

---

# **Tic Tac Toe Game Design Document**

*Rev. A draft 1*

**James Richey**

**Mar 15, 2020**



# CONTENTS

<b>1</b>	<b>Introduction</b>	<b>1</b>
1.1	Purpose of this Document . . . . .	1
1.2	Scope of the Game . . . . .	1
1.3	Overview of this Document . . . . .	1
1.4	Project Objectives . . . . .	2
<b>2</b>	<b>Gameplay</b>	<b>5</b>
2.1	Rules of Tic Tac Toe . . . . .	5
2.2	One Player Mode . . . . .	6
2.3	Two Player Mode . . . . .	6
2.4	Speedrun Mode . . . . .	6
<b>3</b>	<b>User Interface</b>	<b>7</b>
3.1	Game Board . . . . .	7
3.2	Controls . . . . .	9
3.3	Menus . . . . .	9
3.4	Screen Flowchart . . . . .	15
<b>4</b>	<b>Environments</b>	<b>17</b>
4.1	Environment Selection . . . . .	17
4.2	List of Environments . . . . .	17
<b>5</b>	<b>Additional Considerations</b>	<b>33</b>
5.1	Target Audience . . . . .	33
5.2	Target Platforms . . . . .	33
5.3	Licensing and Distribution . . . . .	34
5.4	Similar Games . . . . .	36
5.5	Future Enhancements . . . . .	36
<b>6</b>	<b>Technical Design</b>	<b>39</b>
<b>7</b>	<b>Glossary</b>	<b>41</b>
	<b>Bibliography</b>	<b>43</b>
	<b>Index</b>	<b>45</b>



## INTRODUCTION

### 1.1 Purpose of this Document

This is the game design document for Tic Tac Toe. This document describes in detail the he objectives, requirements, and design considerations of the game providing a central location for this information. This is invaluable for understanding the game's scope, planning the project milestones, and creating the games assets.

Anyone who is involved with the game's development is encouraged to read this document and keep a copy handy.

### 1.2 Scope of the Game

Tic Tac Toe is a game of strategy where two players, X and O, take turns placing their mark in a 3 x 3 gird. The first player to get three marks in a row, column, or diagonal wins the game.

This version of Tic Tac Toe is an unique take on the classic game. Players of all ages battle the computer or each other in a variety of stunning environments. Each environment tells part of the story of Tic Tac Toe from the past, present, and future. Each environment has a strong visual theme and complementary soundtrack.

Tic Tac Toe is fee, open source, no contains no annoying advertisements. It opens runs on Windows, Mac, and Linux. The game launches Summer 2020.

### 1.3 Overview of this Document

This game design document contains chapters covering various aspects of the game.<sup>1</sup> This includes chapters for the game's *Gameplay*, *User Interface*, and various *Environments*.

The *Technical Design* chapter provides technical design details of the games architecture.

Additionally, the *Glossary* defines terms that are used throughout the game design document.

---

<sup>1</sup> The structure of this document is based on [Rogers-2014].

### 1.4 Project Objectives

Creation of the Tic Tac Toe game described in this document is part of the Pound of Rust project. This section describes the project's objectives.

#### 1.4.1 Create Tic Tac Toe Game with Rust

The main deliverable of this project is the Tic Tac Toe game for Windows, Mac, and Linux. This project is the follow-up to the Ounce of Rust project<sup>2</sup> that resulted in the creation of a Rust based Tic Tac Toe logic library, `open_ttt_lib`.<sup>4</sup>

In addition to creating a fun game, the project provides more hands-on experience using Rust and is a showcase for `open_ttt_lib`.

#### 1.4.2 Provide Free of Charge and Under an Open Source License

Tic Tac Toe and all future releases of the game are free, open source, and contain no advertisements or trackers. The game is released under a permissive open source license and its code is available from a public repository such as <https://github.com/>.

Many of today's games casual games are released for free, but include questionable monetization models such as microtransactions, pay-to-win schemes, advertisements, and personal data harvesters. Tic Tac Toe stands apart from these games by respecting players who choose to spend their valuable time playing the game.

#### 1.4.3 Release by RustConf 2020

Tic Tac Toe's initial release is scheduled to coincide with RustConf 2020 on August 21, 2020. RustConf is the annual Rust developers conference; since Tic Tac Toe is developed in Rust this makes an excellent time to launch a Rust based game.<sup>5</sup>

To help meet the target release date, the initial release of the game might contain a subset of the environments described in this document.

#### 1.4.4 Easily Expandable and Modifiable

Playing Tic Tac Toe in a variety of environments is a large part of what sets this version of Tic Tac Toe apart from others. The game is designed such that developers can easily add new environments. This allows developers to focus their time and efforts creating stunning environments. Additionally, this lowers the barrier of entry for users who are interested in modifying the game. Finally, this allows quick turnaround of future releases of the game.

Automated tools, guides, checklists, and detailed documentation are some ways that can help development speed.

---

<sup>2</sup> For details on the Ounce of Rust project see the [Ounce of Rust Project Manual](#)<sup>3</sup>

<sup>3</sup> <https://j-richey.github.io/project-documentation/ounce-of-rust/>

<sup>4</sup> `open_ttt_lib` is available at [https://crates.io/crates/open\\_ttt\\_lib](https://crates.io/crates/open_ttt_lib) and source code is hosted at [https://github.com/j-richey/open\\_ttt\\_lib](https://github.com/j-richey/open_ttt_lib)

<sup>5</sup> For details on RustConf see their website: <https://rustconf.com/>

### 1.4.5 Build Risk Reduction Prototype

The development team creating Tic Tac Toe is new to the Rust programming language and the available Rust libraries for game development. To help mitigate this risk, a throwaway prototype game is created early in the project that explores various technical aspects.

Using the lessons learned from the prototype also helps the development team design a code base that is easily expandable and modifiable per the above objective.





## GAMEPLAY

### 2.1 Rules of Tic Tac Toe

Tic Tac Toe is a game of strategy where two players, X and O, take turns placing their mark in a 3 x 3 grid. The rules for Tic Tac Toe used for each game mode are as follows:

1. Play occurs on a board composed of a 3 x 3 grid of squares. The board starts empty with no marks.
2. The first player places their mark in one of the grid's squares. Traditionally, the mark is the letter *X*.
3. The second player places their mark in one of the grid's empty squares. A square that already contains a mark cannot be updated or altered. Traditionally, the second player uses the letter *O* as their mark.
4. Turns alternate between the players until the game is over.
5. The first player to get three of their marks in a line wins the game. That is: they have three marks in a row, column, or diagonally. Examples of winning games are shown in [Figure 2.1](#).

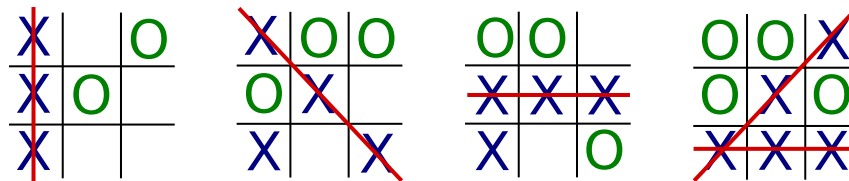


Figure 2.1: Examples of winning Tic Tac Toe games showing player X winning by getting three marks a row, diagonal, and column. The red line shows the squares that contributed to the win. Notice that it is possible to get multiple sets of three marks in a row.

6. The game ends in a draw, known as a cat's game, if no more empty squares remain and a player has failed to get three marks in a line. Examples of cat's games are shown in [Figure 2.2](#).

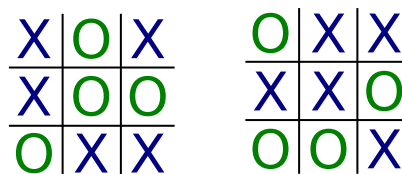


Figure 2.2: Examples of Tic Tac Toe games ending in a cat's game. No player managed to get three marks in a line.

7. The steps above are repeated for a series of games. The starting player alternates between games, that is the second game player *O* gets to make the first move.

### 2.2 One Player Mode

In one player mode the player battles a computer controlled opponent. There are three difficulty settings: **easy**, **medium**, and **hard**.

Easy difficulty is targeted at players who are new to Tic Tac Toe and/or computer games. The computer picks random squares allowing players to learn the game's controls and rules.

Medium difficulty is for players who have some experience with Tic Tac Toe. The computer provides a challenge to the player but games are still winnable.

At hard difficulty the computer plays almost perfect games. The player must capitalize on rare mistakes made by the computer to win. This is the recommended difficulty for experienced Tic Tac Toe players.

### 2.3 Two Player Mode

In two player mode two players take turns placing their marks according to the rules described above.

### 2.4 Speedrun Mode

A speedrun mode provides an additional challenge for experienced players. Players battle a flawless computer opponent through 10 environments completing the games as fast as possible. At the end of the run the total time is displayed along with the previous best times.<sup>6</sup>

The player is disqualified and the run halted if the player loses a game. Since the computer opponent never makes a mistake, each game in the speedrun ends in a cat's game. In other words, each speedrun requires the same total number of moves to complete.

Unnecessary animations are disabled in speedrun mode so they do not get in the way of the speedup gameplay. Additionally, speedrun mode has its own dramatic music that replaces the music tracks of each environment — environments are played so fast there is not sufficient time to appreciate their individual sound tracks.

---

<sup>6</sup> The time it takes the computer to pick a square is not counted towards the player's time. This ensures times are consistent between slower and faster computers.

## USER INTERFACE

This chapter describes the user interface of Tic Tac Toe. This includes the main game board, controls, and all game menus.

### 3.1 Game Board

The game board is where players spend the majority of their time. Additionally, the game loads directly to this view ensuring players get to gameplay as quickly as possible without menus getting in the way.<sup>7</sup> A concept drawing of the game board is shown in [Figure 3.1](#).

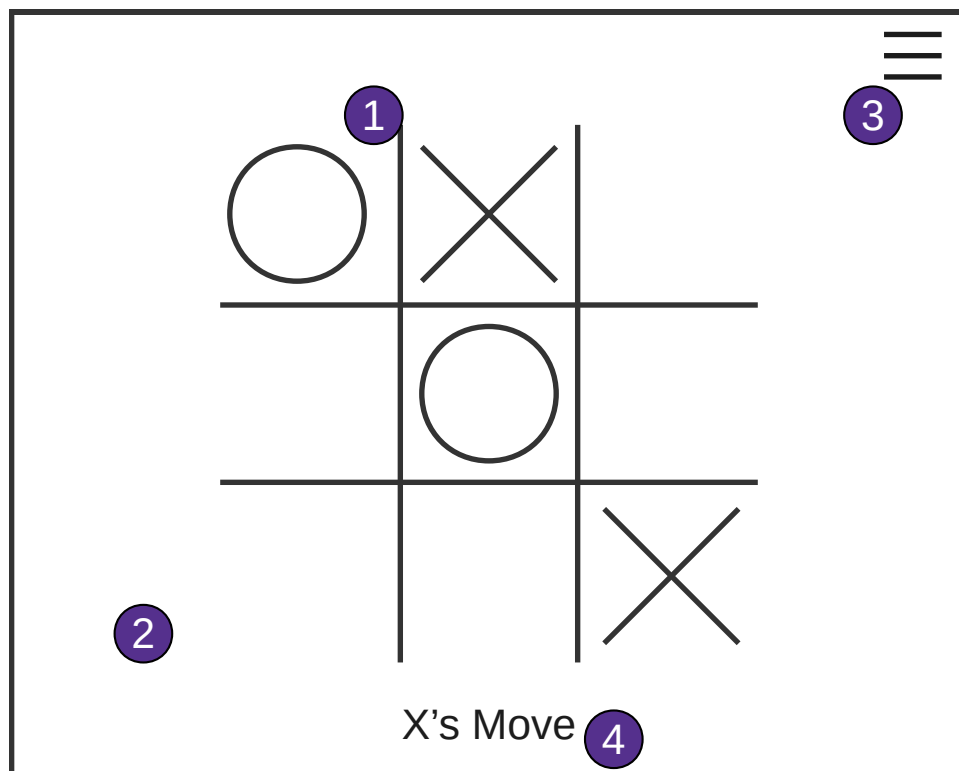


Figure 3.1: Game board concept drawing.

The game board contains the following items of interest:

---

<sup>7</sup> The loaded game is a single-player game using the last difficulty and player mark settings. The defaults for these are Medium difficulty and X marks.

1. The marks and grid. The appearance of these depend on the current environment being played. However, the marks for all environments use the same center point and have the same selectable hotspot. This ensures consistency between environments when using the mouse.
2. The background of the game board depends on the current environment.
3. The hamburger button opens the *Main Menu*.
4. Status text indicates who gets to make the next move and the outcome of the game. Once the game is over buttons to play the next game or return to the menu also appear in this area. The text is outlined or shaded such that it is visible over any possible background.

A major focus of the game is playing Tic Tac Toe in variety of stunning environments that control how the marks, grid, and background look. Thus, a minimalist approach is used for the game board view. The only UI widgets are a menu button and some status text.

### 3.1.1 Speedrun

Additional UI widgets are added to the game board to facilitate speedrun mode. [Figure 3.2](#) shows the speedrun gameboard.

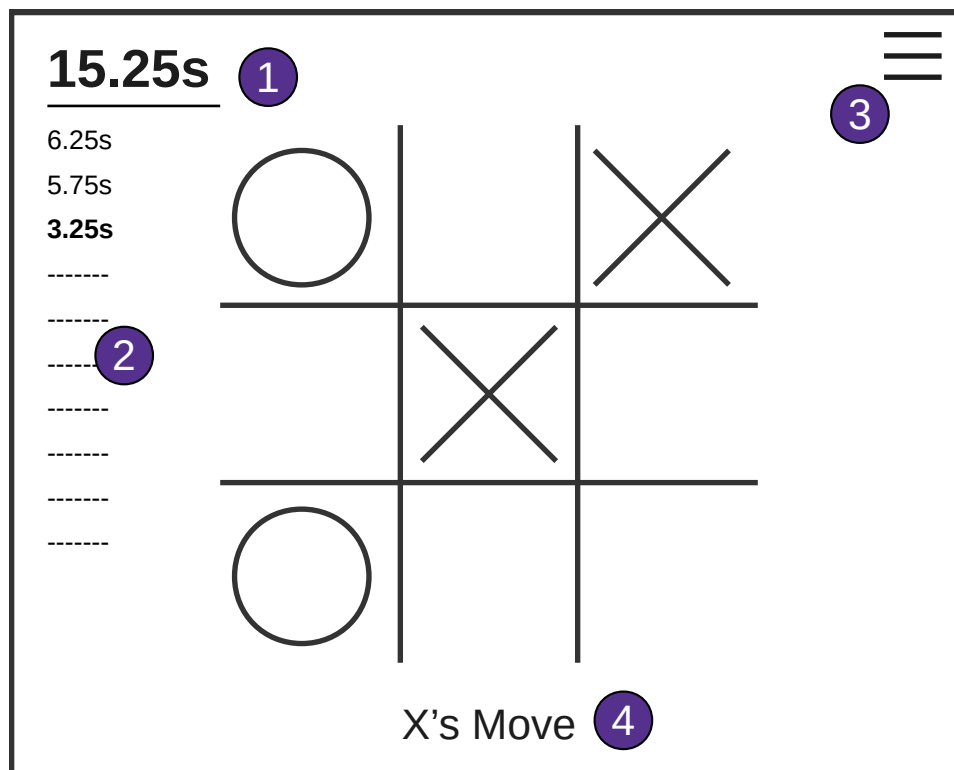


Figure 3.2: Speedrun game board concept drawing.

Items of interest are:

1. The speedrun status display prominently shows the elapsed time of the run. This helps give the player a sense of urgency and lets them see if they are on track to get a best time.
2. Splits for each game are additionally displayed. Dashes or other marks are used for games that have yet to be played. This allows players to quickly visually gauge how many games remain.
3. Opening the game's menu ends the run. The run is disqualified.

4. Status text indicates the current turn. If the player loses a game, the status text notes that the run is disqualified and the player is invited to try the run again or return to the Speedrun menu.<sup>8</sup>

When a game is competed successfully the next game starts immediately allowing players to go as fast as possible through the games.

## 3.2 Controls

The game can fully played with either a mouse or keyboard. New or casual players may prefer to use a mouse where as speedrun players may prefer to use the keyboard.

### 3.2.1 Mouse

Mouse left click is used to select free squares and menu press buttons. Right click and other mouse buttons are unused.

### 3.2.2 Key Bindings

The game supports being played using the keyboard. [Figure 3.3](#) shows the game's keybindings for selecting squares.

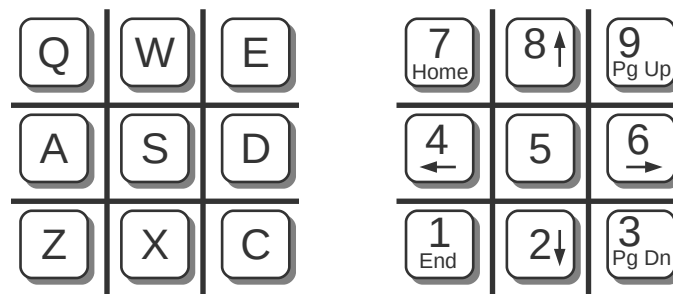


Figure 3.3: Keybindings for selecting squares.

The numpad keys are available for right handed players and the QWE set of keys are available for left handed players. The arrow, ESC, Enter, and Space keys allow users to navigate the game's menus.

## 3.3 Menus

The Tic Tac Toe menus allow players to select the various game modes and to customize the game. The [Screen Flowchart](#) provides details on how the menus and views connect.

Each menu is described in the following sections.

<sup>8</sup> If the player loses a speedrun game, the board remains visible so the player can see where they made mistakes. This allows them to adjust their strategy for next time.

### 3.3.1 General

Unless otherwise noted, the information in this section applies to all menus.

The menus have translucent or blurred background that show the active environment. This includes any animations or environment FX. Additionally, the environment's sound track continues to play while the menus are open.

### 3.3.2 Main Menu

The main menu provides a central point for users to navigate to the game's various modes and settings. [Figure 3.4](#) shows the main menu.



Figure 3.4: Main menu concept drawing.

1. The title of the game is prominently displayed at the top of the menu.
2. The *Resume Game* button is visible if there is a current game in progress. Selecting it closes the menu and returns the player to the game.
3. New game buttons. The *Single-player* button navigates to the while the *Single-player* screen while the *Multi-player* button immediately starts a new multiplayer game.
4. Miscellaneous buttons to open the *Options*, *Credits*, *Help* screens.
5. *Exit* closes the game and returns the user to their desktop.

### 3.3.3 Single-player

The single-player menu, shown in [Figure 3.5](#), allows players start new single-player games.

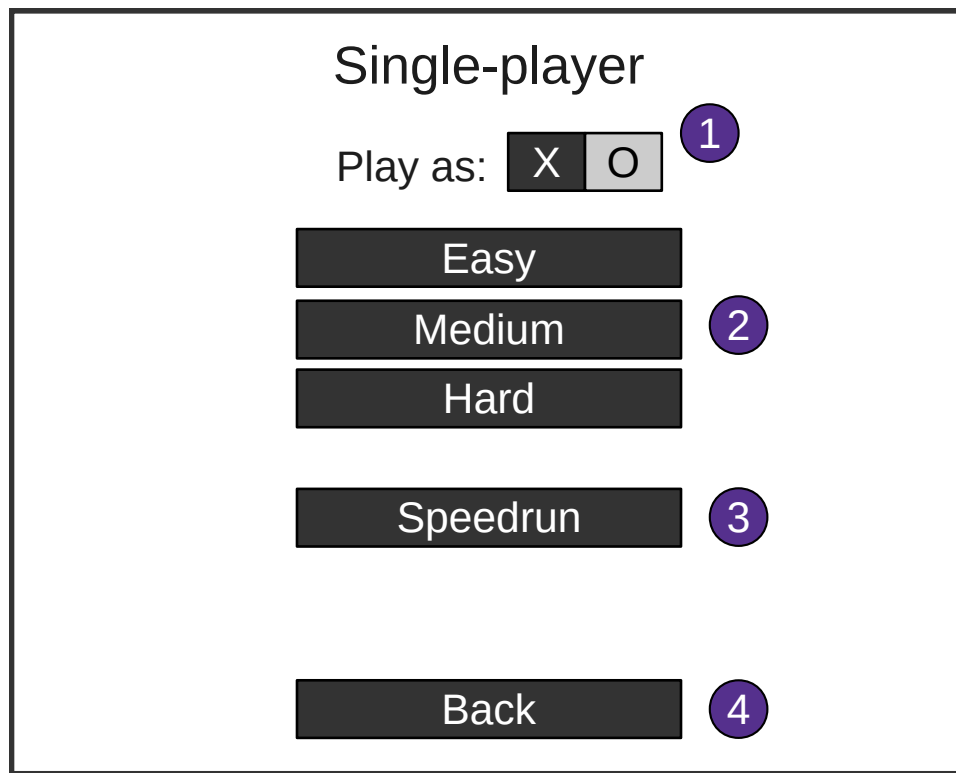


Figure 3.5: Single-player menu concept drawing.

1. The *Play as* selector allows players to select the mark they wish to use throughout the games.
2. The difficulty buttons select the difficulty then start a new single player game. Selecting one of these buttons closes the menu and launches a new single-player game with the requested settings.
3. The *Speedrun* button navigates to the [Speedrun](#) menu.
4. The *Back* button returns to the main menu.

### 3.3.4 Speedrun

The speedrun menu allows players to start a new speedrun and view best times of previous runs. [Figure 3.6](#) shows the speedrun menu.

The speedrun menu contains the following items of interest:

1. Instructional text that provides a short overview of the speedrun rules. Once the run is completed this text is replaced with the run's result and invites the player to play again.
2. *Start* begins the run. This navigates to the [Speedrun](#) game board.
3. Table of previous best times sorted from fastest to slowest.
4. The *Back* button returns to the main menu.

When the speedrun begins, the game board is shown, a prominent three second countdown is show, and dramatic music starts to swell. Once the timer elapses the run begins.

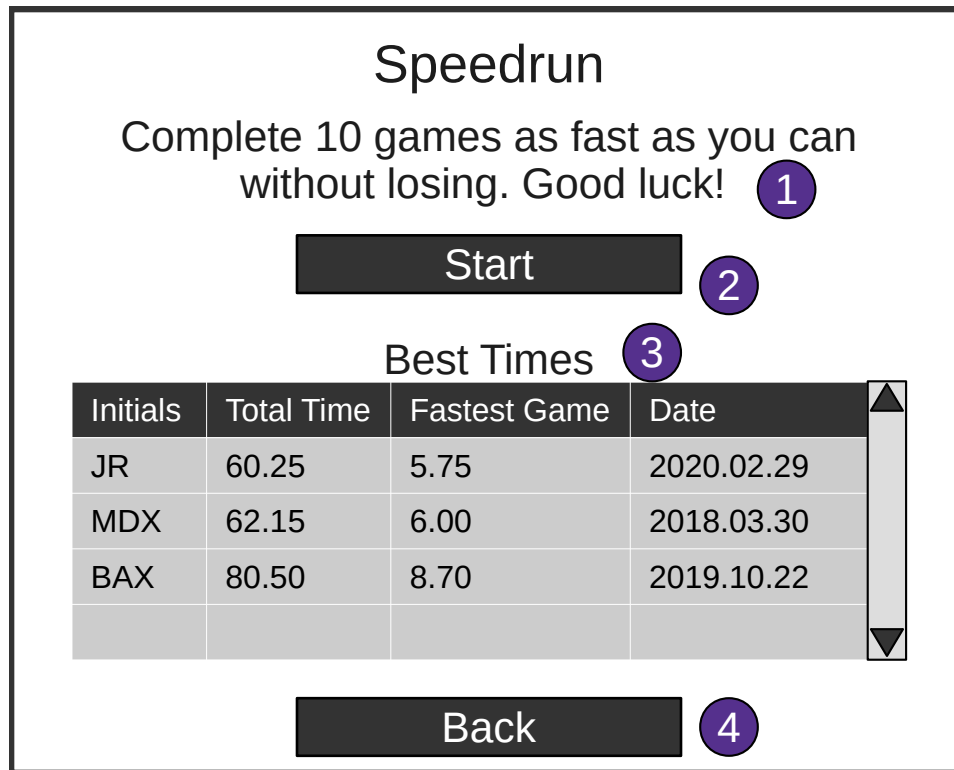


Figure 3.6: Speedrun menu concept drawing.

Once the run is completed the speedrun menu is displayed and shows the result of the run.

If the player gets a new best time the dialog shown in [Figure 3.7](#) is presented to the user.

The best time dialog contains the following items:

1. The speedrun time time.
2. The *Initials* text box allows players to enter their initials so their best time is differentiated from other players that happen to use the same computer. The field remembers the last set of initials entered to save players from having to retype their initials.
3. The *Close* button hides the dialog allowing the speedrun menu to be fully visible.

### 3.3.5 Options

The options screen contains all of the game's player configurable options. [Figure 3.8](#) shows this screen.

1. Music and Sound FX sliders to control the volume of these items. This allows players to mute some or all of the in-game sounds.
2. *Reset to Defaults* rests all options to their default values.
3. The *Back* button returns to the main menu.



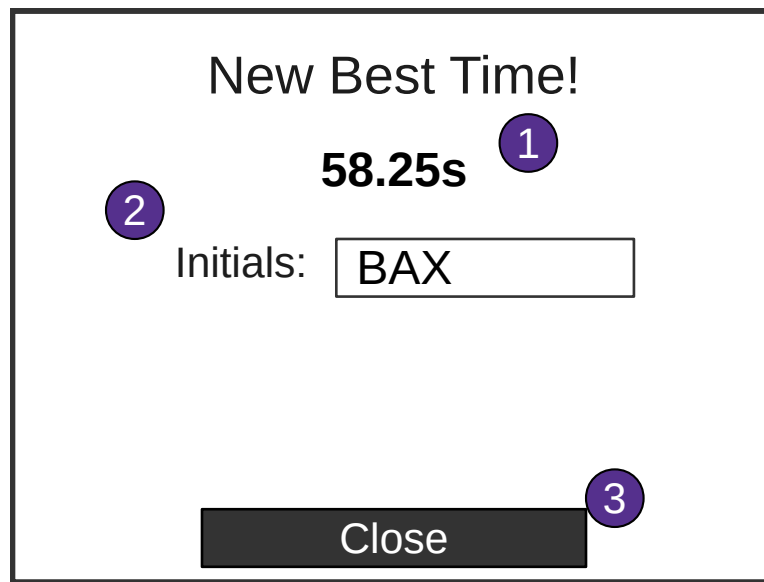


Figure 3.7: Speedrun best time dialog concept drawing.

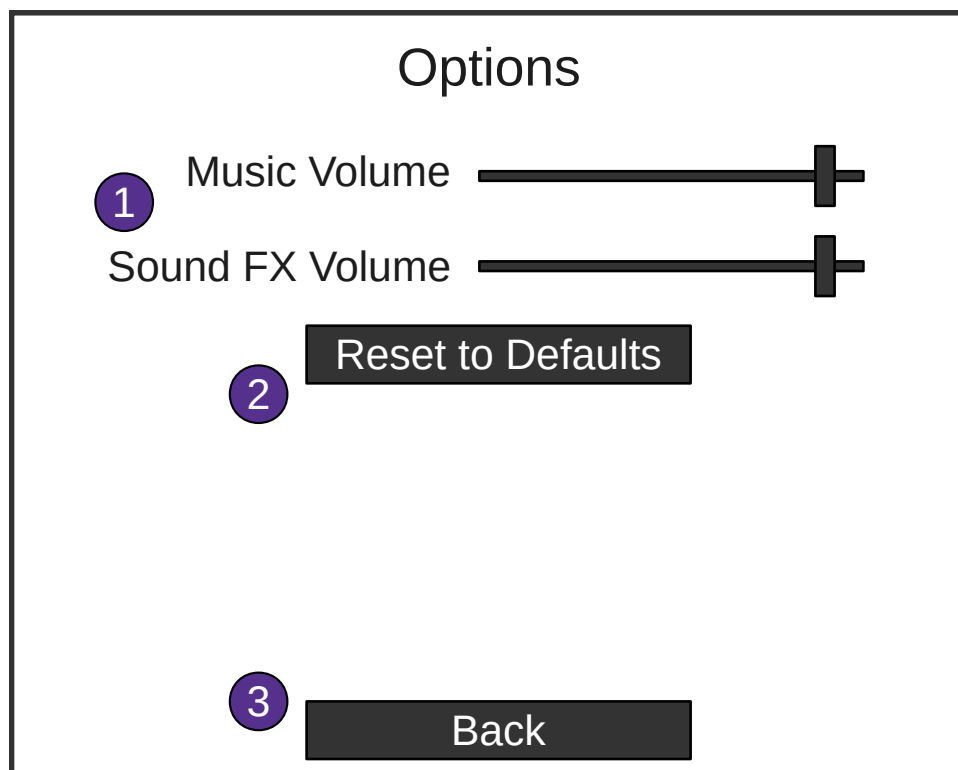


Figure 3.8: Options screen concept drawing.

### 3.3.6 Credits

The credits screen displays information the game's developers and helps fulfill the *Third Party License Compliance* obligations. The credits screen is shown in [Figure 3.9](#).

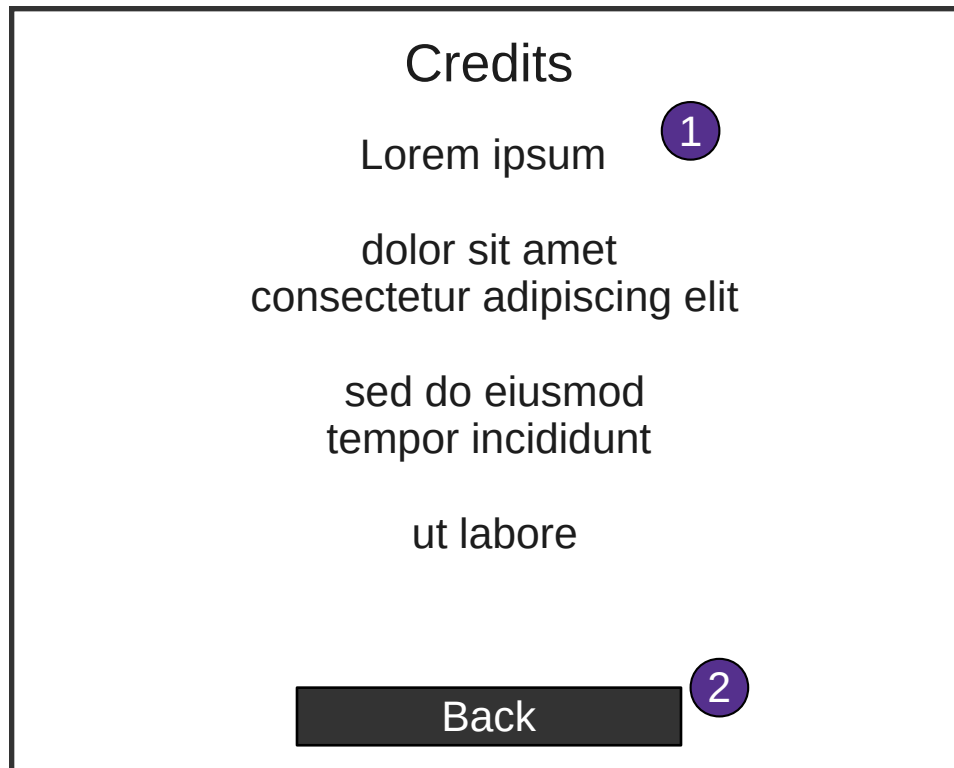


Figure 3.9: Credits screen concept drawing.

The credits screen contains the following items:

1. Scrolling list of developer names, third party assets, and other information about the game.
2. *Back* returns to the [Main Menu](#).

The credits screen uses a different background and soundtrack than the other menus. The background consists of one or more Tic Tac Toe games being played in a variety of environments. Each environment is clearly visible — blurring and other effects are not used on this screen. The environments are changed several times per game. This showcases the many environments of the game.

The credits screen has its own sound track. The music and sound FX of the individual environments are not used.

Once all of the credits have played the screen remains open with Tic Tac Toe games being played in the background.

### **3.3.7 Loading Screen**

If necessary for technical reasons, the loading screen provides feedback to the player while assets are loaded. This screen is only shown when the game first loads.

### **3.3.8 Help**

The help screen provides information on how to play Tic Tac Toe, the application version, how to report bugs, and other information. All information is hosted locally; no internet access is required.<sup>9</sup>

## **3.4 Screen Flowchart**

The flow chart in [Figure 3.10](#) visually shows how the screens and menus are connected.

---

<sup>9</sup> The *Provide Free of Charge and Under an Open Source License* objective mentions not tracking players. Websites often contain trackers, advertisements, and other items that violate this objective.

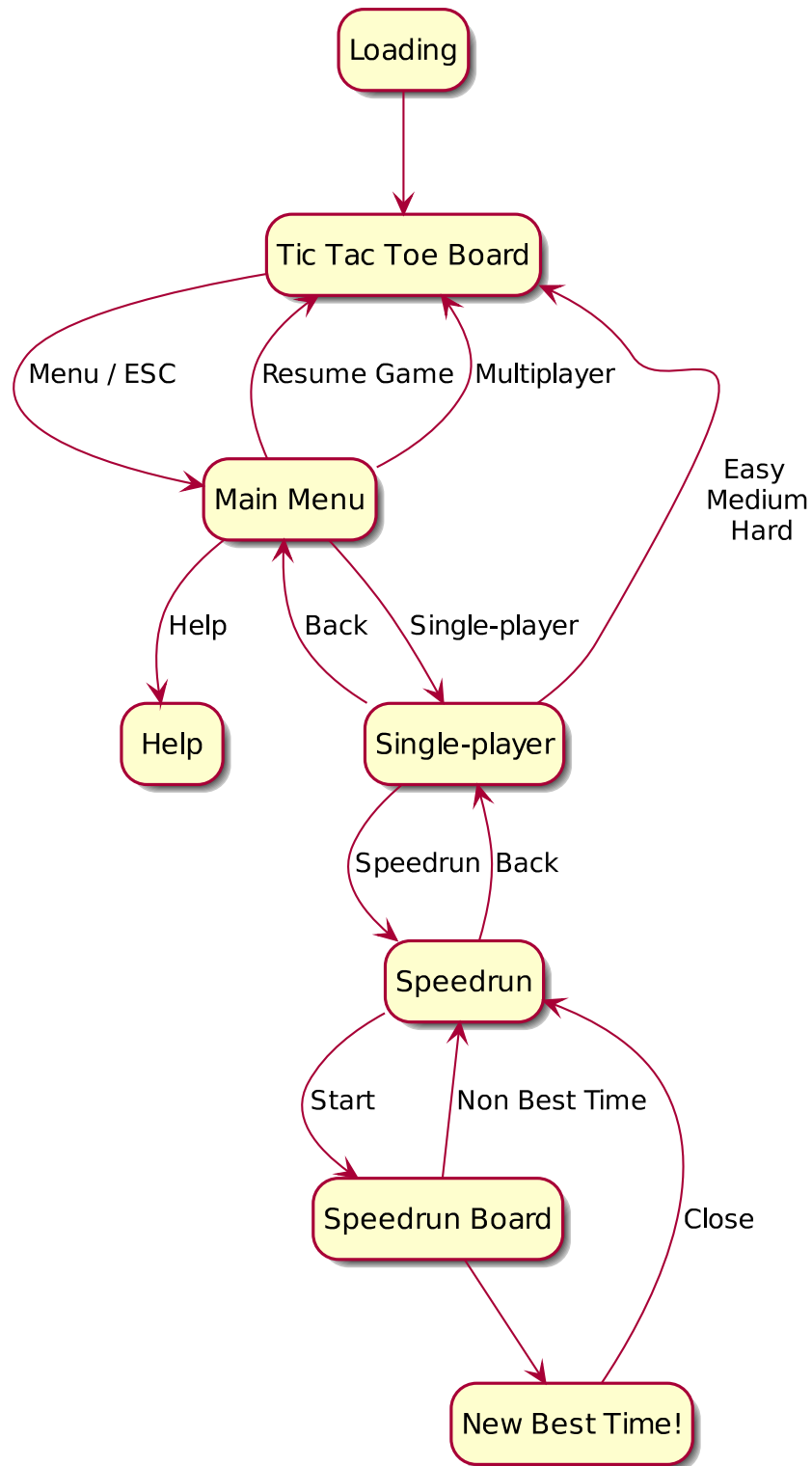


Figure 3.10: Connections between Tic Tac Toe's menus and screens.

## ENVIRONMENTS

One of the main defining features of this take on Tic Tac Toe is the use of multiple environments. Each environment has a strong visual theme and complementary soundtrack.

### 4.1 Environment Selection

Each game takes place on a different environment. Environments are selected using a shuffling algorithm.<sup>10</sup> This ensures players get to experience each environment in a random order.

### 4.2 List of Environments

This section lists each of the game's environments.

#### 4.2.1 Blueprint

##### Marks and Grid

The blueprint uses white text for the marks and white lines for the grid.

The grid can actually be a series of empty boxes but using thicker interior lines.

Hovering over an empty square shows a dotted version of what would be drawn there. Clicking places the mark without further animation.

Winning the game causes a “Winning Marks” label to be placed with arrows pointing to the corresponding marks.

##### Background Scenery

The background is predominantly deep blue. Additional details are included in white.

There is a title block with the game's name, author, version number, etc.

Measurement lines indicate the unit less distance between features.

The grid, squares, marks, etc are pointed to by arrows with labels. Additional notes describe the music being played or point out the key bindings.

---

<sup>10</sup> Wikipedia's [Fisher–Yates shuffle](#)<sup>11</sup> article contains details on various shuffling algorithms.

<sup>11</sup> [https://en.wikipedia.org/wiki/Fisher%E2%80%93Yates\\_shuffle](https://en.wikipedia.org/wiki/Fisher%E2%80%93Yates_shuffle)

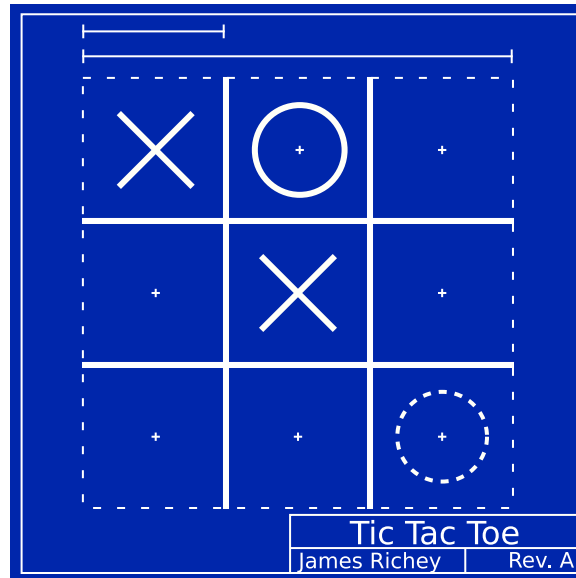


Figure 4.1: Concept art for the Blueprint environment.

### Music Theme

Background or thinking music

### Additional Details

This environment provides an opportunity to point out features of the game that might be overlooked by players such as keyboard support.

This environment makes a template when creating new environments. This is due to it literally being a blueprint.

## 4.2.2 Notebook Paper and Pencil

### Marks and Grid

The marks and grid are dark graphite from a #2 pencil. Each mark is slightly different and the grid lines are not perfectly straight to give it a hand drawn look. Likewise, mark placement is animated like one writing an X or O while a corresponding pencil sound FX is played.

A line is drawn through the winning marks.

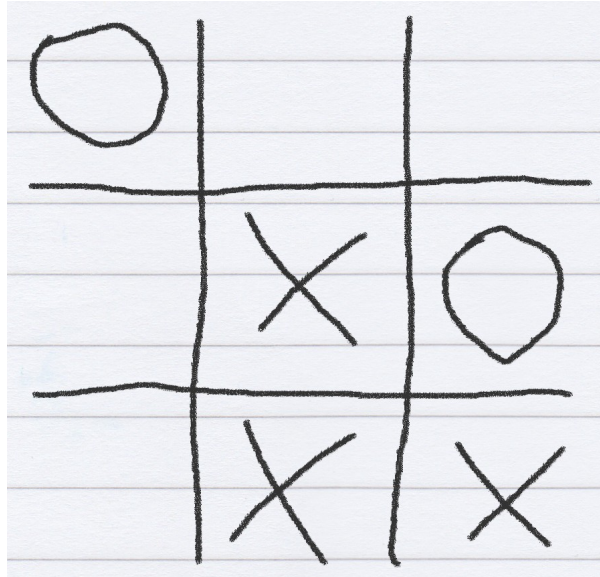


Figure 4.2: Concept art for the Notebook Paper and Pencil environment.

### Background Scenery

The background consists of notebook or engineering paper. Notebook paper is white with gray or blue lines. Often times there also a vertical red margin line.

Engineering paper is yellow with gray grid lines.

### Music Theme

Mild background music

## 4.2.3 Pen on Scrap Paper

### Marks and Grid

The marks and grid are created from blue or red ink. Like the *Notebook Paper and Pencil* environment, each mark is hand drawn with corresponding animations and sound FX.

A line is drawn through the winning marks.

### Background Scenery

Various bits of scrap paper are used for the background. This can include but is not limited to bank deposit slip, various store receipts, or any other paper someone might have in their purse.

Text on the paper should be gray, faded black, or otherwise not interfere with the visibility of the marks.

### Music Theme

Something with a gospel / churchy vibe

### Additional Details

The mark / grid color and the background image is randomly selected each time the environment is played. Blue and red ink is used so the marks stand out from the

The inspiration from this environment comes from me not paying attention in church and drawing on scrap paper my mom happened to have.

### 4.2.4 Neon Lights

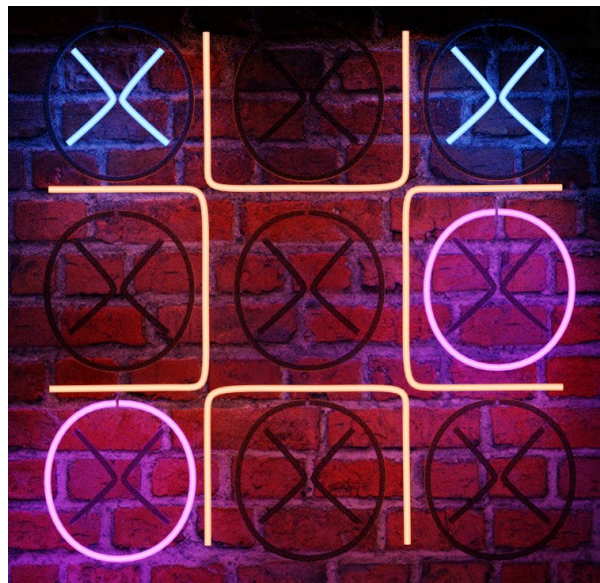


Figure 4.3: Concept art for the Neon Lights environment.

### Marks and Grid

The marks and grid are created from luminescent orange, pink and / or blue neon lights. Selecting a square causes the mark to illuminate with a flickering animation and an electric buzzing sound FX.

All winning marks flash on and off with a mild electric buzz sound FX.

When the environment first loads in non-speedrun mode the lights are not illuminated. Then the grid flickers on as if the transformer is starting to fail.



### Background Scenery

The background is a brick wall holding the neon lights. Ideally, the wall is illuminated by the lights. E.g. when a mark is selected the brick wall behind the mark gains some illumination as shown in the concept art.

### Music Theme

80's electronic

### Additional Details

This environment provides an opportunity to show off the game's graphics including lighting FX.

#### 4.2.5 Early Computer

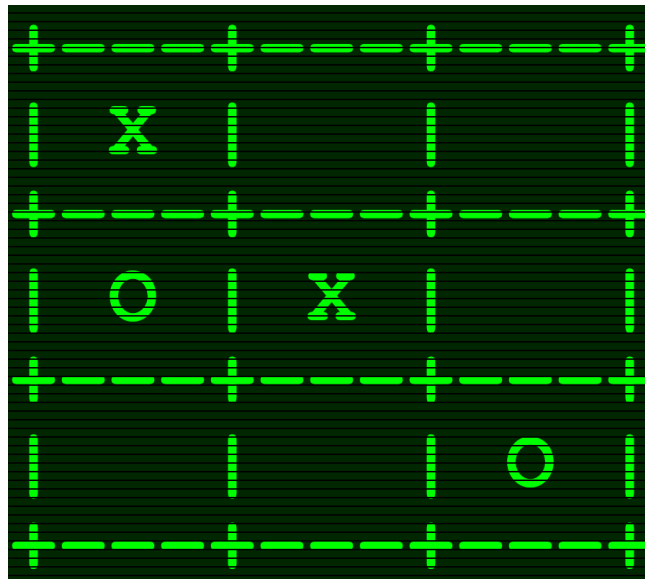


Figure 4.4: Concept art for the Early Computer environment.

### Marks and Grid

The marks and grid are green or amber ASCII characters displayed on a low resolution monochrome computer screen. E.g. individual pixels are distinguishable. The marks in winning squares blink on and off.

A blinking cursor is shown when players mouse hover over a empty squares. The cursor is the full width and height of a character, e.g like the modern insert cursor. A cliché beeping sound FX is played when squares are placed.

When the environment loads in non-speedrun mode each line of ASCII text is drawn as if the game was being loaded over a 300 baud modem.

### Background Scenery

The background is a monochrome computer monitor with the an off white bezel and red LED near the power button. The background color is dark green or dark amber with perhaps the dark pixels still visible.

### Music Theme

8-bit electronic

### 4.2.6 Sidewalk

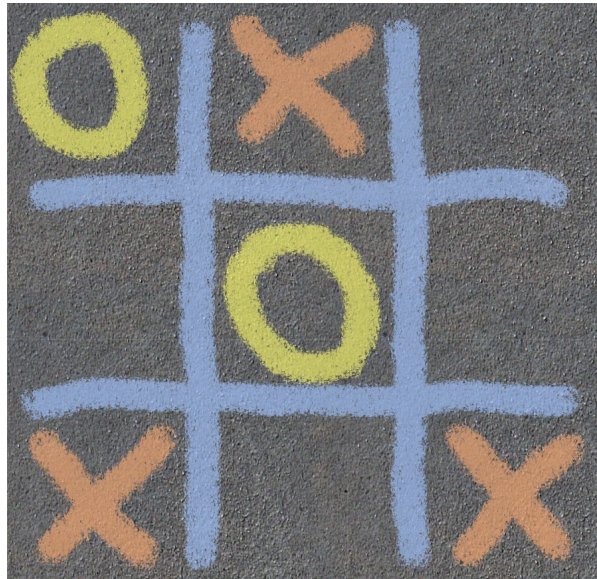


Figure 4.5: Concept art for the Sidewalk environment.

### Marks and Grid

The marks and grid are created from pastel sidewalk chalk. The lines are thick and hand drawn with similar animations as the *Notebook Paper and Pencil* environment.

Pastel colors include red, blue, yellow, pink, green, and orange. The X, O, and grid use different colors that are randomly selected each game.

A line is drawn through the winning squares.

### **Background Scenery**

The background is a gray sidewalk or black asphalt. Imperfections such as cracks and twigs are visible. There might even be bits of other chalk drawings visible.

### **Music Theme**

Hip Hob / R&B

## **4.2.7 Papyrus Paper and Fancy Calligraphy**

### **Marks and Grid**

The marks and grid are drawn from a quill pen with deliberate, fancy strokes. Thick black ink is used. The grid lines and marks are precise but the stroke pressure varies as the mark is being made, thus the lines are thinner near the starting and ending points.

Creating of the marks and grid are animated. A line is drawn through the winning squares.

### **Background Scenery**

The marks and grid are drawn on a faded tan papyrus paper. The paper's rough texture is clearly visible.

The Tic Tac Toe game could be part of a scroll or book. E.g. there is Latin text and one of those fancy first characters of the paragraph.

### **Music Theme**

Snooty classical music.

## **4.2.8 Bathroom Stall**

### **Marks and Grid**

The marks and grid are scratched into the bathroom stall divider. This reveals the metal underneath the paint. Each scratch is a fairly straight line but it takes several attempts to form a line thus there are "whiskers" hanging off the lines. The O marks are especially problematic resulting in malformed O's.

A metal scraping sound FX is played when the marks are being created.

A line is scratched through the winning marks.

### Background Scenery

The background is a pail blue, green, or gray stall divider. There is additional graffiti written in ink or scratched into the paint that one would find in a truck stop bathroom. Nothing inappropriate but adults should find it funny.

### Music Theme

Truck stop music; perhaps a country vibe.

### Additional Details

There is bathroom humor a Easter egg in this environment. A “straining” sound FX is played after a minute or two; past the length of a typical game. Think of that one scene from Austin Powers. The situation becomes more desperate as time goes on. Finally it resolves itself with a large plop / splash sound and a big sigh of relief.

### 4.2.9 Chalkboard

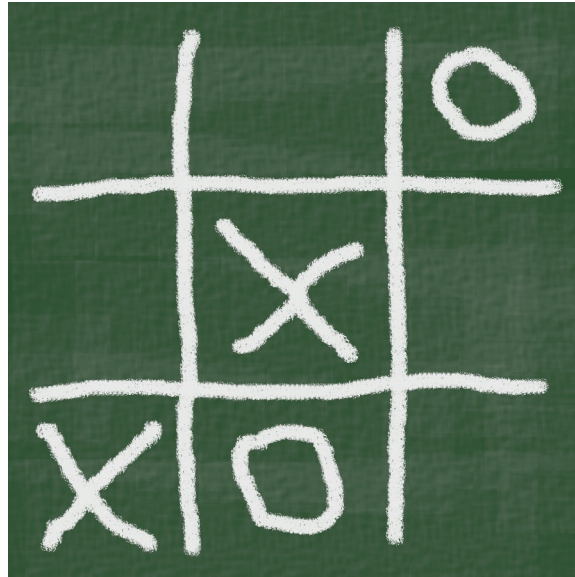


Figure 4.6: Concept art for the Chalkboard environment.

### Marks and Grid

The marks and grid are created from hand drawn white chalk. A stereotypical chalk sound FX is used when the marks are being drawn.

A line is drawn through winning marks.

### **Background Scenery**

A black or dark green chalkboard is used as the background. The chalkboard is well used; there is a lot of chalk dust marks left from the eraser.

To reinforce the chalkboard theme, the tray containing the eraser and extra chalk is visible.

### **Music Theme**

TODO

## **4.2.10 Whiteboard**

### **Marks and Grid**

The marks and grid are created from various colors of dry erase markers. Typical colors include, blue, green, black, red, and purple. The grid and each mark are hand drawn in different colors.

A squeaking sound is made when marks are being drawn.

A line is drawn through winning marks.

### **Background Scenery**

The background is a white dry erase board. The board does not erase well so faded physics equations is still visible in the background.

To reinforce the whiteboard theme, the tray containing the eraser and extra markers is visible.

### **Music Theme**

TODO

### **Additional Details**

Easter egg opportunity: selecting one of the extra markers changes the mark color of the current player.

## **4.2.11 Ancient Alien Ruins**

### **Marks and Grid**

The marks and lines are glyphs carved into rock with glowing light coming from the glyphs.

At first there is a grid, X, and O carvings in the rock. Mouse hovering over a free square causes some mild blue illumination. Placing the mark results in brighter illumination and the surrounding grid to illuminate too.

When the game ends in a win, the winning marks illuminate very brightly and even change color. The grid takes on a uniform illumination. A cats game or single player loss results in the grid to go dark and all the marks to go back to a mild illumination, as if the power has been cut and they are slowly fading away.

A strange alien sound FX is used when marks are placed.

### Background Scenery

The background is gray rock that would be found in a cave deep in the jungle. There are various glyphs carved into the rock. There might even be some moss, vines, or other deep green plants on the rock.

### Music Theme

SciFi

## 4.2.12 Dirty Car Window

### Marks and Grid

The marks and grid are drawn in the window's dust revealing the shiny tinted glass underneath. The marks are fairly fat and despite being hand drawn are mostly straight.

A mild swooshing sound FX is used when marks are drawn. A line is used to indicate winning marks.

### Background Scenery

The background is a dusty car window. The dust is a brown dirt color. Stickers such as the "my family" are trying to peak out from underneath the dust. The back windshield wiper reinforces the car theme.

### Music Theme

Hip Hop with rattling door sounds.

## 4.2.13 16x4 LCD Display

### Marks and Grid

The marks and grid are black or white LCD pixels. Black is used with the green background and white is used with the blue background.

Mousing over a free location causes a cursor, the underscore character (  ), to blink in that location.

Asterisk characters are placed next to winning marks.

### Background Scenery

The background is green or blue backlit 16x4 LCD display that is mounted on a green circuit board. There are other electrical components, wires, and even LEDs around.

### Music Theme

8 bit

### Additional Details

Some artistic liberty with regards to the 16x4 LCD display might need to occur to ensure the might need To ensure the mark's center points are in the correct locations.

## 4.2.14 Ancient Egypt Tomb

### Marks and Grid

The marks and grid are carved into the limestone wall background. A chiseling sound FX is played when marks are being carved. The grid and marks are chiseled with skill, that is they are straight and precise.

A line is chiseled through the winning marks.

### Background Scenery

A limestone wall covered in hieroglyphs is used as the background.

### Music Theme

Something with an ancient vibe.

## 4.2.15 Beach

### Marks and Grid

The marks and grid are hand drawn into sand at the beach. A light swooshing sound FX is played when marks are created.

A line is drawn through the winning marks.

### Background Scenery

Wet sand is used as the background. There are seashells and other shore debris.

### Music Theme

Tropical / Jamaican / steel drums.

### 4.2.16 RGB Led Grid

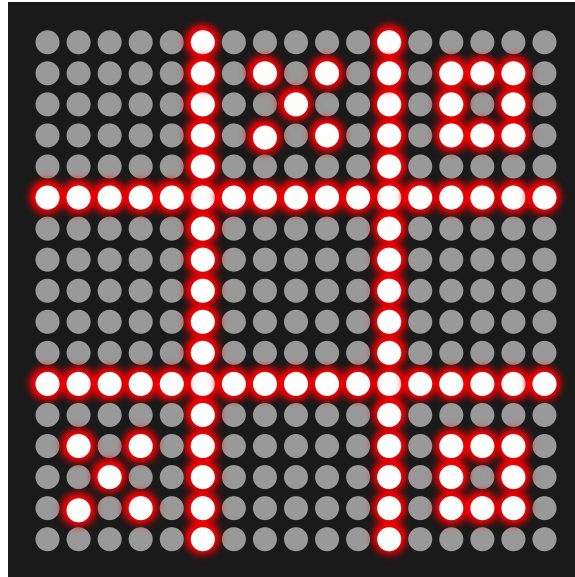


Figure 4.7: Concept art for the RGB Led Grid environment.

#### Marks and Grid

The marks and grid are displayed on a 64x32 RGB LED grid. The marks and grid each use their own colors. This includes primary and secondary colors.

Placing a mark displays an cheesy animation such as stacking up block to form the mark. Winning squares flash between normal and inverted color where the background is illuminated instead of the foreground.

#### Background Scenery

The background is the unilluminated LEDs and the plastic material that separates the LEDs. The LEDs are a lighter color than the almost black spacer material. Occasionally there may be various animations or other FXs played on the LEDs outside the play area.

#### Music Theme

Modern 8 bit or techno

### 4.2.17 Electroluminescence

#### Marks and Grid

The marks and grid are composed of either cyan or orange colored electroluminescent wire. X is assigned one color, and O the other. Placing a mark causes the border of the grid to pick up the marks color.

All of the marks and grid take on the winner's color after a brief flashing transition.



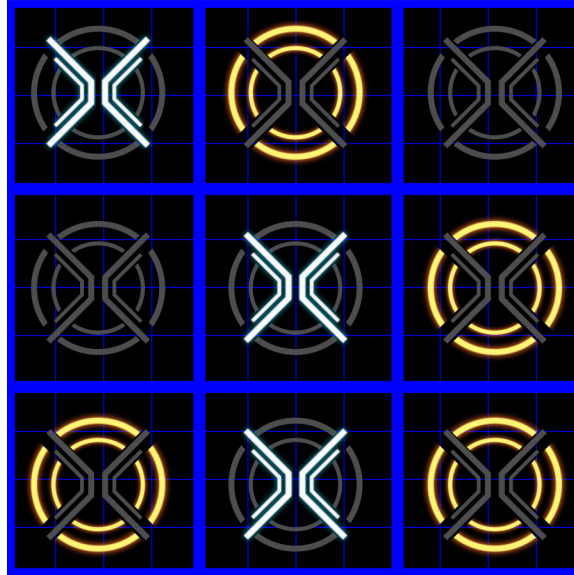


Figure 4.8: Concept art for the Electroluminescence environment.

### Background Scenery

Black with a faint blue or cyan grid pattern. Pulses of light could occasionally travel down the grids.

### Music Theme

Techno

## 4.2.18 Crayon

### Marks and Grid

The marks and grid are hand drawn with different colors of crayons.

A line is drawn through the winning marks.

### Background Scenery

The game is played on drywall. A light switch or electrical outlet is visible to reinforce the location. There are other rudimentary drawings of flowers, houses, or trucks visible.

### Music Theme

Background

#### 4.2.19 Oil Painting

##### Marks and Grid

Thick oil based paint forms the marks and grid. The brush strokes are clearly visible. Strokes are deliberate resulting in fairly straight lines. Stereotypical paint stroke FXs are used where lines from individual bristles are visible as the stroke is letting up.

Various vibrant colors are used; the marks and grid each get their own color.

A line is drawn through the winning squares.

##### Background Scenery

The painting is being drawn on white canvas. The texture is clearly visible. Part of the wooden easel is also visible.

### Music Theme

Phone hold music, e.g. light modern classical.

#### 4.2.20 Bacteria

##### Marks and Grid

The grid is composed of rod bacteria viewed under a microscope. The X and O marks are likewise formed from bacteria of the corresponding shape.

The bacteria can appear purple in color due to Gram staining or as red.

The winning squares are marked by the bacteria inside the multiplying rapidly.

##### Background Scenery

The background is the fluid the bacteria is living in. There are imperfections in the fluid that seem to wander around, many out of focus.

### Music Theme

TODO

### **4.2.21 App**

#### **Marks and Grid**

The marks and grid are made up of simple clean lines. The marks are solid red or blue. The grid is solid gray in color. Animations are not used or used lightly.

A line is drawn through the winning marks. The line is the same color as the marks.

#### **Background Scenery**

The game is being played on a smart phone that is rotated 90 degrees. The background of the play area is solid white.

Additionally, the phone's bezel, speaker, and status bar are all visible. The status bar includes signal level, power level, and time on the right side. The left contains annoying advertisements that advertise fictional products, poke fun at data stealing apps, and promote the other environments in the game. The advertisements are rotated 90 degrees like the phone.

#### **Music Theme**

Casual

#### **Additional Details**

There is an Easter egg opportunity with the advertisements where clicking on one advertising a different level switches the environment to that level.



## ADDITIONAL CONSIDERATIONS

This chapter contains additional information about Tic Tac Toe.

### 5.1 Target Audience

Tic Tac Toe is targeted at casual gamers. There are three general groups that might find the game interesting: new PC gamers, bored adults and modders.

#### 5.1.1 New PC Gamers

Tic Tac Toe can be played by young or older players that do not have much PC gaming experience. The game requires basic strategy and has simple inputs. As players become more comfortable with the game mechanics they can try *Speedrun Mode*.

#### 5.1.2 Players Filling Time

Office workers or people stuck on an airplane can play Tic Tac Toe to make the time pass. The regular game modes do not demand much attention so players can let their minds wonder while playing. The variety of environments should keep players entertained for at least a little while.

#### 5.1.3 Computer Science Students / Modders

Tic Tac Toe is open source and is developed so that others can add their own environments or extend the game. Tic Tac Toe is a simple, but non-trivial application where people learning to program might find it an interesting codebase to examine.

### 5.2 Target Platforms

Tic Tac Toe runs on a variety of popular desktop operating systems including Windows, macOS, and Linux. Being a casual game, it does not require a beefy video card or fast processor. [Table 5.1](#) lists the minimum system requirements.

Table 5.1: Tic Tac Toe minimum system requirements

OS	Windows 10, Linux <sup>12</sup> , macOS
Processor	1GHz dual core
RAM	4GB
Disk	500 MB available space
Graphics	OpenGL compatible graphics adapter
Display	800 x 600 pixel display or larger

## 5.3 Licensing and Distribution

This section describes how the game is licensed, distributed, and other related details of getting the game into player's hands.

### 5.3.1 Game's License

The game is dual licensed under the MIT<sup>13</sup> and Apache-2.0<sup>15</sup> licenses. This is the same licensing scheme used for the Rust language.<sup>17</sup> Under these licenses users are free to obtain, modify, and redistribute the source code for both private and commercial use.

Any original game's assets are licensed under the CC-BY-4.0 license.<sup>19</sup>

### 5.3.2 Distribution

The game is directly distributed to players via the internet. There are two main ways to acquire a copy of the game: downloading an platform specific package, or building the game from source.

#### Platform Packages

Platform specific packages can be downloaded from the game's website. In particular, Windows users expect an installer to install the game on their system. They do not have readily access to compilers or other tools to build software from source. Finally, Windows is the most widely used operating system so a Windows installer is a requirement for this project to be successful.

Linux and Mac users have easier access to compilers but they still appreciate packages for their convenience. Packages for these platforms are created on a best effort basis.

---

<sup>12</sup> Tic Tac Toe is tested on the latest releases of Debian and Fedora. It should work on other popular distributions including Ubuntu. However, testing on other distributions is outside the scope of this project.

<sup>13</sup> Choose an open source license - MIT License<sup>14</sup>

<sup>14</sup> <https://choosealicense.com/licenses/mit/>

<sup>15</sup> Choose an open source license - Apache License 2.0<sup>16</sup>

<sup>16</sup> <https://choosealicense.com/licenses/apache-2.0/>

<sup>17</sup> See Rust's README<sup>18</sup> for licensing details.

<sup>18</sup> <https://github.com/rust-lang/rust#license>

<sup>19</sup> Choose an open source license - Creative Commons Attribution 4.0 International<sup>20</sup>

<sup>20</sup> <https://choosealicense.com/licenses/cc-by-4.0/>

### Build from Source

Because the game is open source anyone can build the game from source.

There are two main ways to build the game from source:

1. Clone the game's source code repository and follow the build instructions.
2. Use Cargo, the Rust package manager, to download, build, and install the game and its dependencies.

### 5.3.3 Monetization

There is no intention to generate revenue from Tic Tac Toe. The is released free of charge. There are no advertisements, micro-transactions, or data stealing spyware that is found in other *free* games.

### 5.3.4 Third Party License Compliance

Tic Tac Toe makes extensive use of third party software libraries and game assets to help reduce the development effort. To use these resources, the game must comply with their various licensing terms. Additionally, the game can only use any third party resources that have licenses that are compatible with the game's license.

#### Software Libraries

Rust libraries that have the following licenses can be used by the game:

- MIT
- Apache-2.0
- The Unlicense

Additionally, the game can dynamically link to libraries with the above licenses plus libraries that are licensed under the LGPL.

A software bill of materials is included in the game's user documentation. This ensures the game complies with the required copyright notices. Also, users can see exactly what is included in the software.

The software bill of materials includes the following information about each library:

- Name
- Version number
- License
- Link to website or source code repository

#### Game Assets

The game can freely use or remix third party music, sound FX, and textures that have the following licenses:<sup>21</sup>

- CC0
- CC-BY

A list of third party assets used is accessible from the game's credits or user documentation to fulfill the attribution requirements. The following information is included for each asset:

---

<sup>21</sup> The CC-BY-SA is incompatible with the the game's licensing terms. It is this author's understanding that a game is considered a derivative work and thus must also be licensed under CC-BY-SA.

- Title - title of the work.
- Author - name or handle of the author.
- License - the specific CC license the work is published under.
- Copyright notice - some works include a copyright notice supplied by the author that must be included in the attribution.
- Source - link to website the asset was obtained from.
- Is derivative - Mention if the work was modified from the original and provide a short summary of changes.

### 5.3.5 Internationalization and Localization

The game is initially released in American English. However, localizations are accepted from contributors to include in future releases. This helps the game be more accessible to a wider audience. To make it easy to add various localizations, the game is designed with internationalization support.

## 5.4 Similar Games

Tic Tac Toe is a popular paper and pencil game that has been played since the times of ancient Egypt. Sandy Douglas' 1952 take on Tic Tac Toe, called OXO, was one of the first computer games.<sup>23</sup>

However, the major inspiration for this variant of Tic Tac Toe comes from James Richey's 2004 [Tic Tac Toe](#)<sup>22</sup> shown in [Figure 5.1](#).

James developed his version of while learning the C++ programming language. Tic Tac Toe provides a fun challenge for learning a new language: it requires both algorithmic and graphical components to implement a small but nontrivial application.

When thinking of a project to use to help learn the Rust programming language, creating a Tic Tac Toe game is a fun choice.

## 5.5 Future Enhancements

There are some additional features that can be considered for future versions of the game. These are described below.

### 5.5.1 Achievements

Achievements provide players goals outside the usual *Gameplay* rules. These provide more depth to the game and encourage players to try different approaches to the game.

A future enhancement is to add achievements to the game. Some achievement might include the following.

- Win a game
- Play a game in every environment
- Win a hard game
- Win a game on every environment

---

<sup>23</sup> See Wikipedia's [Tic-tac-toe](#)<sup>24</sup> article for additional information about the history of Tic Tac Toe.

<sup>24</sup> <https://en.wikipedia.org/wiki/Tic-tac-toe>

<sup>22</sup> <https://www.imaginaryphase.com/ttt.html>



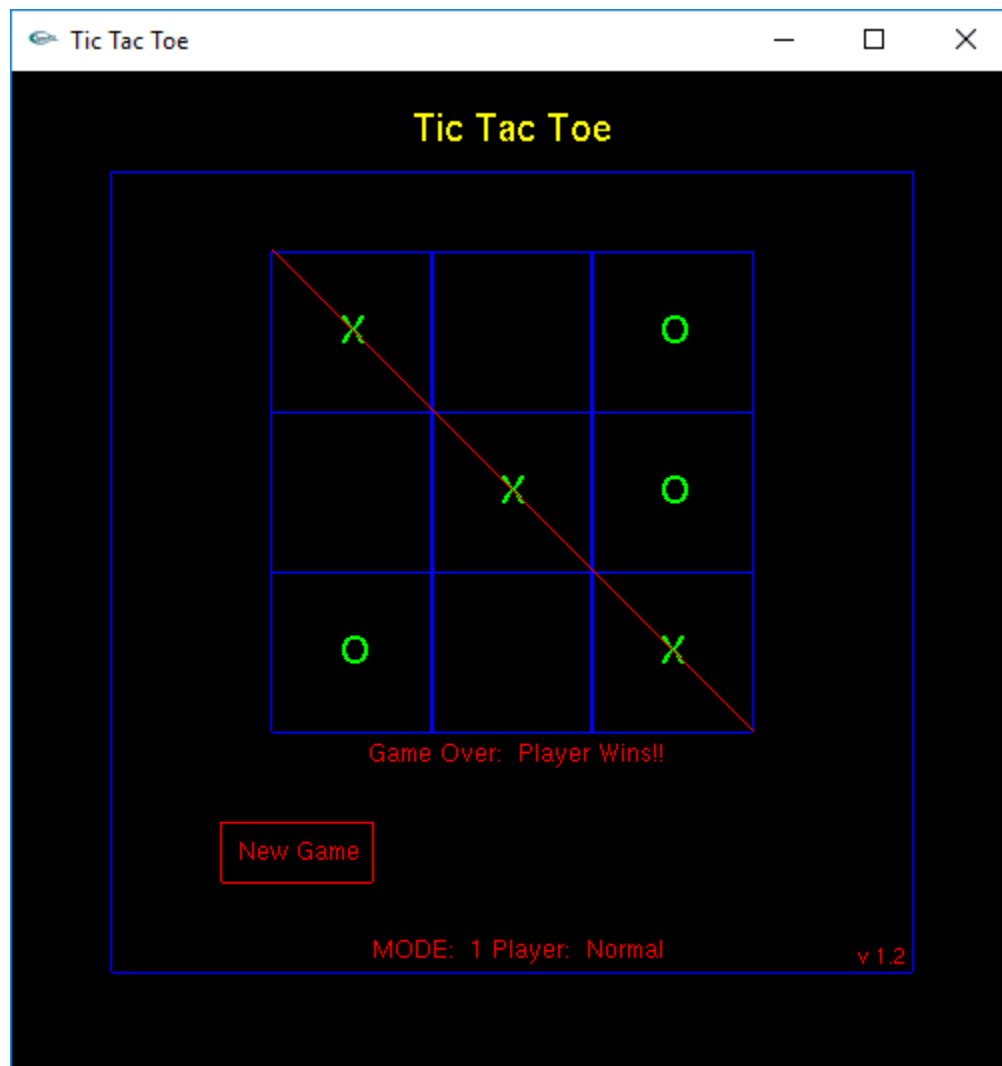


Figure 5.1: Screen shot of James Richey's 2004 Tic Tac Toe.

- Win a hard game on every environment
- Complete a speed run
- Complete a speed run in less than X seconds
- Complete a speed run in less than 1 second!!

### 5.5.2 Additional Environments

A major focus of this version of Tic Tac Toe is the ability to play on a variety of *Environments*. Future versions of the game can add even more environments. Some environments to consider are listed below.

- Lasers. Perhaps a laser cutter or some other use of lasers.
- Shower door or bathroom mirror.
- Futuristic dystopian or cyber punk theme.
- Some playgrounds have a 3x3 grid of squares for playing Tic Tac Toe.
- Winter / snow / frozen lake theme.
- Food theme.
- Light board, like those found in ships or submarines.
- OXO oscilloscope throwback theme.
- Metal or glass forge with lots of glowing hot liquids.
- Clone of James' 2004 Tic Tac Toe.

**TECHNICAL DESIGN**



## GLOSSARY

**cat's game** Term used when a game of Tic Tac Toe ends in a draw where there is no winner.



## BIBLIOGRAPHY

[Rogers-2014] Scott Rogers (2014) *Level Up! The Guide to Great Video Game Design*.





## INDEX

### C

cat's game, [41](#)

cat's game, [5](#)