

# Earth Shard pre-production report



**MALEVOLENT  
CROW**

Version 1.0



# 1 CONTENTS

---

2	Analysis of brief .....	2
3	Project Aims .....	2
3.1	Aims specified in brief.....	2
3.2	Aims of graded unit .....	2
4	Factors influencing project .....	3
5	Constraints and risks .....	3
6	Functional and non-functional requirements .....	4
6.1	Functional .....	4
6.2	Non-functional .....	5
7	Development approach .....	5
7.1	Agile approach with kanban .....	5
7.1.1	Rejected approaches .....	6
7.2	Development Software .....	6
7.3	Development Platform.....	6
8	Asset acquisition.....	7



## 2 ANALYSIS OF BRIEF

---

I have been tasked with creating a pc or mobile game that is to be centred around one or multiple elements of nature (being fire, water, wind or earth). The age rating for the game should be for ages 12 and above. The game can also include themes that are educational in either science or history but must be focus gameplay first as fun is a priority.

Suggested narrative ideas are right a wrong, change the future, gain a remedy or save humanity. But these are not requirements. The game can take on any genre as long as its suitable for the requirements.

Detailed development requirements state it should be produced using object-oriented development environment with direct programming. At least one 3D object is to be included. As well as audio. All assets also must be compliant with copyright laws.

The game must have a god mode or developer tools for easy testing and showcasing. There should also be a full GUI with a menu system that should include a main menu, credits screen, player HUD and an end screen. Developer logos should also be present somewhere in the game. The gameplay should also last around 10-15 minutes.

There is to be no content that could be regarded as offensive or that would be unsuitable for an age range of 12 and up.

## 3 PROJECT AIMS

---

### 3.1 AIMS SPECIFIED IN BRIEF

- Create a PC or Mobile game.
- Have elemental theme with one of, or multiple elements of nature (i.e. fire, water, wind or earth).
- Be suitable for age ranges of 12 and above.
- Be a fully functional game developed with direct programming in an object-orientated development environment.
- Include at least one 3D element.
- Include audio.
- Have a fully functional UI with main menu, credits and end screen.
- Include a HUD.
- Be branded with developer logos.
- Have a god mode or developer tools for fast showcasing and testing.
- 10 to 15 minutes' worth of gameplay.
- Compliancy with copyright laws.
- Contain no inappropriate or offensive content.

### 3.2 AIMS OF GRADED UNIT

The aim of this unit is to show the range of skills I have developed over the past 2 years in this course. It will also be to prepare myself for further education within the games industry and future employment. This unit will also show my ability to work independently and apply a variety of skills and problem-solving techniques to a given project within a limited time scale.



It will also show my ability to push beyond what I have made before and should be the 'Magnum Opus' of my current progress in the games industry. Until I later create something even better in the future whether alone or as a part of a team.

## 4 FACTORS INFLUENCING PROJECT

---

This project is to be completed within 12 weeks with the bulk of the work being done within class.

The brief specifies several requirements that will have impact on what the project will be including theme like the 4 elements. Scale of the game as it must include at least 10 to 15 minutes of gameplay. Age rating of the game as it must be suitable for 12 and above. And other features the game must include such as screens and HUD.

The software required to build this project must be consistent for working on different computers. The unity version is most important for this.

This project will be the biggest and most consistent project I have worked on and will be challenging but within possibility of completion.

Motivation can vary for me around this time of year and could impact the speed of development as working from home can be a big struggle when motivation is short. This is further impacted by holidays around this time of year where I will not be at college and only be able to work from home.

Asset creation and acquisition will be a challenge as creating assets takes a lot of time and finding specific free or affordable assets can be difficult.

## 5 CONSTRAINTS AND RISKS

---

The time constraints for this project limit the scope significantly as I want prioritise quality over quantity. But this will be at the risk of the game possibly feeling small and may impact the length of gameplay. I hope to make up for this by making sure the levels are long even if the mechanics may not be mechanically complex.

The brief itself constrains the scope of the project as there are specific requirements that must be met for it to pass. This can be beneficial though as it gives clear borders on the projects scope which mitigates the risk of scope creep by constraining the scope. There is still the risk of scope creep appearing though, but I have taken the time to make sure this project is sound in its design and features. Where scope creep could occur is with the level design as these designs currently not as sound due to testing of the core abilities of earth shard being required for a clearer design. But the core ideas of what the levels will include are sound.

There are many risks involved with the hardware and software of the project such as crashes or corruption cause detrimental damage. To mitigate this risk, I am using Github to backup all my files. This allows me to revert any potentially damaging changes that could corrupt the project.

This project being one of the biggest projects I've worked on is a huge risk as it will be a challenge. Some features will be a definite challenge as there are no tutorials available for



what I plan on making. As most features will be designed by myself means I have a deeper understanding of the project than if I followed a tutorial meaning the risk of potential game breaking bugs can be mitigated easier as I will be able to find solutions faster.

Burnout can heavily affect the progress of a project and can be rather hard to predict, I have made sure a lot of the projects design is sound to avoid the damaging effects of burnout as it effects my motivation more creatively. If I have a sound plan already in place it will be easy to just follow already laid steps than having to design at the same time. Working from home effects my motivation and can risk the progress. To mitigate this, I will make sure I utilise all my time in class to get large bulks of work done.

A big constraint is the acquisition of assets. Creating assets takes a lot of time and risks other parts of the project by taking up that time. A way around this is to acquire most assets from 3<sup>rd</sup> party sources but this constrains the overall design of the project as it will be harder to keep designs consistent between assets. This is mitigated by a low poly design that is naturally more consistent but also means I am only constrained to low poly assets.

## 6 FUNCTIONAL AND NON-FUNCTIONAL REQUIREMENTS

---

### 6.1 FUNCTIONAL

- Movement and control
  - Directional movement on x and z axis with WASD
  - Jumping for vertical movement on Y axis plus gravity
  - Looking with mouse for aiming
  - Spawning and throwing rocks with mouse
  - Spawning and raising pillar with mouse
  - Keyboard button for ability switching
  - Enemy AI navmesh
  - Enemy move and attack states
- Collisions
  - Player with environment
  - Player with enemies and enemy projectiles
  - Player with ground raise ability
  - Projectiles with environment
  - Projectiles with targets
  - Projectiles with enemies
- Menu navigation
  - Start game
  - Exit game
- Win/Lose conditions
  - Make it to end of level and exit (win)
  - Take too much damage from enemies
- Core features
  - Player rock throw ability (summons and throws rocks)
  - Player ground raise ability (ability to summon rock under self and raise it)
  - Health system for losing health that can lead to lose
  - Switches for activating doors
- Levels



- Enclosed environment
- Enemies present in level
- End to level

## 6.2 NON-FUNCTIONAL

- Controls
  - Controller support for controls
- Sound
  - Player movement sounds
  - Player death sound
  - Rock throw sounds for spawning, throwing and colliding
  - Ground raise destruction noise, spawning noise and raising noise
  - Enemy sounds
  - Enemy attack sound
  - Enemy projectile impact sounds
  - Enemy death sound
  - Music
  - Switch noises
  - Door/event activation noises
- Animation
  - Player hand Movement animations
  - Player hand movements for abilities
  - Rock throw animations for raising
  - Enemy animations for moving and attacking
- HUD
  - Red glow around screen when taking damage
  - Crosshair to indicate selected ability
- Menu system
  - Pause button with return to menu and restart options
  - Main menu with settings, credits and level select screen
- Graphics and models
  - Environmental props
  - Consistent art style
  - Detailed enemy models
  - Detailed switches
  - Detailed rock throw ability rock
  - Detailed ground raise ability pillar

## 7 DEVELOPMENT APPROACH

---

### 7.1 AGILE APPROACH WITH KANBAN

Earth Shards development will use an agile methodology with 3 sprints. Although each sprint will take a heavier focus on a particular aspect. A Gantt chart will be created at the initial stage for setting up a roadmap for development, but Kanban will be utilised throughout the development process as it allows for a much easier time tracking tasks.



The initial Gantt chart will be created with Gannt Project but the rest of the project management side of the project will be handled through Github projects as it can integrate with development making it easier for bug tracking and adding feature too earth shard.

Agile allows for iteration of features, which means the end product can be cleaner and more refined as certain features may sound good on paper but not be as good in practice so by allowing an agile approach allows those features to be improved upon in later iterations, so it functions in a way that is desirable.

Kanban mean 'signboard' in Japanese and does as it says. It breaks tasks down into 5 distinctive boards. Backlog, ready, in progress and complete. This allows for developers to easily understand what is happening with the project immediately. Using kanban on Github also allows developers to click on certain tasks and see the specifics and further detail.

I have experience with this approach, and it has been successful for me in the past, so it is what I have chosen as an approach for Earth Shard

#### 7.1.1 Rejected approaches

Other approaches were considered at first one of the first to be rejected was waterfall as it is far too basic for the complexity of this project. Waterfall is strict and allows less freedom for possibilities of redesign and test features that may not be suitable for a final version of the game.

Scrum is another popular approach especially when working in teams but can be rather complex to learn and manage. Due to this being a solo project and not having enough knowledge of how to use scrum it was rejected in favour of a simpler agile with kanban approach.

## 7.2 DEVELOPMENT SOFTWARE

Earth Shard will utilise mostly free tools as these are what I will have available to me in college and at home.

Tools used will be:

- Github desktop for version control and saving files.
- Github projects for tracking development.
- Unity version 2022.3.36f1
- Microsoft Word for documentation
- Visual studio 2022 for IDE
- Blender for 3D models
- Internet browser for Github and online assets
- Gannt Project for initial project plan

## 7.3 DEVELOPMENT PLATFORM

Earth Shard will be mainly designed for a PC platform as that is what it will be built on. Earth Shard will use mouse and keyboard primarily, but controller support will be available.



## 8 ASSET ACQUISITION

---

Majority of the assets will be acquired from 3<sup>rd</sup> party sources due to asset creating taking a large amount time that will take away from other important parts of the project. I also do not have the greatest skills in assets creation meaning if I want a consistent product I should acquire them from 3<sup>rd</sup> party sources.

Website I will acquire assets from include:

<https://assetstore.unity.com/>

<https://sketchfab.com/>

<https://polyhaven.com/>

<https://freesound.org/>

<https://www.mixamo.com/>

any assets I cannot find online but require I will create myself through blender.