|  |
| --- |
|  |
| Ultimate Tic-Tac-Toe |
| H446-03 Computer Science Programming Project |
|  |
| **Oliver Greaves** |
| **7/9/2018** |

Contents:

[Analysis 2](#_Toc518908252)

[Introduction to the problem: 2](#_Toc518908253)

[Justification of suitability for computational methods: 2](#_Toc518908254)

[Stakeholders: 3](#_Toc518908255)

[George Honeywood – computing student: 3](#_Toc518908256)

[Persona – middle aged adult: 4](#_Toc518908257)

[Persona – retired lady: 5](#_Toc518908258)

[How will my solution meet the needs of these stakeholders? **Error! Bookmark not defined.**](#_Toc518908259)

[Research into existing solutions: 5](#_Toc518908260)

[Website Implementation (www.bejofo.net/ttt): 5](#_Toc518908261)

[Mobile Phone Application (made by Magma Mobile): 11](#_Toc518908262)

[Ultimate Tic-Tac-Toe on Steam (made by Tigerish Games): 16](#_Toc518908263)

[Essential features of the proposed solution: 24](#_Toc518908264)

[Limitations of the proposed solution: 25](#_Toc518908265)

[Requirements specification: 26](#_Toc518908266)

[Hardware requirements: 26](#_Toc518908267)

[Software requirements: 26](#_Toc518908268)

[Specific requirements and justification: 26](#_Toc518908269)

[Measurable success criteria: 28](#_Toc518908270)

[Bibliography 30](#_Toc518908271)

# Analysis

## Introduction to the problem:

The program I am going to create is my take on a game called ‘ultimate tic-tac-toe’, also known as ‘meta tic-tac-toe’ or ‘tic-tac-tic-tac-toe-toe’. Ultimate tic-tac-toe is a game that builds upon the very simple idea of traditional tic-tac-toe, and introduces several new components to make the game much more difficult and dependent on strategy. My program is going to be a user-friendly implementation of this game with the aim of reducing confusion, so that players can more easily enjoy the game. The program will include:

* Two modes: a ‘classic’ mode, which is the original tic-tac-toe in a 3x3 grid; and an ‘ultimate’ mode. Each mode will contain:
  + A local multiplayer mode (two players)
  + A single-player mode against the computer, with several levels of computer AI (artificial intelligence) difficulty
  + A capacity for undoing moves
* A clean user interface that allows the user to:
  + Navigate around the game’s different menus and modes
  + Customise the interface, and change the appearance and colours of the player symbols
  + View archives of previous games, and watch them being replayed

## Justification of suitability for computational methods:

There are several features that I plan to include in my solution that are only achievable by using computational methods:

* **Computer as an opponent:** In order for a single-player mode to work, there will need to be a computer for the user to play against. Whether the automatic opponent just moves randomly or it uses some degree of artificial intelligence, this can only be achieved by using a computer to perform these calculations.
* **Game as a piece of software, instead of physical:** Ultimate tic-tac-toe can be played in ‘real life’ using simply a pen and some paper. Recreating the game in software has many benefits including:
  + Not having to draw a new grid for each game. This saves paper and time, and also avoids having a grid that is messy or too small, which could happen if it was drawn by hand.
  + Adding extra tips and pointers. In a physical pen-and-paper game, the players have to remember their previous move and if they forget the whole game is essentially ruined. In software this could be tracked automatically and the game could provide hints on where the user is allowed to make their next move. The game could also have an in-built set of rules so that players don’t have to search for them elsewhere.
  + Move validation. Stopping an illegal move in a physical game requires one of the players to notice it. In a computer version of the game the software could stop invalid moves so that the rules are always followed. It would also stop a player from trying to cheat.
* **Analytical calculations:** It would be tedious to manually work out the win percentages and write down results from every game – a computer can do this automatically and present the information in a friendly format.
* **Achievements:** The computer could track statistics such as time played, moves made and games won in order to count these towards achievements. The user would not have to worry about keeping track of this since the computer is doing it automatically. These achievements would give an extra goal for players to work towards, making the game more enjoyable than if it were just played with pen and paper.
* **Animations:** To make a visually pleasing game I will need to make use of graphics and sound. This can only be done using a computer.

## Stakeholders:

In order for my program to be enjoyed by all users, it must cater for everyone’s needs. The three main stakeholders in my game are people from different age groups to reflect potential users from several different groups and backgrounds.

### George Honeywood – computing student:

George is a fellow student in my computing class. He has a busy lifestyle studying four A-levels and balancing other commitments with his school work.

#### How will this stakeholder make use of my solution?

George wants to use my solution as a way to take a break from his studies. He wants a game that he can pick up and play quickly, but also use for longer periods of time if he feels like it. He would like the ability to play against friends who are in his class, especially during the lunch hour at school.

#### Why is my program appropriate to this stakeholder’s needs?

The ‘game’ aspect of my solution means that it will feel fun and challenging to play, and the ‘puzzle’ element means that he won’t completely switch his brain off while playing. This means that he will be able to return to his studies at any moment while still in a mental state to do so. However for my game to be appropriate it will be user-friendly to reduce any chance of frustration, which could then go on to affect George’s studies.

#### Individual requirements:

* Well-explained and easy to find set of rules that can be understood quickly and easily
* Clearly labelled menus that can be easily navigated around
* Bright and attractive theme, sounds and animations
* Simple design that reduces confusion
* Smooth and simple control system to avoid potential frustration

### Persona – middle aged adult:

This woman has a busy schedule with work and looking after her son and would like a game that she can pick up and play any time she has a few spare minutes.

#### How will this stakeholder make use of my solution?

Similarly to George, this persona would use my solution as a form of recreation. However, she can get quite competitive and would most likely challenge people to a quick match every day. She would also like to teach her son logic skills through the game, even though he is quite young.

#### Why is my program appropriate to this stakeholder’s needs?

My program will contain a multiplayer mode to allow the persona to challenge friends, and a simple rule set so she can start a game against a beginner easily and quickly. It will also contain several levels of computer difficulty, including an easy mode that her son could use to learn the game.

#### Individual requirements:

* Clean-looking game that is friendly to the user
* Easy to begin a game quickly – games can be saved and resumed later if she needs to suddenly stop playing
* Controls need to be easy to pick up and use
* The game needs to make it obvious what is happening but without too much text which could slow the pace of the game
* Simple multiplayer which allows her to challenge friends or colleagues to a game without having to spend much time explaining the rules

### Persona – retired lady:

This lady is happily retired and has a lot of free time since she no longer works. Even though she is not the most technologically adept, she enjoys spending around an hour each morning playing games to keep her brain active.

#### How will this stakeholder make use of my solution?

This lady will likely use my game most mornings, and keep returning to it over a long period of time. She would like to have the choice of playing against the computer, or against a friend. The game will also be a challenge for her and she will keep returning to it until she has mastered the strategy.

#### Why is my program suitable to this stakeholder’s needs?

This persona will be able to start from a beginner’s level and progress at her own rate, due to the multiple levels of computer difficulty which I will add to the single-player mode. In addition, my game will have an achievements system which increases the longevity and playability. This means that this lady can return to the game many times and there is always a goal she can work towards while playing.

#### Individual requirements:

* Simple, easy to navigate interface
* Several modes that offer variety, to stop the game becoming repetitive and boring
* In-built achievements that she can work towards
* Game statistics such as play time and win percentage stored which can be seen from within the game
* Several levels of AI that help her to improve at the game over time.

### Summary of requirements gathered from stakeholders:

Since my solution will be a game and less of a utility application, its nature means that the features people want are quite open. My stakeholders are all from different ages so my main aim should be to ensure that my game can be played by anyone, as little or as much as they like. Therefore it will be easy to start playing straight away by including easy menu navigation and a concise explanation of the rules, but also include elements of progression such as achievements to make it more fun to play in the long-term.

## Research into existing solutions:

### Website Implementation (www.bejofo.net/ttt):

This website is an online version of the game, that lets two players play against each other, each with their own device. It was made by Yannick Rietz in 2013 as a programming project in their spare time.

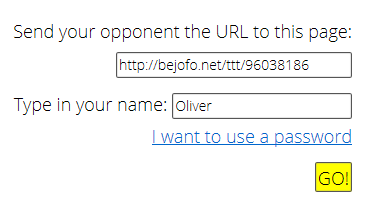


Figure 1 – starting a match

To start a game, one user must click the ‘Start game’ link, which takes them to a menu, as shown in Figure 1. The player can enter a username and can then send the URL to the other player, which lets them join the same game. This URL system comes with a handy bonus – as long as players retain the link to the game, they can stop half way through playing and return to the same game later, by connecting to the same link. However, this system does have its drawbacks.

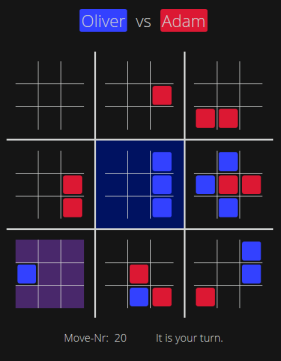


Figure 2 – a typical game

Firstly, it can be tedious to read out an 8-digit number every time you want a friend to play with you. Secondly, it isn’t very practical having to save the URL to come back to later, and when the two players do decide to return, they may reconnect in the wrong order and be assigned to the wrong colour. Having said that, it is a useful system and the ability to return to these save states is an advantage of this solution, since most implementations of this game do not have this capacity built in. An online capability would be outside of the scope of my project, because it adds unnecessary complexity and my knowledge of this field of computing is limited.

The main advantage of this particular solution is the simplicity of the user interface. The playing grid is clearly laid out and the counters are simply squares, which make them easy to see and process mentally. This makes it easier to process what is going on in the game, especially for new players who may struggle to pick up the complicated rules.

When it is your move, the possible squares that you can place a counter in are highlighted in a different colour (shown in purple, in the bottom-left grid in Figure 2). I particularly like this feature as it helps the player to evaluate their options when making their next move. As the game progresses, whole squares are won by the players, and they are set to the corresponding player’s colour. This keeps the design of the board clean, something that I will aim to do with my own program.

Another feature of this website is the ability to choose between several themes. This allows users to customise the game, and also allows potentially colour blind users to find a theme where they can differentiate between the two colours (this is necessary since the counters are all the same shape). I would like to implement a method to cater for colour blind users in my solution.

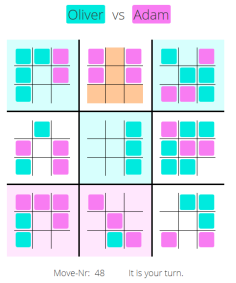


Figure 3 – different themes

Figure 4 – different themes

­­­­After the game has been won, the text updates accordingly to tell both users the outcome. I particularly like the way the board is left on screen, so users can see how the game was won. There is a ‘Rematch’ option that appears in place of the move status, which is very useful because a new game can be started with the same opponent straight away, without having to go through the URL process beforehand. I hope to incorporate my own rematch system for all of the modes in my program.

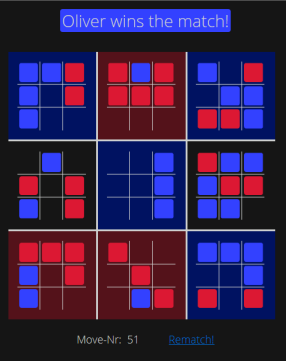


Figure 5 – a finished match

**­­­­**However, this implementation does have its drawbacks. My first wish for this program is a single-player mode that lets the user play against the computer, because this game cannot be played without an opponent (and practically there is not always a friend who is willing to play). Single-player practice could help users to improve at the game without the need of a friend, who may be at a completely different skill level. I would like to add different levels of difficulty to the single-player mode in my solution so that a person at any skill level can still practice and hone their skills.

Another limit to this website is that there is no local multiplayer mode, so in order for the game to be played, two separate devices are needed, and both need an internet connection. There are a lot of criteria that need to be fulfilled in order to play the game, and this reduces the potential player base drastically. My own solution will solve this problem by providing a mode that lets two people play a game by sharing one device, without the need for an internet connection.

I also feel that this game would benefit from having sound effects added into it. At the moment the website feels like it is missing something; the use of sound could not only reinforce the minimalistic theme, but also make the game feel like it has a bit more substance to it.

A more minor factor I would like to improve upon is how the opponent’s previous move is displayed. On this website, it takes several seconds between one user selecting a move, and it appearing on the other player’s screen. This in itself slows down the pace of the gameplay and makes the game more likely to be boring. As a new counter is placed, it blinks white a few times before returning to its corresponding colour. However, after this animation has finished, there is no way to tell where the player just placed their counter. I intend to have a system where the most recent counter is highlighted differently to the rest; this will make it clearer.

Overall, I think this implementation of ultimate tic-tac-toe has been very successful. As well as adopting some of its features in my own solution, I also recognise the need for other modes to build upon the concept and to make the game more smooth, visually pleasing, and enjoyable.

#### Summary of how this research will affect my solution:

|  |  |  |
| --- | --- | --- |
| Aspect of researched solution | Identified approach to my solution with respect to this aspect | Justification of approach |
| Online website-based game | My game will not be Internet-based; instead it will be an offline Windows application. | My knowledge of networking and of compatibility between different search engines and operating systems is too limited for me to attempt this. |
| Online multiplayer URL system to connect players | The only multiplayer compatibility I will be adding will be an offline, local multiplayer mode. | Since my game will be limited to offline, it is not possible for me to use a system like this. |
| Game cannot start without an opponent to tell the URL to (there is no single-player mode) | My solution will have both single-player and multiplayer versions of each mode. | I would like players to still be able to use my game if they are alone; therefore a single-player mode against the computer is needed. However if they do have someone to play against, they will still be able to use the multiplayer mode. |
| Unfinished games are stored online and can be resumed by using the same URL | My application will automatically save logs of each game, and include the option to watch a replay of each match. | It is impractical for the user to have to save the URL in order to resume their game. Storing logs of each game within the program’s files allows the user to either resume or watch a replay easily. |
| Clean and simple user interface | My user interface will be as simple as possible and have clearly laid out menus. | The interface of my game must be visually appealing, otherwise it is hard for the user to navigate the menus and understand the board. I will endeavour to maintain a clean design in my solution. |
| Players can choose a name | Before my game starts, players will be able to choose a name, and if they enter no name, it will automatically revert to a default. | This is a nice little touch that allows the user to slightly customise their game, and is a small feature that is easy to implement. |
| Multiple themes | I will add an option into my game to change between several colour schemes. | Different users will have different tastes, so I want to cater for everyone so that there is a theme that they find visually pleasing – this will ensure that everyone can enjoy the game. |
| Possible next moves are highlighted clearly | I will create a system that clearly displays where the player can place their next counter. | I feel that the system used on this website is good, but there is still potential for confusion which I will try to eliminate further with my own solution. |
| Easy to differentiate between counters and who has won each grid | My grid will have a clear and simple layout, and it will be obvious which counters belong to which player, and who’s turn it currently is. | In order to maintain the clean, simple aesthetic of the game the counter must conform to the themes that I will implement. |
| Message tells which player won the game, and the board remains on screen | My grid will remain on screen even when the game has finished, and any menus will be next to the grid so that the grid is fully visible. There will also be a win/draw message at the end of each game. | I want players to see the post-game board so that they can learn from their mistakes. It may not be immediately apparent who has won the game, so I will include an explicit message. |
| Rematch system, avoids re-entering URL | The menu that appears when a game finishes will include a rematch button that starts a new game with the same settings. | I am including this small quality-of-life feature because users won’t have to change all the settings every game, minimising the time between games so that more can be played. |

### Mobile Phone Application (made by Magma Mobile):

This Android app is another version of ultimate tic-tac-toe that is available for smartphones to download from the Google Play Store. It was made by the developer Magma Mobile, and has a single-player, and both local and online multiplayer modes within the app.

The first thing you see upon opening the app is a colourful title screen (Figure 6), with several options. After pressing the play button, the user is then greeted with 3 different modes, as shown in Figure 7. I particularly like the colourful interface here, because it makes the game more appealing and also easy to navigate. This is a quality that I will strive to maintain in my own program.

When the purple pencil is pressed, the menu in Figure 8 appears. This gives a large variety of options to modify the game to the user’s liking. The computer can be set to a total of 4 difficulties, and the user can choose whether they would like to play first, or second, after the computer. I would like to adopt this level of game customisation into my solution to ensure that the user can play the game exactly how they want to.

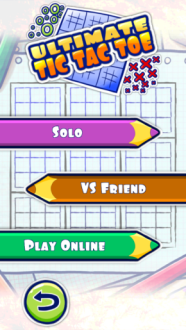
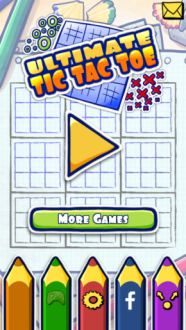


Figure 6 – main menu

Figure 7 – mode selection

The game itself continues with the stationery theme, with a bright and user-friendly interface. The pencil at the bottom of the screen clearly indicates whether it is your turn or not, although the computer moves within a second of you making your move, so most of the time it is as pictured in Figure 9. Magma Mobile has chosen the classic ‘nought’ and ‘cross’ shape for the counters, but here they look like they have been drawn by a pencil, which gives off a more modern vibe. When a square is won, it is highlighted in the corresponding players’ colour. I find this highlighting system to be very clear, and it makes it very obvious that a square has been won, which is a good thing.

This version of ultimate tic-tac-toe quite successfully shows the area in which you can make your next move. In Figure 9, the top-right grid has a green border, showing that player one may place a counter in only that grid. The border also has a rotation animation every few seconds, which helps to make it stand out. There is a slight possibility of confusion, because the border is the same colour as the counters, and it could hinder a player’s thought process. I plan to use a similar method in my solution, but I think I need to make the highlighting colour slightly different to that of the counters.

This app also makes good use of sound, with unique sounds for when a counter is drawn, when one of the small grids is won, and when the user wins or loses the match as a whole. These sounds are not too distracting and they add a nice touch to the application, complementing the stationery theme and giving the game more substance.

An interesting option that is unique to the two offline modes is the ‘eraser’ near the top right. Not only does this let the user undo their previous move, but they can completely restart the game if they press the button enough, since the app seems to store the whole order of play for the game. This is a very helpful feature that can help a person who is trying to improve at the game – if they realise a move was bad, they can simply replay the situation in a different way. It also acts as a provision against accidents; anyone using a small phone to play the game can easily press the wrong square because their screen is so small, but this is not a problem since they can just undo it. I think this would be a very worthwhile feature to implement in my own game.



Figure 8 – changing the game settings

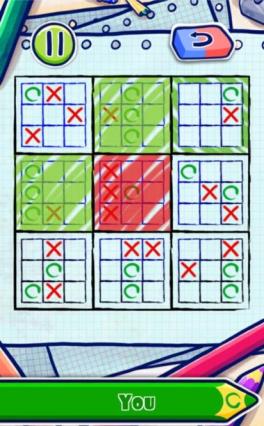


Figure 9 – the playing board

The app also contains two multiplayer modes. Both are very similar to the solo mode, and they are just adapted to facilitate two players. The first, ‘VS Friend’, is very similar to the solo mode, except instead of playing against the computer, two users play against each other taking turns on the same device. I managed to find a bug with this mode – when using the undo button, it does not switch the player correctly, so a player ends up getting an extra move. For this reason, this mode is slightly compromised because you can end up with many more of one counter than the other, and this breaks the fundamental rules of the game. Having said this, I still want to include a local multiplayer mode in my own game, but I must ensure that the rules can’t be broken by any major bugs, and I must take extra care when implementing an undo function.

The final mode in this game is the online multiplayer. It is again similar to the other modes, but this time you can be matched up against a random opponent anywhere in the world. The game also includes the option to log in to Facebook but aside from displaying your Facebook name instead of ‘Anonymous’ as a username, it doesn’t seem to have much of a purpose. The mode itself is exactly the same as the local multiplayer mode, except there is no undo button.

The application also has support for Google Play achievements which allows users to accumulate ‘Play Games’ experience for their account. This is a nice feature - while it doesn’t add much to the game itself, it does give players an extra incentive to keep playing. I think an achievements system would be a fun addition to my own solution.



Figure 10 – online multiplayer

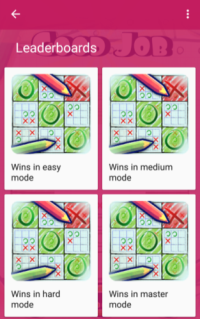


Figure 11 – Google Play achievements

However, implementing support for external applications such as Google Play and Facebook would be beyond the scope of my project because I don’t foresee any reason this would be needed, and there is a long process involved to register any applications with these platforms which would take too much time for such a little benefit.

­­­

A limit of this application is that it is only available on mobile, so it cannot be played on a desktop computer. Magma Mobile also seems to have made this game to earn a profit, and therefore there are multiple adverts which can become annoying as they distract the user from the gameplay. This should not be an issue with my solution because I do not intend to make money from it. I will not, however, create my application for mobile phones because of compatibility issues between languages and operating systems, which would require me to essentially recode the whole game for each platform.

As a whole, this application has a clean and user-friendly interface, making good use of graphics, animations and sound. For this reason, I feel this solution has also been a success, despite a flaw in one of the modes. There are some important features and concepts that I wish to adopt and build upon when I create my own version of this game.

#### Summary of how this research will affect my solution:

|  |  |  |
| --- | --- | --- |
| Aspect of researched solution | Identified approach to my solution with respect to this aspect | Justification of approach |
| Application can only be played on Android devices | My application will only be available on devices that run the Windows operating system. | I have a limited knowledge of coding with other operating systems, and it would take unreasonable amounts of time for me to test each operating system one by one. Therefore I feel having the game on multiple platforms is outside the scope of my project. |
| Colourful stationery-themed interface | I will add multiple themes into my game, and they will not be as complicated as the one in this mobile app. | While this theme does work, I plan to keep to a simpler style with my themes in order to make the game as clean and easy to understand as possible. |
| Game contains advertisements | My game will be free to play, and will not have any advertisements in it. | I am making this game as a non-profit project, so I do not wish to degrade the user experience and make the interface look worse by including advertisements. |
| Easy to navigate menus | My menu system will clearly state where the user currently is and clearly list each button they can press. | This is important because I want users to be able to explore all of the features of my game with ease. |
| Only one theme | My solution will have multiple themes that the user can choose between, instead of just one. | For this mobile application, if the user does not like the theme there is nothing they can do to change it. I am going to add several themes to make the game more flexible for different users. |
| Sound effects play when certain events are triggered | I want to make use of similar sounds in my own game – however I must not overuse them. | Sounds will add more substance to my game and make it more enjoyable, but only if they are used in moderation and fit the style of the application (since otherwise they would feel out of place). |
| Use of simple animations to show where next counter can be placed | I intend to use some simple animations in my application, but they must not interfere with the simple, clean feel of the game. | Animations will be able to complement the theme and make the game feel more complete and professional. As long as this doesn’t interfere with the gameplay I feel this is justified. |
| Three modes – single-player, and both local and online multiplayer | My game will include several modes and they will all be playable with either one or two users. There will not be an online mode however. | I don’t have enough experience with networking and online systems, so I will not attempt this. However, I wish my offline modes to be playable even if a user does not have someone to play against. This will require me to create an option to play against the computer. |

### Ultimate Tic-Tac-Toe on Steam (made by Tigerish Games):

The final existing solution that I have researched is a third version of ultimate tic-tac-toe which is available for download on the Steam platform. It was uploaded to the Steam store 28th April 2015 by Tigerish Games.

Upon opening the game, the user is greeted by a window that looks like a chalkboard, and the main menu. The game’s whole theme is to make everything look like it was drawn by chalk. However, I feel like the green colour of the background does not mix very well with the brown wooden border of the window, and the font is quite a low quality. In my opinion the aesthetics of this game fall short of the standard of the other two solutions I have researched.

Part of this aesthetic problem is that the menus look confusing. The options are very closely grouped and are all the same colour, which makes reading the text very difficult. Choosing an option on a menu then brings up the next menu, which looks nearly the same and can often lead to confusion or a user getting lost in the menus. For example, selecting ‘Single Player’ in Figure 12 leads to a very similar menu, shown in Figure 13

An advantage of this game over the mobile application is that it has a ‘Help’ page (Figure 14) which not only explains the basics of ultimate tic-tac-toe, but also gives the rules for some of the modes which are unique to this solution. The trouble is that it is very difficult to read because so much text is on the screen at once. It would be better if the information was split up into multiple slides that the user can select through.

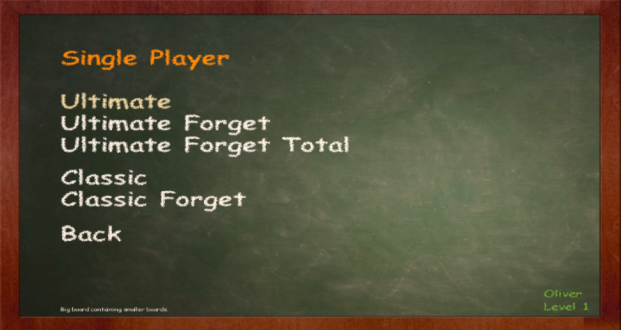


Figure 13 – Choosing a single-player mode

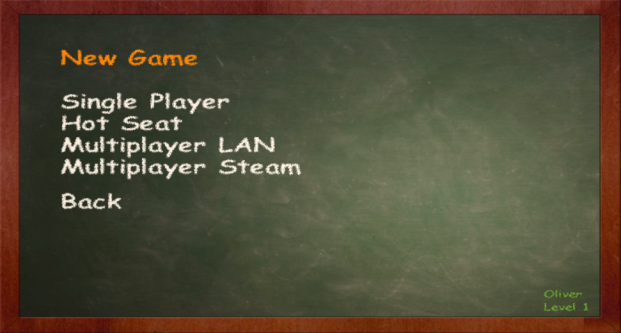


Figure 12 – Choosing which mode to play

This game does include an options page, shown by Figure 15, which can modify the game somewhat, which I like as a feature. A setting unique to this solution is ‘Turn Time’, where the user can choose whether they want to allow only 5, 10 or 15 seconds to make a move (or infinite time). If the timer runs out then the computer chooses a random move for the user.

Like the mobile application, there is also an option to choose the computer’s difficulty, which makes the game accessible to players of all levels. The other options are what I would expect a game of this type to have.

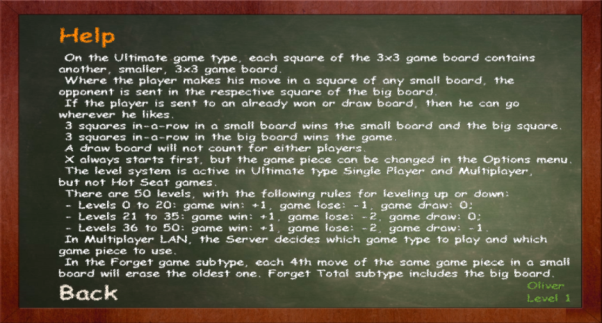


Figure 14 – The rules page

Despite not looking quite as good as the other two solutions that I looked into, this game has the biggest variety of modes out of the three. Firstly, it has the ‘Classic’ mode which is simply the traditional (three by three grid) tic-tac-toe game. The presence of the original game makes sense because it brings the ultimate version of the game back to its roots, and I feel like this makes the solution feel more complete. A finished game of the ‘Classic’ mode is demonstrated in Figure 16.

The second mode is called ‘Hot Seat’, and although by the name it seems like a new mode, it is actually just local multiplayer where the computer is shared between two people. The third mode, ‘Multiplayer LAN’, is again similar but in this version two games are connected with an IP system that is similar to the one used by bejofo.net/ttt, but it is all processed internally by the application rather than relying on website-based support. Therefore, this solution does not have a save states system implemented. However, it does allow multiplayer on either one device, or over two – something my solution should have the capacity for.

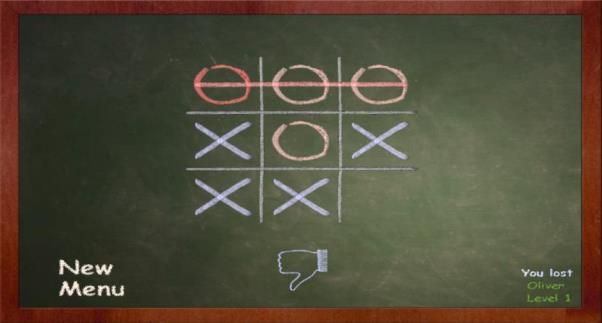


Figure 16 – Losing a game of ‘Classic’ mode

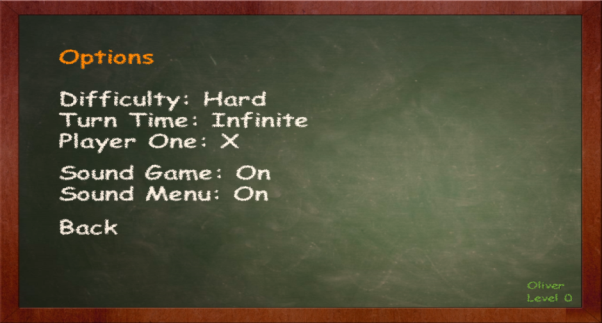


Figure 15 – Changing the settings

This application also features ‘Multiplayer Steam’, where the user chooses the mode they would like to play, and they must wait for the game to match them up with another user somewhere else in the world. This mode only works if each user has an internet connection, and unfortunately you can only play against random opponents. Having said this, the application still goes beyond the scope of my own project by adding internet support, even though there is no way for friends who are not near each other to play together on this program.



Figure 17 – A game of ‘Ultimate’ mode; also note the 5 second move timer in the bottom-right corner

**­­­**

All of these multiplayer modes only have slight subtle differences to the single-player modes, which is good since it keeps the game consistent for the users. Each mode is available for every option on the menu in Figure 12 (so it is available in single=player and all of the multiplayer modes).

The fundamental mode in this solution, like in the other two, is the ‘Ultimate’ mode that can be selected from the menu in Figure 13. This works with the same ruleset and adopts a red and blue colour scheme. As shown in Figure 17, the grid layout is quite clear and the use of colour to distinguish between counters works well, albeit for the dark coloured background. The middle square is highlighted in a slightly blue colour – this is indicating where the user can make their next move. The status messages in the bottom-right corner are informative but perhaps look too subtle or out of place - I found that it was difficult to keep track of how much time I had left on each move because it was so hidden in the corner.

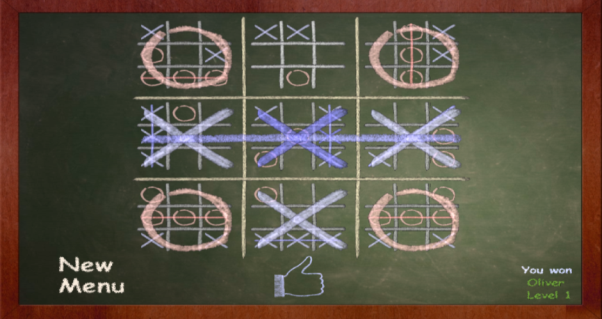


Figure 18 – A complete ‘Ultimate’ game

The main positive is that the game works as it should do, and even incorporates sound and some animations. The animations here are not too over the top since they are only used to show the counters being placed, as if being drawn by a piece of chalk. The drawing sound that plays when a square is selected is a nice touch that reinforces the chalkboard theme. Sound is also used when options are selected in the menus and when the pointer hovers over an option, it glows orange. I also like this small quality of life feature, which makes me want to include something similar in my game.

The ‘Forget’ and ‘Forget Total’ modes are unique to this solution. They are essentially a spin-off from the normal rules of the game. The extra rule in this mode is that previously placed counters will fade away until they disappear four turns after they were placed. The ‘Total’ mode means that this rule applies to both the small grids and the big ones – a nice idea, but it can lead to very prolonged games. Overall, I like this mode but it can be a bit confusing, and there is no way to access the rules while a game is being played, so the user has to remember the rules and if they don’t they may not understand what is going on. I would like to make sure that the user can access the rules in my solution while a game is in progress, to avoid them being unsure of what to do and potentially quitting the game.

Like the mobile app, this Steam game also contains some in-built achievements which also contribute to points for the user’s Steam account. Again, while these do not change the gameplay in any way, they are an incentive to keep the user playing for longer and something worthwhile to work towards. There are roughly 10 achievements, an example of which is shown in Figure 19. I also think the way that the game records your current best is a good idea to include in my game.



Figure 19 – Looking at the achievements

As a whole, I think this Steam application has been successful. Despite falling short on the user interface in comparison with the other two solutions, it offers a wide variety of playable modes and several settings that can be changed to the user’s preference. In my final solution I hope to have a selection of modes just as big, but improve on the user interface to make the game more enjoyable.

#### Summary of how this research will affect my solution:

|  |  |  |
| --- | --- | --- |
| Aspect of researched solution | Identified approach to my solution with respect to this aspect | Justification of approach |
| Application can be downloaded from the Steam store | I do not foresee putting this application online for distribution. | If my application appears to be successful enough then I may consider making it available somewhere online. Until then I would rather focus on the development of the application itself. |
| Chalkboard theme | I will adopt a consistent aesthetic in my solution but it will not be based on a specific theme. | I do not want to limit my interface by having to conform to a particular theme as this may cause me to lose focus on keeping the game simple. |
| Colours do not work that well together | When creating my multiple themes I must ensure that the colours all fit together well. | The game will not be enjoyable to play if it is not visually pleasing, so I will be putting emphasis on ensuring this during development. |
| Single-player mode with various computer difficulties | My solution will have different levels of difficulty when playing against the computer. | If a user does not have someone else to play with I still want them to be able to enjoy my game. Having different difficulty levels will cater for users of all skill levels. |
| Various ways of playing multiplayer | Unlike this Steam game, my own solution will only contain a local multiplayer mode where users share one computer to play against each other. | I am not going to try and involve networking in my program because my knowledge in that field is unfortunately too limited for me to do this to a reasonable degree of success. |
| Both a ‘Classic’ and ‘Ultimate’ mode | I will include both the traditional noughts and crosses game and the ultimate mode within my program. | I like the idea of a classic mode, so that users can practice on a small scale or simply take a break from the complex ultimate mode. |
| Unique ‘Forget’ mode | I do not plan on adding any new modes with different rulesets in my solution. | I found the ‘Forget’ mode confusing, and while I like the concept of a new and exciting mode I would rather focus on making the classic and ultimate modes the best they can be. |
| Steam achievements system | My game will have some achievements and goals for users to work towards. | Having achievements gives users more of an incentive to play the game so I think it would be worthwhile to include some. However I will not use any external services to log the achievements, it will just be tracked and stored within the application. |
| Simple use of sounds and animations | I hope to include some simple sound effects and animations, and maybe background music. | I like the way the sounds complement the theme in this game. Any sounds and animations that I use will be subtle as to not interfere with the gameplay or themes. |

## Essential features of the proposed solution:

Taking into account the input from stakeholders and my research into existing solutions, I have created a list of features that my solution must meet as a minimum for it to be considered a success:

* **Single-player and multiplayer mode,** so that users can play the game whether they are on their own or with a friend.
* **At least three levels of computer difficulty** and an option to choose between them so that the single-player mode is accessible for users of all skill levels, and they can practice against the different levels of artificial intelligence (AI) to get better at the game.
* **Classic and ultimate modes** that can be played whether the user is alone or with a friend.
* **Rules page** that explains the rules of ultimate tic-tac-toe in a concise and easy-to-understand way. This will help new users to understand the game.
* **Logically structured menu system** that explicitly shows the user how to navigate around the application and stays simple to avoid users becoming confused.
* **At least three colour themes** that follow a minimalist design, which the user can choose between. These will allow them to slightly customise their experience and select a theme that helps them to best perceive the playing grid.
* **Achievements** that are tracked by the computer and tell the user when an achievement has been unlocked. This will offer a sense of progression to the game and keep users playing it for longer.
* **System which highlights the possible moves** that the user may make. This is to offer as a guide, and stop players making illegal moves.
* **Players can choose their names** before the game starts.
* **Rematch system** that starts a new game without players having to choose the same settings again.
* **Current score** and information such as move number and win percentages are shown during the game.
* **Records of each completed game are stored** in a text file within the game’s files.
* **Replays of old games can be watched** or suspended games can be continued, by reading in these text files.
* **Moves in single-player can be undone** using a button. This is in case the user accidentally misclicks, or if they are about to lose they could go back and see where they made a mistake.

## Limitations of the proposed solution:

There are some features that I will not try to include in my application, and therefore these may limit the success of my solution:

* **No support for different operating systems.** I anticipate only creating this game for the Windows operating system, and the resulting executable (.exe) file will possibly not work on operating systems such as Linux and macOS. I have no programming experience with these operating systems so it will be too difficult to ensure compatibility.
* **No multiplayer across two devices.** I feel this would be very difficult to implement even on a local area network (LAN), since the game may not be compatible between different operating systems. I also do not wish to spend too much time figuring out how the networking would work when I could be focusing on improving other features of the game.
* **Only playable on desktop computers**, not on mobile. Even though one of my researched solutions was a mobile application, this field again brings up compatibility issues between devices. I have no knowledge of programming languages such as C++ and Java (common languages used with mobile application development) so I would have to learn these and this would take a lot of time.
* **No online multiplayer.** Online networking would be even more difficult than working with a local area network, and since I am not doing the latter this will not be a feature in my solution.
* **No external service used for achievements.** I aim to keep my game independent because I do not want to be restricted by companies’ regulations, and the sign-up process is long and could involve paying money. As a result my game will not contain support for third party applications such as Facebook and Google Play. This means that the achievements will be purely cosmetic and they will not be able to contribute to anything outside of the game itself.

## Requirements specification:

I have listed below the hardware and software that users must have to be able to use my finished application.

### Hardware requirements:

* 1.5GHz or higher CPU clock speed
* 2GB or more of RAM
* Monitor with minimum resolution of 800x600
* Hard disk with at least 1GB of free storage space
* Mouse and keyboard
* Speakers or headphones are optional and will not be needed to be able to use the application

Apart from using graphics and sounds my game should not be too intensive on the computer’s resources, so I have specified low requirements that nearly all modern computers meet. The usual basic computer hardware will be needed to control the application. I plan for my game to be quite simple, so I expect it to not take up that much space, hence the reasonably low storage requirement compared to other games.

### Software requirements:

* Any edition of Windows 7 (Starter, Home Basic, Home Premium, Professional, Enterprise or Ultimate)
* Any other more recent update of Windows

I expect to use several computers to program the application. The oldest of these uses Windows 7, and the others all use newer versions of Windows. Since these operating systems are made to be backwards compatible there should be no issues running the application. I have little experience with other types of operating system so my solution will most likely not be compatible with those.

### Specific requirements and justification:

My finished solution will have the features below. I have also added a justification with each as to why I want to include them:

* **The game will include a ‘classic’ mode.** This mode will be my own iteration of the traditional tic-tac-toe game that uses a 3x3 grid. It will follow the same set of rules. This is mainly for the game to have a complete feel so that the main game can build upon the simple concept of this mode.
* **The game will also include an ‘ultimate’ mode.** This will be the main feature of the solution and will use the rule set of ultimate-tic-tac-toe. The playing area will be a 3x3 grid, with each square containing a smaller playing grid (overall there are 9 small grids). The big grid will be distinguishable from the smaller ones so that the game is as explicit as possible.
* **Both of the modes will have their own set of rules.** These sets of rules will be accessible by a button in the corner of the window. Depending on which mode is being played, only the corresponding rules page will be accessible so that players don’t accidentally read the wrong rules and get confused with what they are doing.
* **There will be two ways to play both of the modes.** The first two options on the main menu of the application will be ‘Single-player’ and ‘Multiplayer’ buttons. Clicking on either of these will bring up the same secondary menu, and then prompt users to choose either classic or ultimate mode. I will include these modes so that the game can be played whether someone is on their own or with a friend.
* **There will be three levels of computer difficulty for the single-player mode.** These levels of artificial intelligence will be called ‘Easy’, ‘Medium’ and ‘Hard’ and can be changed just before starting a single-player game. This is so that there is more of a challenge for skilled players while also making the game accessible for new users.
* **The single-player mode will contain an undo button.** This will allow the user to erase their previous move and try again. This will help to improve the game by allowing users to undo their mistakes.
* **Options will be accessible from the main menu.** Below the single-player and multiplayer buttons will be an ‘Options’ menu that contains settings that can be changed. The settings will include move timer, the option to turn animations, sound effects and music graphics on or off, and theme selection.
* **The game will play music in the background.** A quiet ambient soundtrack will play on loop in the background of the game – this will be able to be turned off if users find it annoying or wish to use listen to their own music, to improve the quality-of-life aspect of the game.
* **Sounds will play when counters are placed and when options are selected on the menu.** This will make the solution feel more like a computer game and set it apart from playing the physical game with pen and paper, slightly improving the user’s experience. If the user does find these sounds annoying they will be able to toggle them in the options menu.
* **Before the game starts, players will be able to choose their names and/or the name of the computer opponent.** This text will have a 16 character limit to avoid this feature being abused and allow it to fit in the window easily. If a name is not chosen the game will set it to a default value.
* **The user will be able to access statistics from the main menu.** As the game is played the computer will automatically record data and present it to the user if they select the ‘Game Statistics’ button. This data will include play time, games played, moves made and percentage of games won. This is for the user’s interest and makes the game feel more professional – plus logging these statistics means they could be used for further features in the future.
* **There will be three themes that the user can choose between.** These themes will be based upon the base colours of white, blue and grey and will adopt colour schemes that make the gameeasier to understand and customise. There will be an option to choose the theme before the game starts.
* **From the main menu users can import text files of previous games and watch them being replayed.** This will allow players to remember their victories and also reflect on their losses, enabling them to improve at the game.
* **In-game achievements.** The final option my game will contain on the main menu is a button that leads to achievements. These will mainly be for fun and an incentive to keep users playing for longer without losing interest. Some achievements will include tiers such as: winning 1, 5, 10 ,50, 200 games; winning ultimate in less than 40, 35, 30, 25 moves; and achieving a win rate of 60%, 70%, 80%, 90% (after having played at least 10 games).

## Measurable success criteria:

Based on all of the research I have conducted, the needs of the stakeholders and my own ideas, I have created some criteria that I will use to test if my solution has been successful:

|  |  |
| --- | --- |
| Success criteria | Justification |
| Every time a counter is placed, the move count will increase by one. | This will mean that the application is correctly tracking the number of moves that have been made. |
| The counter being used will alternate every time a move is made, between ‘noughts’ and ‘crosses’. | For the game to work, this is essential. Without this working, one player may get an unfair amount of moves and this will break the rules of the game. Even if the computer registers which player moved correctly, the visual representation given to the player must reflect this. |
| When there are three of the same counters in a row in a grid, that player wins the square. Users will then not be able to place counters in that grid. | This includes all of the modes that will be in my solution. This is the basic framework of the game since it would not function properly without this completely working. |
| When three bigger squares are won in a row, the game will end and show which player won. | This only applies to the ultimate mode but it is the main focus of the application. The win checking must work for the game to conform to the rules. |
| Selecting each button on the menu will result in a ‘clicking’ sound. | This will give the game a more complete feel and will increase the user-friendliness of the application, meeting the needs of my stakeholders. |
| When requirements for an achievement are met, the achievement will be marked as ‘completed’ in the achievements menu. | The computer will automatically track the data, but during testing I must also ensure that it registers achievements as completed when the requirements are met – otherwise the achievements system that my stakeholders would like will not work properly. |
| Changing the visual theme will not affect the gameplay in any way. | Changing settings in the middle of a game could potentially cause the rules of ultimate tic-tac-toe to be broken. I must ensure that this does not happen. |
| After a game the user will be asked if they want to save the replay or not. Upon selecting ‘Yes’ a text file is created in the game files with all of the game’s information. | Using text files to store the game’s results will be sufficient. I do not need to worry about extra encryption or security in this case because changing the game replay would not affect the tracked statistics. |
| The game will be able to read in text files and allow the user to step through the game at their own pace. | The replay must accurately replay the respective game move for move; otherwise this will not be a success. |
| If a text file is read in but it has the incorrect format the game will let the user know without crashing. | The game should not crash at any point due to bugs or poor design. This feature must be tested so that the game can keep running even if there is an error with the file format. |
| The artificial intelligence in single-player mode will be harder to beat the higher the difficulty level is. | This can be tested by analysing the win rates against each difficulty level – the harder levels should have a lower win rate than the easier difficulties. |
| Selecting single-player or multiplayer on the menu leads to the correct corresponding game type. | It would be very confusing for the users if the computer started the wrong mode. One of the main aims of my solution is to reduce confusion so I must ensure this does not happen. |
| Statistics that involve ‘lifetime’ usage of the application will not reset when the application is closed and then opened another time. | The game must be able to save statistics to the game files, otherwise when they quite they will have to start all over again. I may use simple encryption to deter users from accessing the files to change these statistics and instantly unlock every achievement in the application. |

# Bibliography

Google Play Store

[www.bejofo.net/ttt](http://www.bejofo.net/ttt)

Steam