RESEARCH INTO THE SIMULATION OF SHOCK WAVES

Project Approach

Authors:
Georgina Perera, Tomasz Mackow

Introduction

This document outlines the organisation of the project, how we approached tasks, resources used and coding approaches.

Roles and Responsibilities

Our final year project is made up of five members: Georgina Perera, Tomasz Mackow, Robert McDonnell, Liam Ireland, Michael Jones. We made the decision to assign members individual roles so that one member was in charge of keeping the group on track with the assigned task. Although these roles were set, they were only guidelines and members worked in multiple different areas also. The roles were as follows:

Georgina Perera	Project Manager/Planner, Front end Designer
Tomasz Mackow	Physics Researcher
Robert McDonnell	Quality Assurance, Front end Tester
Liam Ireland	Tech Lead/Architect
Michael Jones	Backend Tester, Tech Architect

All members played the role of developers.

As well as these roles we decided to split our team down further into a backend team and a front end team. This was done so that members could focus on specific areas as per their previous experience in certain fields. The teams were as follows:

Backend Team	**Front end Team**
Tomasz Mackow	Georgina Perera
Liam Ireland	Robert McDonnell
Michael Jones	

Within these teams we programmed using agile software engineering techniques to ensure that everyone was fully involved in the project as well as helping us to manage the work. We also used pair programming to ensure that all code was written to a high standard and to speed up the code development process.

We believed that assigning roles to individuals would help with the productivity of the project as members would be working in their area of interest. It allowed us to work efficiently and also allowed for a better sharing network of skills.

Scheduling

The process for scheduling, changed dramatically throughout the project, causing a number of issues. For more details on this, please view the *“Scheduling Disclaimer”*.

Tools Used

Code and document quality

Github and Sourcetree

To ensure that our code was kept to a high standard we used either GitHub or Sourcetree which allowed us to keep track of changes as well as allowing multiple users to be working on the code at the same time.

Google Drive

For the documentation we used Google Drive for the same reasons as mentioned for GitHub/Sourcetree. Google Drive allowed us to keep track of changes in documents as well as allowing us all to read or edit documents at the same time. We used the “Track changes” and commenting features to help each other to maintain the quality of our documentation. We also created a small information wiki (please see “Developer Guide” document) which contained documents such as Unity tutorials, tips for initial setup as well as various physics calculations.

Communication

Facebook Groups

We created a Facebook group which allowed us to keep track of what was being worked on by the front end and back end teams for a particular week as well as individual tickets which needed closing. This meant that both teams were transparent at all times and made it easier to collaborate on our work. The Facebook group was favoured over Facebook Messenger for important notices or meeting schedules as those could have been lost in the discussions held in Messenger.

Facebook Messenger

Facebook Messenger was our main form of communication. Using this we would discuss plans for meetings as well as communicate any changes or suggestions we had. Facebook Messenger proved to be essential when people were working separately or were unable to meet in person.

Trello

We initially intended to use Trello as a means of storing a list of tickets which needed closing as well as for storing documents such as Unity user guides, in the form of a wiki which would have been useful for the team. We decided that Google Drive provides enough features for us to be able to use it instead for the knowledge wiki and the Facebook group was a more suitable place for keeping track of our tickets.

Code tools

Unity

Unity is .NET based which means our source code would be written in C#, which is very similar to Java, a language the group were all familiar with. In contrast with developing in the Unreal Engine, which would be C++, which the group have limited experience with. Additionally, after completing tutorial videos on the editor software for both engines, Unity was found to be easier to use. Some features which also attracted us are the fact the Unity is open-source and has a well documented API.

Visual Studio

We chose to use Visual Studio as it integrated nicely with GitHub and allowed us to write code in C#.