Osborn, Ben

8659 | Geometric organiser

Project Report

OCR Computer Science H446

Contents

[Analysis 2](#_Toc128246683)

[Problem Outline 2](#_Toc128246684)

[Stakeholders 2](#_Toc128246685)

[Computational methods to solve the problem 2](#_Toc128246686)

[Why is it suited to a computational soloution 2](#_Toc128246687)

[Thinking abstractly 2](#_Toc128246688)

[Thinking ahead 3](#_Toc128246689)

[Thinking procedurally 3](#_Toc128246690)

[Control scheme 4](#_Toc128246691)

[Player count 4](#_Toc128246692)

[GUI menus 5](#_Toc128246693)

[Game states 7](#_Toc128246694)

[Thinking logically 7](#_Toc128246695)

[Thinking concurrently 7](#_Toc128246696)

[Conclusion 8](#_Toc128246697)

[Questionnaire 1 9](#_Toc128246698)

[Audio 9](#_Toc128246699)

[Graphical 10](#_Toc128246700)

[Multiplayer 11](#_Toc128246701)

[Gamemodes 12](#_Toc128246702)

[Progression system 13](#_Toc128246703)

[Summary for questionaire 1 16](#_Toc128246704)

[rESearch in to original tetris 17](#_Toc128246705)

[Features of the proposed soloution 22](#_Toc128246706)

[Design stage 24](#_Toc128246707)

[Decompose the problem 24](#_Toc128246708)

[Justification 26](#_Toc128246709)



# Analysis

## Problem Outline

Using Python 3.10.9, I am going to develop a version of the computer game Tetris. Tetris allows the user to control shaped blocks called Tetro Minos; the goal is to arrange the blocks into a complete line while the blocks fall towards the hard deck or land on top of the previous blocks. While avoiding the blocks stacking to the ceiling of the grid which ends the game. The player can rotate the pieces, move them sideways and drop them to the hard deck.

My version of the game will include a Two player version, a campaign system, options to personalise the game and a rewards system.

To achieve this, the game will require: a method to score the users game, a way to save data between playing sessions, menus to allow players to select game options, a method to allow to players to play together on one computer, profile page with stats/achievements.

## Stakeholders

The graphics in the game will be in 8-bit styling meaning it will not relate with real life. In addition, there won’t be any dialogue in the game; menus will have different language options to prevent language barriers becoming a limiting factor to my target audience, allowing the game to be internationally available. The games main control method is a keyboard/mouse making the game intuitive. Laptop users will be able to play the game ‘on the go’ at launch. The game could be easily adapted to be available on other platforms, as there aren’t many control inputs; this would make the game accessible to more people.

With the listed factors in mind, the game should be designed towards people between the range of 15 and 24. This is because this is ,statistically, the age which plays the most video games.

I have selected a user to personify my target audience, Fergus Young. He is a 17-year-old student whom I share my computer science class with. I have identified that he has a strong passion for games and has played the original Tetris, released in 1984. I will contact Fergus to help to develop my game.

## Computational methods to solve the problem

### Why is it suited to a computational soloution

The problem lends itself to computational methods of finding and implementing a solution for a variety of reasons. The problem is suited towards a keyboard/mouse input as well as the IDEs available on computer systems. Computational methods will allow me to break down the problem and target specific aspects of it independently of the rest.

### Thinking abstractly

(a) It is the process of removing excessive details to arrive at a representation of a problem that consists of only the key features. (b)Abstraction enables more efficient design during software development as programmers can focus only on elements that need to be built into the software. (c)Abstraction is a simplified representation of reality. Real-world entities may be represented using computational structures.

(d)Abstraction for ‘Geometric Organiser’:

* Restrict the game to two dimensions
* Add pixel art graphics style
* Add simple sound effects and game music to indicate details while playing the game
* Add a scoring system to give the players a goal
* Add a GUI menu system to let users change game options
* Add a progression system to give the player a sense of achievement
* Add user-changeable styling options to allow the player to personalise their copy of the game

### Thinking ahead

(a)The inputs/outputs for the game are displayed in the table below.

|  |  |
| --- | --- |
| Inputs | Outputs |
| keyboard used to control the game | Animations, sound effects and music to visualise the gameplay |
| In-game name entered via keyboard | The scoring system aims to give the players a goal |
| Mouse or keyboard used to navigate menus | The account level/campaign stage gives the players achievements |
| Menu options: Play, Profile, Quit game, Settings | If play selected: display different game mode options, difficulty options and single/multiplayer selector |
|  | If the profile menu is selected: display account level, achieved/locked accolades and cosmetic options |
|  | If quit game is selected: the game stops running first saving player data |
|  | If Settings is selected: audio and language options are displayed |

(b)The preconditions in order to program the solution are as follows:

* I plan to use a windows computer with Python 3.10.9 installed this is because I am familiar with python. However, Python isn’t great for game making so Pygame and pyGUI make up for the shortcomings.
* My IDE will be PyCharm as it has useful features to make programming more simple
* Mouse and keyboard program my game and to test it
* Audio output e.g. speakers or headphones this will allow me to test the in-game audio
* A Visual display unit to view my selected IDE which is PyCharm

(c)Caching is storing instructions or values in cache memory after they have been used, as they may be used again.

(d)Commonly used functions are often packaged into libraries for reuse.

## Thinking procedurally

This is going to break down the game into parts and then break those parts into smaller problems or ideas; I have used my chart to break down every part of the problem into smaller problems. By thinking in this way, the game has been broken down into 5 main parts. These are the [control scheme](#_Control_scheme), [player count](#_Player_count), [game modes](#_Gamemodes), [GUI menus](#_GUI_menus), and [game states](#_Game_states).

There is a large [Organisation chart](https://marchesmultiacademytrust-my.sharepoint.com/:w:/g/personal/osborn17b_mar_mmat_co_uk/EW670J_oLPpEpwkX67hQElMBl_3-Xl2-_t6qUxjMivvMHA) linked there. This is a wide breakdown of every factor in the design process of my game. I am going to continue to add to this during my project’s journey. So, this document will remain up to date.

### Control scheme

This section will be used to plan and consider what every input will perform about the game’s functions. I plan to have multiple different control schemes which will be switched through the “settings menu.” The player will be able to interact through menus with a mouse and or keyboard. I have designed the controls to be comparable to other existing games, hopefully making it intuitive for new players to pick up and learn. Also, this use of existing control schemes will allow experienced players to understand the game very quickly.

### Player count

The game will be playable with two different amounts of players: single players and two players. This will allow people to either play alone, share the experience with a friend, or challenge someone to a game. I hope this will make my game appear to a broader audience as local multiplayer can create a similarly fun experience to full online multiplayer games. I think this matters as large multiplayer games are the most popular games at the moment, so it appeals to many people who enjoy these multiplayer games.

The player count will have two options there will be a split screen option, where both users play simultaneously splitting the screen into two playing fields. Or another option is where “Player 1” will set a score within a set amount of time and then “Player 2” will attempt to beat that score. This could prove who is a more skilled player. This game is entirely skill-based, although, there is luck it’s controlled by a skilled player. The only luck is which geometric shape you get for your turn.

This option will be decided in the play section of the “play menu.”

Game modes are an integral part of the playing experience for my game. This is because they change the game in massive ways. Like they change the scoring methods, for example, you can either be racing against a clock or you could be aiming for a higher and higher score by completing lines in the “Geometric playing grid.” I think being able to change how the game changes are extremely important to keep a player’s attention. If a player becomes bored they can change the game mode and this will change the experience enough that it should be able to hold their attention for a longer amount of time.

**In all of the game modes, the games for either player ends, if the geometric shapes hit the top of the playing grid, meaning they have run out of space.**

I have split the game modes into several different options these being:

#### Versus

Versus mode is a two-player-only game mode. This mode splits the screen into two halves, the player then battles each other simultaneously to test who is the most skilled player within the set amount of time.

#### Time attack

Time attack is a single/two-player game mode.

In single-player mode, there is a timer with a small amount of time, the game ends when the timer hits zero. So, the players have to complete lines to gain extra time so they can play for longer and complete more rows of geometric shapes. This means it’s a competition for how long you can make the game last.

Two-player mode is the same game except it’s a competition for who can survive the longest. If one of the players runs out of time the game ends for both players. This would then crown the player who still had time left in their game the winner of that match.

#### Endless

This is a single-player-only mode.

Endless mode is a game where you are scored based on how many completed lines you can get without losing from running out of play space. As stated in this quote “In all of the game modes, the games for either player ends, if the geometric shapes hit the top of the playing grid, meaning they have run out of space.**”**

The game aims to set a “high score” which the player will try to best each time they play. This mode will give the player a clear metric of how they are improving in the game. As if they can suddenly beat their old high score with ease then they are more skilled than they were.

#### Campaign

This is a single or two-player game mode.

In this mode, there is a score set which you have to try and beat within a limited time, but the score will be ambitious to achieve within the time limit. This will be described as levels. So, if you beat the score within the time limit then the score achieved will increase, making the game more difficult the more levels you complete.

### GUI menus

The graphical user interface menus will be used to change between game modes, and settings and to play a game mode. They will be controlled by either a mouse or keyboard which will hopefully make them easy to navigate. The menu will be laid out through different sections, this will help to organise all the different options making it simpler to understand.

This is going to be a description of each different section:

#### Settings

There will be sound options, control options and language options. The settings options will make the game appear to more people and make it more customisable to a broader audience.

* Sound will be used to control the sound effects volume and whether they are muted. Having control over this will hopefully allow the player to tailor the gameplay more to their preferences.
* Colour options will be used to define the menu colours, the colours of playing pieces and the colour of interfaces around the screen. This is important as it allows the user to customise the game more to their preferences.
* Language options will help broaden my game market to more people. It will allow people from different backgrounds and those who speak different languages will be able to navigate the menus and understand all of the interface options.

#### Playing menu

Through this menu, you will be able to change the difficulty of the game modes, and select player count and difficulty options. Having all of the options to play a game should make it intuitive for the player base.

Difficulty options

The difficulty options will change different parts of the game depending on what game mode you are playing. After you click the play button, you will pick a difficulty, which will determine the challenge of each game differently depending on whichever game mode you choose after this option. You will change I have decided to name the difficulties of universal terms from gaming communities, which will make the options obvious.

The three different difficulty options:

* Easy
* Normal
* Hard

Game mode options

After the difficulty screen, you pick will between the four game modes:

* Versus
* Time attack
* Endless
* Campaign

There will be a brief description of the game mode below the four options and then you will go through to the play section.

Player count options

There will only be two options.

* Single player
* Two players

This option determines how many players are playing the game at once, the effect of these options will vary between the different game modes.

#### Pause menu

There will be four different options from this menu: Resume, Restart, settings and return to the main menu.

Resume

This option will simply resume back to gameplay.

REstart

The reset option will reset the currently played game, back to the start.

Settings

This option will link you to an overlay of the settings menu. This will not interrupt the played game as it will be paused in the background. The changes made within the settings will take effect during the game.

Return to the main menu

Will end the currently played game without saving any data and then send you back to the main starting menu.

### Game states

Game states break the game down into different running conditions.

The main game will have three different playing states:

#### GAME PAUSED

Wil freezes the currently played game in time, allowing a user to take a brief while mid-way through the game.

#### Game running

Game running is the state the game will be in when a user is playing the game.

#### Game over

The state the game is in after a player has lost or won his match. This state will tell you if you have won the game if that applies to the currently played game mode.

## Thinking logically

This section will help me consider the logic of my program so I can understand what actions may branch to different game states or update the score.

In my game, a constant iteration will be run in the background to verify if any of the conditions have been met to alter the score. The condition which is universal between all of the game modes is that if a row of shapes is completed then it will increase the score. However, depending on the game mode the ways of scoring may alter. I will discuss the game modes specifically later in this document.

Another constant iteration will be checking if the game pieces have reached the ceiling of the playing grid. If this was true then the game state would change to the game over.

I also need to decide what conditions will cause the game state to change. But this will vary identically to the scoring system. Game paused will be universal around every game mode though. The game state will define which branches and conditions are running at a given time.

## Thinking concurrently

This will be useful to make my solution to my game problems more efficient. The game will be updating the score or ti

## Conclusion

Through this section, I have demonstrated that computational methods are applicable methods to break down my problem. This breaking down into sections will give my design process structure to ensure that I don’t forget features that I wanted to add to my program. It will also ensure that my code is kept to a high level as I can consider each area of my code independently of the other areas of code.

## Questionnaire 1

After reviewing my game’s concept, I have concluded that I need to gather public opinion. I need feedback on ideas for game features. As the target market for my game is broad, I will gather a large amount of data from varying age groups and interests. To make the surveying process efficient I will use “Microsoft Forms” as it is an efficient method for collecting responses and it is intuitive to make a professional-looking survey.

A plan for the interview can be viewed below:



The results of this survey can be broken into several sections:

1. Audio
2. Graphical
3. Multiplayer
4. Game modes
5. Progression system

Within each section, I will present/analyse the data relating to the heading and I will conclude with features and opinions to guide my design process further. I will further research each topic to decide the implementation and viability of each concept.

This survey is not inclusive and it’s likely as this project progresses further, I will need to interview about different topics.

Age range

I need to preface the following data with information for the age ranges from the findings.

As you can see most participants are aged below 18, although, there is a small (proportionally) amount of people from age groups higher than that.

### Audio

I test-played some games in a similar genre and I noticed a theme between all of them. All of them include some kind of sound effects or game music. As of this, I included some questions to investigate whether it’s valid to include audio and what form the player base would like this in.

,

After collating the data, I could quickly see that the respondents agree with every point from above; I will carry every concept through to the design stage where I’ll discuss how to properly execute the ideas.

### Graphical

I decided that the graphical style of the game is a personal preference. So, I couldn’t make the decision alone and thus it has been included in the survey

As you can see 76% of people interviewed believed that the game’s background should change. This shows how most of the target market believes the background while playing the game should change.

As of this, I will include backgrounds that dynamically update with your gameplay. The execution of this concept will be explored further during the design stage.

The results show that exactly fifty per cent of respondents think the game should be drawn in a pixel art game style. This is the same style as the game I’m abstracting from ‘Tetris’ so my game may have similar art styles. In my research for a game, I found that a game must be unique to make a user want to play. This style should be used for all menus and gameplay elements. So, I will find ways in which to alter the graphics style to be engaging and interesting while also being a new design. I will explore this during the design stage.

### Multiplayer

I had the idea of adapting my initial game idea to be suitable for multiplayer. I wasn’t sure of the best way to implement this so I have included two survey questions to find out how others think that a multiplayer mode should work.

The only viable method I could think of to include multiplayer was by split screen. This is where you divide the screen into two halves and each player has a side of the screen on which they can play. I didn’t consider online or local multiplayer to be in the scope of this project. Online multiplayer requires a game server which is costly and impractical for a non-commercialised game. However, for local multiplayer I couldn’t find a method to implement it into a python program; I don’t believe it is worth the time investment as it wouldn’t ‘make or break’ the game. The graph shows that sixty per cent of people believe the best way to have a multiplayer system is by ‘Pitting players against each other.’ This will be the method that most multiplayer aspects will be handled.

However, I think working together still has value. In future questions, I introduce the concept of a campaign. A co-op campaign could enhance the campaign experience. A campaign in which players are against each other is not a common game idea. Which could help to make my game unique; I haven’t thought of an idea to make the campaign against another player while still following the campaign’s storyline. I will explore both a cooperative campaign and a versus during the design stage.

This graph clearly shows that the target market wants difficulty options while playing multiplayer. I will discuss what the difficulty options will specifically change about the gameplay during the design stage.

### Gamemodes

This section of my survey is brief. This is because I had limited options for the type of game modes I could think to implement for the main gameplay of the game. As of this most of the game mode ideas I accepted as being essential to the game as I couldn’t think of better viable alternatives.

However, I thought of implementing a campaign mode into the game. I wanted to understand what kind of reception this would get and a brief idea of how respondents think this should be implemented.

The graph to the left shows that most respondents believe that a campaign should be included in the game. I consider this good news as I think a campaign has a lot of potential to be a great addition to my game.

I wanted to explore the initial idea I had for the goal of the campaign system. This can be seen in the pole furthest to the right. The respondents agree that there should be a main ‘villain’ who has accomplices and that defeating them should be the target of the campaign game. I will explore designs for the campaign system and villain concept during the design stage.

I also wanted opinions on the gameplay for the campaign system. Most voted for ‘The speed of gameplay increases as you progress through the campaign’ out of the two options. This will be the win/progression system used for campaigns’ gameplay,

The graph shows that three-quarters of respondents believe that the difficulty of the game should increase over time. This means that while playing in endless mode, everyone will start at the same speed and then whoever can cope at the fastest speed will survive the longest and thus achieve the highest score. I will explore how this works during the design stage.

### Progression system

I researched what basic features make a good game. Both replayability and a goal were reoccurring results. I think a progression system is a good way of including both of these. My progression system was formulated with a levelling system, accolades system and unlockable cosmetics. In this section, I ask for opinions on the listed concepts.

Accolades

An accolade is an award granted as an acknowledgement of merit. In my game, I plan to have them as rewards which will act as a milestone. For example, if you achieve a very high score then you will receive a badge which can be used to display the achievement.

Firstly, I wanted to establish whether the respondents would agree with the idea to include accolades.

The graph shows that the high majority feel accolades would motivate them to play the game. This means I will continue this feature concept through to the design stage.

This will be a good way to encourage players to come back to the game and develop their skills further

Secondly, I wanted to understand whether the respondents feel that the badges should increase account level. As this could encourage people to complete these accolades to reach an inflated account level.

The graph shows sixty-five per cent of users feel that this accolade system should increase account level. I will explore by what margin each accolade will increase account level during the design stage.

Thirdly, I was interested in whether respondents feel that these accolades should be achievable by a beginner player or just more advanced players. This is because I wasn’t sure whether respondents would feel that they should be exclusive rewards for the people trying to be better at the game.

The graph shows six-tenths feel that they should be achievable by beginner players. As of this, I will design my accolades to cater to skilled and lesser-skilled players.

Account levelling

I wanted to test if respondents would feel there is value in an account levelling system and a simple requirement of this.

The graph shows that most respondents agree that a levelling system would add value to the game.

I will explain the design of the levelling system during the design stage and what requirements are needed to increase the tracker.

The graph shows that respondents believe that the account level should continue infinitely. This means that the game could truly never be completed which adds replayability to the game.

Cosmetics

The third idea for the progression system was cosmetic elements.

The graph shows that users believe that cosmetics would be a good addition to a progression system.

I will explore specific cosmetic items during the design stage.

I wanted to establish what cosmetic items people would be interested in. The graph shows that the respondents would be happy with all of the options above. None of the options received a negative response. As of this, I will explore the implementation of avatar styling, new game backgrounds, profile customisation and alternative patterns for playing pieces during the design stage.

With these positive responses I needed to establish whether the users would enjoy the implementation of an avatar system. The main purpose of this would be to add an out put for the cosmetic ideas I have.

The graph shows that over three quarters of respondents believe that a playable avatar would be a good feature. I will explore how an avatar will integrate into the gameplay during the design stage as well as how to integrate them with cosmetic elements.

### Summary for questionaire 1

I have collated the results from the questionnaire into a table, this will help me to effectively follow the guidelines given by this questionnaire.

|  |  |
| --- | --- |
| To be explored further | Justification |
| Game music that responds to the gameplay | Most respondents agree |
| Sound effects | Most respondents agree |
| Volume slider menu | Most respondents agree |
| Dynamically updating backgrounds | Most respondents agree |
| The game should employ a pixel art style | Most respondents agree |
| Difficulty options while playing multiplayer modes | Most respondents agree |
| Preferably multiplayer where players are against each other. Or working with each other | Both methods of multiplayer have potential value in different areas |
| A campaign system with a villain and gameplay speed increasing parallel to level count. | Most respondents agree |
| The difficulty of endless mode should increase over time | Most respondents agree |
| Accolades that increase account level | Most respondents agree |
| An infinite account levelling system | Most respondents agree |
| Cosmetics for avatar styling, new game backgrounds, profile customisation and alternative patterns for playing pieces | Most respondents agree |
| A playable and customisable avatar | Most respondents agree |

## rESearch in to original tetris



Push start informs what button is to be pressed in order to start the game.

The title screen for the game shows the colour scheme and style of the gameplay



A scoreboard of each players score. It orders the players with the highest scores and displays what difficulty they achieved this on.

Level allows you to pick the speed of the game

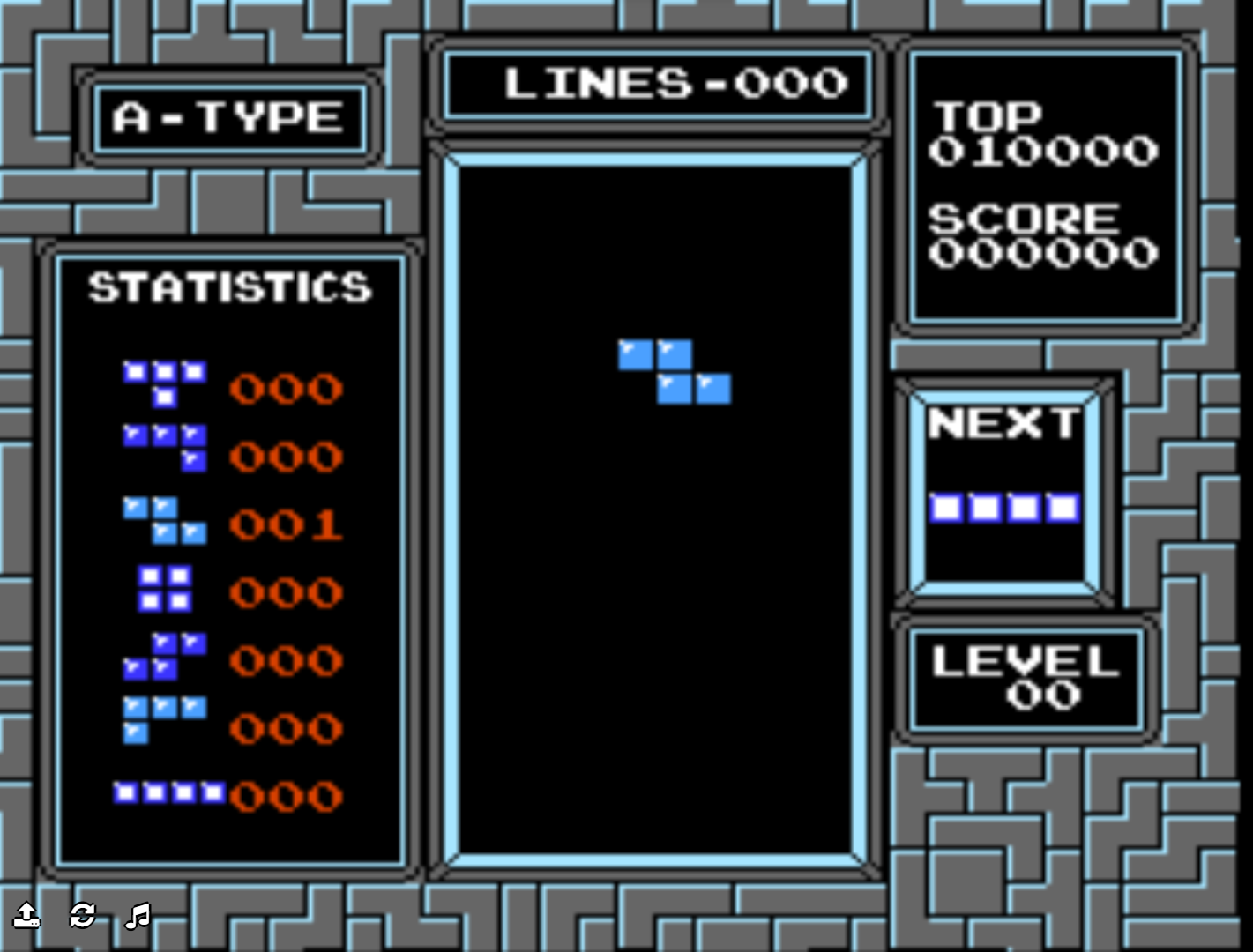
This section allows you to pick between game type a and type b

Allows you to select which type of music for the game to use

Shows how many completed lines by the current playing player

Shows the game mode which is currently being played

Top shows the highest score achieved by any players



Playing pieces which the player can rotate move left and right and down

Score shows the current score achieved by the player.

Next shows the next piece after you place the last one

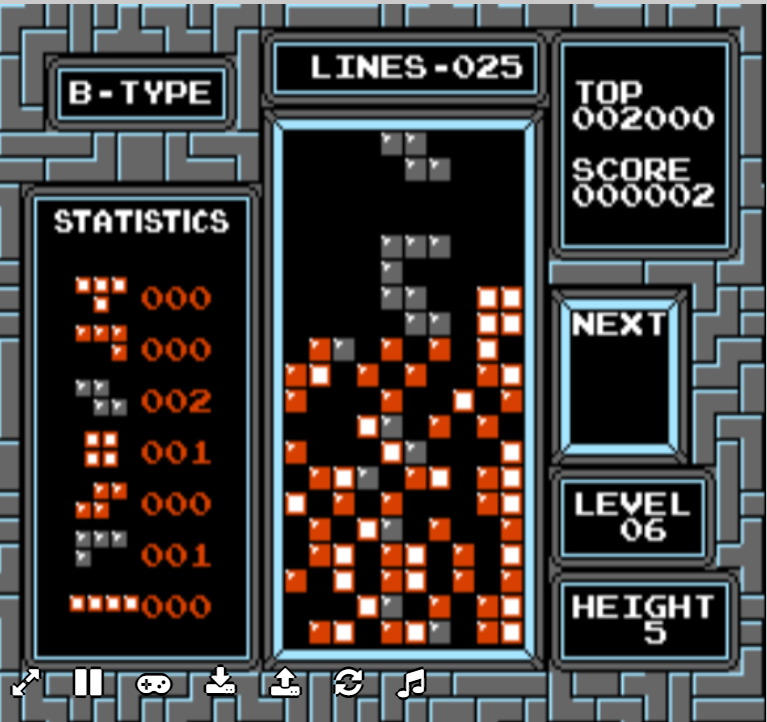
Level displays the difficulty of the game

Statistics shows the number of each pieces which you have been given

The playing grid where the pieces gradually moved down and stop the bottom



A scoreboard of each players score. It orders the players with the highest scores and displays what difficulty they achieved this on.



Adjusts the level of pieces placed at the start in order to make the game more difficult.

Adjusts the speed that the playing pieces move towards the ground.

Shows game type-B has been selected

A picture containing text, scoreboard

Description automatically generated

The decoy pieces in red/grey which are placed depending on the height option

Top shows the highest score achieved by any players

Playing pieces which the player can rotate move left and right and down

The playing grid where the pieces gradually moved down and stop the bottom

The selected height of decoy pieces on the playing grid

Level displays the difficulty of the game

Next shows the next piece after you place the last one

Score shows the current score achieved by the player.

This number counts down with every completed line. Once it reaches zero the game ends

Shows the game mode which is currently being played

Statistics shows the number of each pieces which you have been given



A simple pause screen activated by pressing escape

## Features of the proposed soloution

|  |  |
| --- | --- |
| Menus | Menus will be used to select the game options and view progression |
| Accolades | Accolades will provide progression for advanced players and small goals for starting players to give them an idea of their skill level |
| Level | Leveling will be used to track playtime and the skill of the player and also provide the player with a goal to work towards |
| Next piece queue | Shows the playing pieces that the player will receive next so that they can plan their current move with this knowledge. Will be mainly for more skilled players |
| Hold piece box | A tactical option for skilled players which allows them to store a single piece swapping it with the next piece in the queue. |
| Survival timer/completed lines display | Calculates the amount of lines completed per minute by the player. Based off how long they’ve played and how many lines have been completed |
| Completed lines | Displays the number of lines which have been completed by the player |
| Campaign/endless game mode | Multiple game modes should help keep the player interested as there will be multiple ways to play |
| Cosmetic items | Allows the player to personalise the game by changing colours and backgrounds |
| Sound effects/music | Will add tension to the game by adjusting to the situation of the gameplay |
| Personal best | Act as a tracker so that the players can see how they’ve improved in ass their personal best increases |
| Game paused with countdown | A pause button will stop the game completely. This will add flexibility to the user allowing them to come back if they are interrupted. A countdown allows the players to adjust back to the game before the high-octane gameplay starts again |
| As the game continues the pieces fall faster | This adds difficulty as the game plays as an easy game that won’t hold the player’s attention. It also means the players can use the survival timer to track how they’ve improved as they survive longer and longer. |
| Pixel art | This will be the graphical style for the it is simple and should complement the design of the game |

# Design stage

## Pre-Alpha testing

My game requires a difficulty system for the players of differing skill levels. In order to get the levels accurate to the players skill level I am going to conduct a pre-alpha test. I will measure the skill level of players on a prototype of my game that I have made. This data will then define the level for Easy, Normal and Hard difficulties.

Easy will be the lowest quartile of player achieved scores.

Normal will be the Interquartile range between player achieved scores.

Hard will be the Upper Quartile of player achieved scores.

To make the testing accurate: I will first explain how to play the game and the goal of the game. They will then have 5 minutes to get used to the game; the next 5 minutes I will calculate the median score of their results.

## Decompose the problem

I am using a top-down diagram as it will help me to identify all the parts of the problem. It will keep every stage independent and help me to more efficiently design my game.

### Justification

#### Startup

Method that initialises on startup of program.

* **Displays games title screen**
* **Menu music**

Initialises the games’ menu music

* **Pick playing profile**

Allows player to change between saved profiles.

* **Start Button**

When pressed changes game state to Main menu.

#### Game over

Runs after the game has ended.

* **Game result**

Displays result of validation for game won/lost

* **XP reward calculated/displayed**

Calculates the amount of XP based off of playing performance

* + **Game summary**

Displays tracked scores from the game

#### Main menu

Used to link option for the game

* **How to play section**
  + **Explanation for goal of each game mode**

A guide to explain how to play each game mode, the aim and also some tactics

* + **Control scheme diagram**

A diagram which shows the user what every button does in relation to the game

* + **Back to main menu**

Returns the player to the main menu

* **Play**

Pick options which affect the specific game mode

* + **Player quantity**

Allows you to pick between one player and two player gaming

* + **Pick gamemode**

Lets you pick between campaign and Endless mode

* + **select difficulty**

Lets you change the difficulty of the game, applying to both campaign and endless mode.

* + **Back to main menu**

Returns you to main menu without saving options made

* + **Start game**

Changing the game state to playing while inputting values from all the variables in the ‘play’ method

* **settings**
  + **Audio options**

Adjust the music/sound effects volume level while muting both with master control

* + **Back to main menu**

Changes game state to menu while inputting values from variables that were inputted to

* **Profile**
  + **Locker**

Allow the user to change their cosmetic elements

* + - **Validation for unlocked items**

Determine which cosmetics have been unlocked determining which ones the player can use

* + **Set username**

Lets the user identify their profile by giving them an identifer

* + - **Validation against other profile names**

Used to ensure that their profile name is unique

* + **View unlocked and locked accolade**

Display the achievements of the players and the ones which are left to be unlocked with how to unlock them

* + **View account level**

Displays the players level with a progress meter to show how far they are from the next stage

* + - **Next reward**

Shows the player what they will unlock when they go up a level

* **Back to main menu**

A menu button that returns the player to the menu state

* + - **Save or quit without saving**

Lets the player decide whether they want to save their progress or not

#### Game playing

* **IF campaign:**
  + **Level generation**
    - **Score per level method**

Uses the difficulty setting to generate levels with gradually increasing difficulty between each level.

* + - **Unique background generator**

Algorithm to apply a different background for the level which is being played

* + **Save state**

This will act as management for saved progress. This will be linked to the players in-game account. The save will store the level which the player is on, the scores achieved and their unique generation of the campaign.

* + - **Load progress**

Loads the save state that the player selects

* + - **Delete save**

Deletes the save that has been made the player

* + - **Create save**

Creates a new save and allows the player to name it. Saves are specific to the players profile

* + **World map**

The world map shows the player which level they are on and all of the levels ahead of them. Allowing them to replay past levels and achieve a better score.

* + - **Previous scores**

Displays the scores achieved by the players on each individual level

* + - **Progress meter**

A bar which shows how far the player is towards the final level

* + - **Back to main menu**

Returns the player to the main menu

* + - * **Save or quit without saving**

Questions whether the user wants to save their progress or not

* **Endless/campaign**

These are features of the running game that apply to both game modes

* + **Difficulty**

It allows the user to adjust the games difficulty to their skill level

* + **Controls**

The inputs the player will have during the game

* + - **Game piece movement** 
      * **Rotate right**

The ‘E’ key activates the ‘rotate\_right’ function

* + - * **Rotate left**

The ‘Q’ key activates the ‘rotate\_left’ function

* + - **Pause game**

The ‘ESC’ key will change the game state to pause

* + **Scoring**

Metrics of the gameplay that will be recorded while the user is playing

* + - **Completed lines**

The value of ‘linesCompleted’ variable will be showed on the gameplay screen. ‘linesCompleted’ is the number of completed rows per a round.

* + - **Lines completed per minute**

Calculation performed with the ‘timer’ and ‘linesCompleted’ variable to calculate how many completed lines per length of time. Which Is the value of the ‘linePerMin’ variable.

* + - **Game round timer**

‘timer’ variable which starts when the game is run and pauses with pause state.

* + **Game info**
    - **Next 3 pieces displayed**

A display for the user of the three next shaped pieces they will receive

* + - * **Sequence of pieces**

‘imageSeq’ variable which records ‘image’ value of the next three to be outputted.

* + - **Held piece**

Copies the value of the ‘held’ variable with the ‘image’ variable.

* + **Audio**
    - **Sound effects**
      * **Game over sound**
      * **Line completed sound**
    - **Music**
      * **Increases in speed as the game does**
      * **Changes if game has been won**
  + **Graphics (8-bit styling)**

#### Game paused

* + **Return to main menu**

Returns to main menu without saving the games progress

* + **Restart**

Restarts the game being played

* + **Resume**

Returns to the currently paused game

## Algorithims

### Settings

Diagram

Description automatically generated

Diagram

Description automatically generated

Diagram

Description automatically generatedDiagram

Description automatically generatedDiagram

Description automatically generated

### game over

### Diagram Description automatically generatedDiagram Description automatically generated

#### Xp reward

Diagram, schematic

Description automatically generated

NOTES

https://coolors.co/a6311d-cb3c23-ff4b3e-ff6639-ff7b39-ffad43-ffe759