MineBuster Project

Project Management Document

Table of Contents

1. Introduction

MineBuster is a python game that loads a map with resources on. These resources all have different values and can be mined by pressing ‘SPACE.’ Mining the resource gives the player score the resources can be sold at the shop for upgrades. Once the player has mined all the resources on the level the player can proceed to the next level door where a new map will be loaded.
The game features different difficulty modes such as (In all modes the shop and the inventory are automatically sorted by value/price):

Easy: Player can click on one of the resource icons in the objectives menu to go to the nearest source of that resource. The resource can then be mined a pathfinding algorithm will be used to achieve this.
Normal: The player has full control over the robot but cannot click to auto path.
*Hard: The player has full control over the robot the objectives menu is disabled cannot click to auto path. All shop upgrades for the drill (Increases the chance for the resource to ‘drop’ and not be destroyed) are disabled. All shop prices are 50% higher.
*Extended goal

1.1 Project Requirements

For the project to be considered a success these requirements must be met at 13 March 2016 (23:55):

- All deliverables must be completed (see 1.3)
- The program must reach the Minimum Viable Product (see 1.2)
- The project much score at least 4 marks on the Assessment Criteria

Full Assesment Criteria is available here:
https://drive.google.com/open?id=0BxGBQsFdGnloZm9rdm5BRWswNWM
Program requirements found here (page 2):
https://drive.google.com/file/d/0B7dNe5318ceFM2JwZTY2MkNlWm8/view

1.2 Minimum Viable Product (MVP)

The program must fulfill a set list of criteria for it to be considered a success:

- 1 Sort OR Search algorithm is implemented
- The GUI should match the documentation and have simple design concepts and principles applied.
- The UML Activity diagram should match the way the final system works
- One or two of the elements of the application should have been updated based on the evaluation.
- Should include one additional feature which should fit in with the overall ethos of the system but not really be seen as innovative or complex.
- A program which enables a virtual robot to search a map for resources which contains a graphics user interface (GUI)

- Different locations have resources each of which have a name, type price and quantity
- A list of items that the user can select in order for the robot to search for resources
- Enable the user the ability to specify an area for the robot to search for these resources
- The ability for the items to be sorted by value
- The ‘robot’ should move physically on screen.
- To display sorted items on screen
- To allow for the user to specify how the items are sorted. Ascending or descending order according to certain criteria. Like name or value which is determined by the user.

1.3 Deliverables

Because of the nature of the project we have two types of deliverables project deliverables and individual deliverables. Individual deliverables make up the deliverables that are all required for the project to be considered a success.
Individual Deliverables:

Deliverable	Deadline
A fully working program that fits the project and program requirements. (See 1.1) Github commented code, includes algorithms and external libraries/APIs with screen shots of key functionality testing.	13 March 2016 (23:55)
Graphic interface design prototype	13 March 2016 (23:55)
A report of what design concepts and principles that have been considered and applied to the prototype design. (500 words max)	13 March 2016 (23:55)
A report of the research on search algorithms (500 words max)	13 March 2016 (23:55)
UML activity diagrams for the program flow which includes the sorting algorithms	13 March 2016 (23:55)
An up to date PWP (Project Work Portfolio)	13 March 2016 (23:55)
Program usability testing report (max 2 pages) on the evaluation of the program using one usability method.	13 March 2016 (23:55)
Program testing and results document/presentation	13 March 2016 (23:55)

Full ‘Activity’ Breif available:
https://drive.google.com/file/d/0B7dNe5318ceFM2JwZTY2MkNIWm8/view?usp=sharing

Project Assessment Deliverables:

Deliverable	Deadline
Individual Viva	14 - 24 March 2016
Project Submission (Where the project is declared completed)	13 March 2016 (23:55)

1.4 Extended Requirements

These are optional features we have chosen to include:

Feature
Difficulty System (Easy, Normal, Hard)
A choice of what sorting Algorithm/searching algorithm to use
A Fully functioning save/load system
A fully functioning “Start” Menu that loaded before the game so the player could choose what to do (New game, Load, options, exit)
The ability pause the game
Different levels
Shop system to purchases upgrades and fuel
Fuel system (if you run out game over)
Time specific objectives (collect a list of items before the time runs out or game over)
3 different searching and sorting algorithms.

2. Project Scope

The scope of the project really depends on how far we get with development it is very likely this will change:
The game must fit both the MVP and score above 4 on the full specification (the assessment criteria.) The ‘game’ should have no more than 8 resources to collect and make use of sorting or searching algorithms. We have six weeks to code according to the project

timescale. Therefore all coding must be completed before the deadline which is March 13th 2016.

2.1 Success Criteria

The project will be considered a success when the project requirements (see 1.1) are met. However we would like to do better than the basic project requirements and as a group we would consider the project a success if it scores above 7 marks on the project assessment criteria (see 1.1.)

3. Resources

Below is a breakdown of every resource we have access to we have considered using at some point.

3.1 Hardware

Item	Description	Quantity
HP Z230 Tower Workstation	OS: Windows 7 Enterprise 64 bit CPU: Intel i5- 4950 3.30ghz (4CPUs) GPU: NIVIDIA Quadro K2200 VRAM: 4GB RAM: 8GB	5
HP Compaq LA2206 Wide LCD monitor	1920x1080 Res	5
Phones with camera	Personally owned phones for video capture	5

3.2 Software

These are all pieces of software we have access too we may not use it all.

Item	Description	Quantity
GitHub / Git	A website and software utility for version control	5
Python 3.4 and IDLE IDE	The python programming language and standard libraries installed on each workstation	5
Microsoft Office 2013	Office 2013 tools	5

	Access 2013 Excel 2013 Infopath Designer 2013 Infopath filler 2013 OneNote 2013 Outlook 2013 Powerpoint 2013 Project 2013 Publisher 2013 Visio 2013 Word 2013	
Google Drive	A cloud storage and document creation service: Google docs Google sheets Google slides Google forms Google Drawings Third Party: Lucidchart (Tool for creating diagrams and flowcharts) SmartSheet (Gantt chart and spreadsheet tool)	5
Microsoft OneDrive and Microsoft Office Online	Microsoft's Cloud service and office online Office 365 (student edition) Microsoft outlook emails Powerpoint Online Onenote Online Sway Word Online OneDrive	5

3.3 Human

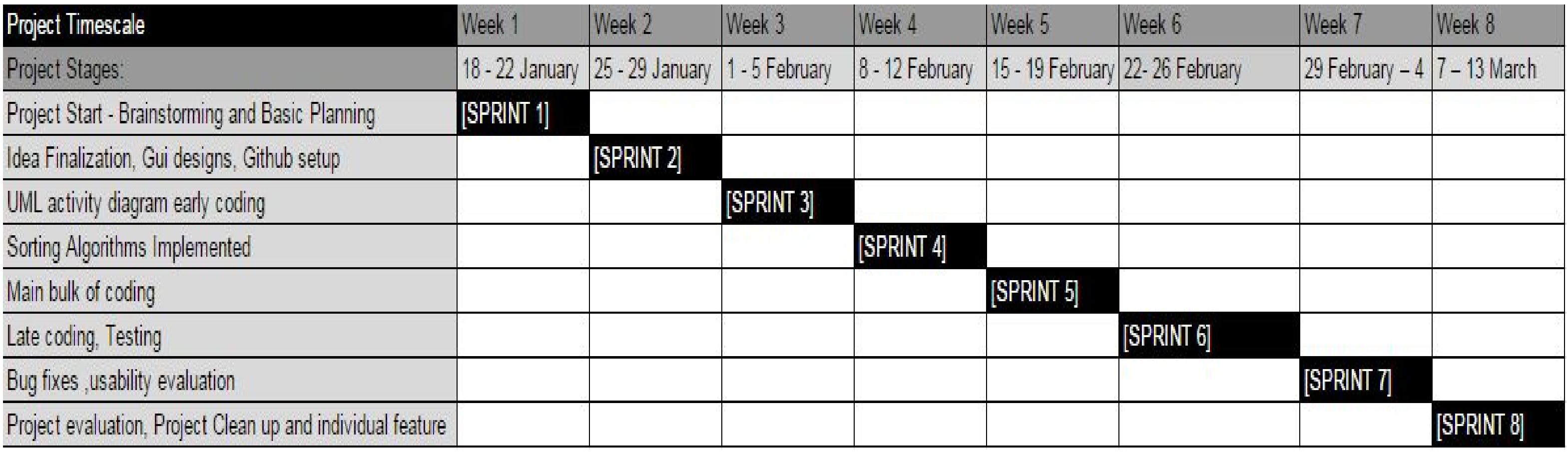
In total we have 5 group members

Name	**Role**
Bogdan	Designer and Programmer
Martyn	Project Manager
Ivo	Start Menu Designer / Programmer
Sam	Programmer/Debugger

Tom	Programmer/Debugger

4. Timescale

This is the timetable we will be sticking to during the project



Project Timescale	Week 1	Week 2	Week 3	Week 4	Week 5	Week 6	Week 7	Week 8
Project Stages:	18 - 22 January	25 - 29 January	1 - 5 February	8 - 12 February	15 - 19 February	22- 26 February	29 February – 4	7 – 13 March
Project Start - Brainstorming and Basic Planning	[SPRINT 1]							
Idea Finalization, Gui designs, Github setup		[SPRINT 2]						
UML activity diagram early coding			[SPRINT 3]					
Sorting Algorithms Implemented				[SPRINT 4]				
Main bulk of coding					[SPRINT 5]			
Late coding, Testing						[SPRINT 6]		
Bug fixes ,usability evaluation							[SPRINT 7]	
Project evaluation, Project Clean up and individual feature								[SPRINT 8]

5. Project Methodology

Whilst the project structure reflects a very logical sequential flow we feel it necessary to follow an agile methodology for which we have chosen SCRUM. Scrum blends all development activities into each iteration, adapting to discovered realities at fixed intervals (http://scrumreferencecard.com/scrum-reference-card/ .) Each iteration is called a sprint. These sprints allows a team to modify and produce a working prototype each sprint run. For each sprint the team and the product 'owner' looks at a list of requirements for the 'product' called a product backlog to decide which goals are to be added to the sprint backlog. The sprint backlog is a set of goals and tasks that the team aims to achieve each sprint. After each sprint the product is evaluated by the product owner and the feedback helps steer the direction of the project.

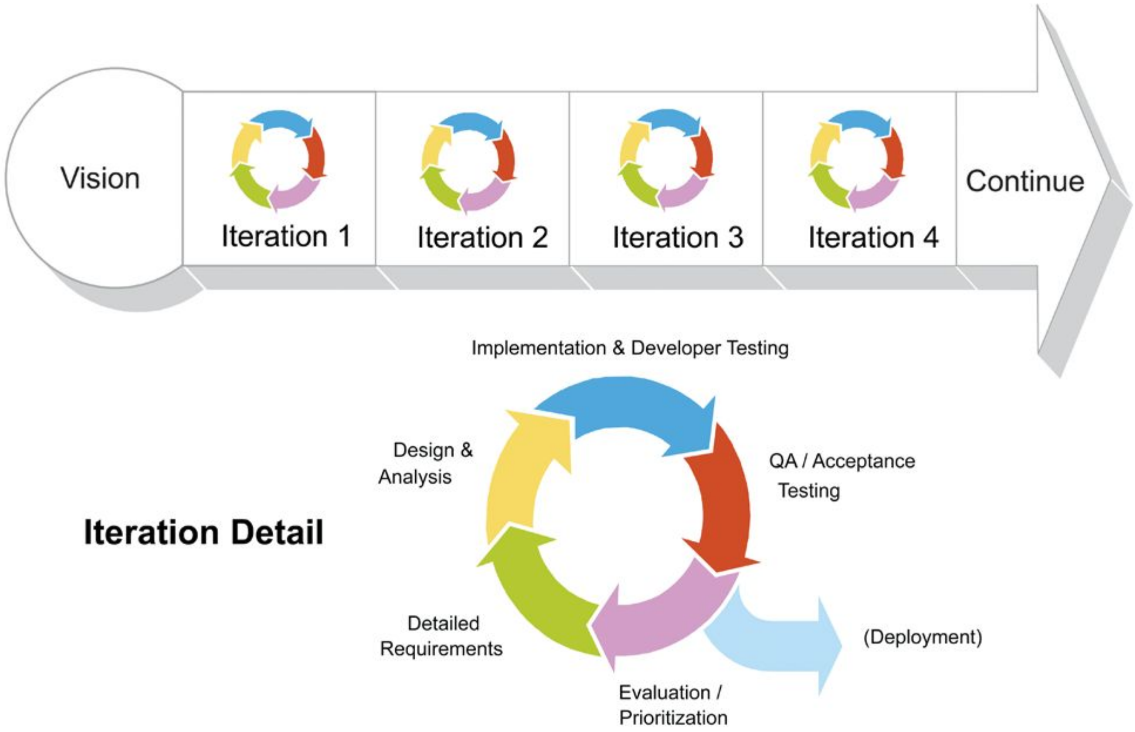


Fig 1. The stages involved in each sprint (Courtesy of http://scrumreferencecard.com/scrum-reference-card/)

Advantages this will grant:

Saves us time

Will help keep the project focused after each sprint we will be going back and looking over our requirements in detail.

Keeps us in touch with what each other are doing. We will be having a scrum meeting not daily but at the start and end of each sprint (On the monday and the friday.) We will also be meeting individually discussing issues we occur while working on the project.

Because of the frequency of these meetings problems are identified quickly and solved quickly.

Disadvantages:
“Scope creep” - We have a clear deadline however depending on our progress we might decide to add additional features this will inturn expand our project scope. If this happens and we have a major issue that takes us a while to fix we might waste 1 or 2 sprints and not deliver on another project goal.

Heavily dependent on the dedication of all team members

We believe in this case the advantages outweigh the disadvantages and will allow us to produce a higher quality program. (For defined roles see 3.3)

6. Conclusion

As project manager of MineBuster I feel that the project was a success this is because we have included everything that is required by the MVP. Additionally we have eclipsed those requirements see 6.1 features we added for some additional features we added. However we had a fair few issues along the way. One major issue was the with inventory system it took us 2 sprints to sort out properly. Programming issues aside I feel that one issue was either project members would not tell me when they had done a task or that they were doing it last minute. Whichever this was meant I sometimes did not know until the scrum meetings that were held at the end of each sprint. This meant I was sometimes late assigning a new task to someone which meant a slight loss in productivity.

I feel the hardest part of the project was the programming side of it this is because we were learning pygame’s functions as well as its inner workings and quirks. For example in order to display something on the screen you have to create a surface for that object blit that surface onto another surface and then you attach the surface to the screen. It's little things like that which were confusing to understand at first and even more annoying to bug fix.

6.1 The “State” of the game

The following is a list of features we added and we did not add to the program but we wish we did. The reasons we didn’t add certain features was because they we etheir ran out of time or because we didn’t assign someone to the task early enough for it to implemented and work properly:

6.1.1 What we managed to do:

Features we added
Everything under the MVP
2 searching (pathfinding) algorithms, 2 sorting algorithms
“Smart” Buttons (Button which have two states and are group only one can be active)
UI (Description system, Select searching/sorting algorithm, score, total worth of inventory)
Takes time to pick up resources a progression bar is displayed (each resource has a toughness)

Inventory system as a hole
Robot levels up the higher the level the quicker the mine speed.

6.1.2 What we didn’t manage to do:

Features we didn’t get round too
Shop System
“Bank” / Deposit system
Objectives system
Difficulty system