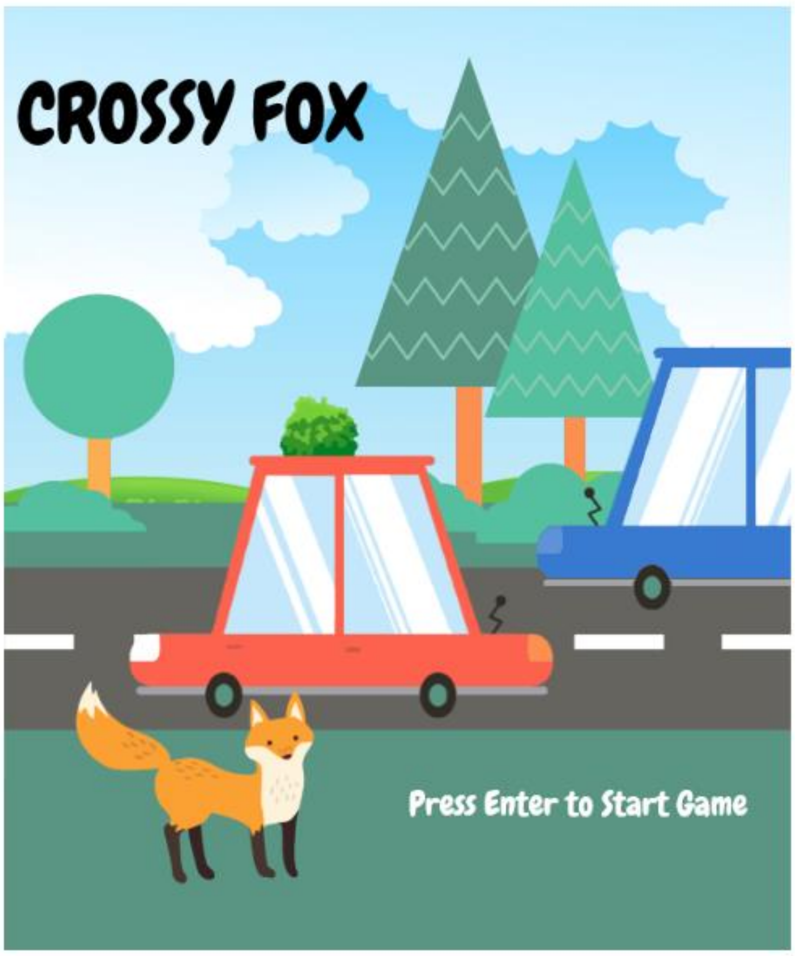
Crossy Fox: An Interactive Game Developed in JavaScript



Candidate : 105870
Computer Science BSc
Multimedia Design and Applications
2016
Word Count: 2100

Contents

Preface

This project was developed using the Mozilla Firefox (Version 46.0) web browser. Furthermore, it has been thoroughly tested using Internet Explorer 11, Google Chrome (For windows and Mac), Safari, and Microsoft Edge. For best results, play Crossy Fox in Mozilla Firefox.

The game is available to play online at http://davethings.com/Game.html

Please note: The game contains depictions of violence and some cases of (censored) bad language.

The project was developed without the use of any game engines or external code, except where explicitly stated.

1 Introduction

The following report details the functionality and implementation of an interactive game created using JavaScript, CSS and HTML5 Canvas. The game was created as a part of the Multimedia Design and Applications module with the following core requirements.

- Movement of an object using the keyboard arrow keys within the bounds of the display.
- Collision detection between the player controlled object and other objects,
- A life/health system.
- A video
- A scoring system

2 The game: Crossy Fox

In order to satisfy these core requirements, it was decided to build a game in the style of Crossy Road [1]. Gameplay involves controlling a fox character as it tries to cross a series of roads while avoiding the traffic. The player succeeds when they reach the forest on the far side of the roads. Along the way, the player can collect coins for added points and eat chickens to replenish their health.

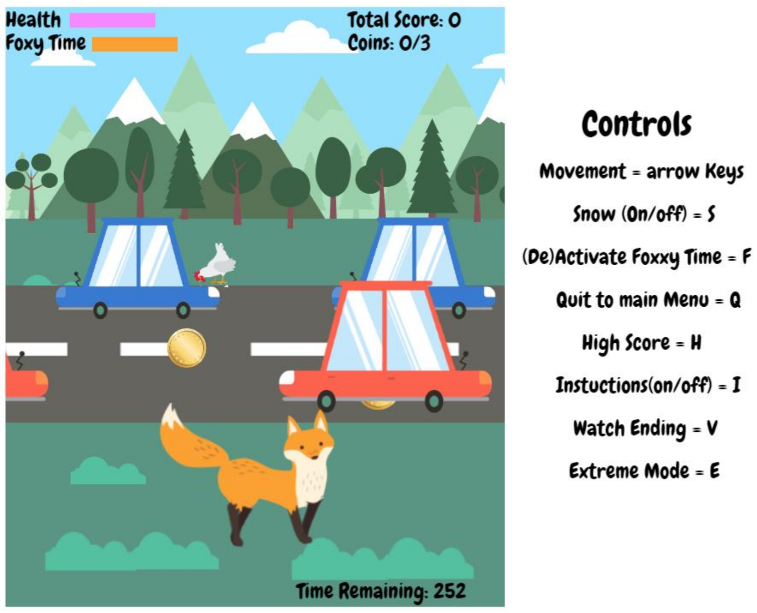


Figure 1: Screen-shot of the game (level 1). The image displays the playable fox character, collectable coins and chickens, and the vehicles which move back and forth across the screen. Also shown is the Game's HUD; displaying the players health, foxy time, total score, coins collected and time remaining.

2.1 Code overview: An object oriented approach

Implementation of Crossy Fox took an object oriented approach. Following a tutorial from phpied.com [2], it was discovered that functions could be treated as class constructors by using the "new" keyword. Methods could then be added by appending functions to the prototype of the constructor function. This process was used for the car, fox, coin, chicken, road, forest and snow objects.

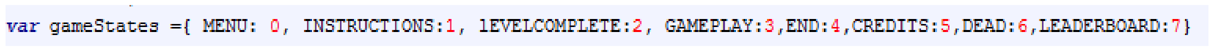
As an example, the car object is described in the table below.

Car Object
Variables:
Speed
Direction
Image
Start position
Current position
Size
Methods:
Check collision
Draw
Move

Figure 2: Figure showing the methods and variables associated with the Car object.

Game States

The game itself has a series of "states" that it can be in. the game can only be in a single state at any time. The game draws the correct information to the screen, and updates the necessary objects depending on its current state.



```
var gameStates ={ MENU: 0, INSTRUCTIONS:1, lEVELCOMPLETE:2, GAMEPLAY:3,END:4,CREDITS:5,DEAD:6,LEADERBOARD:7}
```

Figure 3: Possible game states in Crossy fox.

Level creation

Each level within the game contains pseudo randomized object locations[1], so no two levels are alike. Each level creates a certain number of road objects and pushes them on to a road array. Then for each road, a number of left facing and right facing car objects are instantiated. These too are pushed on to two separate car arrays. Finally, the coins and chickens are created and stored in separate arrays.

Each game loop, these objects are iterated through; their positions are updated, they are redrawn and they are checked for collisions.

Once a level is completed, all arrays are cleared and a new set of objects, for the following level, are instantiated.

All moving objects are limited to the bounds of the display, if an object goes over the bounds on one side, it reappears on the other side of the display.

[1] Objects have random positions within set bounds. For example, cars start each level with a minimum of one car width space between them.

3 Core features

This section discuss the core requirements and their implementation within the game.

3.1 Movement

The player controlled fox can move in all four directions and is controlled via the arrow keys. The code used for this was modified from the module lab classes [3]. In which all four directions have Boolean values, when the directional Boolean is true, the fox is moved in the respective direction. When the fox travels too far horizontally, it appears on the opposite side of the screen (wrapping). The Fox cannot travel over the top and bottom borders of the canvas.

3.2 Collision detection

Collision detection occurs between the playable fox character and the vehicles on the road, as well as with any collectables, the end of level boundary, and all screen boundaries. This collision detection code is a modified version of that taught in the module lab classes [3].

The hitbox for the fox and car are shown below. The hitboxes only cover a portion of each object to ensure collisions occur when the correct part of the fox collides with the correct part of the car. The player would not expect the fox to be hit by a car when it is not standing on the road itself.

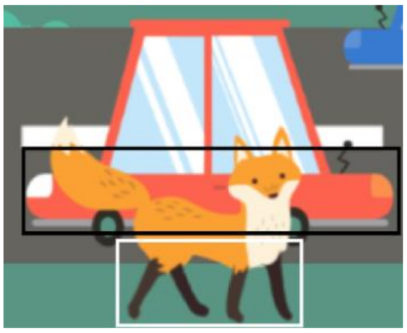


Figure 4: Diagram showing the hitboxes for the fox (white rectangle) and car (black rectangle). In this case, the objects do not collide.

3.3 Health system

The player has a health bar which depletes whenever the fox collides with traffic. This health bar is made up of a series of filled rectangles (one for every unit of health the player has). When a unit of health is depleted, the health bar decreases in length. Health can be replenished by eating chickens.

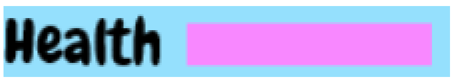


Figure 5: In game health bar.

Once depleted the player dies and is shown a game over screen and returned to the main menu.

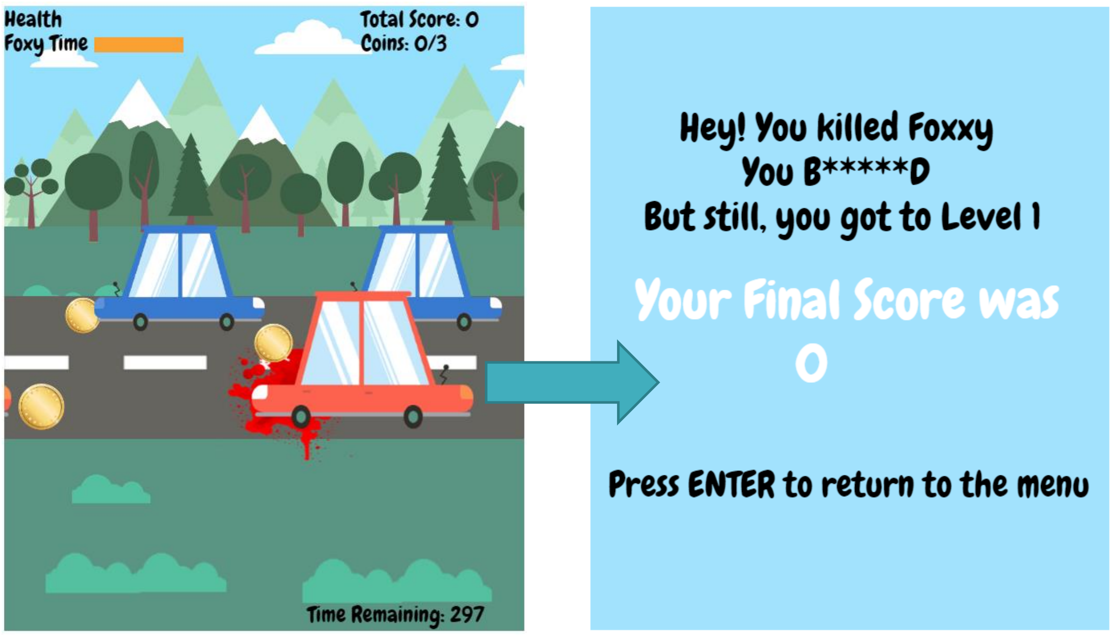


Figure 6: screenshot showing the player has zero health remaining and the fox has died. On the right hand side, the game-over screen is displayed.

3.4 Video

A video plays when the player completes the game[2]; originally, a YouTube video [4] depicting the ending to the game Starfox 64(Lylat Wars in the UK) [5]. This video has been downloaded and trimmed using editing software so that it begins at the correct place. When the video fades to black the game displays a "thanks for playing" screen. The player can then hit enter to return to the main menu. The video is encoded in both WebM and MP4 (H.264) formats to ensure it plays on all web browsers.

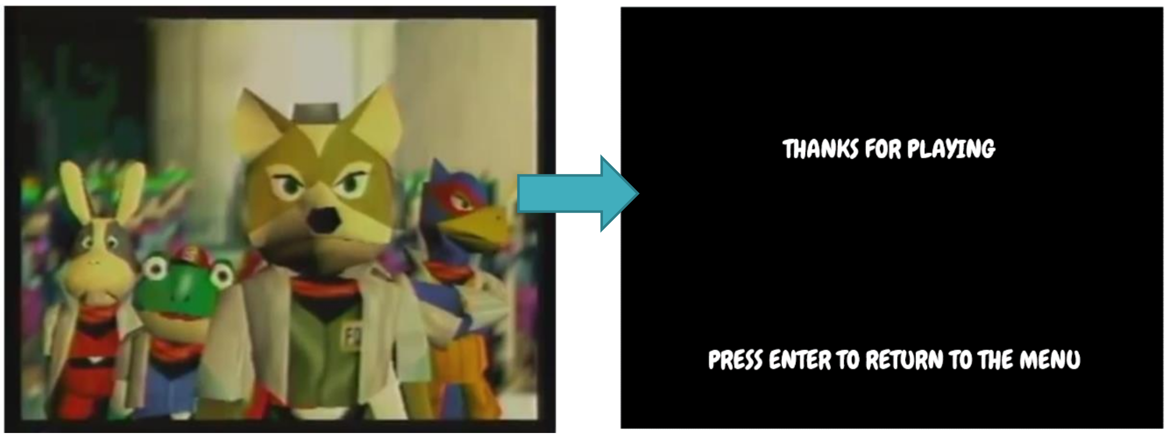


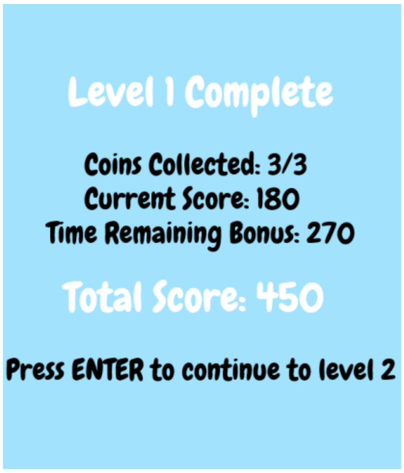
Figure 7: screen shot of the video that is played once the player completes the game. This is followed by the "thanks for playing" screen.

[2] The ending video can be watched by pressing "V" at any time, In case the user cannot complete the game.

The audio from the video continues to play until the user returns to the main menu, this ensures the is no jarring halt to the audio when the video fades to black.

3.5 Scoring system

The player is scored based on a number of factors; the speed in which they complete a level, the amount of coins they collect, and the number of chickens they eat. The player is awarded 50 points for every coin collected and 30 points for every chicken eaten. At the end of each level, the time remaining on the clock is added to the player's current score. As the player is given more time for progressively more difficult levels, the player will receive more points in the later levels of the game.



Level 1 Complete

Coins Collected: 3/3
Current Score: 180
Time Remaining Bonus: 270

Total Score: 450

Press ENTER to continue to level 2

Figure 8: screenshot of the level completion screen.

The score carries over from level to level, until the player either dies, or completes the game.

4 Additional functionality

4.1 Foxy time

In addition to the health bar, the player also has a "Foxy Time" bar. Enabling foxy (using the "F" key) time causes the traffic in game to slow down, while the fox remains at the same speed. This power up is akin to "bullet time" popularised by the movie "The Matrix" [6]. Time appears to slow down and the fox seems to become more agile. Foxy time can only be enabled for a certain amount of time on each level, denoted by the length of the foxy time bar.

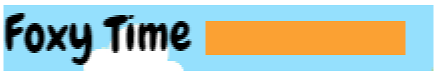


Figure 9: Foxy time bar, denoting the amount of foxy time remaining for the player to activate.

4.2 Multiple ways to die

In addition to reaching zero health from too many car collisions, if the player takes too long to complete a level they will be attacked and killed by an eagle. The eagle swoops in from off screen and kills the fox on collision with it.

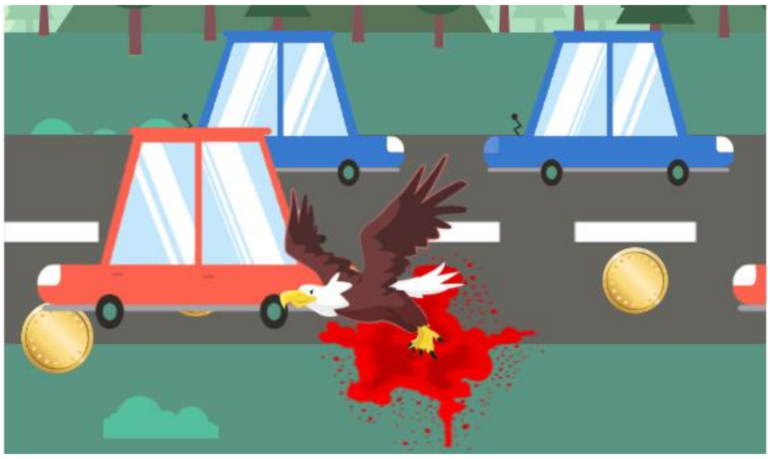


Figure 10: screenshot showing the eagle killing the fox as the time has run out.

4.3 Scaling: Images and speed

The size and speed of all objects within the game are scaled dynamically depending on their distance from the bottom of the screen. This gives the optical illusion of depth with objects further away appearing smaller and slower. The fox itself becomes smaller as it moves away and its speed decreases too. This scale depends on the number of roads in each scene, so a scene with many roads will appear to cover a much larger distance.

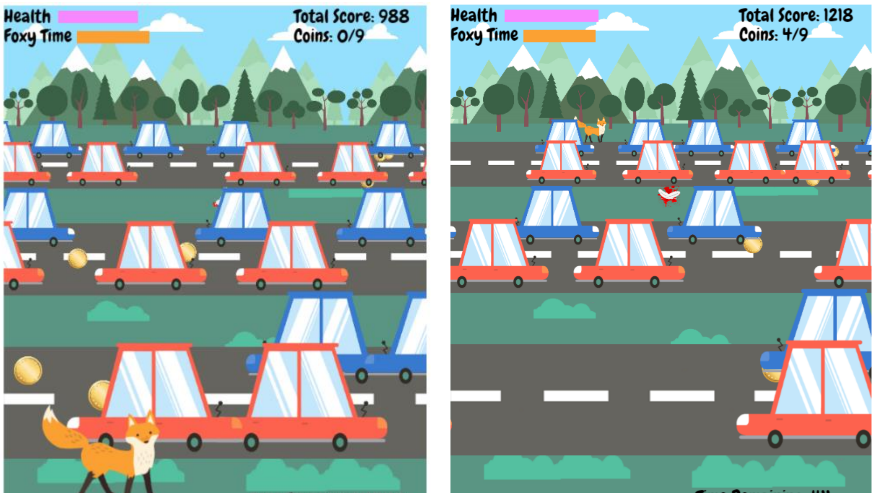


Figure 11: game images showing the fox getting smaller as it moves away from the camera.

4.4 Ensuring the fox is drawn behind the correct objects

To ensure the fox is drawn behind some cars and in front of others, a second round of collision detection occurs between the fox and the top half of the cars. If a collision occurs, the car is added to a new array of objects to be drawn after the fox has been rendered. The diagram below shows this in action.

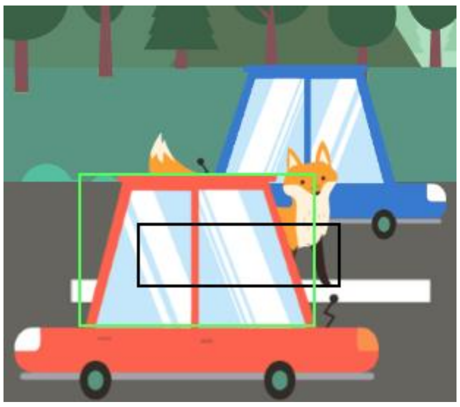


Figure 12Diagram showing the hit box of the car (green rectangle) and fox (black rectangle) colliding. This collision causes the car to be drawn in front of the fox.

4.5 Artwork and animations

The game contains 17 pieces of unique artwork, and eight additional sprite sheets (each containing 10 separate images)[3]. The fox itself has three separate visual stages; healthy, injured and badly hurt. To create the fox Sprite sheet, the legs were separated from the original fox image, and 10 separate images were created (for each sheet) with the legs in different positions. These were converted into a sprite sheet using a GIMP [7] plugin created by Brian Schultheiss [8]. Each stage of health has two sprite sheets (left facing and right facing) which are used to animate the fox as it walks. Every time the fox moves, a new frame of the sprite sheet is rendered, giving the impression of a moving image. All these images were created from one original fox image shown below.

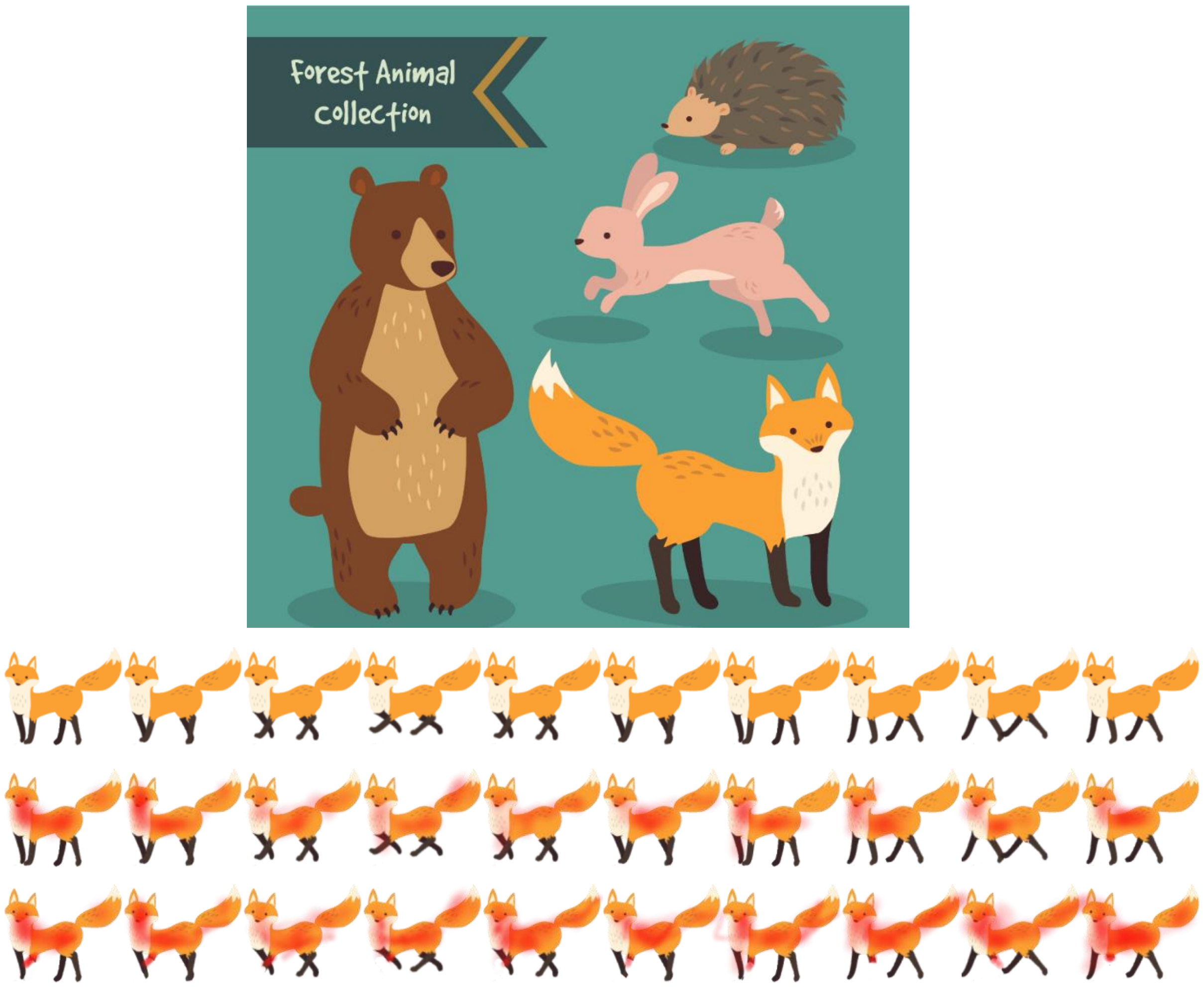


Figure 13: Original fox image (top) [18] along with the three (left facing) sprite sheets showing the three stages of health the fox can have.

[3] All of the original artwork, along with all the new pieces created from it, can be viewed in the appendix. All images were acquired from freepik.com, with the right to reuse with modification

4.6 Audio

Every action within the game is paired with its own piece (or pieces) of audio. The backing track consists of two pieces of audio that were cut to length, layered on top of each other and begin playing on loop at the start of the game. If the backing track has not loaded at the start of the game, a 2 second timeout is started, after which the audio is loaded again.

Below is a full list of actions and their associated sound effects. All sounds where cut to size from their original tracks, and the amplitude normalized.

Action	Sound
Game start	Backing track (birdsong [9] and uplifting melody [10])
Collision with car	Car horn [11] Fox Whimper [12]
Collision with coin	Classic "Mario-esque" coin sound [13]
Collision with chicken	Chicken "Pwarp" [14] Fox growl [15]
Eagle appears	Hawk cry [16]
Player dies	Mourning theme [17]
Level complete	"Yay" sound [18]
Activate foxy time	Ticking sound [19]

Figure 14: Table showing the sounds associated with actions within the game.

4.7 Snowy Mode

The view of the game can be enhanced at any time (even during gameplay) by enabling the snow mode. Enabling this mode creates a snow storm with moving particle effects. Furthermore, the roads and cars all become frosty. This mode is persistent between the main menu, each level and even the end game video.

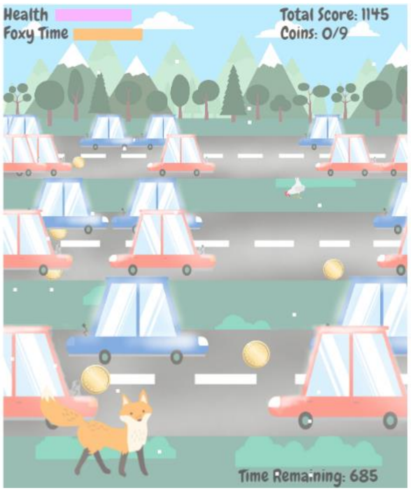


Figure 15: In game screenshots showing the activated snow storm; both during a level (left) and the end game video (right).

4.8 Increasingly harder levels and difficulty settings

The game features five unique levels, each progressively more difficult. The car, coin and chicken placement of each level is pseudo randomized to ensure each level is never exactly the same. When the player finishes a level they are shown a brief level completion screen displaying some level statistics. The user can then hit "enter" to start the next level.

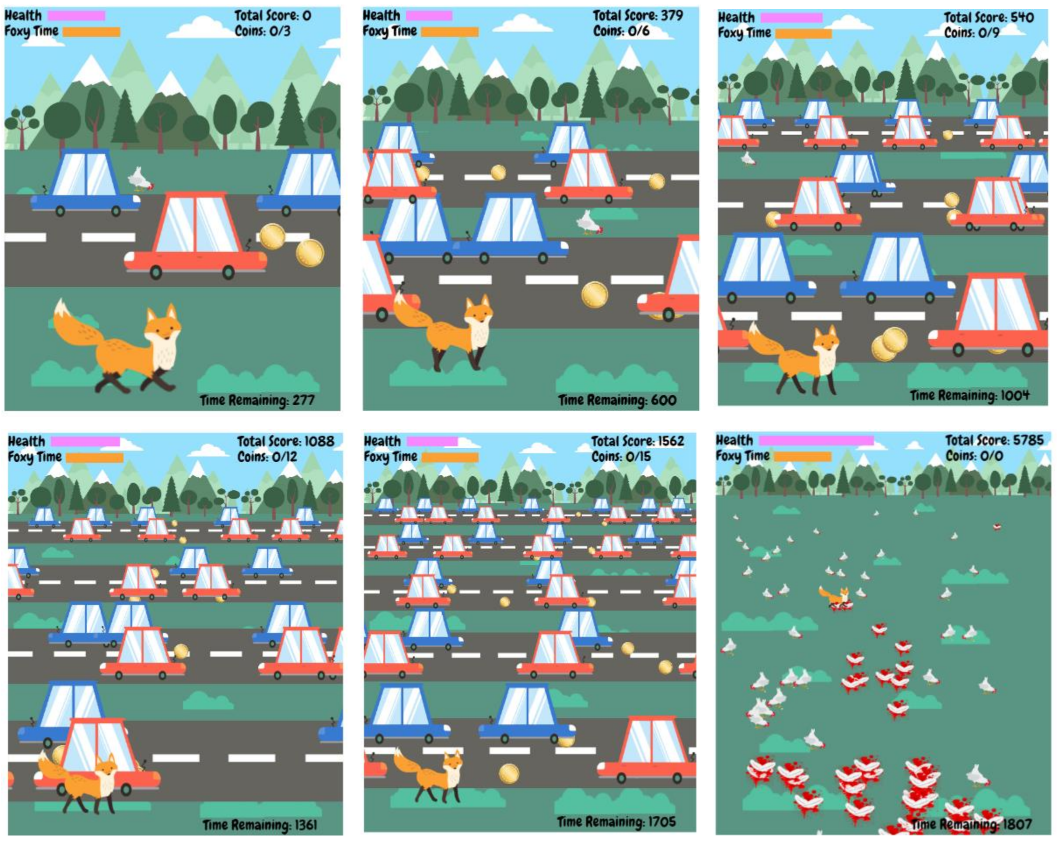


Figure 16: screenshots showing every level within the game, including the bonus level full of chickens.

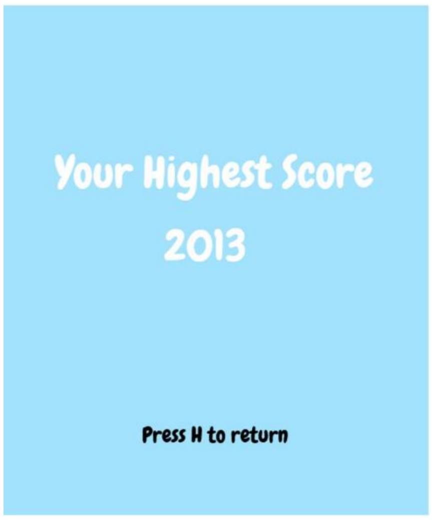
If the player collects 100% of the coins on every level, they are treated to a bonus level populated with 70 randomly placed chickens. While being a bit of a blood bath this level is a great way for players to achieve a much higher score.

Extreme Difficulty setting

The game features an extreme difficult mode (which can be activated at any time). In this mode, the player cannot ever move backwards. Making it much harder to collect coins or dodge vehicles. In addition, the timer depletes much more quickly, giving the player less time to collect all items and finish the level. This gives the user an ultimatum, do the try to collect every coin before the time runs out, hoping to reach the bonus level for extra points. Or do they ignore the coins to ensure they at least finish the level.

4.9 Persistent high score

The user can view their highest achieved score at any time by pressing the "H" key (if pressed during gameplay, the game is paused). This score is persistent, meaning the player can refresh their browser (or even close and re-open it) and the score is still saved. This is achieved using HTML5's local storage. This is not compatible with Internet explorer 11 when offline[4], so in this bowser, the user is shown a message stating this aspect is unavailable.



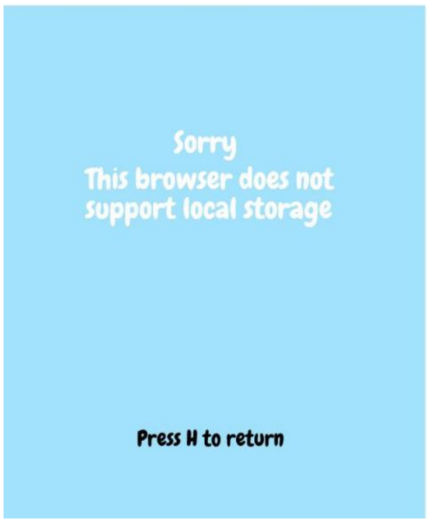


Figure 17: screenshots showing the high score screen; in a browser supporting offline local storage (left) and in a browser not supporting the feature (right).

5 Conclusion

The game created and documented in this report achieved all of the core requirements set out, and also contains several additional features. However, there is always more work that could be completed. The following features are planned for implementation in future versions of the game:

More Levels
Level creation is a very quick process, due to the nature of the object oriented approach taken. A new level could be created by instantiating a new road object and pushing it onto the road array. This would automatically scale to the correct size, with the correct amount of cars.

A procedurally generated endless runner mode.
With some tweaking, the game could be modified to be an endless runner, with the fox staying somewhere within the middle of the display, and distant objects moving towards the fox[5]. The game would then effectively last forever if the player was skilled enough.

An online leaderboard
A simple PHP script could link the game with an online database. This would encourage competition between players and extend the lifespan of the game.

[4] For compatibility with local storage in Internet Explorer 11, the game can be played online at http://davethings.com/Game.html
[5] Akin to the pseudo 3D racing games of the 90's such as "Lotus Ultimate challenge" [21]

6 References

[1] Hipster whale, *Crossy Road (Android),* Australia: Hipster Whale, 2014.

[2] S. Stefanov, "3 ways to define a JavaScript class," phpied..com, September 2006. [Online]. Available: http://www.phpied.com/3-ways-to-define-a-javascript-class/. [Accessed 03 May 2016].

[3] P. Newbury, "Multimedia Design and Applications Labs," 2016. [Online]. Available: https://studydirect.sussex.ac.uk/course/view.php?id=24401&topic=2. [Accessed 03 May 2016].

[4] navgtr, "Star fox 64 (Easy Ending)," Youtube, 21 June 2007. [Online]. Available: https://www.youtube.com/watch?v=y71O3ZKMruA. [Accessed 03 May 2016].

[5] N. EAD, *Lylat Wars (Nintendo 64),* Kyoto, Japan: Nintendo, 1997.

[6] L. W. Lilly Wachowski, Director, *The Matrix.* [Film]. United States: Warner bros, 1999.

[7] The GIMP Development Team, "GIMP- GNU Image Manipulation. Version 2.8.16," 21 November 2015. [Online]. Available: https://www.gimp.org/. [Accessed 03 May 2016].

[8] B. Schultheiss, "Sprite Sheet / Animation tiles," GIMP plugin registry, 02 December 2009. [Online]. Available: http://registry.gimp.org/node/20943. [Accessed 03 May 2016].

[9] Vonora, "Cuckoo & The Nightingale Duet.MP3," freesound.org, 11 April 2015. [Online]. Available: https://www.freesound.org/people/Vonora/sounds/269570/. [Accessed 03 May 2016].

[10] FoolBoyMedia, "Sky Loop," freesound.org, 16 Februrary 2015. [Online]. Available: https://www.freesound.org/people/FoolBoyMedia/sounds/264295/. [Accessed 03 May 2016].

[11] Keweldog, "car horn.wav," freesound.org, 30 May 2013. [Online]. Available: https://www.freesound.org/people/keweldog/sounds/182474/. [Accessed 03 May 2016].

[12] Qubodup, "Whining Dog.flac," freesound.org, 14 April 2013. [Online]. Available: https://www.freesound.org/people/qubodup/sounds/184808/. [Accessed 03 May 2016].

[13] ProjectsU012, "8-bit Video Game Sounds » Coins 1," freesound.org, 30 March 2016. [Online]. Available: https://www.freesound.org/people/ProjectsU012/sounds/341695/. [Accessed 03 May 2016].

[14] shawshank73, "cartoon_chicken.wav," freesound.org, 10 July 2010. [Online]. Available: https://www.freesound.org/people/shawshank73/sounds/100884/. [Accessed 2016 May 2016].

[15] Tails1942, "Dog sounds » DogGrowl2.wav," freesound.org, 17 August 2016. [Online]. Available: https://www.freesound.org/people/Tails1942/sounds/163278/. [Accessed 03 May 2016].

[16] reidedo, "RAM_Mouth Hawk_dry_v1.wav," freesound.org, 28 April 2016. [Online]. Available: https://www.freesound.org/people/reidedo/sounds/344446/. [Accessed 03 May 2016].

[17] rdholder, "2dogSound_player_death1_4s_2013jan31_CC-BY-30-US.wav," freesound.org, 04 february 2013. [Online]. Available: https://www.freesound.org/people/rdholder/sounds/177123/. [Accessed 03 May 2016].

[18] qubodup, "Cheerful Character," freesound.org, 15 May 2013. [Online]. Available: https://www.freesound.org/people/qubodup/sounds/188432/. [Accessed 03 May 2016].

[19] olver, "Clock ticking," freesounds.org, 03 October 2011. [Online]. Available: https://www.freesound.org/people/olver/sounds/130388/. [Accessed 03 May 2016].

[20] D. C. (Squid), "Google fonts - Chewy," Sideshow, 2016. [Online]. Available: https://www.google.com/fonts/specimen/Chewy. [Accessed 03 May 2016].

[21] Freepik, "Hand drawn cute forest animal collection," freepik.com, 2016. [Online]. Available: http://www.freepik.com/free-vector/hand-drawn-cute-forest-animal-collection_844311.htm. [Accessed 03 May 2016].

[22] Magnetic Fields, *Lotus Ultimate challenge (Sega Genesis),* Electronic Arts, 1992.

[23] Freepik, "Cute flat forest landscapes," freepik.com, 2016. [Online]. Available: http://www.freepik.com/free-vector/cute-flat-forest-landscapes_842612.htm#term=cute flat forest&page=1&position=2. [Accessed 03 May 2016].

[24] Freepik, "Cartoon Car on road," freepik.com, 2015. [Online]. Available: http://www.freepik.com/free-vector/cartoon-car-on-the-road_798408.htm#term=cartoon car on road&page=1&position=6. [Accessed 03 May 2016].

[25] Freepik, "Countryside landscape vector," freepik.com, 2015. [Online]. Available: http://www.freepik.com/free-vector/countryside-landscape-vector_757698.htm. [Accessed 03 May 2016].

[26] Freepik, "Eagle illustrations," freepik.com, 2015. [Online]. Available: http://www.freepik.com/free-vector/eagle-illustrations_795083.htm#term=eagle illustration&page=1&position=6. [Accessed 03 May 2016].

[27] Freepik, "Blood stain vector backgroud," freepik.com, 2015. [Online]. Available: http://www.freepik.com/free-vector/blood-stain-vector-backgroud_724214.htm. [Accessed 03 May 2016].

[28] Freepik, "Black and white feathers collection," freepik.com, 2015. [Online]. Available: http://www.freepik.com/free-vector/black-and-white-feathers-collection_758761.htm#term=black and white feathers&page=1&position=21. [Accessed 03 May 2016].

[29] Freepik, "Farmer in a farm landscape with chickens," freepik.com, 2016. [Online]. Available: http://www.freepik.com/free-vector/farmer-in-a-farm-landscape-with-

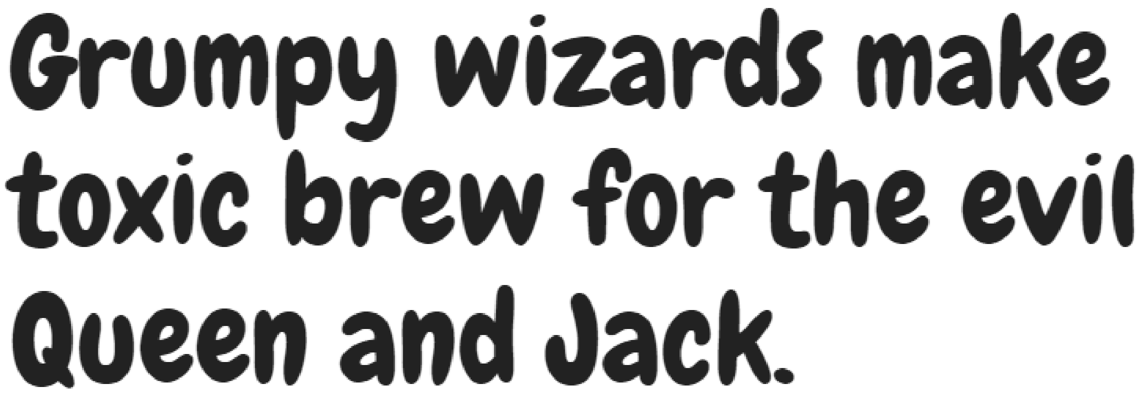
chickens_849681.htm#term=farmer in farm landscape with chickens&page=1&position=2. [Accessed 03 May 2016].

[30] Freepik, "Shiny golden coin vector," freepik.com, 2015. [Online]. Available: http://www.freepik.com/free-vector/shiny-golden-coin-vector_754242.htm. [Accessed 03 May 2016].

7 Appendix

7.1 Game font

The game font used throughout the game is the google font Chewy [20]



Grumpy wizards make toxic brew for the evil Queen and Jack.

Figure 18: "Chewy"- A Google font.

7.2 Original Artwork

Figure 26: Cartoon car on road. [23]

Figure 27: Countryside landscape vector. [24]

Figure 20: Eagle illustrations. [25]

Figure 24: Hand drawn forest animal collection. [20]

Figure 23: Blood stain vector background. [26]

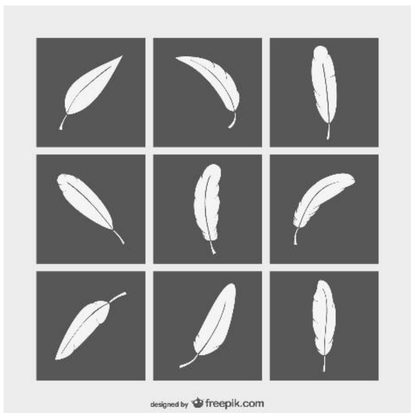


Figure 25: Black and white feathers collection. [27]

Figure 22: Farmer in a farm Landscape with chickens. [28]

Figure 21: Cute flat forest backgrounds. [22]

Figure 19: Shiny gold coin vector. [29]

7.3 Game art assets

The following assets were created using the original artwork referenced above.

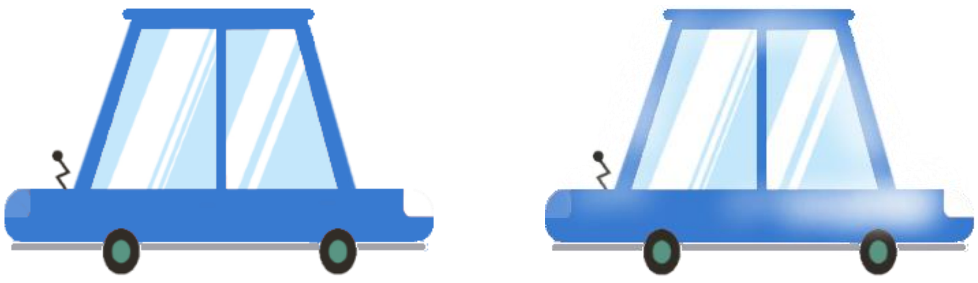


Figure 29: Blue car sprite (left) and snow covered blue car image (right).

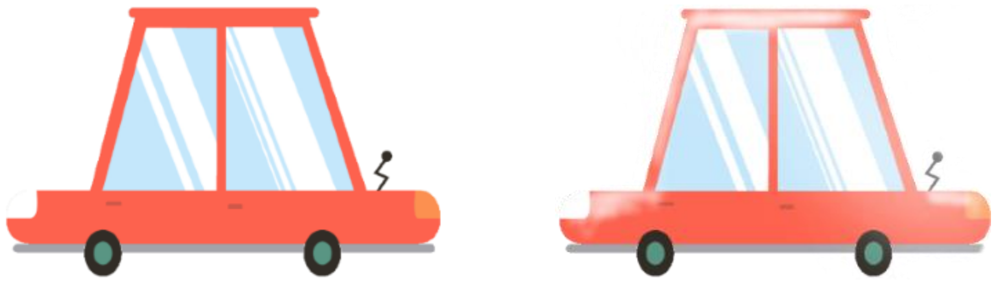


Figure 30: Red car sprite (left) and snow covered red car sprite (right).

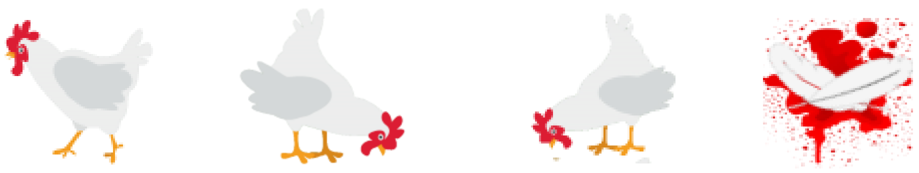


Figure 28: three chicken sprites and dead chicken sprite (far right).

Figure 34: Coin sprite.

Figure 33: Eagle sprite

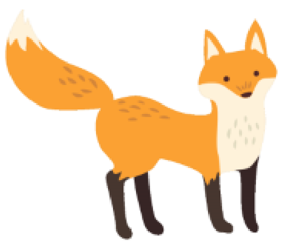


Figure 32: single fox sprite

Figure 31: Crouching fox sprite

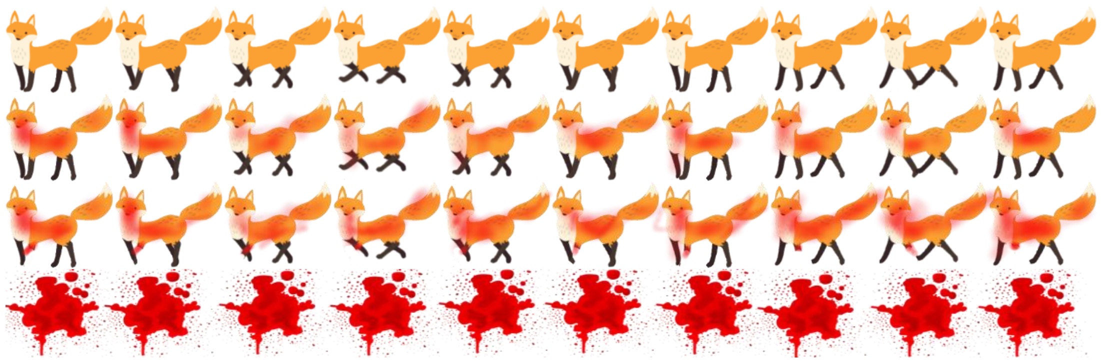


Figure 41: Left facing fox sprite sheets.

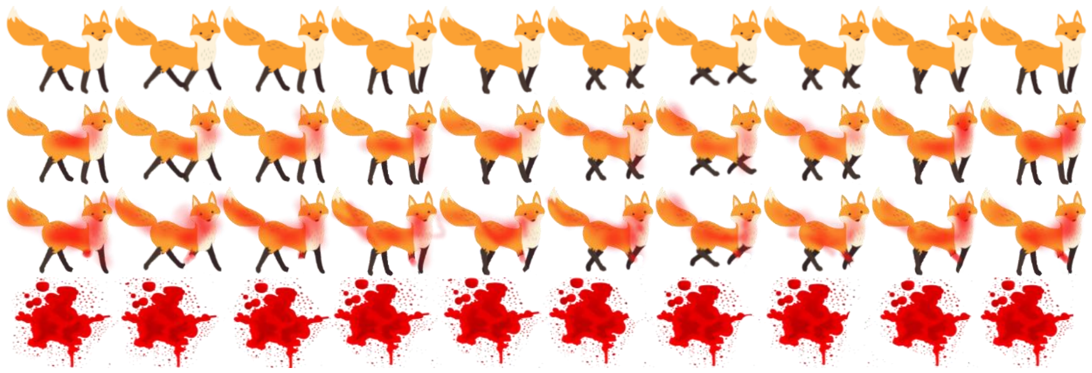


Figure 40: Right facing fox Sprite sheets.

Figure 39: Forest sprite (end of level).

Figure 38: Road sprite.

Figure 37: Frosty road sprite.

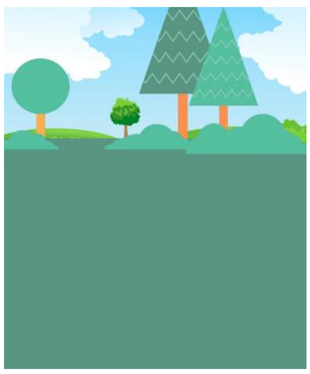


Figure 35: Main Menu background.

Figure 36: Level background