

## **COSC345 Assignment One**

**Caitlin Dyas 6455215, Max Freeman 7874528, Nat Moore 7603374, Rhys Davie 5774730**

### **What will we build?**

We are going to build an autorunner-rhythm game in Kotlin using Android Studio, where the player guides an avatar through a simplistic abstract 2D environment. Obstacles will arrive in pre-designed patterns that are synchronised to features of accompanying music, and the player will have to act in time with the music to avoid the obstacles. The music will consist of songs that we will produce, and each song will consist of multiple different sections that can be pseudo-randomly shuffled to create different permutations of the same song. The specific design of the game is intended to make it highly replayable, encouraging the user to waste as much time in it as possible (corresponding to the theme of this year's projects).

### **Who is in the team?**

Our team is made up of Rhys Davie, Caitlin Dyas, Max Freeman, and Nat Moore. We formed our team with the aim of having a diverse set of abilities and perspectives; we wanted a team where one person's weaknesses would be compensated for by another's strengths (and vice versa).

Our team members have experience or interests in the following areas:

- Rhys Davie - Gameplay design, programming, and music
- Caitlin Dyas - Art & visual design, programming, and communication
- Max Freeman - Art & visual design, programming, and music
- Nat Moore - Programming, writing, and documentation

The nature of our team's skillset is the main factor that led us to pick a music-based video game as our project. Two of our team members have experience with music production, which will help us to create original music for our game. We also have experience in visual communication and graphic design, which will help us make our game appealing and easy to play. We also all have a shared interest in video games and game development.

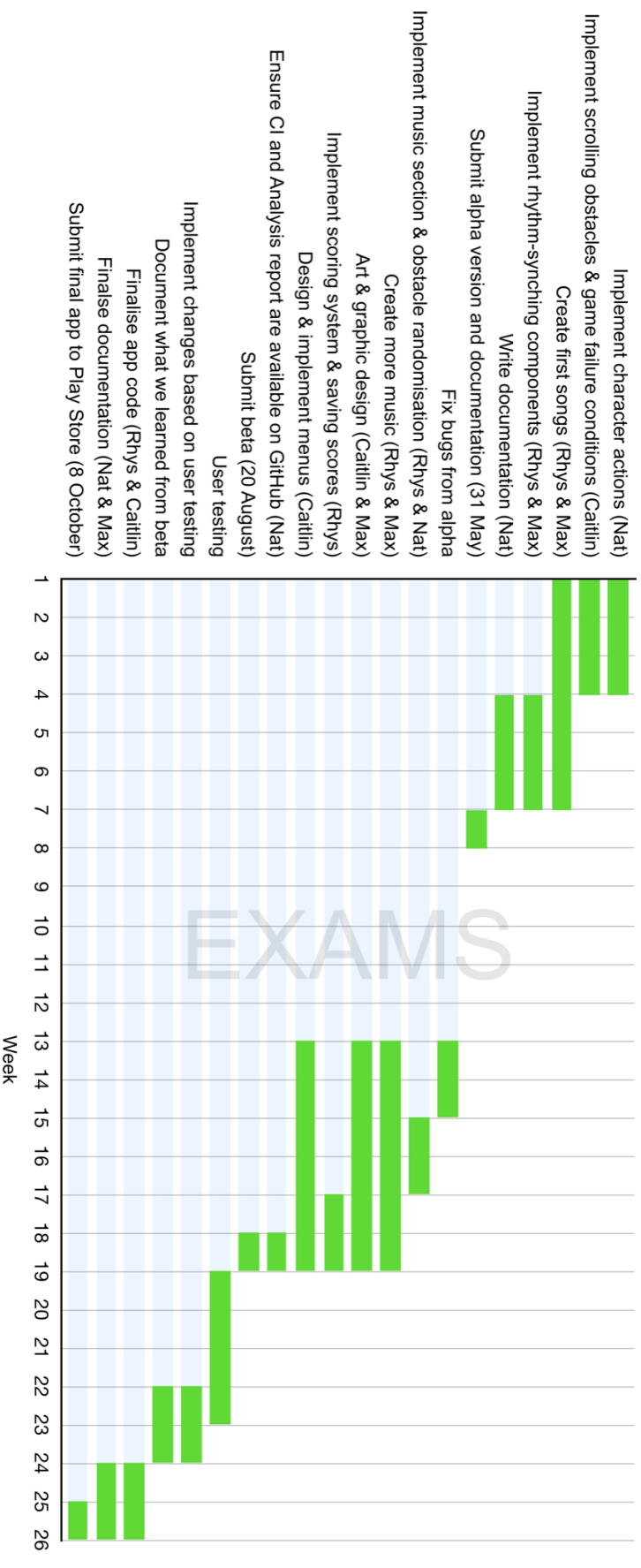
### **How will we build our app?**

We will develop our app in multiple phases, focusing on different aspects of its technical design.

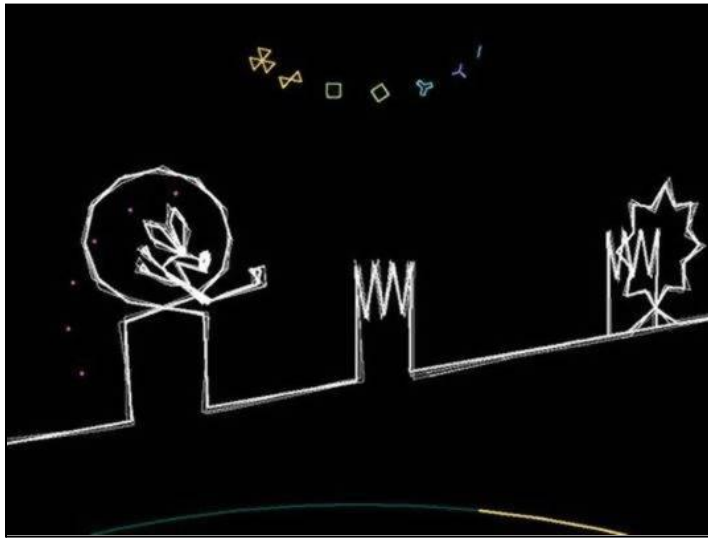
- The core of the design is based on the "2D autorunner" game model, in which a player avatar moves to the right along a flat plane. The plane contains obstacles such as walls and pits that the player avatar must avoid. We will develop this framework using simple placeholder graphics (e.g. grey boxes and lines); the most important objective will be making the player inputs feel responsive and satisfying.
- After the autorunner core is functional, we will implement the game's rhythm-based elements, which will involve making obstacles appear on screen at a rate that correlates to the beat of the music (the delay between obstacles will be able to be derived from the known beats-per-minute of each song). Then we will design "clusters" of obstacles, each having a unique section of music associated with it, which we will use as a reference to

design the timing of the obstacles - obstacles will generally occur on accents within the music.

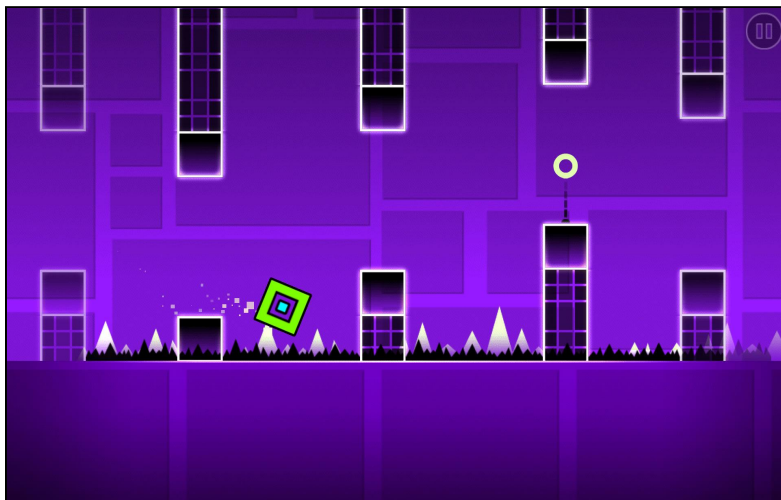
- After these elements are implemented, the game will be technically functional, and we will submit it as our alpha version. The next steps will be to make the game have a win/lose state, implement a scoring system, add menus, and replace placeholder content (such as turning the player avatar from an abstract shape into an identifiable character), leading us to a beta version. If we have time, we could also focus on adding further base content such as more music and obstacle patterns.
- After the game is completely playable (e.g., it contains functional menus, can save player scores, and etc.) and we have submitted our beta version, we will enter a polish stage. This will involve fixing bugs (ideally with the help of playtesters such as friends, family, or classmates), completing our documentation, and reflecting on what we learned from the feedback given on our beta version.



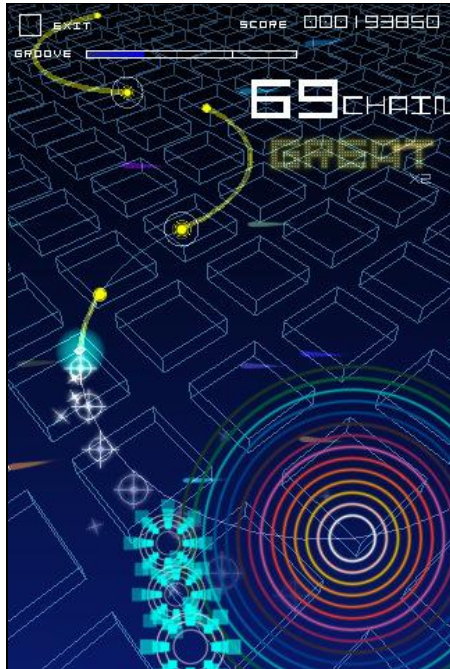
## What similar software exists already?



Vib-Ribbon (1999) is an early example of a game in the genre we are working in, where the player avatar runs sideways and avoids obstacles synchronised to music. It features a minimalist, abstract presentation similar to what we are aiming for. However, its gameplay is slow-paced by today's standards, and the obstacles are randomised, meaning it is not very engaging as it lacks meaningful connection to the music. Despite this, it retains a small following due to its appealing avatar character and music, demonstrating the aid that these elements can bring to a game.



Geometry Dash (2013) is a mobile game in which the player controls a square that constantly moves to the right, and must tap the screen to jump over obstacles. Similarly to our concept, obstacles are timed to synchronise with the game's music, and they arrive constantly, keeping the player on their toes. One of the game's main appealing factors is that players can create their own levels using music from the media-sharing website Newgrounds, although this is beyond the scope of our project.



Groove Coaster (2011) is a simple mobile game where a ball moves along a line, and the player must tap the screen every time the ball touches an obstacle embedded on the line, which occurs at intervals synchronised to the music. Sometimes obstacles are chained together, so the player must repeatedly tap the screen to get a higher score. On higher difficulty modes, the game requires a “flick” action, which is a swipe to the left or right on the screen, and a “scratch” action, where the player repeatedly swipes left and right over the obstacle.

### **What sets us apart?**

Each level of our game will feature a different song, with each song built out of sections, which are played in a pseudo-random order. This will give a more unique experience on each playthrough, because although the inputs required of the player are synchronised with the music (and hence predictable), the wider structure of the music is unpredictable, creating an appealing tension between the predictable and the unpredictable. This approach to level design will keep players on their toes and keep them interested in our app for longer -- our goal is to provide an outlet to waste the user's time, after all.

### **Who is our market? Why will they care?**

Our target market is people who are interested in electronic music, platform games, and wasting time, as well as developing their sense of rhythm or musical recall. Currently there are few apps that combine elements of autorunner games, rhythm-based games, and platform games. Many of the current similar apps on the market aim to challenge the user to tap faster and faster, ending the game when they cannot keep up. In contrast, our goal is to use randomised music segments to encourage the user to focus on their mastery of rhythm and musical recall, rather than sheer reaction speed. This encourages them to spend/waste as much time in game as possible, rather than simply ascend the game's levels until they reach the end.