

Project Hero

Feasibility Study

01001000
01000101
01010010
01001111

Team: DJORDJEVIC Filip
REICHL Markus (*Project Manager*)
TEKIN Abdurrahim
WELLNER Florian

Supervision: DOLEZAL Dominik

January 16, 2017

Abstract

Project Hero is a **bullethell roguelike multiplayer RPG** providing a very unique storytelling and gameplay experience. Its main goal is to create a solid base for a video game ready for further development. The core should be highly abstract, stable and easy to extend.

This document goes into detail with the projects technical feasibility and reviews different approaches resulting into a clear conclusion and the best way to go.

Contents

1	Introduction	2
2	Project Goals	3
2.1	Targets	3
3	Technical Feasibility	4
3.1	Team	4
3.2	Requirement Analysis	5
3.2.1	Functional Requirements	5
3.2.2	Technical Requirements	6
3.2.3	Conclusion	6
3.3	Engines and Frameworks	7
3.3.1	Monogame	7
3.3.2	Unity	7
3.3.3	Vulkan	7
3.3.4	Comparison	8
3.3.5	Conclusion	8
4	Economic Feasibility	9
4.1	Market Research	9
4.1.1	Conclusion	12
5	Schedule Feasibility	13
6	Conclusion	13

1 Introduction

Project Hero is a **bullethell**^I **roguelike**^{II} **multiplayer**^{III} **RPG**^{IV} providing a very unique storytelling and gameplay experience.

It is highly inspired by previous titles in the roguelike and roleplay game genres, the most noteworthy being [Realm of the Mad God](#), [Titan Souls](#) and [Enter the Gungeon](#).

While all mentioned games are using a topdown pixelart setting, the Hero Project is drawn in a very unique combination of high and low resolutions.

^I A video game sub-genre where the screen is usually covered in bullets.

^{II} A video game sub-genre which, based on the [Roguebasin Interpretation](#), is defined by

- **Permanent Failure:** The player is encouraged to take responsibility for the risks he takes.
- **Procedural Environments:** Most of the game world is generated and provides complexity in resources and other elements of the game.
- **Resources:** The player can manage a limited amount of resources.

The [Roguebasin](#), [Berlin](#) and [Rogueteple](#) Interpretation may give a more detailed explanation on this subject.

^{III} A multiplayer game allows but does not require clients to play together. The Hero Project is not meant to be played online only!

^{IV} A roleplay game is a game in which the player assumes the roles of characters in a fictional setting and takes responsibility for his acting either through literal acting.

2 Project Goals

The main goal of Project Hero is to create a solid base for a video game ready for further development. This base includes all elements listed in the *Functional* and *Technical Requirements* sections found in the *Requirement Specification* document.

2.1 Targets

Before development can start the aim of the project has to be as clear as possible. Therefore the following targets and non-targets have been declared.

Targets

A core game A prototype ready for further development including all Functional and Technical Requirements. Date of submission is set on 17th of January 2017.

A prototype for presentation A simple setup including a player, an enemy and items including effects to declare an attack.

Non Targets

A full game The core game is not meant to provide a full game including graphics, gameplay and multiplayer. This goal is not reachable in time and may not be in budget.

A demo version The core game should demonstrate how the game works and how it can be extended. Not how the game will look like in production,

3 Technical Feasibility

With a video game being an enormous software project this study is mostly technical which requires the reader to have at least basic knowledge about video games and software development. Technical terms will still be explained as footnotes at the bottom of each page.

3.1 Team

Because the majority of the product will be developed by a rather small team each members technical experience and skill has a huge influence on the projects schedule and quality. The Hero Projects core as a school project is developed by a static team of 4 members and 1 professor listed below.

DJORDJEVIC Filip

Roles	Sound Designer
Areas of expertise	Music and Sound Design

REICHL Markus

Roles	Project Manager (<i>Product Owner</i>)
Areas of expertise	Concept Art and System Architecture

TEKIN Abdurrahim

Roles	Artist
Areas of expertise	Art and Design

WELLNER Florian

Roles	Software Developer
Areas of expertise	Software Development

DOLEZAL Dominik

Roles	Professor (<i>Supervisor</i>)
Areas of expertise	Software Development

While providing a wide variety of skills the majority of the team is also well attuned. Everyone has at least some experience with video games but only a few have already worked on one.

3.2 Requirement Analysis

The following section analyses all Functional and Technical requirements found in the Requirements Specification document and gives some short information about the teams thoughts on their technical feasibility.

3.2.1 Functional Requirements

Artificial Intelligence

Als are basically characters controlled by the game, which means that an AI differs from a character only by his controlling unit.

Character

A character, according to the Requirements Specification document, defines an entity controlled by the player. It should be able to move, take actions and carry an inventory. The first two depend directly on the users input, which is hard to keep platform independent assuming that the game is not using another engine. Implementing a levelling system for the player which interacts with a characters attributes should not be a problem.

Effects

In order to keep the core clean while adding lots of new features to the game effects have to be abstract and highly encapsulated. They should not depend on the identity of their owning entity and therefore work out of the box. A decoupling pattern called the Component Pattern can be used to manage multiple effects and keep the game extendable. Most popular engines also rely on this pattern.

Items

In order to store items for multiple players at least some kind of a database is required. There are multiple solutions for this problem, lots of game engines even bring their own database implementations. Items also have to be store- and even equipable by the player which makes them great candidates to hold effects the player can activate. Items should only usable by the player because they only interact through their effects which also covers the requirement *Item Interaction*. This is possible because all other entities use effects as components directly.

Attributes

Attributes are simple values and can easily be interacted with. They should also be treated like components and not be restricted to characters only.

3.2.2 Technical Requirements

Cross-Platform

Cross-Platform is hard to guarantee while not using an engine which is designed for multiple devices. Most programming languages nowadays can be used on any popular operating systems which at least fits the requirements. Using an engine would allow the game to be ported to more platforms including consoles, smartphones and tablets.

Extendable

In order to keep the core extendable abstraction and decoupling are really important. All parts of the game have to work independently including later extensions.

Multiplayer Ready

Assuming a server while developing the core is not as easy as it seems but possible when considered from the start. Here too an engine can help and improve the project lot.

3.2.3 Conclusion

The development process is quite intense and might not fit the schedule when every part of the game is hard coded. The team suggests using a framework which already provides essential components. Possible candidates are engines like Gamemaker and [Unity](#) or frameworks like [Monogame](#) or [Vulkan](#).

3.3 Engines and Frameworks

Because of the volume of this project the team highly recommends using an engine which already provides key elements and can be easily extended. After comparing lots of possible options the team decided on 3 major possibilities.

3.3.1 Monogame

Monogame is an open-source framework based on Microsofts XNA-4¹. It provides cross-platform support and is optimised for 2D. Popular games using Monogame include the indie game [FEZ](#).

It allows but also requires the developer to write critical game code and is very lightweight.

3.3.2 Unity

Unity is a huge game engine and built for 3D games while also providing 2D support. It comes with a lot of features and a solid framework for both newbies and professional game developers. Lots of popular games were developed in Unity recently including [Ori And The Blind Forest](#), [Rust](#) and one game also found in the Requirement Specification document [Enter the Gungeon](#).

3.3.3 Vulkan

Vulkan is the next level of OpenGL released in 2016. Using Vulkan basically means working from scratch. It allows the user to work very close to the computers hardware and offers a lot of possibilities not found in OpenGL.

¹ XNA Is Not Acronymed is a video game framework developed by Microsoft

3.3.4 Comparison

To compare these options the following table rates all features relevant for the project from a minimum of 1 to a maximum of 3.

	Monogame	Unity	Vulkan
Features	2	3	1
Documentation	1	3	1
Performance	2	2	1
Cross-Platform	3	3	0
Popularity	1	3	1
	9	14	4

Looking at this short comparison it seems rather clear which framework is the one to choose.

3.3.5 Conclusion

Comparing all these engines and frameworks the team got really excited looking at the games running on Unity and its amazing features which would otherwise be really hard to implement. While Monogame and Vulkan give the developer a lot of freedom, Unity gives a whole swiss army knife. Looking at the teams schedule and the features the project needs Unity seems to be a great option and maybe the only option.

4 Economic Feasibility

Although the core game itself is not sold to any user base the team did some research and gathered information about the market. Right now there are many unknown and few popular bullethell roguelikes available including the four mentioned in the previous documents.

4.1 Market Research

A comparison of those games, their players and economics, can be found below. All information was gathered on the 26th September of 2016.

Realm of the Mad God



Realm of the Mad God, or ROTMG for short, is a bullethell MMORPG¹ currently played by 1.7 billion accounts. It is free to play and makes profit from in game transactions which brought in 1.36 billion Euros in total since 2011.

While the game still has a high user base a lot of players actually wish for a better game.

¹ A massively multiplayer online role-playing game is a story-driven online video game in which a player interacts with a large number of other players.

Titan Souls



Titan Souls is a roguelike RPG released in 2015. Since then it gathered 170 thousand players and acquired 1.65 billion Euros in total. It is available on steam for 14.99 Euros at the moment.

It uses a clear pixelart style and has basically no interface which improves gameplay a lot and makes a great game.

Dungeon Souls



Dungeon Souls is also a bullethell roguelike RPG released in 2015 and counts about 35 thousand players. It was developed by a indie studio and is now available on steam for 12.99 Euros which is quite a lot for this genre. The game gathered 300 thousand Euros since its release.

Enter the Gungeon



Another bullethell roguelike RPG released in 2016 with a great art style and a huge user base of 414 thousand players which is quite a lot for an indie game. It is now sold for 14.99 Euros and gathered 4.095 billion Euros since its release.

4.1.1 Conclusion

The information gathered while researching for these games shows lots of possibilities but also sets high quality standards. There are many small indie developers in this genre but only few who rise to the top mostly because of their creativity or storytelling.

5 Schedule Feasibility

Time is a very important factor in this project because it can become very consuming and may even take too long to be completed. Due to the size of the team a single member can influence the schedule a lot.

Important milestones are the **prototype deadline** and the **project approval** on the 17th December of 2017 which gives the team a hundred days, excluding sun- and holidays, to provide the required specifications.

Although a hundred days would be quite enough while working 40 hours per week the team only has a regular work-time of 4 hours a week what leads to 60 hours per person which definitely is not enough to complete the project without a framework.

6 Conclusion

All features required by the previous document are viable and can be completed in time even though the development process will might become very intense. An engine would help a lot, even more when the core game is finished and has to be extended. The team suggests using Unity for this purpose due to its large amount of features and solid base.

This said Project Hero is considered ready to start.