DESIGN AND IMPLEMENTATION OF TETRIS GAME

BY

**OJUKWU,** Kamachi David BU/18B/IT/3171

DEPARTMENT OF COMPUTER SCIENCE BAZE UNIVERSITY

ABUJA

May, 2021

Design and Implementation of Tetris Game

Thesis Submitted in Partial Fulfilment of the Requirement for the Degree of

B.Sc.

In

Computer Science

By

**OJUKWU,** Kamachi David BU/18B/IT/3171

To

The Department of Computer Science Baze University, Abuja

May, 2021

## DECLARATION

This is to certify that this Thesis entitled DESIGN AND IMPLEMENTATION OF TETRIS GAME which is submitted by me, Kamachi Ojukwu in partial fulfilment of the requirement for the award of degree for B.Sc. in Computer Science to the Department of Computer Science, Baze University Abuja, Nigeria, comprises of only my original work and due acknowledgement has been made in the text to all other materials used.

Date: May, 2021 Name of Student: Kamachi Ojukwu

**APPROVED BY** …………………

**HOD**

Dept. of Computer Science

## CERTIFICATION

This is to certify that this Thesis entitled Design and Implementation of Tetris Game which is submitted by me, Kamachi Ojukwu in partial fulfilment of the requirement for the award of degree for B.Sc. in Computer Science to the Department of Computer Science, Baze University Abuja, Nigeria is a record of the candidate’s own work carried out by the candidate under my/our supervision. The matter embodied in this thesis is original and has not been submitted for the award of any other degree.

Date: Supervisor: Dr. Morolake Lawrence

## APPROVAL

This is to certify that the research work, Design and Implementation of Tetris game and the subsequent preparation by Kamachi Ojukwu with Student ID, BU/18B/IT/3171 has been approved by the Department of Computer Science, Faculty of Computing and Applied Science, Baze University, Abuja, Nigeria.

By

|  |  |  |
| --- | --- | --- |
| Dr. Morolake Lawrence  1st Supervisor |  | Date |
| Ms Ruqayya Muhammad  2nd Supervisor |  | Date |
| Dr. C. V Uppin | Date |  |
| Head of Department |  |  |
| Prof M. B. Hammawa | Date |  |
| Dean, Faculty of Computing and Applied Science |  |  |
| Prof. A. B. Garko | Date |  |
| External Examiner |  |  |

## DEDICATION

This project is dedicated to God Almighty, my creator, my strength, source of inspiration, wisdom, knowledge, and understanding. Throughout this program, He has been my source of strength, and I have only been able to hold him for direction even in the hardest of times to lead the way. I dedicate my project to my mother and father for doing everything to make sure my stay in school and University life was worthwhile, providing for my resources and giving me enough to push forward. I also dedicate this work to my girlfriend Erika for giving me inspiration and working by me and with me through thick and thin. I dedicate this work to all my friends; God bless you all. I appreciate my Supervisors, Mrs Morolake Lawrence and Ms Ruqayya Muhammad for helping and seeing this project through, guiding, watching and correcting I really appreciate the Unity and Indie game development community for providing help as much, as fast and efficient as they could. God bless and Thank you to everyone mentioned.

## ABSTRACT

Tetris is a very popular game that has served as an inspiration for many puzzle-style, tile- matching video games we have. It is a classic. The fan base is growing every year because there are many variations of the game each with its own adjustments and different ways of gameplay. Each of these variations have concepts such as, speed increase, special effects, music and theme differences, colours, world environments, game boards and number of tetrominoes/shapes possible to clear rows, new bonus awards other than getting a Tetris in the game, for example DodecaTetris and OctoTetris.

Recently, the Tetris industry provided another variation called Tetris effect that heavily focuses on effects and somewhat time manipulation which has helped the Tetris community experience another addition of the game.

This project seeks to provide a new variation of the Tetris game, a harder and more compelling version. Previous variations kept the hold and next feature which just makes the game too easy, especially for the Experts. This Tetris project will remove them and may only make them available by earning scores and levels for a temporary time. This project will also include an augmented reality integration to give its final edge.

The current literature on this Tetris game variation is examined in order to clearly describe what has been done, what has worked, and the challenges that have arisen in this context. To achieve the intended goal, a methodology and design approach were established based on the literature research.

## TABLE OF CONTENTS

[**ABSTRACT**](#_bookmark0)[**VII**](#_bookmark0)

[**LIST OF TABLES**](#_bookmark1) **IX**

[**LIST OF FIGURES… X**](#_bookmark2)

[**LIST OF ABBREVIATIONS… XI**](#_bookmark3)

[**LIST OF TECHNICAL TERMS… XI**](#_bookmark3)**I**

**CHAPTER 1: INTRODUCTION** ERROR! BOOKMARK NOT DEFINED.

* 1. [OVERVIEW 1](#_bookmark4)
  2. [BACKGROUND AND MOTIVATION 1](#_bookmark5)
  3. [STATEMENT OF THE PROBLEM 2](#_bookmark6)
  4. [AIM AND OBJECTIVES 2](#_bookmark7)
  5. [SIGNIFICANCE OF THE PROJECT 3](#_bookmark8)
  6. [PROJECT RISKS ASSESSMENT 4](#_bookmark9)
  7. [SCOPE/PROJECT ORGANIZATION 5](#_bookmark10)

[CHAPTER 2: LITERATURE REVIEW](#_bookmark11) [6](#_bookmark11)

* 1. [INTRODUCTION 6](#_bookmark12)
  2. [HISTORICAL OVERVIEW 6](#_bookmark13)
  3. [RELATED WORK 7](#_bookmark14)
  4. [SUMMARY 7](#_bookmark15)

[CHAPTER 3: REQUIREMENTS, ANALYSIS, AND DESIGN](#_bookmark16) [9](#_bookmark16)

* 1. [OVERVIEW 9](#_bookmark17)
  2. [METHODOLOGY 9](#_TOC_250000)
     1. [Agile METHODOLOGY 9](#_bookmark18)
     2. [Waterfall METHODOLOGY 11](#_bookmark19)
     3. [COMPARISON OF METHODOLOGIES FOR GAME DEVELOMENT 12](#_bookmark20)
  3. [TOOLS AND TECHNIQUES 14](#_bookmark21)
  4. [ETHICAL CONSIDERATION 15](#_bookmark22)
  5. [REQUIREMENT ANALYSIS 15](#_bookmark23)
  6. [REQUIREMENTS SPECIFICATIONS 15](#_bookmark24)
     1. [Functional Requirement Specifications](#_bookmark25) [16](#_bookmark25)
     2. [Non-Functional Requirement Specifications](#_bookmark26) [16](#_bookmark26)
  7. [SYSTEM DESIGN 17](#_bookmark27)
     1. [Application Architecture](#_bookmark28) [18](#_bookmark28)
     2. [Use Case 119](#_bookmark29)
     3. *Activity Diagram* ***Error! Bookmark not defined.***
     4. [*Class Diagrams*](#_bookmark30)[*20*](#_bookmark30)
     5. [*User Interface*](#_bookmark31)[*21*](#_bookmark31)
  8. [*Summary*](#_bookmark32)[*24*](#_bookmark32)

[**CHAPTER 4: IMPLEMENTATION AND TESTING**](#_bookmark33)[**25**](#_bookmark33)

* 1. [OVERVIEW 25](#_bookmark34)
  2. [MAIN FEATURES 25](#_bookmark35)
  3. [IMPLEMENTATION PROBLEMS 25](#_bookmark36)
  4. [OVERCOMING IMPLEMENTATION PROBLEMS 27](#_bookmark37)
  5. [TESTING 28](#_bookmark38)
     1. [*Tests Plans (for Unit Testing, Integration Testing, and System Testing)*](#_bookmark39)[*30*](#_bookmark39)
     2. [*Test Suite (for Unit Testing, Integration Testing, and System Testing)*](#_bookmark40)[*30*](#_bookmark40)
     3. [*Test Traceability Matrix (for Unit Testing, Integration Testing, and System Testing)*](#_bookmark41)

[*31*](#_bookmark41)

* + 1. [*Test Report Summary (for Unit Testing, Integration Testing, and System Testing). 31*](#_bookmark42)
    2. [*Error Reports and Corrections*](#_bookmark43)[*31*](#_bookmark43)
  1. [USE GUIDE 33](#_bookmark44)
  2. [SUMMARY 33](#_bookmark45)

[**CHAPTER 5: DISCUSSION, CONCLUSION, AND RECOMMENDATIONS**](#_bookmark46)[**34**](#_bookmark46)

* 1. [OVERVIEW 34](#_bookmark47)
  2. [OBJECTIVE ASSESSMENT 34](#_bookmark48)
  3. [LIMITATIONS AND CHALLENGES 34](#_bookmark49)
  4. [FUTURE ENHANCEMENTS 35](#_bookmark50)
  5. [RECOMMENDATIONS 36](#_bookmark51)
  6. [SUMMARY 36](#_bookmark52)

[**REFERENCES**](#_bookmark53)[**37**](#_bookmark53)

[**APPENDICES**](#_bookmark54)[**38**](#_bookmark54)

## LIST OF TABLES

Table 1.1 Project Risks Assessment… 4

Table 3.1 Advantages and Disadvantages of the Agile Methodology 10

Table 3.2 Advantages and Disadvantages of the Waterfall Methodology 12

Table 3.3 Comparison of Agile Methodology and Waterfall Methodology 13

Table 3.4 Functional Requirement Specifications 16

Table 3.5 Non-Functional Requirement Specifications 17

Table 4.1 Test Traceability Matrix 31

Table 4.2 Test Report Summary 32

Table D1 Test case TC-001(UI Interaction) 48

Table D2 Test case TC-002(Core Game Mechanics) 50

**LIST OF FIGURES**

Figure 3.1 Agile Methodology… 10

Figure 3.2 Waterfall Methodology… 12

Figure 3.3 Application Architecture of the Tetris Game 18

Figure 3.4 Use Case Diagram of the Tetris Game 19

Figure 3.5 Activity Diagram of the Tetris Game 20

Figure 3.6 Class Diagram of the Tetris Game 21

Figure 3.7 User Interface of the Tetris Game running in Unity Engine 22

Figure 3.8 User Interface of the Tetris Game running in final Build for Windows 23

Figure 3.9 The AR view of the game environment in an office sample Simulation View…23 Figure 3.10 AR view of the game environment in an office sample Device View 24

Figure 4.1 Shape leaving boundaries 26

Figure 4.2 Shapes overlapping one another 26

Figure 4.3 The Board Class Fixing limitations and check methods 27

## LIST OF ABBREVIATIONS

|  |  |
| --- | --- |
| CPU | Central Processing Unit |
| ERD | Entity Relationship Diagram |
| IT | Information Technology |
| SDK | Software Development Kit |
| AR | Augmented Reality |
| UI | User Interface |
| IDE | Integrated Development Environment |
| PC | Personal Computer |
| UML | Unified Modelling Language |
| VR | Virtual Reality |

**LIST OF TECHNICAL TERMS**

* + 1. Tetrominoes Shapes that fall from the Top of the Game Board
    2. Inspector The window in Unity Engine that gives every single detail of a game object and allows for removing or adding of extra components
    3. Sprite A sprite is a two-dimensional bitmap that is used to create a larger scene, most commonly in a two-dimensional video game.
    4. Algorithm a method or set of rules to be followed by a computer in calculations or other problem-solving processes.
    5. Programming The process of producing a set of instructions that inform a computer how to complete a task is known as programming. A multitude of computer programming languages can be used to program.

# CHAPTER 1: INTRODUCTION

## Overview

Tetris is a tile matching, line clearing, fun puzzle video game, it has different variations but all follow the major rules of players completing lines or rows by moving and rotating different shapes called tetrominoes which fall from the sky, the game board that shows the grid at which everything is cleared. The players advance to the next levels by completing different number of lines, these lines vanish and give the player a specific number of points. If a player can complete four (4) lines at once they have been granted a super point called a Tetris. A variation of the game has a higher value called hexa-Tetris, octa-Tetris and some more. It is a game over when the board is filled up. Tetris has stood the test of time since 1980 till now, different variations and game modes have been made and its popularity is unmatched as it is still one of the most sold game till date, numbering at three (3) on the list (Electronic Arts, 2018). The purpose of this project is to make a different variation of Tetris with reductions and additions to gameplay, progression and augmented reality and take its place among great Tetris games.

## Background and Motivation

Tetris is a video game designed to be fun and tasking, it is designed to make people happy. Since the dawn of the first version till date different variations of the game has been made and played. Tetris has stood the test of time and that makes it a classic. The motivation for this project was the fact I had played so many variations of Tetris, from the black and white Gameboy advance version, to the latest ones such as Tetris effect which I found enjoyable and the fan base agrees. There was a time when Tetris was number one on the best-selling video game list and truly it deserved the spot even now which is currently third on the list according to Electronic Arts and

steam statistics, surely the numbers don’t lie and proves it’s a very fun game and worth it. Because of this I have decided to make my own variation of the game and make it my own by removing, twisting and adding new features. There are major components of the game that makes it what it is and there are some that can be removed such as the hold, and next extras which make the game even easier, so to make it more challenging and rewarding they were removed, and or made a bonus feature where players reach certain scores to earn them. The addition of augmented reality hasn’t been published across platforms yet but adding these features makes it stand out from the older variations of the game.

## Statement of the Problem

This variation of the Tetris game hasn’t been done yet, tile matching games like Tetris, or any other game becomes easily adaptable for players which is a great thing, after some point when the players level up what feasibly is next? The features to be removed should make it more tasking for expert players and new commers that have leveled up in the game. The hold features to be removed made it more fun and rewarding, this feature allows a player to hold a shape for some time until he or she finds the next best time to release the shape to play making it easier. The next feature to be removed makes it easy for the players to know a next shape to be spawned beforehand, removing this would make the players think more and not make bold moves so easily. It is possible to keep these features but only if it is a bonus method where the players have to expend points or in-game currency to use them at some given point in time this can be achieved as a downloadable content or a future update. The augmented reality feature gave Tetris a push in this time just like Pokémon go did when it first came out. This version of Tetris will inculcate AR and can be played on any area in our modern-day homes and relaxation points which will increase its popularity the more.

## Aim and Objectives

The aim of this project is to design an enhanced version of Tetris game by making it competitive,

rewarding and adding new experiences to the game. To give experienced players more and to give new players ready to join the Tetris family something to look forward to by adding and removing features that can make this feasible.

* + - To increase player experience by addition of AR
    - To make the game more challenging
    - Removing the hold feature
    - Removing the next feature
    - Making bonus features derived from the removal of the hold and next features
    - Addition of Augmented reality to the game

## Significance of the Project

The implementation of this project has potential to increase gameplay and challenges, reduce UI robustness and to bring more players to the Tetris community. This research and implementation further more experience to the puzzle gaming community by bringing in tasking challenges that were not made available before in previous variations of the game. Furthermore, a detailed report on each phase of the implementation of the project and room for better enhancements and updates such as augmented reality will be available. A new Tetris game variation made for more made for new experiences will draw in more players who want more in their gaming experience.

With a new and more rewarding Tetris variation, the Tetris fans and new players play a better and more breath-taking game. This variation of Tetris does enough to make it so, it creates a fun and real-life experience with the help of AR, more challenging experience with the reduction and addition of components and features such as the hold and next, with more updates to come as research and implementations continue, thereby bringing another push to the puzzle genre experience on some platforms, windows and smartphones. This project will aid in increasing the number of gamers in the gaming community as a whole.

The current and previous variations of published Tetris games all have the same features that will be removed or updated as a new feature in this project, with the addition of AR as an update to match a real-life experience for players ready to have a new and challenging experience in Tetris.

## Project Risks Assessment

**RISKS**

## Table 1.1: Project Risks Assessment

|  |  |
| --- | --- |
| Inability to carry out research due to loss of hardware/software resources | Be aware of and observe school IT security procedures  Secure Android mobile phone when not in use. |
| Loss of work due to equipment failure /loss | Weekly data backup to Hard drive and cloud |
| Software availability (Unavailability of API’s) | Alternative API’s will be checked for. Software requirements will be identified in good time for possible contentious software |
| Inadequate supervision | Report to higher authority such as assigned course lecturer or the HOD |
| Health related issues | Report to the supervisors and medical attendants |
| Late delivery of hardware component | Hardware requirements will be identified in good time to be able to order them in good time |

## Scope/Project Organization

The scope of this project is concerned with creating a successful variation of Tetris that functions almost the same way as other variations of Tetris but with the new reductions and additions that make it different from the rest. The Chapter 1 is the Introduction where there is Overview, Background and Motivation, Current System, Statement of the problem, Proposed system, Aim and Objective, Significance of the project, project Risk Assessment and Scope. The Chapter 2 contains Literature Review which includes Historical Overview, Current Findings and studies, Summary. This provides readers with knowledge and information about the Tetris gameplay and how it differs from other games. In the Chapter 3, the Requirements, Analysis and Design are involved which contains the Overview, Requirements Specifications, Design Overview, Detailed or low-level design, Summary. This provides a software development life cycle, also known as SDLC, which is used to develop the proposed system and provides the merits and benefits of using the selected methodology for the system. The methodology chosen is fully explained by its stages or phases, the strengths, and weaknesses of the software development life cycle. It would explain the method used to gather the data used in developing the proposed system. Here, a questionnaire and interview were chosen to do the main research. The Chapter 4 is the Implementation whereby the Overview, Development Environment, Implementation Stages, Database Setup, Major technical problems, overcoming technical problems and Testing are involved. And the final chapter which is Chapter 5 has the Evaluation, Conclusion and Recommendation: -Overview, Achievements, challenges, Future enhancements, Recommendations, Summary. There is also the References and Appendices page available.

# CHAPTER 2: LITERATURE REVIEW

## Introduction

Tetris is a tile matching video game that gamers and new commers will play for its adaptability easy interface, challenges and entertainment, especially with the addition of features that have make it so. Tetris is the third most sold game in the world according to research and statistics (Electronic Arts, 2018). Over the years different variations have been made all having adjustments and additions from the original, features to be removed adjusted and added haven’t been inculcated at once the way this project is meant to accomplish. By doing so the game will be more rewarding and users of old and new will have a new experience. All can be accomplished with the right set of tools and technologies as would be described.

This chapter is based on the literature review of the techniques and technology used in this thesis. Section 2.2. is based on the Historical Overview of Tetris game. Section 2.3 gives descriptions of some of the literature available with regards to the application of C#/Unity/AR in solving the research question. Finally, section 2.4 is a summary of the entire chapter.

## Historical Overview

Tetris is a tile-matching video game developed in 1984 by Russian software engineer Alexey Pajitnov (Vadim Gerasimov (n.d)). Several corporations have published it, most famously during a dispute over the appropriation of the copyright in the late 1980s. The rights were reverted to Pajitnov in 1996, who co-founded The Tetris Company with Henk Rogers to oversee licensing, after a significant period of publication by Nintendo.

Tetris established itself as one of the great early video games, based on simple rules and requiring intellect and abilities. As of December 2011, it had sold 202 million copies, about

70 million physical units and 132 million paying mobile game downloads, making it one of the best-selling video game franchises of all time.

Over the years the Tetris name has withstood the test of time and has earned the title of a classic, different variations are still being ported and published almost every year, such as Tetris mobile the new mobile version and Tetris effect. Therefore, it should not be a surprise because Tetris is still being played till date. The work presented in this project is centered around creating a different variation of the game and be among great versions to be published and played.

## Related Work

There are published works and variations of the game that technically relates to mine, in what should be altered and worked on this project such as the features that has been enhanced removed and added in mine will be a step out of previous work’s direction. The features to be removed are that of previous variations such as Tetris online, Tetris mobile new, and Tetris effect have similar features as they have to because of that’s what makes Tetris, Tetris. Similar works in the concept of the addition of AR that would be used in this project have been found on git but built different and haven’t been published on main stream platforms. Some ideas may be derived from one or two versions that were found on git, and that of the published recent Tetris effect. With the exception of errors, the end result of the Tetris version, Tetris+AR online solves the problem of AR but not enough as it isn’t complete, buggy in functionality and uses a different game engine and SDK. Links to the project and published version of Tetris will be provided at the end of this thesis.

## Summary

The literature reviewed in this chapter 2 shows that the Tetris game has been made with different technologies and different features for each variation. It shows that there is still room for adjustments and improvements, room for a more enjoyable and eye-catching user experience in this project.

Furthermore, it is consequential to explore and research on different game versions that are published on main stream platforms of old and new, also shared among websites and teams such as the one found on git, to give a better understanding and knowledge for a new game version of Tetris.

Chapter 3 will present the requirement analysis and the methodology adopted in solving the project objectives for giving Tetris a new experience and challenge.

# CHAPTER 3: REQUIREMENTS, ANALYSIS, AND DESIGN

## Overview

Chapter 3 is comprised of the requirement, analysis and design of the Tetris game. In chapter 3 Requirements, analysis and the design, the methodology or methodologies used will be discussed, the consequential choice of method used will be highlighted. The requirement system specifications, functional and non-functional requirement, application architecture, schematics and user interface will be discussed in this chapter.

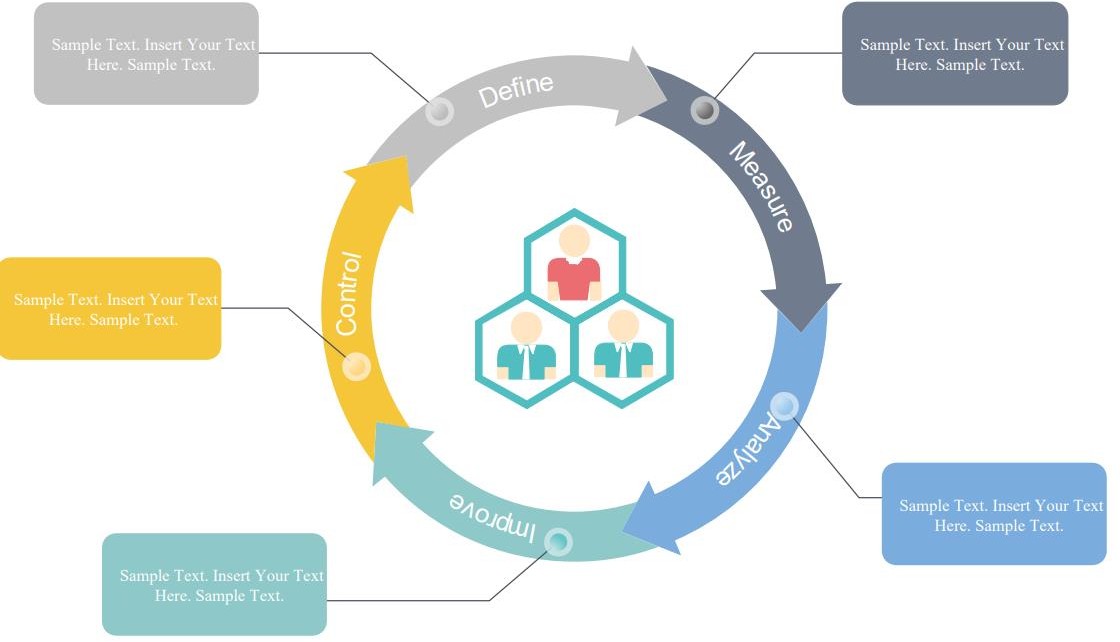
## Methodology

The methodology will describe the development process and testing of the Tetris game and how each aspect of the game will be introduced and deployed. The chosen methodologies are the Agile and Waterfall methodology. Each proposed method will have its definition and reason for choice.

## Agile Methodology

Agile methodology is a simple and very effective iteration that transforms a vision into a real solution. It will involve continuous planning, refactoring, improvement and early deployment of the project. The Agile philosophy is a series of concepts that emphasize versatility and adaptability. Agile focuses on allowing one to execute in manageable increments in order to provide greater responsiveness to changing business needs. The key concept is to create small features for the current project in small increments of time rather than the whole project from start to finish. The outcomes of each iteration are used to modify the project plan in this way: each iteration is like a small project in and of itself, using "inspect and adapt" practices to adjust priorities and assess progress. In the game there are four main phases in this methodology which are stated: Discover, Design, Create, and Test are the steps in the process. If a process is finished, we go over the initial

plan again, starting from the beginning and planning the guided work until the final target is met. When the game is done and we're ready to send it to our customers, we risk running into a major issue: we have the perfect game that no one has ever played. At this point, we are compelled to alter our plans and try to satisfy the desires of our users. As a result, our End was unable to achieve the original Target. This means that when we begin developing a game, we are confronted with a large "Cone of Uncertainty." Small targets for short periods of time are part of agile preparation, which allows us to go through all four phases each iteration.



**Figure 3.1: Agile Methodology (Wondershare, 2021)**

## Table 3.1: Advantages and Disadvantages of the Agile Methodology

|  |  |
| --- | --- |
| **Advantages of Agile Methodology** | **Disadvantages of Agile Methodology** |
| The delivery of software is unrelenting in the Agile approach. | In Agile methodology the documentation is less |

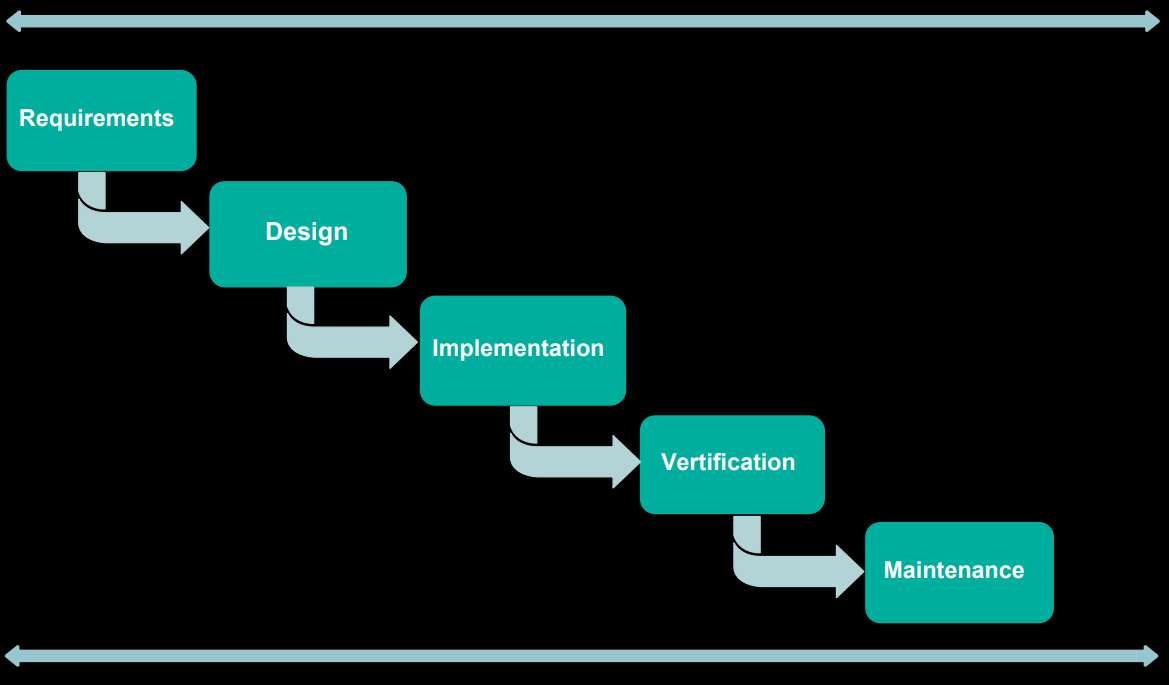
|  |  |
| --- | --- |
| Consumers are happy because each iteration brings them a new working function of the product. | Sometimes in Agile methodology the requirement is not very clear hence it’s difficult to predict the expected result |
| Customers will view the operational function that met their needs | It's difficult to quantify the actual effort involved in a few projects at the start of the software development life cycle. |
| The successful design of the product is prioritized in this approach. | There is always the possibility of an ever- lasting project due to the ever-evolving features |
| Also, in the later stages of production, changes in specifications are recognized. | It's difficult to estimate the amount of time and resources needed for complex projects. |
|  |  |

Table 3.1 describes some of the advantages and disadvantages of agile methodology in some level of detail

## Waterfall Methodology

When it comes to software creation, the waterfall technique takes a step-by-step approach. The project is broken down into a series of activities, with phases being the highest-level grouping. The waterfall technique is a structured approach that includes a list of specific tasks, as well as documentation and exit requirements for each step. SDLC methodology products are frequently required by larger enterprises, particularly in larger IT application projects.

In video games, since it treats the project as a collection of certainties and creation as essentially an act of translation, it is a widely despised method of working, especially in games. The truth is that most game projects are attempting to find a game dynamic that works (a method known as "finding the fun"), which is an inherently messy and creative process. As such most developers prefer to work in an agile.



**Figure 3.2: Waterfall Methodology (Wondershare, 2021)**

**Table 3.2: Advantages and Disadvantages of the Waterfall Methodology**

|  |  |
| --- | --- |
| **Advantages of Waterfall Methodology** | **Disadvantages of Waterfall Methodology** |
| Suitable for smaller projects with well-defined specifications | It's not a good idea for a complicated project with a lot of requirements that change regularly. |
| At any stage of the software development cycle, detailed documentation is generated. | Documentation occupies a lot of time of developers and testers |
| The requirements are stated explicitly and precisely, and they remain constant throughout the project's progress. | All specifications must be identified prior to development, which causes a significant delay in the start of the project. |
| With minimal client involvement, the project is entirely dependent on the project team. | Client feedback is not possible to incorporate into the ongoing development process. |
| Any program modifications are made during the production process. | Small changes or errors in the finished software can cause a slew of issues. |

Table 3.2 describes some of the advantages and disadvantages of waterfall methodology in some level of detail

## Comparison of Methodologies for Game Development

**Table 3.3: Comparison of Agile Methodology and Waterfall Methodology**

|  |  |  |
| --- | --- | --- |
| **Features** | **Agile** | **Waterfall** |
| Progress | Progress is calculated in terms of defined and delivered functionalities by using an agile model. | Progress is usually calculated in terms of the number of completed and tested pieces such as requirements in the Waterfall model. |
| Process | The agile model is a method of incremental delivery in which each incrementally distributed component is created by iteration. | The waterfall model is highly organized and follows a set of steps to collect specifications, analyze them, prepare SRS documents, design, code, and  test them. |
| Customer Interaction | There is a high level of customer engagement. An incremental version of the product is delivered to the consumer after each iteration. | Customer interaction is extremely limited. After the overall production is finished, the product is shipped to the  consumer. |
| Team | Agile team can be a one-man method or a team of 2 to 9 members | The Waterfall model allows for a larger team to be formed, but contact between members  is minimal. |
| Life Span | Even if a project is scrapped in the middle, the agile model always leaves the consumer with some useful code that might have already been placed into production. | If a waterfall-model project is cancelled in the middle of production, the abandoned project leaves just a few documents behind. |
| Flexibility | The agile model allows specifications to be changed after the development process has begun, making it  more flexible. | Waterfall model is rigid, it does not allow to change requirements after the developments process starts. |

In Waterfall methodology, it treats the project as a collection of certainties and creation as essentially an act of translation, it is a deeply unpopular method of working, especially in games. The truth is that most game projects are attempting to find a game dynamic that works (a method known as "finding the fun"), which is an inherently messy and creative process. As such developing with Agile methodology is preferred.

## Tools and Techniques

The following are the methods and applications that were used in the creation of this video game:

UNITY GAME ENGINE: Unity is a cross-platform game engine developed by Unity Technologies. It was first revealed and published as a Mac OS X-exclusive game engine in June 2005 at Apple Inc.'s Worldwide Developers Conference. The engine had been expanded to fit more than 25 platforms as of 2018. The engine can be used to make three-dimensional, two-dimensional, virtual reality, and augmented reality games, as well as virtual reality and augmented reality games. Other industries, such as film, automobile, architecture, engineering, and building, have adopted the engine.

VISUAL STUDIO 2019: is a Microsoft integrated development environment (IDE). It's used to make blogs, web applications, web services, and smartphone apps, and video games together with UNITY. Windows API, Windows Shapes, Windows Presentation Foundation, Windows Store, and Microsoft Silverlight are some of the Microsoft software development tools used by Visual Studio. It has the ability to generate both native and managed code. Visual studio will be used interchangeably with the UNITY game engine for scripting/programming for the entire game to help bring game objects to life.

ADOBE PHOTOSHOP: is a raster graphics editor developed and published by Adobe Inc. for Windows and macOS. Other than image and raster editing, it will be used for digital art for the Tetris game to create game pieces and style environments and effects. Each file art asset created in photoshop will be imported in the unity game engine for further configurations and assignments.

AUDACITY: Audacity is a free and open-source digital audio editor and recording application software. In addition to recording audio from multiple sources, Audacity can be used for post- processing of all types of audio by adding effects such as normalization, trimming, and fading in

and out. Editing of sound effects and background music for the game will be handled with audacity where needed.

## Ethical Consideration

The video game would not damage any player or their system for the sake of ethical consideration. Players who download the game will not be asked for any personal details. All assets (2D art, textures, audio, sound effects) were either available in the open-source websites or created from scratch via the tools mentioned in section 3.3 of chapter 3. During the creation of the application, the required Unity software license was obtained lawfully, and no pirated software was used.

## Requirement Analysis

The requirement of this game application was determined by analyzing the information derived from the literature review. From this a large repository of information of various previous Tetris game versions was discovered. The flaws of these applications were identified using these resources, and functional and non-functional specifications were prompted to address these deficiencies, as outlined in the requirements specification section below.

## Requirements Specifications

This section defines the functional and non-functional requirements, which are the requirements that have been made necessary for the game to be defined as efficient and the requirements that increase the project's quality.

## Functional Requirement Specifications

The functional requirements define the video game application's goals from the user's perspective. It specifies the roles that the game must fulfill in order for it to be considered a success.

**Table 3.4: Functional Requirement Specifications**

|  |  |  |
| --- | --- | --- |
| **Req.**  **No.** | **Description** | **Type** |
| FR-101 | The game will run on Android, IOS, and windows  PC | Configuration |
| FR-102 | The application shall include a user interface. | Functional |
| FR-103 | The game will have player input configurations | Configuration |
| FR-104 | The game will have a high score save sequence | Functional |
| FR-105 | The game will run in an Augmented Reality form | Configuration |
| FR-106 | The game will have level advancement with time  and scoring system | Functional |
| FR-107 | The game will have bonus effects and features for  more engagement | Functional |

## Non-Functional Requirement Specifications

Non-functional specifications are features that make the game effective, engaging, safe, and reliable to use; these are the specifications on which players base their game satisfaction.

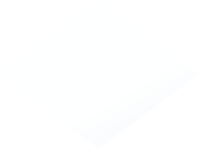
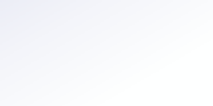
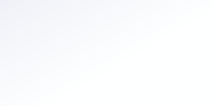
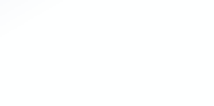
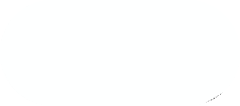
**Table 3.5: Non-Functional Requirement Specifications**

|  |  |  |
| --- | --- | --- |
| **Req.**  **No.** | **Description** | **Type** |
| NFR- 101 | When launched, the game shall stay running unless  there is an intentional shutdown of the application or the platform. | Performance |
| NFR-  102 | The game shall run on any machine of minimum  requirements | Performance |
| NFR-  103 | The UI for the gameplay and menus shall be  simplified | Usability |
| NFR-  104 | The game shall be available at any time | Availability |
| NFR-  105 | The game loading will be done in less time | Performance |

## System Design

This section uses different diagrams to demonstrate how the application's functionalities are implemented, including Application Architecture, Use Case Diagrams, Activity Diagrams, Data-Flow Diagrams, Entity-Relationship Diagrams, and User Interface Design of the game.

## Application Architecture



Splash Screen

Game Starts

Close Game

Level and Score Advance

Game Over

Filled Rows Clear Lines

Response to Key Inputs

Response to Key Inputs

Response to Key Inputs

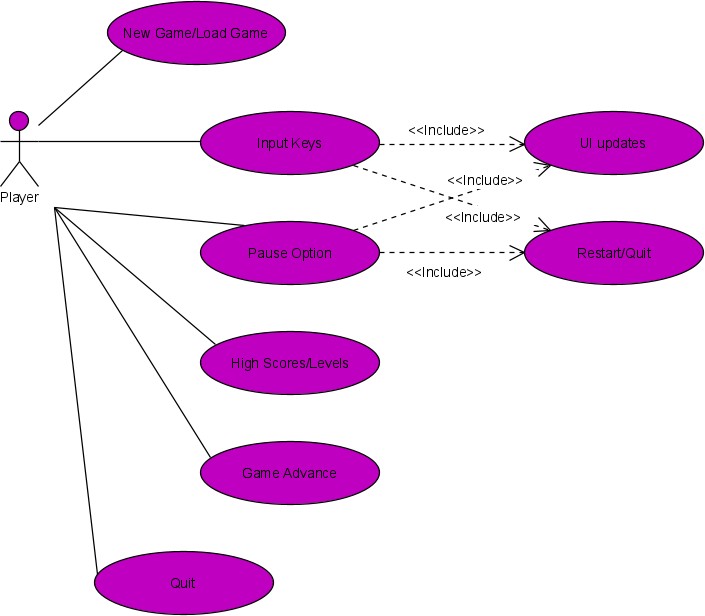
Response to Key Inputs

Response to Key Inputs

Spawned shapes Drop

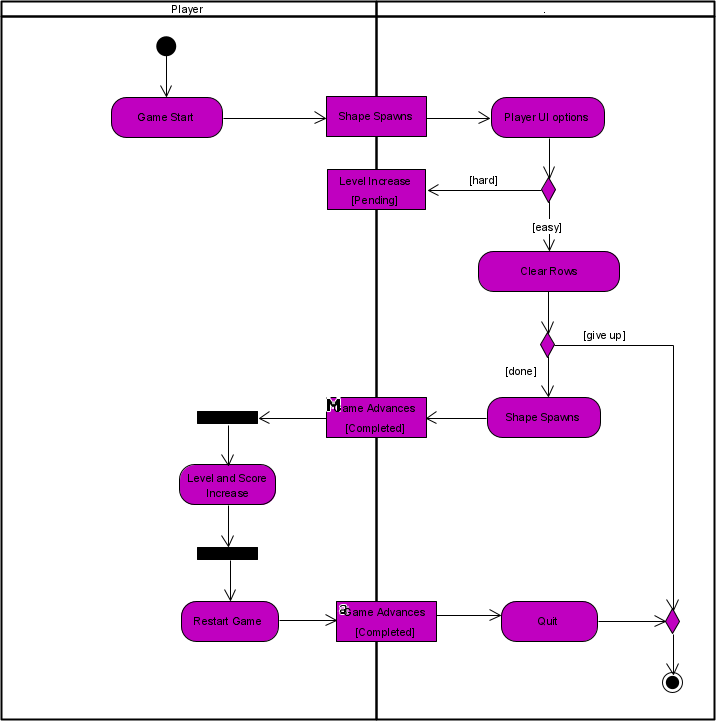
**Fig 3.3: Application Architecture of the Tetris Gam****e**

## Use Case



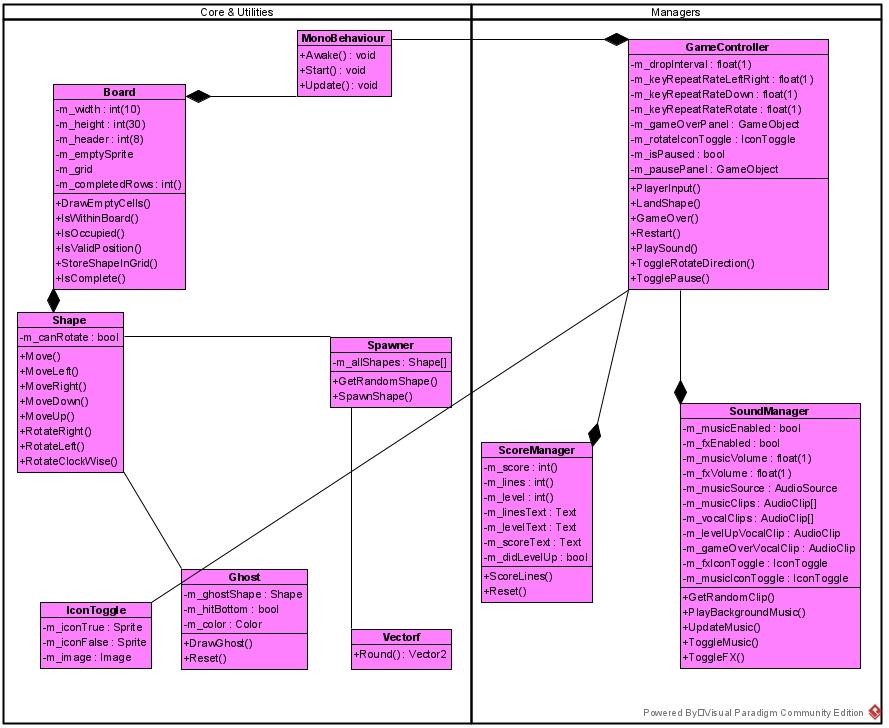
**Fig 3.4: Use Case Diagram of the Tetris Game**

## Activity Diagram



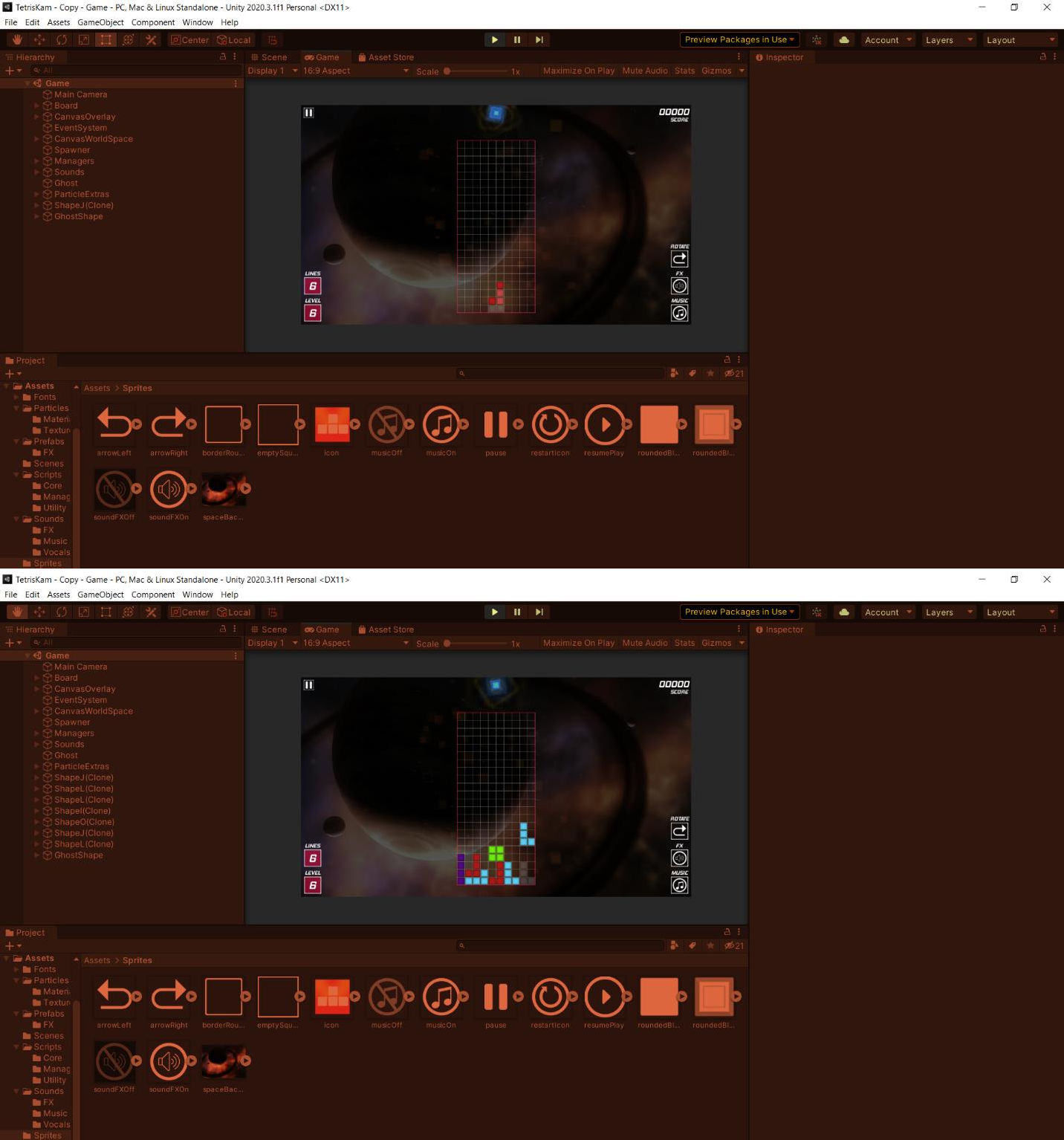
**Fig 3.5: Activity Diagram of the Tetris Game**

## Class Diagram



**Fig 3.6: Class Diagram of the Tetris Game**

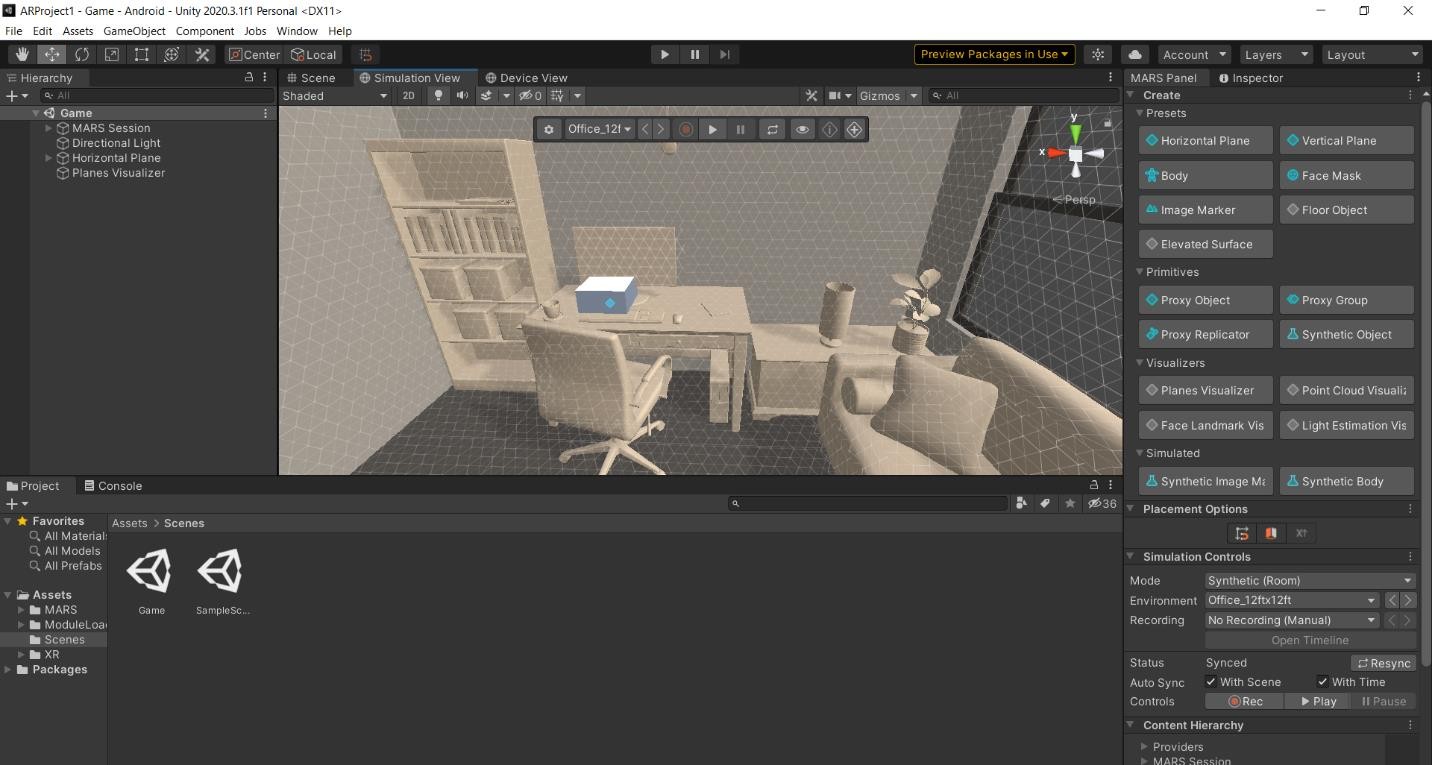
## User Interface Design



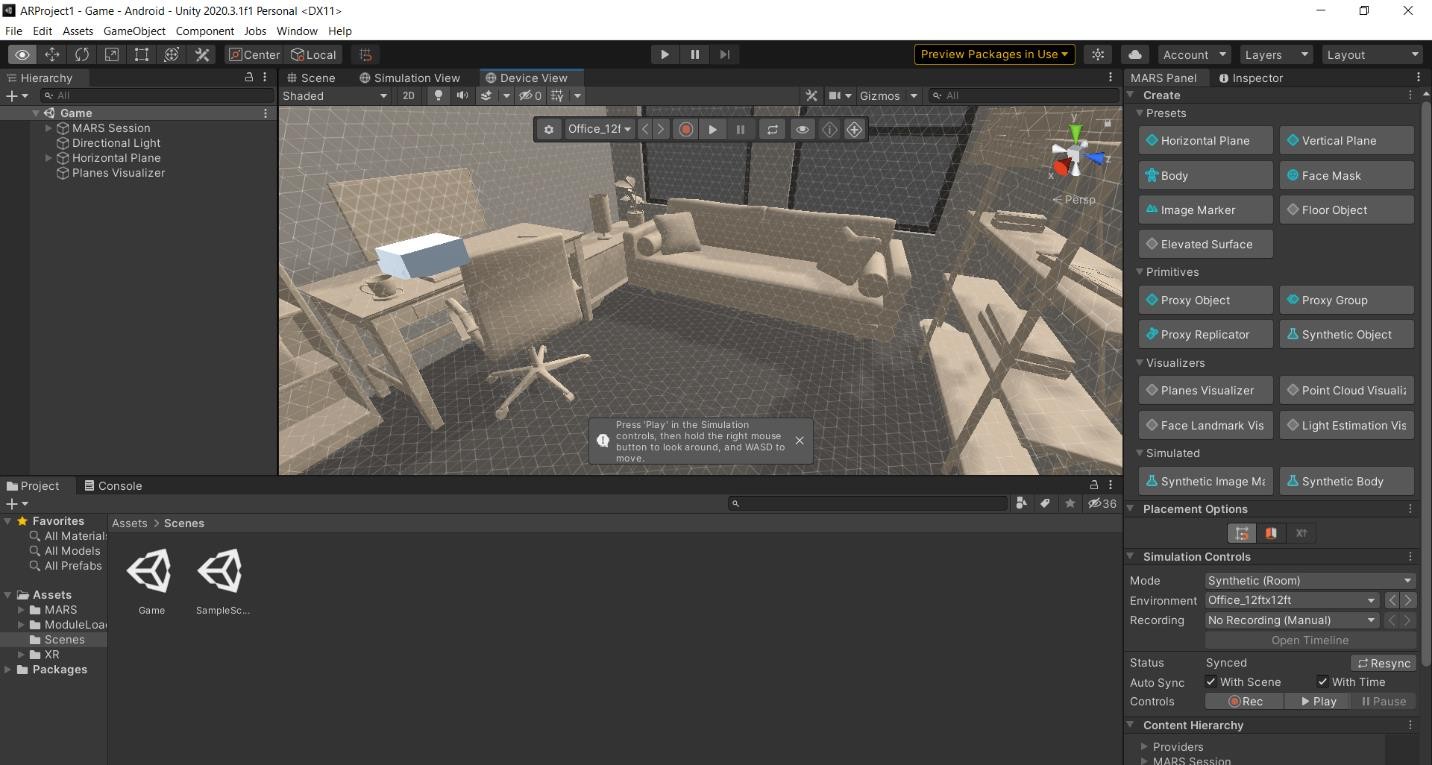
**Fig 3.7: User Interface of the Tetris Game running in Unity Engine**



**Fig 3.8: User Interface of the Tetris Game running in final Build for Windows**



**Fig 3.9: The AR view of the game environment in an office sample Simulation View**



**Fig 3.10: AR view of the game environment in an office sample Device View**

## Summary

This concludes the chapter 3 of the system requirements and designs of the Tetris game. The methodologies that were compared and best chosen for the development and best practices which was Agile methodology. The different game design schematics ranging from the application architecture to the class diagrams were stated in this chapter to give a preview of the in and outs working of the Game and the necessary design implementations without revealing too much. The next chapter will discuss furthermore the true implementations of the game from the design to final build and release.

# CHAPTER 4: IMPLEMENTATION AND TESTING

## Overview

This chapter describes further the detail at which the implementation, development and testing through the Unity game engine of the Tetris game was done. The following aspects of this chapter are given below, they are the main features of the game, problems such as bugs and implementation problems, plans, reports and how everything was solved and ready for deployment.

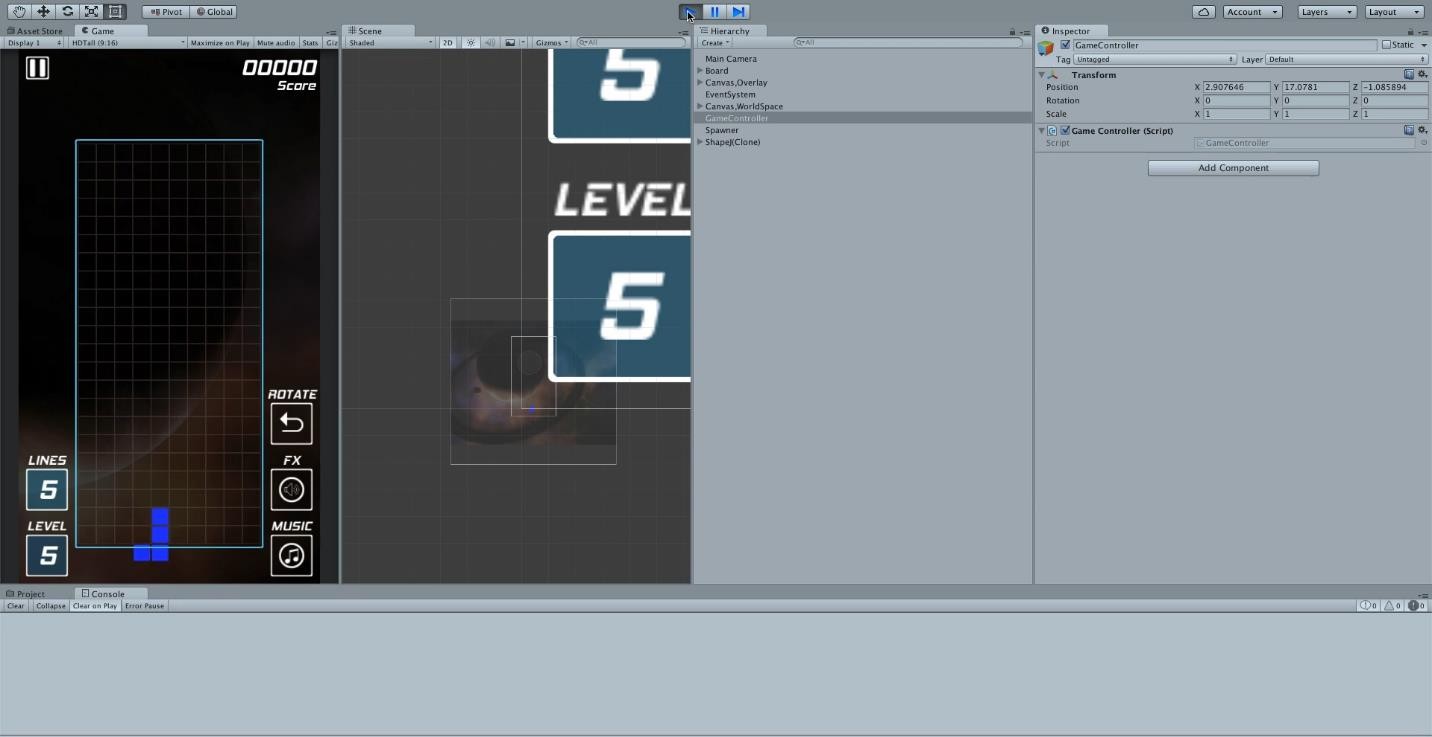
## Main Features

This Tetris game features absence of the hold and next features, instead be bonuses for players to earn, features the ghost ability making it apparent where shapes will drop. Special effects in the background to add a level of detail to the game visuals. A time system that increases per player level making it faster and more daunting for players. The players movement rotations and clearing of rows. The next subsections explain how they were implemented.

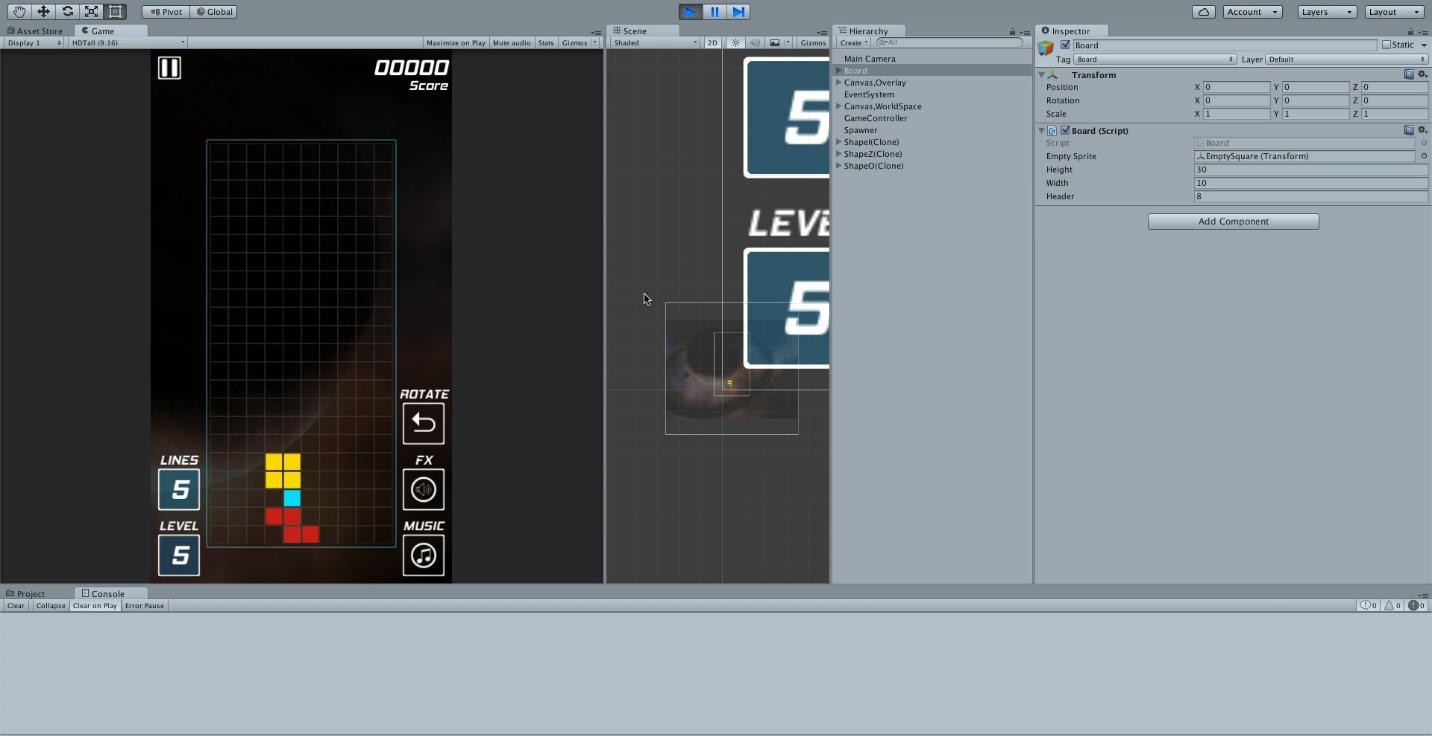
## Implementation Problems

Several bugs were discovered during the game's development. However, this was to be anticipated, as issues would inevitably arise during the development of the game. At the beginning development stages, where the game board and the spawned shapes fall, the shapes never stopped falling because the boundaries and limitations were not scripted yet. The shapes were overlapping over each other which was also a problem at first, it was a bug that had to happen gradually in other to know where to script. When the shapes fill up over the top y axis of the game board there wasn’t a way to end the game, there wasn’t a game over and stoppage of the game when the player loses. The vocals did not loop the way they were intended to during the initial sound phase of the game, they either played at the same time and interrupted the background music or they didn’t play at all. The user interface background layers overlapped the wrong way especially when it came to

the augmented reality integration of the game as expected but the solution was found from the Unity documentation and forums.



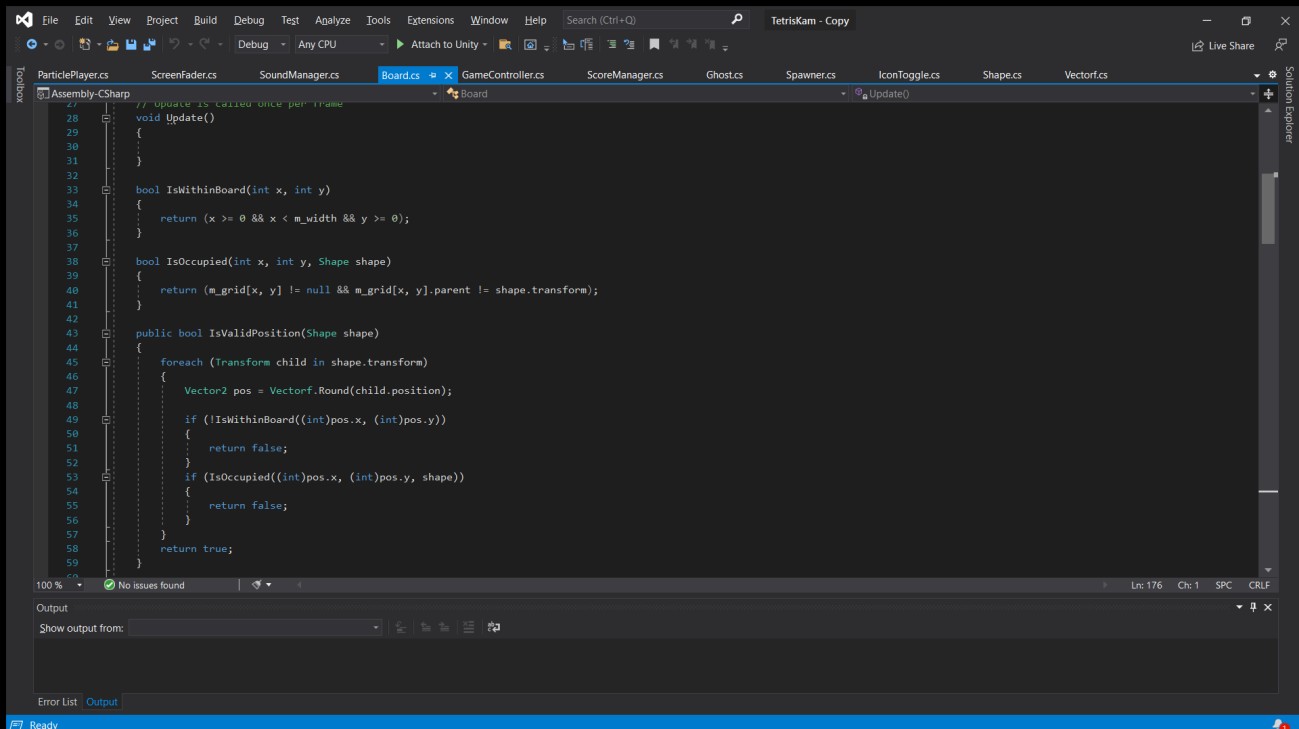
**Fig 4.1: Shape leaving boundaries**



**Fig 4.2: Shapes overlapping one another**

## Overcoming Implementation Problems

Problems were overcome by careful analysis of each game component, starting with the scripts of the Board class and the Shape class where methods for checking limitations and boundaries were implemented. This was done in the Board class and referenced in the game controller by giving it the ability to answer questions such as is the shape in a valid position, is each component of the shape within the boundaries, and is there any other shape occupying a part of the Board. Each of these concepts of validations were created in methods to check if the Board and shapes are behaving like they are meant to. By switching between the IDE and the Unity engine to observe changes it was declared solved. The user interface clashes were solved by reading through the Unity documentation for layer priorities, small iterations were made just by using the Unity engine and no coding was needed for this aspect of the game. The sound system for the game was controlled by a Sound Manager class which needed to be fixed in some blocks in the class. It turned out there were loop clashes of when each vocal or music clip needed to run in the game per action. Simple recoding of the game logic for that block of code.



**Fig 4.3: The Board Class Fixing limitations and check methods**

## Testing

Quality assurance is needed in the final stages of development and deployment of the game. Testing differs from the rest of the game development process in that it prevents the game from failing. Nothing is being made here, yet everything is being staked. Testing ensure that all parts of the game are available, that elements are shown correctly from all sides, and that various functions are implemented correctly. The testing of this game was divided into groups for different aspects. Stress testing, for example, did the best to see how the Tetris game performed when its functionality is pushed to its limits. Other testing performed a complete replay of the game, with all achievements and rewards, to ensure that everything is working properly. Others were assessing the game's entertainment factor, determining whether it was too straightforward or too difficult to keep the player engaged until the end of the session. The sub-chapters that follow go into the features such as scoring, leveling up, music, vocals and effects, and overall game pieces functionality that will be checked as well as the output of each part. At the conclusion, a test report is given.

## Tests Plans (for Unit Testing, Integration Testing, and System Testing)

The following subsections details the types of game test strategies used for the Tetris Game. The Unity Engine made everything less tasking due to its major modes, the Scene mode and Game mode

## Game Testing strategies used step-by-step:

* + - * + Functional testing
        + White-box testing
        + Performance testing
        + Compatibility testing

## Functional Testing

Functional testing concerns such as game mechanics, reliability, and game asset integrity that can arise in the game or its user interface and graphics. User interface research ensures that the game is user-friendly.

Features Tested:

* + - * + Checking of Colors
        + Checking of Backgrounds
        + Checking the screen aspect rations and resolutions for different devices
        + Font sizes in the UI
        + User interface in general especially when it came to the Augmented reality version
        + In Game Camera and how Players view in normal or in AR

## White-box Testing

The architectural, integration, and device aspects of the game are the subject of White Box Testing for Games.

Features Tested:

* + - * + Code inspection: Tested the classes for the game, the source codes and programming logic in each game scenario that may have been possible
        + Path and flow testing: the correct sequence at which the game ran was tested from the IDE and Unity Game mode such as the Shape spawning and leveling up of Players
        + Algorithm specific testing: A scenario was created in the game where the clearing of the game board rows exceeded the maximum number of lines which are four (4), increasing the limit forcefully incurred bugs because the shapes were not greater than four (4) components which made the Engine hang, but was reverted afterwards

## Performance Testing

The game's overall performance was examined. Performance tuning was done to improve the speed and optimization of the game especially during testing of the AR feature.

Important parameters checked during performance testing:

* + - * + Game load performance, tested in the Unity engine and built windows platform
        + Graphics performance of the Game at low settings to higher quality such as the 1080p High definition
        + Processor constraints: The performance of the game was tested on different hardware during the finished windows build and all processors were able to play the game seamlessly, AR version tested but only on android devices that were available, requires specific android devices that are AR capable such as a Samsung Galaxy Tab S6 and above.
        + Battery power: The game ran for long periods of time without draining the battery of devices tested

## Compatibility Testing

Checking if the game was playable on a variety of platforms and with various hardware and software configurations. Because of the Build once and redeploy on any platform Unity has this part of the testing was made possible. When it came to the Augmented reality builds, the limitations were found from the Unity site where a list of AR ready portable devices were already given and their limitations which was found when tested too.

## Test Deliverables

* + - * + Test Cases
        + Test Report
        + Traceability Matrix
        + Test Results
        + Error Report

## Test Suite (for Unit Testing, Integration Testing, and System Testing)

**Test Suite Performed**

Testing in Unity, unit, integration and system is done through the Modes that Unity possesses such as the Debug Mode, Scene Mode and Game Mode. The errors if any are generated at compile time from the Play mode, which pop up and always stops the game from running. Testing for Bugs other than that of just programming can be done in Debug mode, which helps developers see what the parts of code has been altered on the Unity engine. The scene mode helps with placements of Game objects and tests where different Game pieces will be visible to a Player. The overall Gameplay via the Game mode shows the Player’s perspective during testing, if the Gam mechanics work the way they are meant to.

## Test Traceability Matrix (for Unit Testing, Integration Testing, and System Testing)

**Table 4.1: Test Traceability Matrix**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Requirement** | **Description** | **Priority** | **Test**  **Case** | **Test**  **Result** |
| FR-101 | The game shall have a working board | High | 1 | Pass |
| FR-102 | The game shall have a working game manager | High | 2 | Pass |
| FR-103 | The game shall have a working sound manager | High | 3 | Pass |
| FR-104 | The game shall have a working score manager | High | 4 | Pass |
| FR-105 | The game shall have a working UI none AR  version | High | 5 | Pass |
| FR-106 | The game shall have working Effects and Visuals | Medium | 6 | Pass |
| FR-107 | The game shall have working AR implementation | Medium | 7 | Pass |

## Test Report Summary (for Unit Testing, Integration Testing, and System Testing)

The following is a list of the tests that were performed.

**Table 4.2: Test Report Summary**

|  |  |
| --- | --- |
| **Summary of executed tests** | **Results** |
| Number of functions tested | 7 |
| Number of functions not tested | 0 |
| Number of tests passed | 7 |
| Number of tests failed | 0 |
| Percentage of tests passed | 100% |
| Percentage of tests failed | 0% |

## Error Reports and Corrections

* Shapes not being made visible even though the script had no errors. This was caused by placement of the Z-coordinate of the main camera object in the unity inspector.
* Shapes neglecting game board boundaries and other shapes. This was caused by the code logic and fixed appropriately
* Augmented reality crashes. This was caused by hierarchy mismanagement. Still was not perfect as UI’s conjoin as the simulation views positions changed.
* UI not updating, such as level up, scoring and other UI buttons. This was fixed by creating a score manger script that handles updates. Then going back to the unity inspector to place the instances appropriately.

## User Guide

The user guide explains how to operate and interact with the game pieces in the game environment. The menu's user interface was created to be basic and intuitive, with all of the buttons clearly labeled and easy to grasp what they do. The game begins with the player entering the main menu for quick setting of controls and screen resolutions. Navigation across the game and game UIs are done by Player input, button clicks and key presses, this was accomplished by Unity’s inbuilt input manager alongside programming. In the Appendix section, you'll find a diagram of the controls and directions for navigating the menu's user interface.

## Summary

The key components of the game were clearly defined in this chapter, as were the approaches utilized to incorporate them. Issues that arose during implementation and testing were carefully examined, and approaches for resolving them were discussed. The test plans, processes, and results, as well as a user guide, were detailed. During testing, bugs and failures were documented. Bugs and errors discovered during testing were noted, and necessary adjustments were implemented. The conclusion, overall discussion, and recommendations for the video game project are included in the next chapter.

# CHAPTER 5: DISCUSSION, CONCLUSION, AND RECOMMENDATIONS

## Overview

This is the thesis's final chapter, and it will include an overall assessment of the project, as well as information on the obstacles and constraints faced through its development life cycle, as well as suggestions for how the project might be improved in the future. Finally, proposals for the project's direction will be considered depending on the project's findings.

## Objective Assessment

Although the fact that the project did not meet up to every single plan exactly, specifically the integration of the Augmented reality version, all of the video game's functional requirements were met. The video game is built on a stable foundation that will easily enable for future enhancements and features to be integrated into the game's mechanics. This game is now in the early access stages of video game release for the Steam store, Play store, Windows store and Apple store where Players may try out the game and give feedback on what they think of it in its current condition, what modifications they think will benefit the game, and any other ideas they have and it will be available for purchase thanks to the build once and redeploy feature Unity Engine has on all platforms listed.

## Limitations and Challenges

The biggest challenges hindering the full implementation of this project was the budget, time and experience with augmented reality as there are different kits with few tutorials and documentations made available. Coming into this game development put me in the indie game world where making a game that are usually done by masses of individuals of different skills such as Game designers, Level Designers, Music Composers, Art Creators, Assets Creators, Programmers and Publishers, had to be done by me. I had researched on this community of making games and that was another

reason I decided to make a puzzle game like this Tetris that could actually be completed and ready for deployment.

This game was close to completion just based off of the objectives of the thesis. Regardless lack of time and knowledge still made it impossible to actually meet up to a proper augmented reality addition, although it still worked technically but harbored issues of User interface misplacements therefore not ready for deployment for the augmented reality version. The Unity game engine was created to make it easier for small studios, one-man developers and teams whether for game development, architecture, animations and movies to create their own content without the requirement for a huge development team. While it makes game production more accessible and simpler, it is still best done as part of a team even if it’s just two people examples the creators of games like Fez, Super Meat Boy, and Undertale.

I had knowledge of programming and digital art, some game design. However, there is a great requirement for knowledge and expertise in UI design and layer management, Augmented reality, game level design, effects and animations. As a result, a good amount of time was spent learning about these fields as opposed to developing the game in completion.

The budget that gaming studios normally have for a project was the final obstacle to overcome. This was a lot easier to overcome because there were so many free audio, visuals, and game assets on Unity to choose from. Even so, sprites had to be developed, animated, and assembled.

## Future Enhancements

This project has a lot of potential for development, and future releases will include enhancements to the existing built-in capabilities. The first priority for future development will be to integrate a 100% working AR feature of the game without crashes, clashes of UIs and smooth detection of real-life environments and object relative to the Tetris game.

Addition of UI options for the top scores made from playing a game session, dynamic features for the UI will make the game look further interesting.

The reason for making the game which was to make it harder and rewarding by removing the hold and next features of the game and increased level speed. There will be bonusses in the future versions where random objects appear in the Game board that make the game slow time for some

seconds, objects that bring the next feature temporarily, but only when Players reach certain scores and levels.

In terms of effects there will be background switches and more special effects when players clear rows and other effects that come randomly

## Recommendations

Following the findings of this work, it has been determined that a thorough comprehension of Unity's documentation, as well as the use of its pre-built packaging system, is an essential core part of building games in the most efficient way possible. A lot of features Unity provides that other engines does not especially when there is ample knowledge of C# programming language. Augmented reality needs to be pushed to as many industries as possible to give Players and Creators new ways to express themselves.

## Summary

The project documentation comes to a close with this chapter. It went over the project's goals, scope, risk assessments, requirements, analysis, and design, as well as the methodology utilized throughout the development life cycle and how it was executed and tested for real-world application. Finally, changes that could be done in the near future were considered, as well as recommendations. Finally, changes that could be implemented in the near future, as well as ideas for the gaming industry, were considered.

# REFERENCES

* + - Gerasimov, V., 2021. Original Tetris: Story and Download. [online] Vadim.oversigma.com. Available at: <https://vadim.oversigma.com/Tetris.htm> [Accessed 4 February 2021].
    - Marionette Studio. 2018. Agile Game Development - A Quick Overview - Marionette Studio. [online] Available at: <https://marionettestudio.com/agile-game-development- quick-overview/> [Accessed 13 January 2021].
    - Steamcommunity.com. 2021. Steam Community: Discussions. [online] Available at:

<https://steamcommunity.com/discussions/> [Accessed 30 April 2021].

* + - Song, S., 2021. yiming95/AR-Tetris. [online] GitHub. Available at:

<https://github.com/yiming95/AR-Tetris> [Accessed 2 March 2021].

* + - Technologies, U., 2021. Solutions | Unity. [online] Unity. Available at:

<https://unity.com/solutions> [Accessed 27 January 2021].

* + - Technologies, U., 2020. [online] https://unity.com/. Available at:

<https://forum.unity.com/?\_gl=1\*syx94h\*\_ga\*MTM2ODE5MDIwNS4xNjEzNjY2NTE w\*\_ga\_1S78EFL1W5\*MTYyMjE0MjcwNC4xLjEuMTYyMjE0Mjg1Mi42MA.&\_ga=2. 25458557.1998700642.1622142705-1368190205.1613666510> [Accessed 21 March

2021].

* + - Tetris Game Surpasses 100 Million Paid Mobile Downloads, Was the Best-Selling Mobile Phone Game of All Time". Electronic Arts. January 21, 2018. Archived from the original on March 16, 2021. [Accessed December 28, 2020]
    - Tetris. 2021. Tetris | The addictive puzzle game that started it all! [online] Available at:

<https://tetris.com/> [Accessed 11 December 2020].

* + - Wondershare, 2021. [image] Available at: <https:/[/www.edrawsoft.com/template](http://www.edrawsoft.com/template-dmaic-)-[dmaic-](http://www.edrawsoft.com/template-dmaic-) model.html> [Accessed 15 March 2021].
    - Wondershare, 2021. [image] Available at: <https:/[/www.edrawsoft.](http://www.edrawsoft.com/template-)c[om/template-](http://www.edrawsoft.com/template-) waterfall-model.html> [Accessed 15 March 2021].

## APPENDICES

**Appendix A - Project Document**

The project’s documentation for the analysis, design and implementation of the Tetris video game projects:

**IN-DEPTH PROJECT DOCUMENTATION**

**Full Candidate Name**: Kamachi Ojukwu

**Student ID**: BU/18B/IT/3I71

**Tetris**

A Tile-matching puzzle Windows and Android game.

**Course of Study**: B.Sc. Computer Science

**Objectives and Background Study**

Tetris is a video game designed to be fun and tasking, it is designed to make people happy. Since the dawn of the first version till date different variations of the game has been made and played. Tetris has stood the test of time and that makes it a classic. The motivation for this project was the fact I had played so many variations of Tetris, from the black and white Gameboy advance version, to the latest ones such as Tetris effect which I found enjoyable and the fan base agrees. There was a time when Tetris was number one on the best-selling video game list and truly it deserved the spot even now which is currently third on the list according to Electronic Arts and steam statistics, surely the numbers don’t lie and proves it’s a very fun game and worth it. Because of this I have decided to make my own variation of the game and make it my own by removing, twisting and adding new features. There are major components of the game that makes it what it is and there are some that can be removed such as the hold, and next extras which make the game even easier, so to make it more challenging and rewarding they were removed, and or made a bonus feature where players reach certain scores to earn them. The

addition of augmented reality hasn’t been published across platforms yet but adding these features makes it stand out from the older variations of the game

**Statement of the Problem**:

This variation of the Tetris game hasn’t been done yet, tile matching games like Tetris, or any other game becomes easily adaptable for players which is a great thing, after some point when the players level up what feasibly is next? The features to be removed should make it more tasking for expert players and new commers that have leveled up in the game. The hold features to be removed made it more fun and rewarding, this feature allows a player to hold a shape for some time until he or she finds the next best time to release the shape to play making it easier. The next feature to be removed makes it easy for the players to know a next shape to be spawned beforehand, removing this would make the players think more and not make bold moves so easily. It is possible to keep these features but only if it is a bonus method where the players have to expend points or in-game currency to use them at some given point in time this can be achieved as a downloadable content or a future update. The augmented reality feature gave Tetris a push in this time just like Pokémon go did when it first came out. This version of Tetris will inculcate AR and can be played on any area in our modern-day homes and relaxation points which will increase its popularity the more

## Appendix B - Questionnaire

The Questionnaire carried out is as follows; Project Questionnaire

1. What is your gender?
   * Male
   * Female
2. How old are you?
   * Below 18

o 18-21

o 22-25

o 26-30

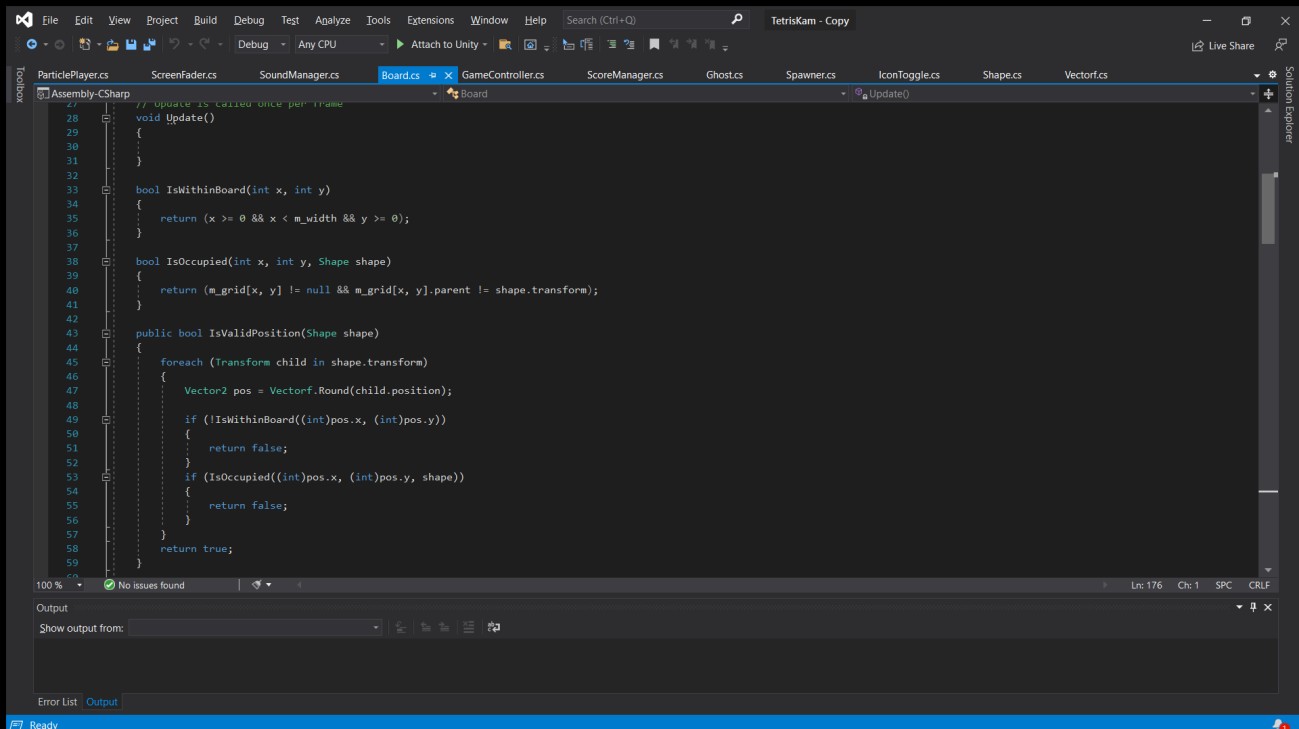
* + 31 and above

1. Do you play games?
   * Yes
   * No
2. What video game genres are you interested in?
   * Puzzle
   * Action
   * Adventure
   * Sandbox
   * Indie
   * Racing
   * Fighting
   * Platformer
   * Shooter
   * Visual Story Tellers
3. Have you ever played a puzzle game?
   * Yes
   * No

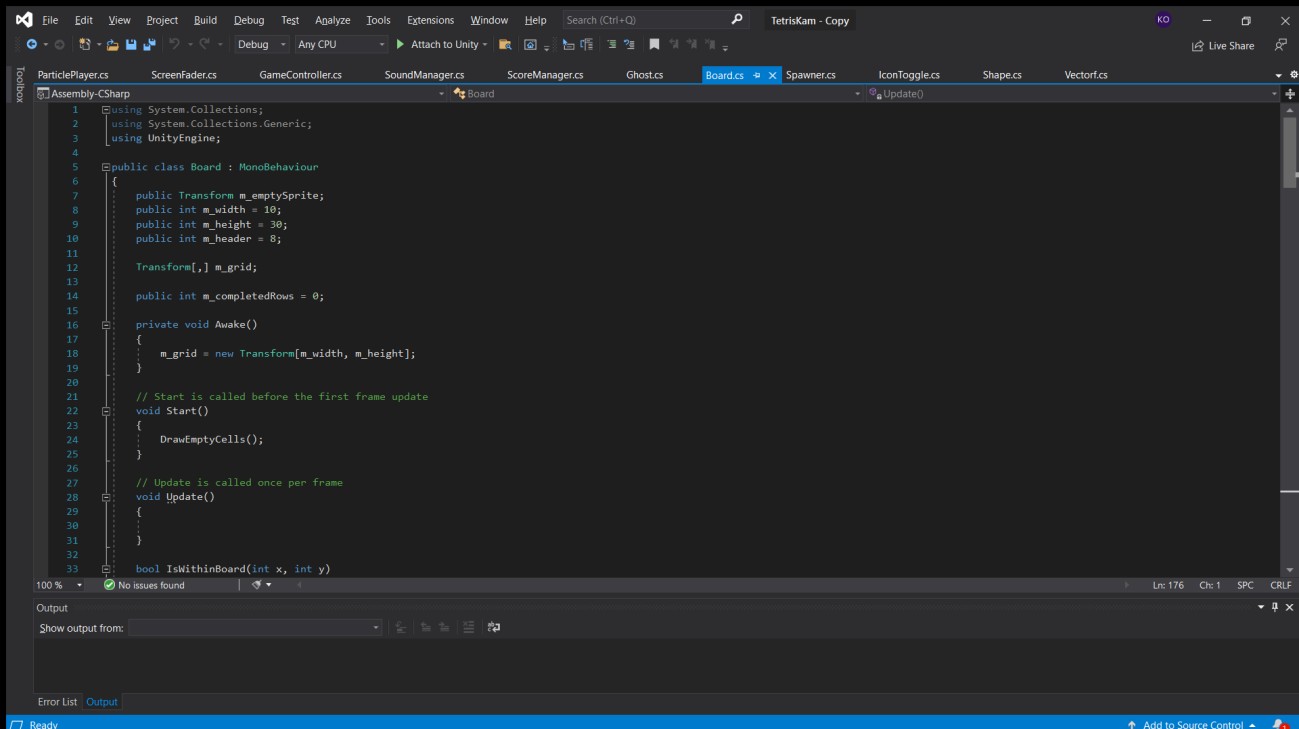
o

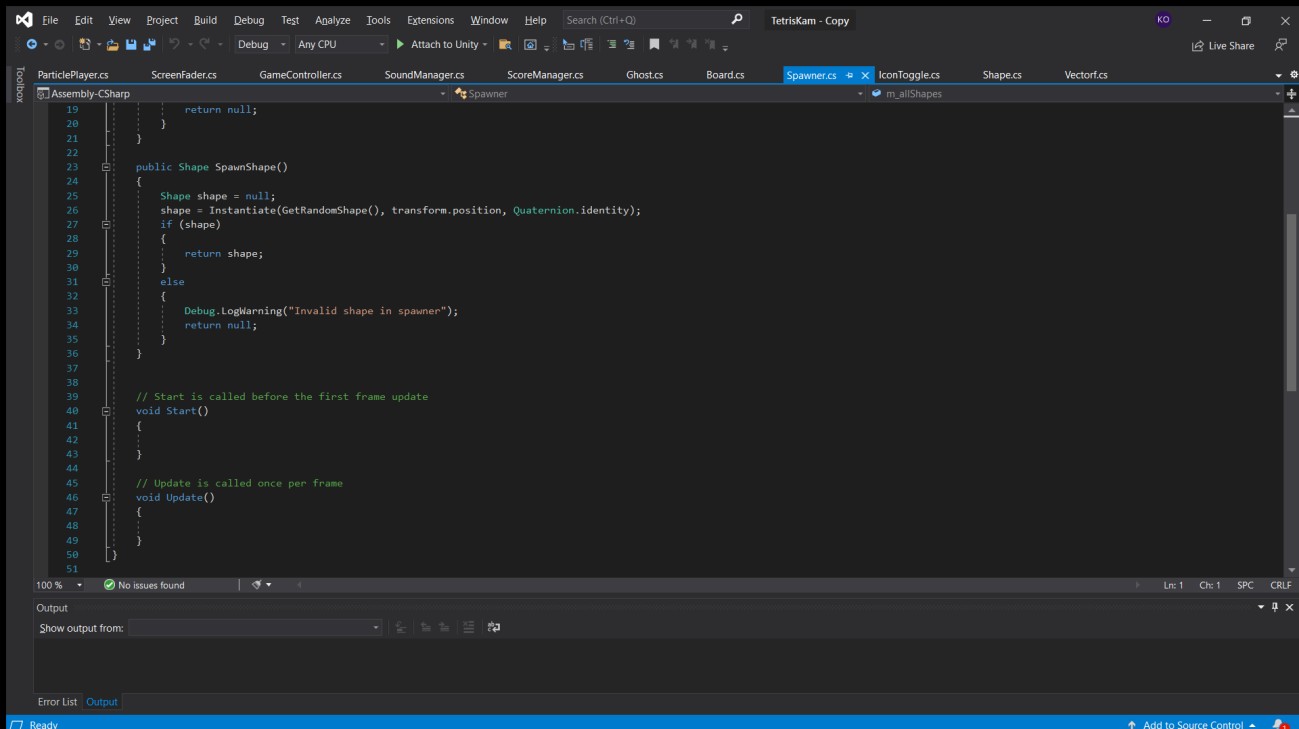
1. Have you ever played a variation of Tetris?
   * Yes
   * No
2. What level do you consider yourself at video games?
   * Beginner
   * Intermediate
   * Expert
   * Kami
3. What level do you consider yourself at Puzzle games?
   * Beginner
   * Intermediate
   * Expert
4. What level do you consider yourself at Tetris games?
   * Beginner
   * Intermediate
   * Expert
5. Have you ever experienced Augmented Reality?
   * Yes
   * No
6. Have you ever experienced Virtual Reality?
   * Yes
   * No
7. Do you own a smartphone?
   * Yes
   * No
   * Obviously
8. Is your smartphone AR or VR compatible?
   * Yes
   * No
9. Would you ever try out a Tetris game?
   * Yes
   * No
10. Would you ever try out Game development?
    * Yes
    * No

## Appendix C – Source Codes

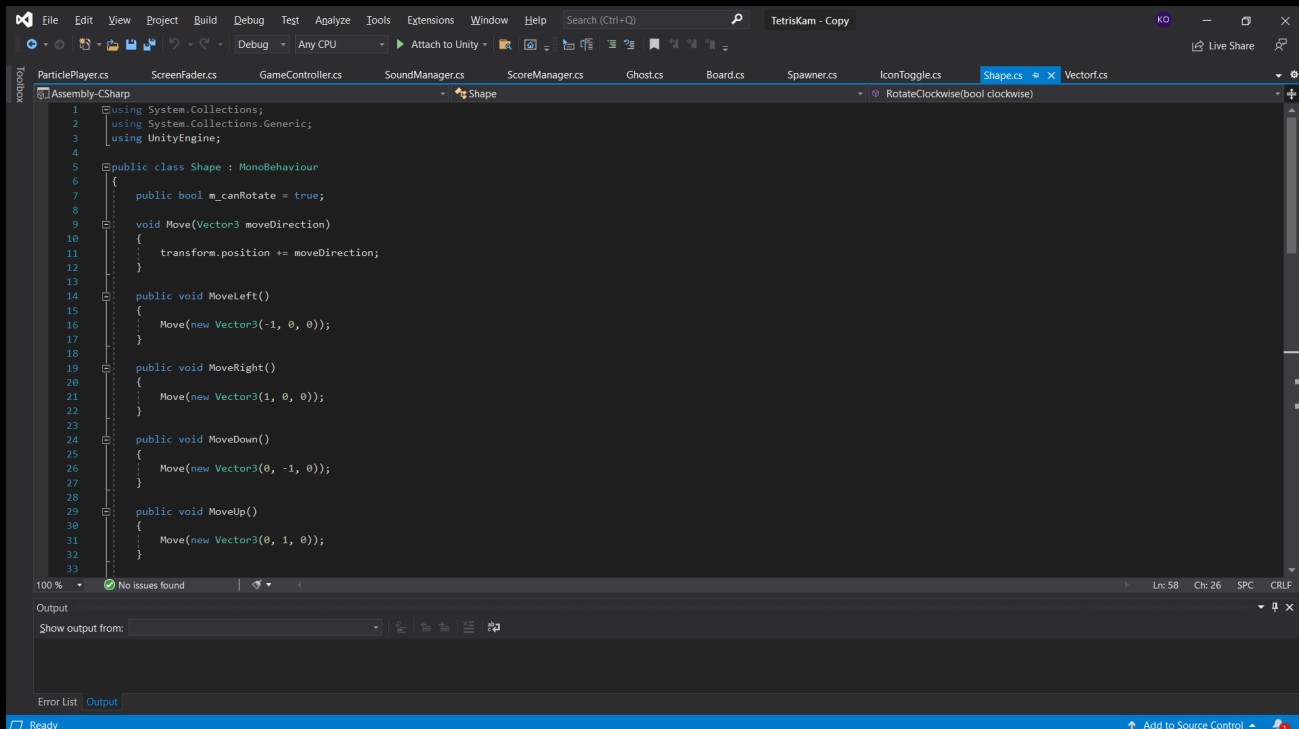


Board class

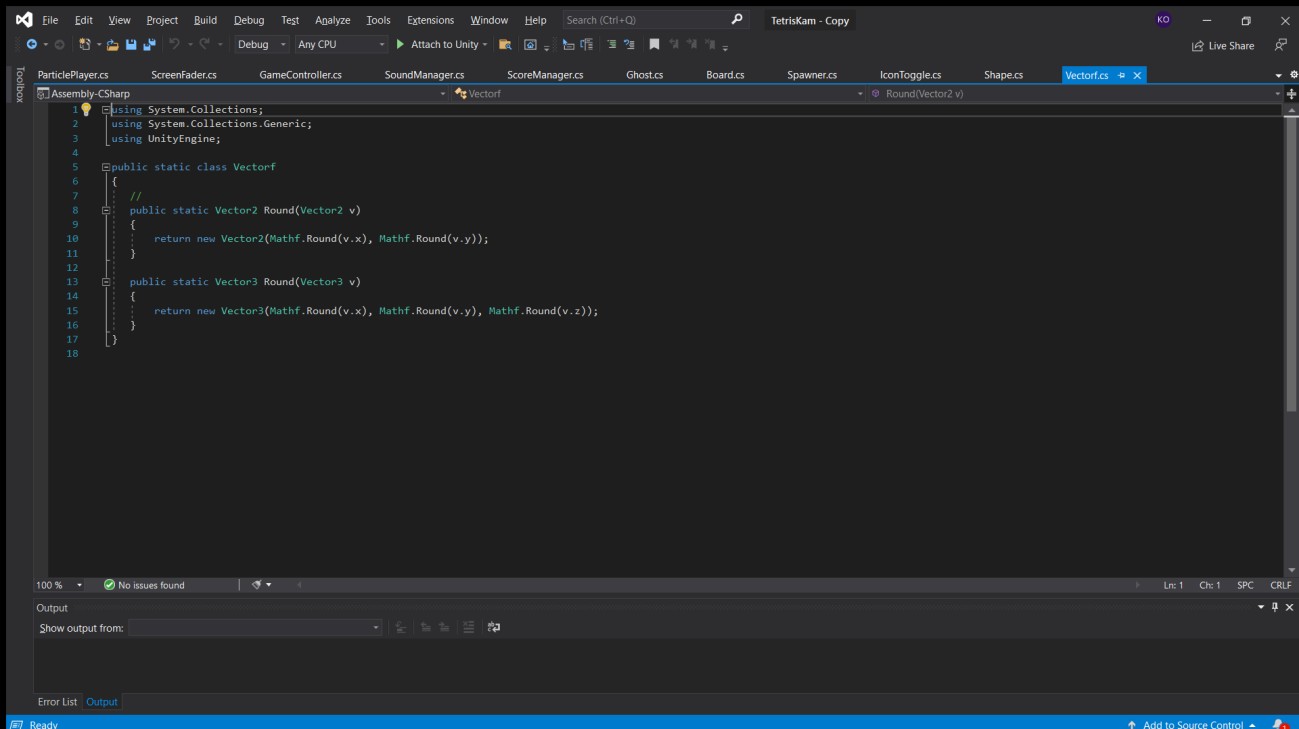




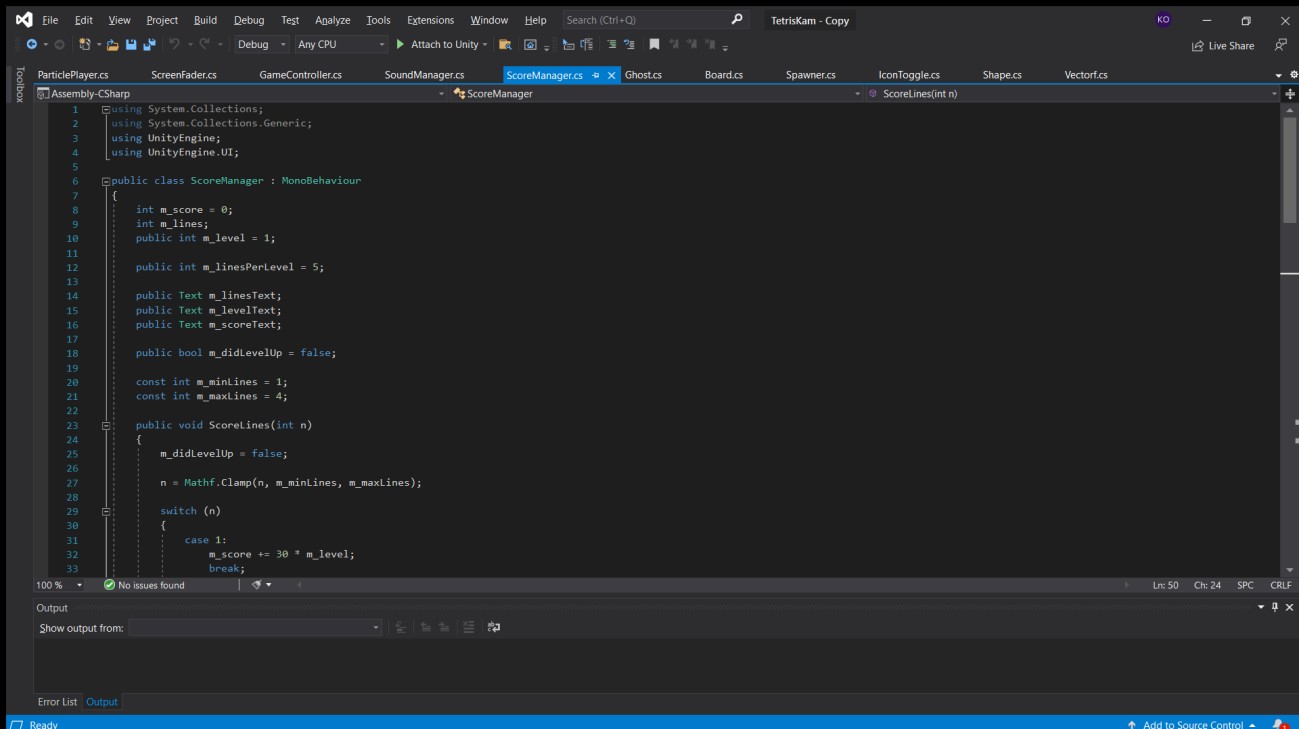
Spwaner class



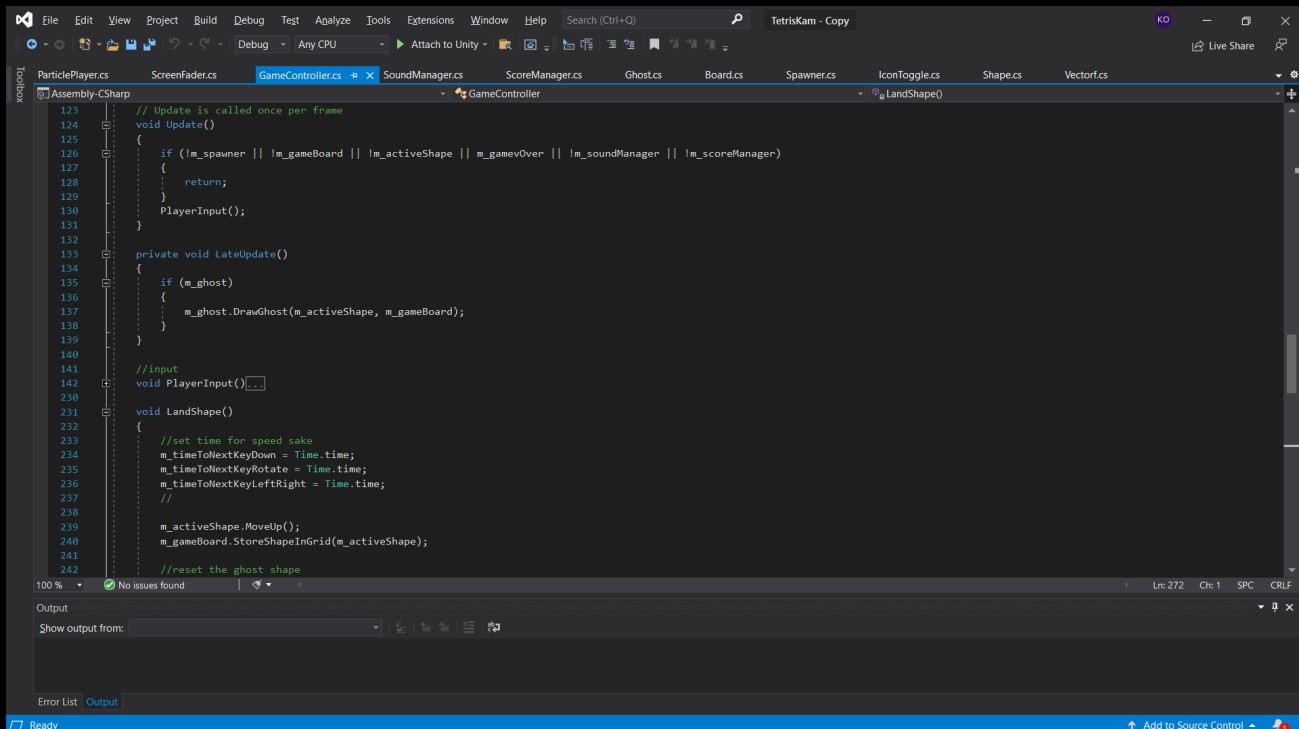
Shape class



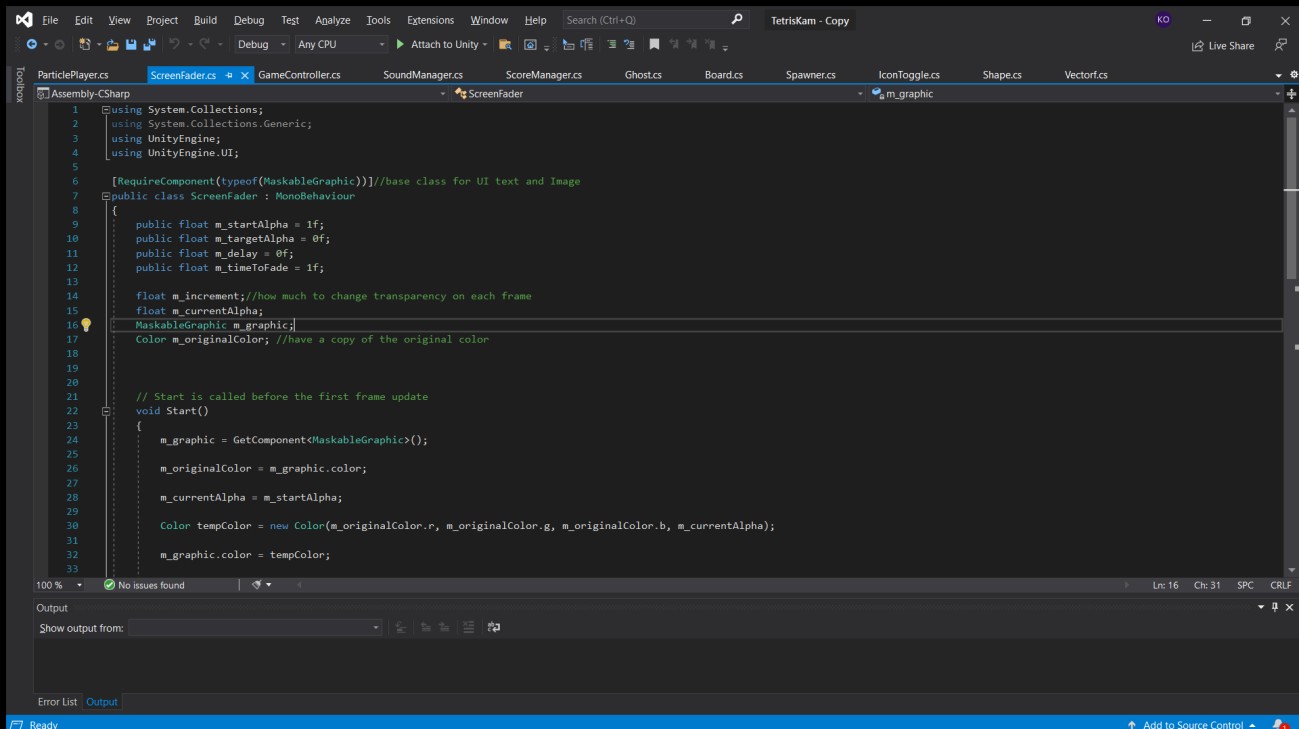
Vectorf class



ScoreManager class



GameController



ScreenFader class

## Appendix D – Test Cases

All tests were run within the engine rather than on a working game (this is why the test methods start with “Run from Play/Game Mode”); doing it within the engine enables for easier troubleshooting and highlighting of where the issues occurred.

**Table D1: Test case TC-001(UI Interaction)**

|  |  |
| --- | --- |
| Test Suit ID | R-100 |
| Test Case ID | TC-001 |
| Test Case Summary | Ensure that the basic UI worked as intended |
| Related Requirement | R-1\*\* |
| Prerequisite | The game must have UI Canvases set to Overlay and World space in the Game Engine before final build |
| Test Procedure | Run Game in Play Mode  Click the UI buttons to ensure all scripts and effects assigned work as intended |
| Test Data | From key presses vs output |
| Expected Result | The UI buttons work in correspondence to the Player Input scripts from the Game controller |
| Actual Result | The UI buttons work as expected (none AR version) |

|  |  |
| --- | --- |
| Status | Test succeeded |
| Remarks | Test succeeded; any future update can easily be altered from now |
| Creator | Kamachi Ojukwu |
| Last Date checked | May 27, 2021 |
| Executed by | Kamachi Ojukwu |
| Date of Execution | May 27, 2021 |
| Test Environment | Software: Unity Engine  Hardware: Zbook studio x360 xeon and Quadro p, Samsung galaxy tab s6 lite, |

**Table D2: Test case TC-002(Core Game Mechanics)**

|  |  |
| --- | --- |
| Test Suit ID | R-101 |
| Test Case ID | TC-002 |
| Test Case Summary | Ensure that the game mechanics work and given room for future upgrades and integration (i.e., time manipulation) |
| Related Requirement | R-1\*\* |
| Prerequisite | The game must have game objects instances ready in the Unity Engine and assigned scripts with appropriate algorithms and logic working in relation to the game object |
| Test Procedure | Run Game in Play Mode  Maneuver falling shapes to create complete rows of blocks. These are called lines and they disappear in exchange for points  If the shapes stack up the game board it’s GAME OVER  Move tetrominoes left and right using the directional keys. Press up to rotate  Use the down key to force drop or make the shapes drop faster |
| Test Data | From key presses vs output |

|  |  |
| --- | --- |
| Expected Result | The game should follow the rules and overall project goal in the mechanics aspect of gameplay |
| Actual Result | The Game mechanics work, it is a lot harder than the previous variations |
| Status | Test succeeded |
| Remarks | Test succeeded; any future update can easily be altered from now |
| Creator | Kamachi Ojukwu |
| Last Date checked | May 27, 2021 |
| Executed by | Kamachi Ojukwu |
| Date of Execution | May 27, 2021 |
| Test Environment | Software: Unity Engine  Hardware: Zbook studio x360 xeon and Quadro p, Samsung galaxy tab s6 lite, |

## Appendix E – User Guide/Manual

**Tutorials, Controls:**

* The descending shapes are called Terominoes. Once they land at the bottom of the board they stop moving, they become blocks
* Manoeuvre falling shapes to create complete rows of blocks. These are called lines and they disappear in exchange for points
* If the shapes stack up the game board it’s GAME OVER
* Move tetrominoes left and right using the directional keys. Press up to rotate
* Use the down key to force drop or make the shapes drop faster and harder
* Use the UI buttons to make configurations for gameplay such as rotate clockwise to counter clockwise.

## Appendix F – TURNITIN REPORT/SCORE

