Method Selection and Planning

Our chosen software development model is Scrum. We believe Scrum is the perfect fit for us for a few reasons. The first is that Scrum can easily adapt to changes in requirements. It is important we can adapt to changes in requirements because we are not creating a critical system where the requirements are stable and predictable, we are creating a game for a customer where they could change their mind of what they want halfway through the planning or designing of the game. We may also realise that a certain requirement is unfeasible so we would have to change that requirement to satisfy the customer. If we were using a different, more planned model then the change in requirements could be a harder challenge to overcome.

The second reason for choosing Scrum is because development of a project is done in 'sprints', of up to a month in length. This is ideal for us because we are working towards very specific deadlines that cannot be overrun, so having a method within which each 'sprint' or iteration takes a few weeks is ideal for our deadline.

We have also chosen to use Scrum as it relies on daily meetings to monitor progress and plan for the next period of work. We felt that regular meetings and communication would keep our team organised and would ensure we kept to deadlines. We also believed that making sure that everyone always knows what they are doing and can freely ask for help would lead to a good quality product with well-organised documentation. Since we are busy students and cannot always meet every day, we have one big weekly meeting, as well as short meetings during practicals and regular communication both in person and via social media.

Development and Collaboration Tools

As this is a team project all members of our team need to be communicating regularly and need to be able to share and store files and ideas.

Facebook:
All team members have an active Facebook profile and therefore we have set up a group chat in order to keep in contact with each other at any time throughout the project. This is vital for easily arranging meetings with each other and for quickly following up any queries.

Google Drive:
We have chosen to use Google Drive to store and share all files we are using for our project. This is convenient since all team members have Google accounts. Google Drive is an excellent resource as it allows multiple people to edit the same file at once, so it allows a group of us to be editing while participating in a Skype call to aid organisation. Google Drive also allows us to see which group members create and edit files, as well as keeping a log of

all edits made by each team member, which helps us monitor the work of each of us to ensure that workload is being shared fairly between us, and that each of us is pulling our weight.

Skype:
We are using Skype to hold group calls to allow for better communication and organisation when group editing a file, and to hold meetings when we cannot all meet in person.

GitHub:
We will be using GitHub during the development process as a means to share code between each other. This will be particularly useful during the Christmas holidays, where we plan to begin writing the code for our game. Another reason we are using GitHub is to ensure changes and errors are easily trackable and reversible, using version control. This is essential, as if multiple people are changing files at the same time and something in the code gets broken, it is easy to find the cause of the problem and revert to a working state. GitHub also allows people to work and change their local repository while offline, which could prove to be useful.

Language and Platform

In order to produce our game, we will need to select a programming language along with a platform with which to implement it in. We decided to avoid dedicated game engines such as Unity. While they can be used to produce 2D games, we felt that it would be a hindrance to use Unity as we would have to schedule a significant portion of time to allow us to acclimatise to the user interface and how games are structured in Unity, as well as learning how to code in C# or Javascript.

We settled on implementing our game using Java, through the Eclipse IDE. We chose this as we had all already had experience with Java and Eclipse itself to some degree. In addition, Eclipse is an easily accessible platform, as it is installed on all computers in the computer labs, and is a free to download software so we can use it on our home computers. Furthermore, there is a Java library LibGDX, which can be used to create 2D games. We have decided to use this library as it is a well documented, open source library with a lot of flexibility. It is compatible with our plan of using tiles to represent the map. Tiled is a program that we can use to create tiled maps which are useable by LibGDX, and has been used to create very successful games, such as Shovel Knight.

Team Organisation

Because it is such a small team, we decided that the team should take on a fairly flat organisation, with each member equally responsible for a certain part of the project, but some members specialising in certain roles.

Kim Miller took on the role of scrum master. She will be responsible for running the meetings and making sure they are on task and productive.

Edward Hathaway took on the role of project planner. He will be responsible for keeping the team as close to the plan as possible and, if there are delays, making changes to the plan to accommodate for these delays.

Thomas Yau took on the role of lead developer. He will be responsible for keeping the implementation of the game on track, and ensuring that there is traceability from our requirements through to our product.

Christos Wild, Michael Rothery and Tristan Owen are team members available to work on any aspect of the project.

Systematic Project Plan

ID	Task	Duration (days)	Earliest Start Date	Latest Finish Date	Prerequisites
Assessment 1					
1	Planning	7	09/10/2015	16/10/2015	-
2	Requirements	13	16/10/2015	29/10/2015	1
3	Method	8	16/10/2015	08/11/2015	1
4	Risk Assessment	6	16/10/2015	08/11/2015	1
5	Architecture	10	29/10/2015	08/11/2015	2
6	Website	3	16/10/2015	11/11/2015	1
7	Review	3	08/11/2015	11/11/2015	ALL BUT 6
Assessment 2					
8	Plan Review	3	11/11/2015	14/11/2015	-
9	Update Requirements	7	14/11/2015	21/11/2015	8
10	Update Risk Assessment	7	14/11/2015	05/12/2015	8
11	Update Methods	7	14/11/2015	21/11/2015	8
12	GUI Report	5	21/11/2015	26/11/2015	9
13	Architecture Report	14	21/11/2015	05/12/2015	9
14	Familiarisation	8	21/11/2015	29/11/2015	11
15	Implementation	35	05/12/2015	09/01/2016	10,12,13,14
16	Testing Report	20	05/12/2015	09/01/2016	10,12,13,15
17	Website	3	14/11/2015	29/11/2015	8
18	Review	5	09/01/2016	20/01/2016	ALL BUT 17
Assessment 3					
19	Selection of New Product	5	20/01/2016	25/01/2016	-
20	Change Report	7	25/01/2016	01/02/2016	19
21	Update Previous Reports	7	25/01/2016	01/02/2016	20
22	Implementation	11	01/02/2016	12/02/2016	21
23	Update Testing Report	7	01/02/2016	12/02/2016	21
24	Website	3	25/01/2016	12/02/2016	20
25	Review	5	12/02/2016	17/02/2016	ALL BUT 23
Assessment 4					
26	Selection of New Product	5	17/02/2016	22/02/2016	-
27	Change Report	7	22/02/2016	29/02/2016	26
28	Implementation	30	29/02/2016	30/03/2016	27
29	Update Testing Report	12	29/02/2016	30/03/2016	27
30	Project Review Report	23	30/03/2016	22/04/2016	28
31	Website	3	22/02/2016	22/04/2016	26
32	Review	5	22/04/2016	27/04/2016	ALL BUT 31

Assessment 1 Critical Path: 1, 2, 5, 7

Assessment 2 Critical Path: 8, (9/10/11/12/13/14), 15, 18

Assessment 3 Critical Path: 19, (20/21), 22, 25

Assessment 4 Critical Path: 26, 27, 28, 30, 32