

2010

# Tetris Project

Software Requirements Revised



*Class:* CECS 491  
*Instructor:* Dr. Alvaro Monge  
*Term:* Fall 2010  
*Date:* September 24<sup>th</sup>, 2010  
*Authors:* Damon Chastain  
Matthew Urtnowski



1 Definitions	5
1.1 Tetris	5
1.1.1 Gameplay	5
1.1.2 Classic Tetris	5
1.1.3 3D Tetris	5-6
1.2 XNA	6
1.3 C#	6
1.4 Tetris Piece	6
1.4.1 Tetrominoes	6
1.4.1.1 I Block	6
1.4.1.2 J Block	6
1.4.1.3 L Block	6
1.4.1.4 O Block	7
1.4.1.5 S Block	7
1.4.1.6 Z Block	7
1.4.1.7 T Block	7
1.5 Colors	7
1.5.1 Red	7
1.5.2 Magenta	7
1.5.3 Yellow	7
1.5.4 Cyan	7
1.5.5 Lime	7
1.5.6 Green	7
1.5.7 Gray	8
1.5.8 Orange	8
1.5.9 Blue	8
1.5.10 Purple	8
1.5.11 Black	8
1.6 Visual Studio 2008	8
1.7 HUD	8
1.8 Keyboard	8
1.9 Game Mode	8
1.10 Splash Screen	8
1.11 2D	8
1.12 3D	8
1.13 Game Authors	8
1.14 Course Instructor	8
2 Introduction	9
2.1 Preface	9
2.2 Description	9
2.3 Scope	9
2.4 Health Hazards and Dangers	9
2.4.1 Seizures	10
2.4.2 Repetitive Stress Injury	10

2.4.3 Eye Strain _____	10
2.4.4 Neglect of Real Life Responsibilities _____	10
2.5 Security _____	10
3 Game _____	11
3.1 Interface _____	11
3.1.1 Game Flow _____	11
3.1.2 Splash Screens _____	11
3.1.2.1 Metadata _____	11
3.1.2.2 Health Hazards and Dangers _____	11
3.1.2.3 Visuals _____	12
3.1.3 Main Menu _____	12
3.1.3.1 Visuals _____	12
3.1.4 Gameplay Mode Menu _____	12
3.1.4.1 Visuals _____	12
3.1.5 View High Scores Menu _____	13
3.1.5.1 Visuals _____	13
3.1.6 Game Mode High-Score List _____	13
3.1.7 Options Menu _____	13
3.2 High Score _____	13
3.3 Options _____	13
3.3.1 Music Volume Option _____	13
3.3.2 Game Volume Option _____	13
3.3.3 Key Control Remapping _____	13
3.3.4 Difficulty _____	14
3.3.5 Level _____	14
3.4 Gameplay Modes _____	14
3.4.1 3D Classic Tetris _____	14
3.4.1.1 Description _____	14
3.4.1.2 Tetris Blocks _____	14
3.4.1.3 Interface _____	14
3.4.1.3.1 Splash Screen _____	15
3.4.1.3.2 Game Menu _____	15
3.4.1.3.3 HUD or Heads Up Display _____	15
3.4.1.3.4 Game Field _____	15
3.4.1.3.5 Controls _____	16
3.4.1.3.6 Camera Angles _____	16
3.4.1.4 Gameplay _____	16
3.4.1.4.1 Clearing Lines _____	16
3.4.1.4.2 Scoring _____	17
3.4.1.4.3 Music _____	17
3.4.1.4.4 Effects _____	17
3.4.1.4.5 Levels _____	17
3.4.2 3D Tetris Challenges _____	18
3.4.2.1 Description _____	18

3.4.2.2 Tetris Blocks _____	18
3.4.2.3 Interface _____	18
3.4.2.3.1 Game Menu _____	18
3.4.2.3.2 HUD _____	18
3.4.2.3.3 Game Field _____	18
3.4.2.3.4 Controls _____	19
3.4.2.3.5 Camera Angles _____	19
3.4.2.4 Gameplay _____	19
3.4.2.4.1 Clearing Lines _____	19
3.4.2.4.2 Scoring _____	19
3.4.2.4.3 Music _____	19
3.4.2.4.4 Effects _____	19
3.4.2.4.5 Levels _____	20
3.4.3 3D Tetris Time Trial _____	20
3.4.3.1 Description _____	20
3.4.3.2 Interface _____	20
3.4.3.2.1 Game Menu _____	20
3.4.3.2.2 HUD _____	20
3.4.3.2.3 Game Field _____	20
3.4.3.2.4 Controls _____	20
3.4.3.2.5 Camera Angles _____	20
3.4.3.3 Gameplay _____	20
3.4.3.3.1 Clearing Lines _____	20
3.4.3.3.2 Scoring _____	20
3.4.3.3.3 Music _____	20
3.4.3.3.4 Effects _____	20
3.4.3.3.5 Levels _____	21
3.4.4 3D Classic Tetris Marathon _____	21
3.4.4.1 Description _____	21
3.4.4.2 Tetris Blocks _____	21
3.4.4.3 Interface _____	21
3.4.4.4 Gameplay _____	21
3.4.5 Additional Game Mode Ideas _____	21
3.4.5.1 3D Tetris with a 3D board _____	21
3.4.5.2 Dark Tetris _____	21-22
3.4.5.3 Item Tetris _____	22
3.4.5.4 Local Machine Two-Player Classic Tetris _____	22
3.4.5.5 Online Two-Player Classic Tetris _____	22
3.5 Performance _____	22
3.6 Hardware Requirements _____	22
4 Referenced Documents _____	23
4.1 Rules of Game: Game Design Fundamentals _____	23
4.2 C# Language Specification _____	23
4.3 Tetris Wikipedia Page _____	23

4.4 Tetromino Wikipedia Page _____	23
4.5 Simple Tetris Clone _____	23
4.6 XNA Tetris _____	23
4.7 .NET 1.1 Class Library _____	23
4.8 XNA Framework Class Library _____	23
4.9 XNA Game Studio 3.1 _____	23
4.10 XNA Game Studio 3D Tutorial _____	23
4.11 XNA Creators _____	23
4.12 Repetitive Stress Injury Wikipedia Page _____	23
4.13 CECS 491 Fall 2010 Website _____	23
4.14 Writing Software Requirements Specifications _____	23
4.15 How Video Games Effect Health _____	23
4.16 Official Tetris Website _____	23
4.17 Microsoft XNA Wikipedia Page _____	24
4.18 Visual Studio Wikipedia Page _____	24
4.19 Youtube Tetris Gameplay Videos _____	24
5 Duties _____	25
5.1 Damon _____	25
5.2 Matt _____	25

## 1 Definitions

### 1.1 Tetris

Tetris is a puzzle video game designed by Alexy Pajitnov and is currently the intellectual property The Tetris Company.<sup>4,3</sup>

#### 1.1.1 Gameplay

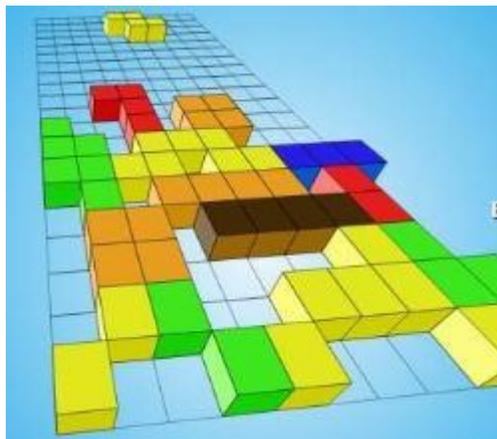
A random sequence of shapes, composed of square blocks, fall down a playing-field over time. The objective of the game is to manipulate these shapes, by moving each one sideways or rotating it by 90 degrees, to create a continuous horizontal line of blocks connecting the sides of the playing field. When a line of blocks is successfully created, that line disappears and any blocks above the deleted line will move down one line. The player receives points every time they cause a line to disappear. However, if lines are not successfully cleared they will begin to pile-up and fill the playing field. When the pieces stack, without being cleared, and reach the top of the board the game ends. The objective of the game is to receive as many points as possible before the game ends.

#### 1.1.2 Classic Tetris

The Tetris video game was released in 1989 for the Nintendo Entertainment System by the Nintendo Publisher.

#### 1.1.3 3D Tetris

A variant of the Classic Tetris which uses identical gameplay, contained in a 3D environment. Tetris blocks will still move on a 2D grid which will be contained within the 3D environment. The image below is a visual representation of how they game field may be 2D while the pieces are 3D.



This variation will include aspects which were not originally included into the original Tetris game such as hard drops, where the player accelerates the downward movement of the piece to make the piece touch the ground (or other pieces) faster.

## 1.2 XNA

XNA is a set of tools with a managed runtime environment provided by Microsoft that facilitates computer game development and management.<sup>4.17</sup>

### 1.2.1 Version

Tetris Project uses XNA version 3.1

## 1.3 C#

C# is a simple, modern, general-purpose, object-oriented programming language included in the .Net Framework.<sup>4.2</sup>

### 1.3.1 .NET Version

Tetris Project uses .NET Framework 4

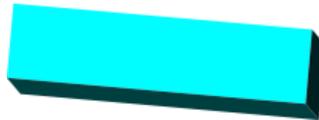
## 1.4 Tetris Piece

A shape composed of square blocks.

### 1.4.1 Tetrominoes

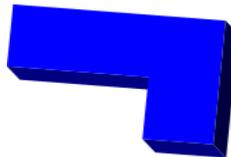
A geometric shape composed of four blocks, connected orthogonally.<sup>4.4</sup> The colors shown in the the sections below may not actually represent the colors used in game.

#### 1.4.1.1 I Block



four blocks in a straight line.<sup>4.4</sup>

#### 1.4.1.2 J Block

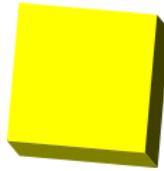


a row of three blocks with one added below the right side.<sup>4.4</sup>

#### 1.4.1.3 L Block



a row of three blocks with one added below the left side.<sup>4.4</sup>

**1.4.1.4 O Block**

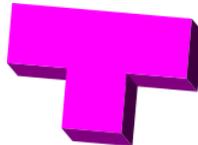
four blocks in a 2x2 square. <sup>4.4</sup>

**1.4.1.5 S Block**

two stacked horizontal dominoes with the top one offset to the right. <sup>4.4</sup>

**1.4.1.6 Z Block**

two stacked horizontal dominoes with the top one offset to the left. <sup>4.4</sup>

**1.4.1.7 T Block**

a row of three blocks with one added below the center. <sup>4.4</sup>

**1.5 Colors**

All colors are defined using the RGB color scale.

**1.5.1 Red**

RGB(255,0,0)

**1.5.2 Magenta**

RGB(255,0,255)

**1.5.3 Yellow**

RGB(255,255,0)

**1.5.4 Cyan**

RGB(0,255,255)

**1.5.5 Lime**

RGB(0,255,0)

**1.5.6 Green**

RGB(0,128,0)

- 1.5.7 Gray**  
RGB(128,128,128)
- 1.5.8 Orange**  
RGB(255,165,0)
- 1.5.9 Blue**  
RGB(0,0,255)
- 1.5.10 Purple**  
RGB(128,0,128)
- 1.5.11 Black**  
RGB(0,0,0)

## **1.6 Visual Studio 2008**

Microsoft Visual Studio is an integrated development environment (IDE) from Microsoft. It can be used to develop console and graphical user interface applications along with Windows Forms applications, web sites, web applications, and web services in both native code together with managed code for all platforms supported by Microsoft Windows, Windows Mobile, Windows CE, .NET Framework, .NET Compact Framework and Microsoft Silverlight.<sup>4.18</sup>

## **1.7 HUD**

HUD stands for 'Heads Up Display' and is a 2D interface that overlays the main game field. It is used to display 2D graphics such as text, artwork, or score.

## **1.8 Keyboard**

Any keyboards in this document refer to a 104-Key PC QWERTY style keyboard

## **1.9 Game Mode**

A variant of the Classic Tetris game

## **1.10 Splash Screen**

The initial graphic which appears when the user accesses an area of a program is defined to be the splash screen. Typically splash screens are used to display disclaimers, messages, or create a visual effect.

## **1.11 2D**

This is meant to mean "two dimensions" or "two dimensional." 2D refers to anything that is built using only an X & Y coordinate plane which excludes the Z coordinate.

## **1.12 3D**

This is meant to mean "three dimensions" or "three dimensional." 3D refers to anything that is built using the X, Y & Z coordinate plane.

## **1.13 Game Authors**

This project is authored by Damon Chastain & Mathew Urtnowski.

## **1.14 Course Instructor**

Dr. Alvaro Monge is the CECS 491 course instructor.

## 2 Introduction

### 2.1 Preface

This document acts as a Software Requirements Specification for the project. It defines the programs functionality, capabilities, performance, and limitations.

### 2.2 Description

The objective of the project is to build a working, three-dimensional, Tetris game that can be played free of bugs with alterations to the classic game to make it more user-friendly, fun, and challenging. The game is meant to be open source and will be created under GNU General Public License v3 guidelines. The game will be playable as a single player game and meant to act as an enhanced version of the classic. Players will be able to enjoy various game modes allowing them to play the classic Tetris or new modes that put a new “spin” on the classic game. The game play changes for the new modes may include items that help or harm the player, importable music selection, different block shapes, a selection of skins, a marathon mode and time-trial mode.

### 2.3 Scope

The program will be written in the C# programming language as a stand-alone application. C# is a language that has gained popularity and is intended to be a simple, modern, general-purpose, object-oriented programming language. The program will also make use of new technologies such as the Microsoft XNA Framework. XNA is a toolset that was developed to provide an environment for game developers to focus primarily on “high-level” game development and less on “low-level” process development. The interface will use both 2D and 3D aspects of the XNA framework. The actual gameplay will exist on a 2D-grid that is placed inside of a 3D “world”. The pieces will be made in 3D and will be played within the 2D-grid. The 2D grid is only used as a visible guideline for the player to use when measuring where each piece will go. The user interface display, including menus, buttons, score, and other notifications will be in 2D. This overlaying 2D user interface will act as a type of lens that the user is constantly looking through as the user plays the game. The 2D interface is not meant to hinder the users experience by blocking sight of the 3D “world” in any way. The design, layout, audio, graphic textures, and flow of the interface will all have to be developed.

### 2.4 Health Hazards and Dangers

While some people may not consider video games dangerous or trivialize their risks, it is important to be a responsible developer by considering them and inform the user of any risks. Although we will not take any specific actions to protect the user from these risks, we will inform them briefly of the risks in the opening splash screens of the game. It is ultimately up to the user to use the program safely and responsibly.

**2.4.1 Seizures**

A seizure is a serious medical ailment can results in unusual and potentially dangerous brain activity. Some people are at risk for epileptic seizures when they see rapid flickering or rapid screen changes happening in the game. The risk of seizures is not dependent on the length of gameplay. Seizures can occur after even a few minutes of play.<sup>4.15</sup>

**2.4.2 Repetitive Stress Injury**

Using a keyboard, mouse, or game controller can cause stress injuries if used chronically or used for extended periods of time.<sup>4.12</sup>

**2.4.3 Eye Strain**

Sitting in front of a computer screen can cause eye strain on the user. It is recommended the user take a five minute break every hour of play or if they experience eye strain, headaches, or blurry vision. If the user is experiencing eye strain, headaches, or blurry vision they should stop playing until these symptoms stop.<sup>4.15</sup>

**2.4.4 Neglect of Real Life Responsibilities**

Playing video games can cause people to neglect their real life responsibilities. It is important to remember to play responsibly.<sup>4.15</sup>

**2.5 Security**

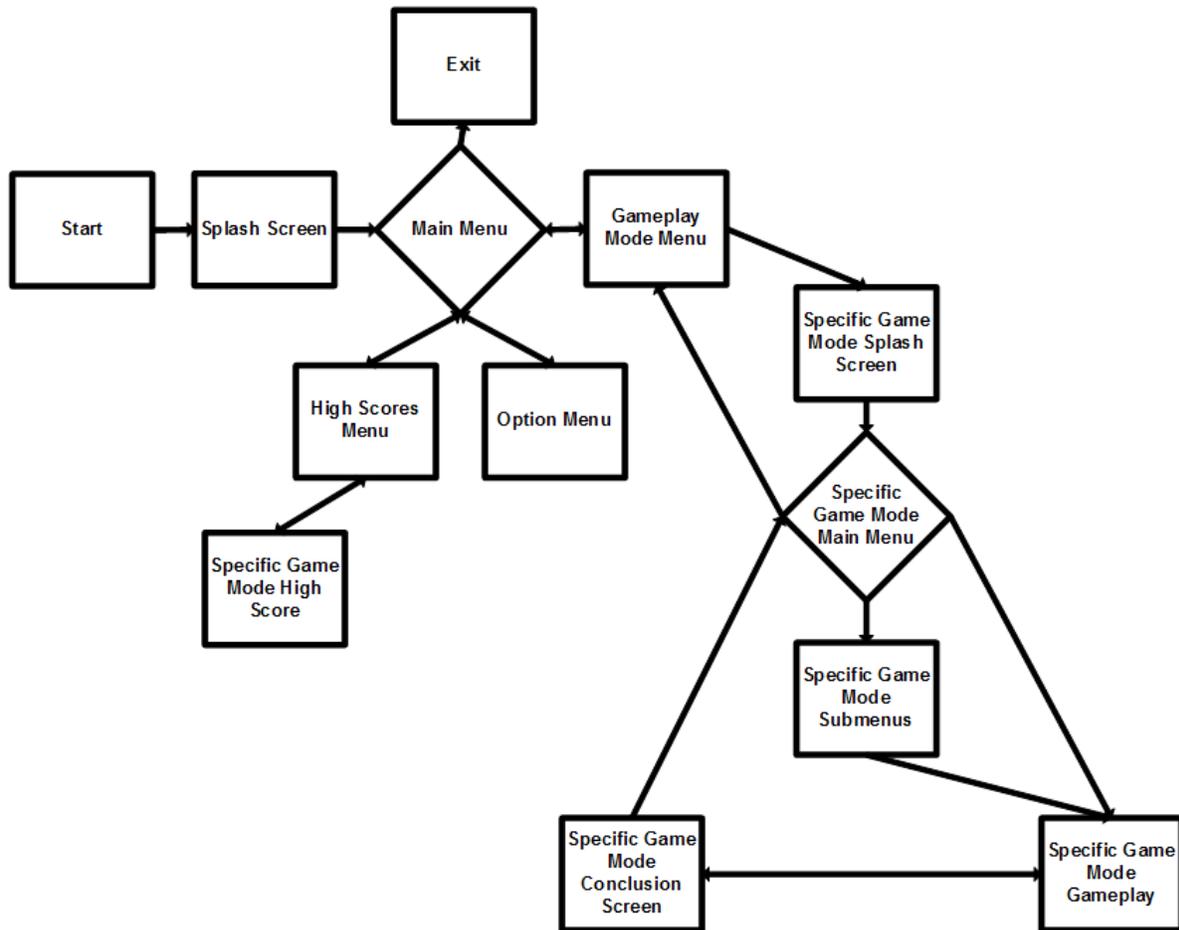
There are no security or privacy requirements for this project.

### 3 Game

#### 3.1 Interface

##### 3.1.1 Game Flow

A finite state machine showing the general flow of how the program is navigated and progresses is as follows:



##### 3.1.2 Splash Screens

When the program initially boots up the user is brought to a series of splash screens. These splash screens are used to inform the player of the metadata about the game, defined in section 3.1.2.1. The splash screen also informs the user of any health hazards or dangers about playing the game.

###### 3.1.2.1 Metadata

This metadata includes the game authors, CSULB name, CECS 491 course title, CECS 491 course instructor, approximate development date, and the intended purpose and appropriate uses of the game

###### 3.1.2.2 Health Hazards and Dangers

Health hazards and dangers are defined in section 2.4 of this document.

### 3.1.2.3 Visuals

The splash screens for the displays may or may not be a single screen. An appropriate number of splash screens will be used to display all information described in section 3.1.2.1. An appropriate number is the number needed to create a pleasing visual effect while conveying the information. The user may not skip any of these splash screens by pushing a button. Each splash screen will be rendered long enough so that the information they show can be read comfortably. After this time the next splash screen will be shown. When all of these splash screens have been displayed the game will then show the Main Menu defined in section 3.1.3.

### 3.1.3 Main Menu

The Main Menu provides the user with the ability to navigate to the Game Play Mode Menu, High Scores Menu, Option Menu, or exit the program.

#### 3.1.3.1 Visuals

The Main Menu should list all of its navigation choices defined in section 3.1.3. These navigation choices should be displayed as text and readily navigated via use of the keyboard. The user should be able to highlight a particular navigation choice, by the use of arrows or a joystick, and then navigate to it by pressing a button. The button to initiate the navigation should be convenient and intuitive to the user. The user should be able to transition between the navigation choices using a convenient and intuitive button. The navigation choices should overlay a visually appealing graphical background.

### 3.1.4 Gameplay Mode Menu

The Gameplay Mode Menu provides the user with the ability to navigate to all of the gameplay modes choices implemented in the program. The user should also be able to navigate back to the Main Menu by pressing a button that is convenient and intuitive to the user.

#### 3.1.4.1 Visuals

The Gameplay Mode Menu should list all of the game mode choices in program. These choices should be displayed textually or with icons. The user should be able to select a particular choice and then navigate to the splash screen for that game mode with the push of a button. The button to initiate the navigation should be convenient and intuitive to the user. When a gameplay mode is highlighted a description of that gameplay mode should appear. The navigation choices and gameplay descriptions should overlay a visually appealing background.

### **3.1.5 View High Scores Menu**

The High Score Menu provides the user with the ability to navigate to all of the high-score lists for each game mode. The user should be able to navigate back to the Main Menu by pressing a button that is convenient and intuitive to the user.

#### **3.1.5.1 Visuals**

The View High Scores Menu should list the titles of all of the game mode high-score lists in the program. These titles should be displayed as text. The user should be able to highlight a particular list and then navigate to it by pushing a button. The button to initiate the navigation should be convenient and intuitive to the user. The navigation choices will overlay a visually appealing background.

### **3.1.6 Game Mode High-Score List**

The game mode high-score list shows the high-score list for a single gameplay mode. The list displays the current top 10 scores. The user should be able to navigate back to the Main Menu by pressing a button that is convenient and intuitive to the user.

### **3.1.7 Options Menu**

The options menu provides the user with a set of options for altering gameplay. The changeable gameplay options will include difficulty, level, and an option to turn off the music or sound effects.

## **3.2 High Score**

At the end of each game, the final score is displayed along with the top 10 recorded high scores for that game mode. If the final score is greater than any of the top 10 high scores then the final score is added to the top ten, and the lowest score on the list gets dropped while the others are altered to be in numerical order. If two scores are tied the score that occurred first is superior.

## **3.3 Options**

A set of changeable game features within the Options Menu. Options allow the user to alter various aspects of the game in an attempt to suit the needs of all users.

### **3.3.1 Music Volume Option**

The user may adjust the music volume level on a scale of 0 to 10; where 0 is no volume and 10 is full volume.

### **3.3.2 Game Volume Option**

The user may adjust the volume level of any audio features or effects on a scale of 0 to 10; where 0 is no volume and 10 is full volume.

### **3.3.3 Key Control Remapping**

As an optional requirement the user may be able to redefine the buttons used to control the game.

### 3.3.4 Difficulty

Difficulty refers to the speed at which the pieces are moving toward the bottom of the board. If the difficulty is small, numerically, then the speed of the falling pieces is also small. The difficulty will range from 0 to 29 and is subject to change at the whim of the programmers in order to maximize the enjoyment level for the users.

### 3.3.5 Level

The level refers to the difficulty that the user is currently playing on. In essence, the level is simply a display for the difficulty. For instance, if the level is 0 then the difficulty is 0. However, each level may speed up the music, change the colors on screen, or more but the difficulty remains one-to-one with the level.

## 3.4 Gameplay Modes

The following 3.4.\* sections are pre-defined variations on how the game is going to be played. The gameplay modes will be similar to that of the classic Tetris game, but may offer their own “spin” on the game. The only required game modes are 3D Classic Tetris, 3D Tetris Challenges, 3D Tetris Time Trial, and 3D Tetris Marathon defined in sections 3.4.1, 3.4.2, 3.4.3, and 3.4.4. As an optional requirement if time permits, 3D-Tetris plans to be extended to new, exciting, gameplay modes which have never been done before.

### 3.4.1 3D Classic Tetris

#### 3.4.1.1 Description

The classic Tetris game built in a 3D environment. Every viewable Tetris piece is drawn in 3D. The gameplay and game rules are intended to be the same as Classic Tetris. For a brief description of how Classic Tetris is played refer to 1.1.1.

#### 3.4.1.2 Tetris Blocks

All Tetris blocks are built in different color variations and positioned into pieces as shown in 1.4. This game mode uses the Tetris Blocks I, J, L, O, S, Z, and T as defined in section 1.4.1. The I block is Cyan, the J block is blue, the L block is orange, the O block is grey, the S block is lime, the T block is purple, and the Z block is red. All colors are defined in section 1.5.

#### 3.4.1.3 Interface

The display method used to give feedback to the user during gameplay. It is supposed to have a similar feel and design to the Classic Tetris game to create a retro video game feel.

**3.4.1.3.1 Splash Screen**

The Splash Screen shows the metadata about this game mode. The metadata contains the title of the game mode, programmer's names, and a visually appealing graphic. The splash screen cannot be skipped by the user and will be displayed long enough for all text on it to be read comfortably. After the Splash Screen the game should progress to the Game Menu defined in section 3.4.1.3.2.

**3.4.1.3.2 Game Menu**

The Game Menu allows the users to navigate to the features of this game mode. 3D Tetris has the options to start a new game or go back to the game mode selection menu.

**3.4.1.3.3 HUD or Heads Up Display**

By design, the interface will be divided into two parts: the game field and the 2D heads up display that displays in-game information. The in-game information for this game mode includes current score, current level, number of lines cleared, and an image of the next piece the user will control.

**3.4.1.3.4 Game Field**

This is where the 3D environment is displayed and the actual gameplay is drawn onto the screen. The pieces will interact with other pieces and the board in this field. The interaction between pieces involves one piece colliding with another piece and resting directly above, or next to the piece, due to the collision. When a piece is resting it is no longer moving. The game field for this mode is a 2D Grid is 10 wide by 20. The size for the 2D grid is dynamically allocated within the game depending on the position of the camera. In this instance of the game, all pieces are built to be one unit tall, 1 unit wide and 1 unit deep. These units, like the size of the grid, are scaled depending on their distance from the camera. The closer they are to the camera, the larger they become. These units are built in the game and have no real comparison to normal units of measurement such as inches or feet.

#### **3.4.1.3.5 Controls**

The controls for this instance of Tetris are identical to that of the classic Tetris game. To move a piece left we will use the left arrow. To scroll down in the menu we will use the down arrow. The user will have convenient and intuitive keys to move the pieces left and right. The user will also be able to rotate the pieces 90 degrees clockwise, as is the direction of the Classic Tetris that this imitates. The user will have the option of making the piece fall faster toward the bottom of the board. Though it's not in Classic Tetris, the user will also be able to "slam" the piece to the bottom of the board which instantly drops the piece down as far as it can go until it hits either another piece or the bottom of the board. The user cannot move a piece through another block or off the game field. The user may also pause the game.

#### **3.4.1.3.6 Camera Angles**

The user will be able to move his view point of the play field by using an intuitive button during gameplay. By changing his view he will be able to change the angle at which he can see the game field and blocks. The user will be unable to change the camera view while the game is paused or within a menu. The number of camera angles may range from a select few static angles to an infinite number of free "floating" angles depending on what the programmers feel suits the game best.

### **3.4.1.4 Gameplay**

#### **3.4.1.4.1 Clearing Lines**

When a user places enough pieces to completely fill a row, horizontally, on the board a line is cleared. Once this line is completed the squares in that given row are removed and the user's score is increased by a set amount. Any blocks directly above the cleared line get moved down to take the place of the newly cleared line.

**3.4.1.4.2 Scoring**

Score is a numerical value which gives visual feedback to the user pertaining to the quality of the user's performance. With each cleared line, points are added to the user's score as a reward. Upon clearing a line, 100 points are added to the player's score. If the user clears five lines at once, called a Tetris, the users is rewarded with 700 bonus points on top of the 500 points for clearing five lines.

**3.4.1.4.3 Music**

The audio selected to enhance the quality and replay value of the game. Music plays a pivotal role in keeping the user's focus as well as giving the user audio feedback during gameplay. The music will be similar to the Classic Tetris in an attempt to add a retro feel to the game.

**3.4.1.4.4 Effects**

Any graphical alterations which provide added visual feedback is an effect. Changing the color of the board and the score could be an effect to indicate that the user has entered a new level of difficulty. When a line is cleared a visual effect will be implemented where the blocks vanish.

**3.4.1.4.5 Levels**

The speed at which the game is played is increased by increasing positive integers which are defined as levels. The levels for this game mode range from 0 to 29. Level 0 has the blocks move very slowly and should be easy for the user to play, while Level 29 is very fast and challenging. Every time the user clears 10 lines, the user will progress to the next level. The speed of the pieces from levels 0-29 are going to closely imitate that of Classic Tetris however, the given speed is not yet known but will be determined as the game is developed.

## 3.4.2 3D Tetris Challenges

### 3.4.2.1 Description

The challenges are designed to be variations of the standard classic experience. They are designed to challenge the user to accomplish specific design-driven tasks. This could range to any number of possibilities and has no limitations. An example of a single Tetris challenge would include a board that already has Tetris blocks arranged in a challenging fashion on the game field and challenges that change the distribution of the Tetris pieces. The distribution is originally randomized, however, for some challenges we will limit, or remove, the number of certain pieces altogether. As of now, the challenges are all still in the planning phase and have no set boundaries or guidelines.

### 3.4.2.2 Tetris Blocks

The blocks used in the various challenges are intended to take the exact shape of the previous blocks stated in 1.4.1, which are; blocks I, J,L,O,T,S, and Z.

### 3.4.2.3 Interface

The display method used to give feedback to the user during gameplay.

#### 3.4.2.3.1 Game Menu

This menu will act exactly as that of the 3D Tetris game menu, as stated in 3.4.1.3.2, but with restrictions based on each given challenge. For the challenges provided, the restrictions are minimal such as; not allowing an 'I' block to be produced. Each challenge will have the given restrictions placed on the menu and/or the splash screen.

#### 3.4.2.3.2 HUD

The HUD will act just as the HUD in 3.4.1.3.3 but with added information, or less information, in the 2D-display depending on each given challenge. For instance, a given challenge may simply include a countdown timer and not include the ability for the user to see which piece comes next.

#### 3.4.2.3.3 Game field

The game field for the challenges will remain exactly as previously specified in 3.4.1.3.4.

**3.4.2.3.4 Controls**

Refer to 3.4.1.3.5 unless a challenge specifically changes the controls to add a challenge. One challenge might be that the controls are reversed or scrambled altogether. Forcing a complete random switch in the control assignments would make the game exceedingly difficult because the player may be trying to rotate the piece but end up dropping the piece to the bottom of the board.

**3.4.2.3.5 Camera Angles**

Refer to 3.4.1.3.6 unless a challenge specifically restricts the camera angles to create a greater challenge. In which case, the camera may rotate inconveniently in an attempt to make the user lose the game. The user will have no control over the camera if the challenge is based upon the camera being a hindrance to the player.

**3.4.2.4 Gameplay****3.4.2.4.1 Clearing Lines**

Refer to 3.4.1.4.1.

**3.4.2.4.2 Scoring**

Scoring challenges may be non-existent since many challenges are simply pass or fail and not dependent on a score. For pass or fail challenges the user will either be successful or unsuccessful. If scoring is necessary for any given challenge, the scoring system will be determined based on each individual challenge. For instance, if a challenge requires that you clear a certain number of lines within a given time-limit then the score will be based on an integer number of lines cleared within that time frame.

**3.4.2.4.3 Music**

Refer to 3.4.1.4.3, however the soundtrack for certain challenges may differ from that of the 3D Classic Tetris game depending on the specific challenge selected.

**3.4.2.4.4 Effects**

Refer to 3.4.1.4.4.

#### 3.4.2.4.5 Levels

Levels in challenges will typically be set and unchangeable to give the player a programmer-defined goal that will not be easy to achieve. Refer to 3.4.1.4.5.

### 3.4.3 3D Tetris Time Trial

#### 3.4.3.1 Description

The 3D Tetris Time Trial adds a new challenge the 3D Classic Tetris and 3D Tetris Challenges modes. The 3D Tetris Time Trial takes elements of the 3D Classic Tetris and 3D Tetris Challenges, but instead of scoring them by points the game is scored by time. The faster the user can complete a particular challenge or faster they can clear a certain number of lines the higher their score.

Refer to 4.15.

#### 3.4.3.2 Interface

##### 3.4.3.2.1 Game Menu

The game menu has the same navigation functionality as 3D Classic Tetris and 3D Tetris Challenge.

##### 3.4.3.2.2 HUD

See section 3.4.2.3.2

##### 3.4.3.2.3 Game Field

See section 3.4.1.3.4

##### 3.4.3.2.4 Controls

Refer to 3.4.1.3.5

##### 3.4.3.2.5 Camera Angles

Refer to section 3.4.1.3.6

#### 3.4.3.3 Gameplay

##### 3.4.3.3.1 Clearing Lines

Refer to 3.4.1.4.1.

##### 3.4.3.3.2 Scoring

Instead of receiving points for clearing lines, the user receives a timestamp based on how fast they complete the level or challenge predefined for them. In the event of a tie between two player's score, the first player to achieve that score is superior.

##### 3.4.3.3.3 Music

The music will be similar to the Classic Tetris music.

##### 3.4.3.3.4 Effects

Refer to 3.4.1.4.4.

#### **3.4.3.3.5 Levels**

Each level in this mode will present a new challenge to the user. The user may be asked to clear a certain number of lines in a certain amount of time. The user may also be presented with a game field with blocks already on it and then be asked to clear all of the given blocks as fast as possible. The user will not be responsible for clearing blocks that appear after the level starts. The levels are selected from the menu by the user.

### **3.4.4 3D Classic Tetris Marathon**

#### **3.4.4.1 Description**

This game mode is identical to 3D Classic Tetris, but instead of restricting the game to 29 levels the user may play up to 2,000,000 levels.

#### **3.4.4.2 Tetris Blocks**

Refer to 3.4.1.2

#### **3.4.4.3 Interface**

Refer to 3.4.1.3

#### **3.4.4.4 Gameplay**

Refer to 3.4.1.4

### **3.4.5 Additional Game Mode Ideas**

If time permits additional gameplay modes may be implemented. Descriptions of each will be brief due to the high probability that they may not happen.

#### **3.4.5.1 3D Tetris With A 3D Board**

A game mode very similar to the Classic Tetris in 3.4.1, but the Tetris pieces move downwards onto a 3D plain instead of a single row. For example, the 2D board consisting of 10x20 may be expanded to 10x20x5. This enables the player to move forward and back, in depth, and create different kinds of strategies. The specifics of this mode are undefined at this point in development.

#### **3.4.5.2 Dark Tetris**

A game mode almost exactly like Classic Tetris in 3.4.1, but we add a challenge to the game by engulfing the board in a fog-like darkness. Without proper lighting, the board cannot be seen by the player. We then embedded light sources inside random blocks and illuminate a localized area around the block. XNA defines this type of lighting as ambient lighting which could be described as a "light bulb" effect. This would add an interesting challenge and visual effect considering that a

player may be ready to clear a given line but will choose not to clear it considering that the only light source will be cleared as well.

#### **3.4.5.3 Item Tetris**

We embed items into random blocks. When a line is cleared, which contains any given item, that item is activated. Activated items can have a variety of temporary effects on the game. It may speed up the game, slow down the game, change the next block to appear, manipulate the camera angle, manipulate the game control, change how the game is scored, change the nature of other items in the game field, and other interesting events. Items can either be helpful or harmful to the player.

#### **3.4.5.4 Local Machine Two-Player Classic Tetris**

Two users play Classic Tetris each controlling their own game field displayed, in a split-screen fashion, on the monitor. The users compete to get the highest score.

#### **3.4.5.5 Online Two-Player Classic Tetris**

Similar to 3.4.5.4, but instead of played split-screen user are able to play on separate machines connected through the internet. The user may or may not need to split the screen depending on if the programmers decide the player would like to see the other player's board.

### **3.5 Performance**

The game should run free of any stuttering, lagging, or skipping when played on a computer with a CPU of at least a dual core 2Ghz processor and 2Gigabytes of RAM. Gameplay should appear seamless and not distract the user with bugs or glitches.

### **3.6 Hardware Requirements**

Aside from the 2Ghz processor and 2 gigabytes of RAM on the machine running the program there are no other hardware requirements.

## 4 Referenced Documents

The following documents and webpages were used as research material for this project

### 4.1 Rules of Game: Game Design Fundamentals

By Katie Salen, Eric Zimmerman

<http://books.google.com/books?id=UM-xyczrZuQC&printsec=frontcover>

### 4.2 C# Language Specification

By ECMA International

<http://www.ecma-international.org/publications/files/ECMA-ST/Ecma-334.pdf>

### 4.3 Tetris Wikipedia Page

By Wikipedia Community

<http://en.wikipedia.org/wiki/Tetris>

### 4.4 Tetromino Wikipedia Page

By Wikipedia Community

<http://en.wikipedia.org/wiki/Tetromino>

### 4.5 Simple Tetris Clone

<http://code.google.com/p/simple-tetris-clone/>

### 4.6 XNA Tetris

<http://code.google.com/p/xnatetris/>

### 4.7 .NET 1.1 Class Library

[http://msdn.microsoft.com/en-us/library/d11h6832\(v=VS.71\).aspx](http://msdn.microsoft.com/en-us/library/d11h6832(v=VS.71).aspx)

### 4.8 XNA Framework Class Library

<http://msdn.microsoft.com/en-us/library/bb203940.aspx>

### 4.9 XNA Game Studio 3.1

[http://msdn.microsoft.com/en-us/library/bb200104\(v=XNAGameStudio.31\).aspx](http://msdn.microsoft.com/en-us/library/bb200104(v=XNAGameStudio.31).aspx)

### 4.10 XNA Game Studio 3D Tutorial

[http://msdn.microsoft.com/en-us/library/bb203897\(v=XNAGameStudio.31\).aspx](http://msdn.microsoft.com/en-us/library/bb203897(v=XNAGameStudio.31).aspx)

### 4.11 XNA Creators

<http://creators.xna.com/en-US/>

### 4.12 Repetitive Stress Injury Wikipedia Page

[http://en.wikipedia.org/wiki/Repetitive\\_stress\\_syndrome](http://en.wikipedia.org/wiki/Repetitive_stress_syndrome)

### 4.13 CECS 491 Fall 2010 Website

<http://www.cecs.csulb.edu/~monge/classes/491/2010Fall/>

### 4.14 Writing Software Requirements Specifications

<http://www.techwr-l.com/techwhirl/magazine/writing/softwarerequirementspecs.html>

### 4.15 How Video Games Effect Health

<http://www.gameinnovator.com/health.php>

### 4.16 Official Tetris Website

<http://www.tetris.com/>

**4.17 Microsoft XNA Wikipedia Page**

[http://en.wikipedia.org/wiki/Microsoft\\_XNA](http://en.wikipedia.org/wiki/Microsoft_XNA).

**4.18 Visual Studio Wikipedia Page**

[http://en.wikipedia.org/wiki/Microsoft\\_Visual\\_Studio](http://en.wikipedia.org/wiki/Microsoft_Visual_Studio).

**4.19 YouTube Tetris Gameplay Videos**

[http://www.youtube.com/results?search\\_query=tetris+gameplay&aq=f](http://www.youtube.com/results?search_query=tetris+gameplay&aq=f)

**5 Duties**

Each member of the team will bring a specific set of skills to the project which, when properly allocated, will greatly enhance the final product. In response to this, a list of individual duties based on areas of expertise will only benefit the project. However, due to the size of the team, and the nature of the project, it is understood that the workload needs to be evenly shared despite most strengths and weaknesses. With that, most of the common jobs will not be noted here and this will only emphasize the variations from the normal workload based on strengths and weaknesses.

**5.1 Damon**

Damon will be responsible for leading the overall design of the game. He is responsible for setting the correct “foundation classes” for the game which are essentially the classes which will be reused consistently in creating Tetris. Such foundation classes include the creation of a single cube which will be fundamentally important and consistently reused in the program. The overall design of the game will follow a step-by-step checklist that Damon will create and jobs will be allocated from there. Specific tasks will be allocated on a week-by-week basis pending progress, knowledge, and allowance of time. Additionally, the music and textures for the game will be, primarily, selected or created by Damon. His knowledge of XNA, Photoshop, and recording devices will all be put to use.

**5.2 Matt**

Matt will be responsible for the majority of the technical writing and documentation for the project. His primary focus in development will be on 2D sprites and animations that typically make up the user interface. He will also focus on development of the menus the user navigates while using the program. Matt will create a series of XNA Sprite tools to help development. He will also be in charge of designing the system to test and verify the program requirements. His weakness lie in implementing 3D graphics in the XNA environment, but is strong in understanding XNA game flow and preserving game data for reuse after the program closes.