Final Report.

*Introduction:*

As a group we chose to stick with the original concept of RogueRun, despite there being some name changes the basic premise was always the same. An action-adventure car shooting game where the player had to overcome waves of enemies in order to progress.

We had other ideas such as having dynamic levels i.e. timed survival waves, point capped waves etc. but these would theoretically be implemented in other levels where appropriate to the story.

*Project Management:*

From the beginning we each recorded the time spent working on the project. Initially before we had any plans we’d meet, discuss and throw ideas out before sticking to anything. One of the first inspirations of the game came from Phillip, giving the idea that the map could be similar to a toy car map, and the player is tasked with shooting self-driverless cars through waves.

We decided to use GitHub since the local and remote repository would help us collaborate from across desks/home.

Our group regularly met around 2-3 times a week. Recording how much time we spent working by timing how long we spent on projects (such as AI, audio etc.)

After Nick completed the Gantt chart, he suggested we work in an agile development cycle. We would recap what everyone was doing before each meeting, and discuss what we needed help with on our current task.

Alongside this we all created stories in the sprint backlog on our GitHub projects tab on our GitHub repository (<https://github.com/nike-sh/CarProto>) Fig.1 Project Backlog

After this we assigned our desired tasks to each other depending on skill and preference.

Brainstorming:

All of us split the different categories between us so the workload would be easier for example; Phillip doing the Map and Power ups, Theo doing the character movement, the buildings (alongside Phillip) and the UI, Steve doing the Enemy Behaviour and waves (with Theo), Projectiles (with Nick doing player projectiles) and Nick doing the Music, Menu and Health system.

Theo said we should include Pixel art as he was learning to get better in pixel art and it seemed fitting for the project. He later went on to teach Phillip a lot on Photoshop and they made a lot of the sprite assets together.

Each time we met we would try to keep on topic with the Gantt chart, using it as a basis for what work would be completed that day.

Anything we couldn’t do we had to move back into the backlog and keep it for a later date. This happened with a few stories such as implementing a minimap, lighting, story panels and destructible buildings. These can all be seen in the first column on the GitHub project sprint board and spoken about later on in the document.

*Implementation:*

*Discussion:*

Gameplay Mechanics:

Comparing the final prototype to the GDD we found there to be many differences.

Most we can say didn’t come to fruition due to time however there were some that had us encounter problems so we had to scrap the idea.

Some of these could be implemented in the full game.

Key differences include the use of a minimap, this was going to help the player find enemies on the larger map and add to the UI. We couldn’t get round to making this since we didn’t have the technical knowledge nor time to learn it to do so.

Another aspect that would be put into the game was the dynamic scoring. We had ideas to have different levels play differently, some as waves, some to reach a certain score and some timed levels. However since we only made one level we couldn’t show the use of all these scoring controls.

After playtesting:

According to the questionnaire sent out we had many responses highlight key points the game could be improved upon. The most common criticism was the player movement mechanics. Testers found the car would slide too much and had a hard time shooting enemies due to this, it’s something we will look to improve and build a better system for.

The second most useful feedback was the players dies too quickly. To solve this we’ve increased the initial player health. Although this was a simple fix, luckily the way the game works, if difficulty was increased/decreased the health and damage would scale.

slo mo too long

hazards were not used effectively

went through enemy (collision)

potential fixes:

2more drag

1more health/less damage

1changing slo mo trigger/adding pause/unpause

2more hazards, slomo first time & less slo mo

2add enemy collision box

1boundaries around map

Appendicies

Fig1