Tears in The Rain

*By Jordan Harrison and Jack Finn*

Contents

[Overview 1](#_Toc513893115)

[Artwork: Sprite sheets and tile sets 2](#_Toc513893116)

[Storyboard 3](#_Toc513893117)

[Resource Catalogue 3](#_Toc513893118)

[UML - Class Diagram 4](#_Toc513893119)

[Learning Experience 5](#_Toc513893120)

[Appendix 6](#_Toc513893121)

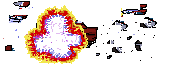
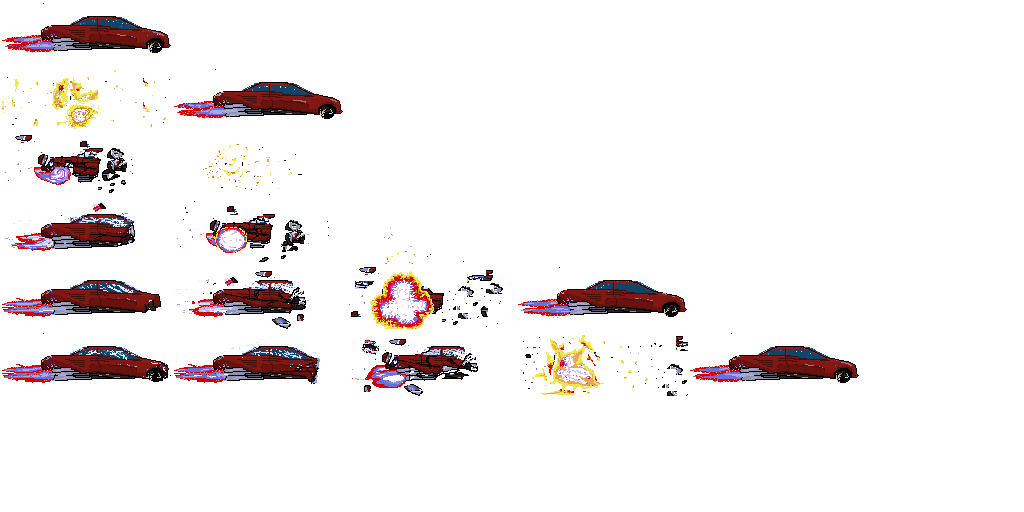
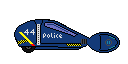
## Overview

Our game is a 2D Platformer based around the Blade Runner Universe. The first level starts with the player pursuing one of the Nexus-6 replicants, finding themselves lost in busy and chaotic traffic. The player needs survive the wave of oncoming traffic, before moving to the next level and on with the chase.

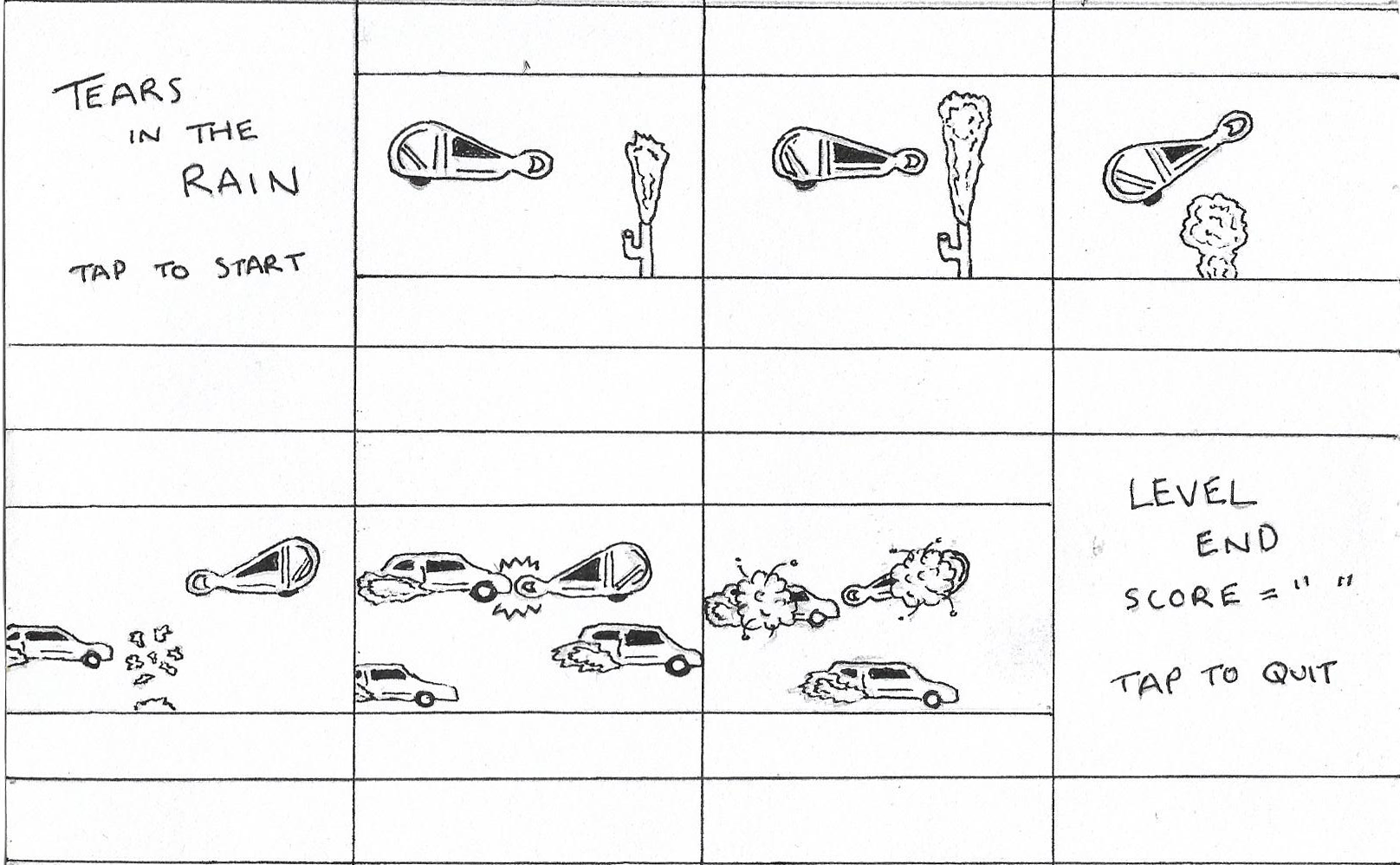
There are only three controls used in this first level; Up, Left and Right. The Down button is disabled during most ‘Spinner Missions’ but will be active during others when some ‘Air Vent’ obstacles will push the player up, requiring them to press Down to resist.

The player has to survive for as long as possible, receiving ten points for every traffic car that spawns during gameplay. The player starts with 5 lives and loses one everytime they collide with a car, the game ends when either the player has died or the time has run out.

## Artwork: Sprite sheets and tile sets



## Storyboard



lives timer score

lives timer score

lives timer score

lives timer score

lives timer score

lives timer score

*INTRO SCREEN*

*GAME SCREEN*

*TAP BUTTON TO MOVE*

*TRY TO DODGE HAZARDS*

*ENEMY CARS SPAWN OFF SCREEN*

*MULTIPLE WAVES OF CARS TO DODGE*

*PLAYER DIES IF ALL LIVES ARE LOST*

*GAME END SCREEN*

-1life

player

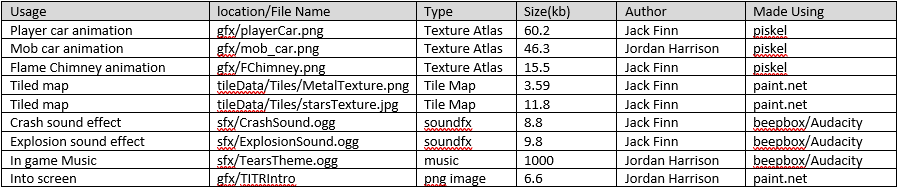
fire sprite

Move left or right

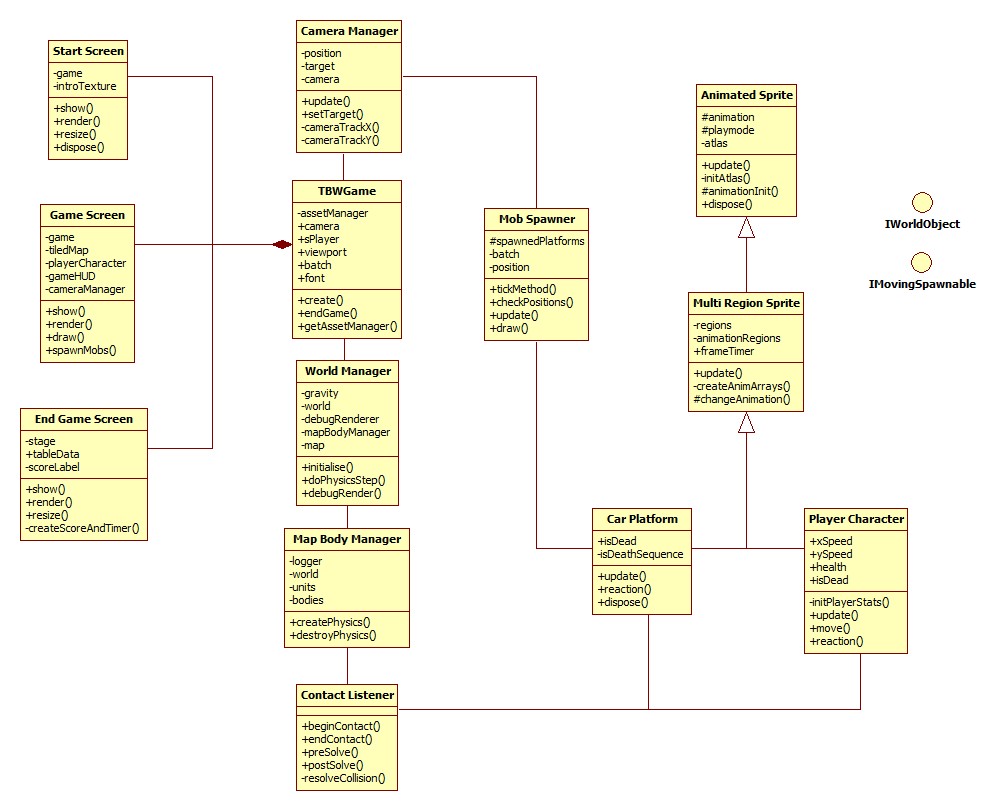
collision

enemy

## Resource Catalogue



## UML - Class Diagram



## Learning Experience

**Jordan Harrison:**

I completed the following things towards this assignment: Car platform class, Player character class, Contact listener, Adaptations to screens, Moving spawnable interface, MobSpawner, Animated Sprite and Multi-Region-Sprite class.

I learned a significant amount about the libGDX suite through university workshops and reading documentation found on the bad logic website. One of the most helpful sections was regarding box2D which is where I learned how to implement the contact listener class. I found using and implementing our own interface through the mob spawner and moving spawnable interface very valuable. It provided a practical example on how typing through interfaces can be used to link similar functions on different classes.

One of the most valuable experiences was through the group orientation of this project. This not only lessened my personal workload through segmenting the project but helped clarify ideas through discussion of implementation.

**Jack Finn:**

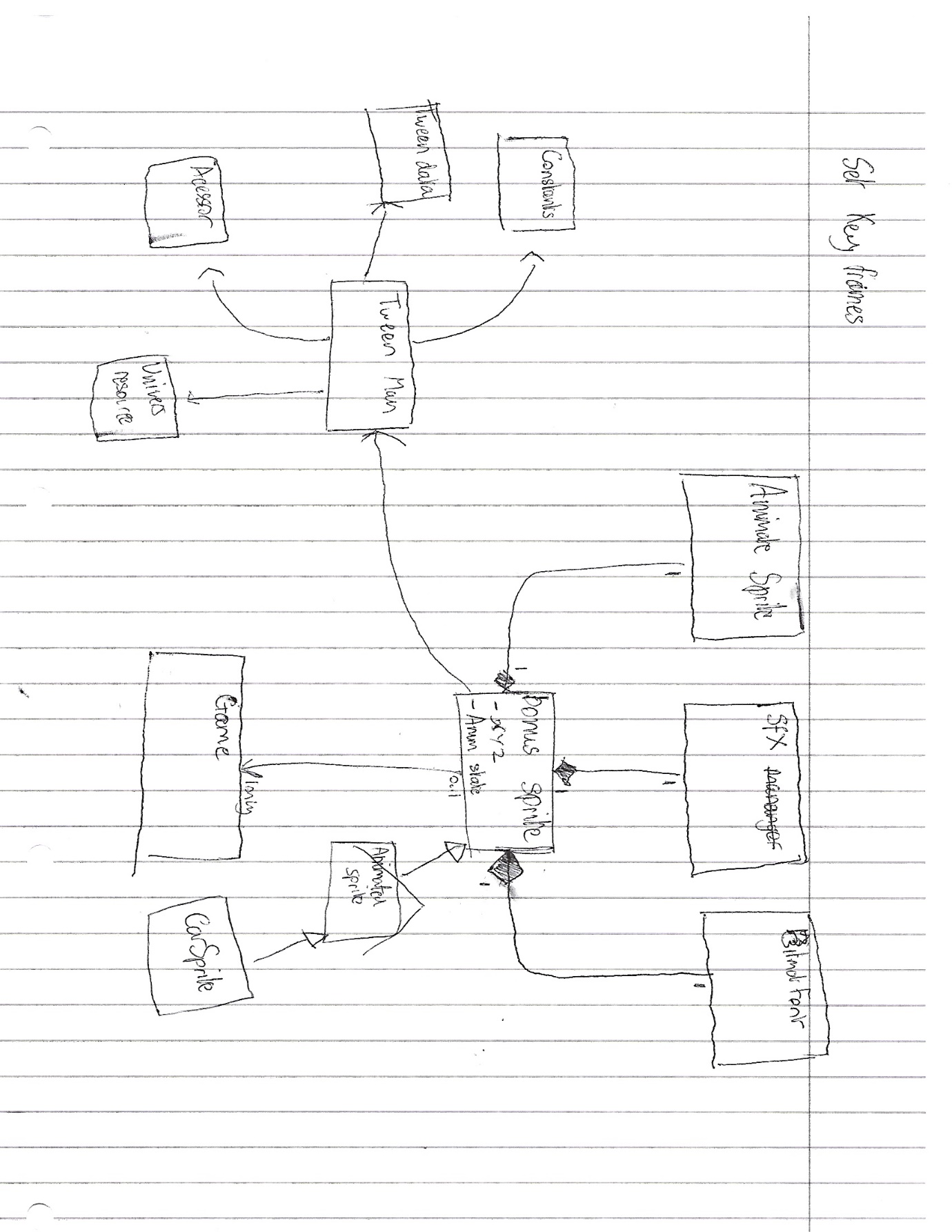
After discussing our individual strengths and weaknesses we tried to share the tasks in a way where neither had more than they could do while still being able to contribute to each area. For this project, I completed the following tasks:

* FireChimney class
* SoundPlayer class
* Tilemap assets
* Storyboard
* Class Diagram

During this project, I learned quite a lot about how classes interact, and have a slightly better understanding of static methods and different access levels, and the part they play in ensuring classes are loosely coupled while remaining flexible to maintain, reuse, or enhance, if needed.

I also have a better appreciation for the design aspect of a project, and how the degree of success or quality of one development phase can directly influence the processes and performance during another, especially concerning the compatibility of assets across multiple development platforms.

## Appendix



Initial class diagram made during a workshop to aid discussion on classes.