keep the system down for longer than 10 minutes.
FIT CRITERION: A timer will be used to check for adherence to this limit.

MSR2.   The software will be supported with bug fixes for a minimum of one year.
FIT CRITERION: A timer will be used to check for adherence to this limit.

3.6   Security Requirements

SR1.   The system source code shall only be modified by the developers of the system.
FIT CRITERION: All modifications of source code will be managed through Gitlab, allowing only the development team to make code changes.

SR2.   The system shall enforce an encrypted connection between users and the server.
FIT CRITERION: Connection between users and the server will be done using Hypertext Transfer Protocol Secure (HTTPS).

3.7   Cultural Requirements

CR1.   The iconography and language will be inoffensive to as many users as possible.
FIT CRITERION: The system shall not contain any hate symbols defined by the Anti Defamation League (ADL). [4]

3.8   Legal Requirements

LR1.   The system must obey applicable laws and regulations.
FIT CRITERION: The system will obey Canadian federal laws,

Ontario's provincial laws, and Hamilton's municipal by-laws.

3.9 Health and Safety Requirements

HSR1. The system must not cause the users any adverse health effects related to prolonged play or use from a game session.
FIT CRITERION: The total in game play time of any game session will not exceed 10 minutes.

HSR2. The system will not cause exposure to undo epileptic risks.
FIT CRITERION: The system will conform with the guidelines set out by WCAG. [5]

This section is not in the original Volere template, but health and safety are issues that should be considered for every engineering project.

4 Project Issues

4.1 Open Issues

Issue 1. Currently we have not decided on how to fully implement the multiplayer component and that could have a significant impact on the user experience.

4.2 Off-the-Shelf Solutions

Currently there are many options for playing Tetris online and on mobile devices. Examples such as https://tetris.com/play-tetris along with numerous knockoffs offer similar product experiences and features. Features such as, user interface and menu, multiplayer, and scoring and leaderboards. However although the products are similar, no other product we found gave the user the option to hold a Tetromino.

4.3 New Problems

No new problems.

4.4   Tasks

1. Create game menu

2. Create game interface

3. Create game model

4. Create multiplayer mode

5. Create leader board and scoring system

6. Proof of concept testing and demonstration

7. Conduct first code revision

8. Further Tasks to be posted at revision

4.5   Migration to the New Product

Not applicable.

4.6   Risks

The only current major risk is related to our lack of experience as developers with and implementing a multiplayer game feature. We will be mitigating this, by beginning to work on that aspect immediately so we have additional time to deal with unforeseen issues. Other implementation risks are considered less relevant as there are many examples of similar working products.

4.7   Costs

The development of this system will have no cost since all libraries and source code that is used is open source.

4.8   User Documentation and Training

The system will be run as a web application that can be found through an internet browser that is HTML-5 compatible. There will be no documentation for this section as we assume the user knows how to operate and navigate a web browser. There will be a section integrated within the game to teach the user how to play Tetris.

4.9 Waiting Room

Features to be considered for future releases:

1. Interchangable UI skins

2. In-game chat

3. User accounts to hold statistics

4. Audio/music

5. AI opponent

4.10 Ideas for Solutions

A possible solution for issue 1 the multiplayer component: is to use web sockets and JavaScript to facilitate the communication between the game clients.

5   Appendix

5.1   Symbolic Parameters

N/A

5.2   References

[1] "Tetris guideline." [Online]. Available: https://tetris.fandom.com/wiki/Tetris_Guideline

[2] "United states department of labor." [Online]. Available: https://www.osha.gov/SLTC/etools/computerworkstations/components{_}monitors.html

[3] [Online]. Available: https://www.w3.org/WAI/WCAG21/Understanding/contrast-minimum.html

[4] "Hate on display™ hate symbols database." [Online]. Available: https://www.adl.org/hate-symbols

[5] "Seizures:understanding guideline 2.3." [Online]. Available: https://www.w3.org/TR/UNDERSTANDING-WCAG20/seizure.html