# Project Hero

Functional Specification

```
01001000
01000101
01010010
01001111
```

**Team:** Djordjevic Filip
Reichl Markus *(Project Manager)*
Tekin Abdurrahim
Wellner Florian

**Supervision:** Dolezal Dominik

January 15, 2017

**Abstract**

Project Hero is a **bullethell roguelike multiplayer RPG** providing a very unique storytelling and gameplay experience. Its main goal as a school project 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 defines all features the product has to provide, as a response to the *Requirement Specification* document, using the results of the teams *Feasibility Study*.

# Contents

# 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.

# 2 Project Goals

The main goal of Project Hero as a school project 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 Specifications sections found later in this document.

## 2.1 Targets

**Targets** The team is going to provide a core game as mentioned earlier and also a prototype for presentation.

**Non Targets** The team is not thought to provide a demo version or even a complete game.

[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 Roguetemple 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.

# 3 Functional Specifications

## 3.1 /FS10/ Artificial Intelligence

Entities[I] similar to a character and controlled by the game can interact with their environment by using effects. The game itself will differentiate between two main types of AIs, NPCs[II] and Enemies, for most of the time. This does not imply that an AI is either a NPC or an enemy, they just differentiate through their behaviour.

**NPCs** NPCs may interact with the player through conversation or providing resources.

**Enemies** While NPCs are most likely to be friendly, enemies will use their effects to harm the player. They can also be used to provide items and other resources for the player when an enemy is killed.

**Other AIs** AIs different from NPCs and enemies can be objects provided by the environment which interact with the player by using effects.

## 3.2 /FS20/ Character

Characters are entities directly controlled by the client or the game. Every character relies on attributes and has the ability to move, level and use effects.

### 3.2.1 /FS20.1/ Movement

The character's movement can be controlled by the game or the client using an input of his choice.

### 3.2.2 /FS20.2/ Levelling

A character is able to gain levels which directly modify his attributes. Attributes can also influence AIs, mostly by modifying their attributes,

[I] Every interacting object in the game is referred to as a (game) entity

[II] A non player character is a character which is not controlled by the player spawn rates[I] or resources. Levels can be given by default or increase on specific actions. Common ways for the player to improve his level include fighting enemies and completing quests[II].

## 3.3 /FS30/ Effects

Entities can be given effects allowing them to interact with their environment. They can be activated, enabled or interact permanently. Common effects include attribute manipulation, conversations and also attacks. Players should be able to control their effects using an input of their choice.

### 3.3.1 /FS30.1/ Attack

A specific effect, which is also implemented in the core game is the attack. Attacks are used to reduce another entity's *life* attribute leading to its death. Attacks interact by firing projectiles.

**Projectiles** Projectiles are used by attacks to harm other entities. Short range attacks are implemented as a single projectile with a bigger hitbox[III].

## 3.4 /FS40/ Resources

Resources refer to all collectable objects in the game. They do not interact by themselves but can be interacted with through a character's inventory.

## 3.5 /FS41/ Items

Items are resources that can be given effects, which can only be activated by another entity. They should also be ready to be displayed to the player although this feature will not be implemented in the core game.

[I] Entities created in a limited amount of time

[II] A search or pursuit made in order to find or obtain something. In this case a quest describes a task for the player to complete.

[III] An area in which one object interacts with another

### 3.6 /FS42/ Inventory

An inventory keeps a limited amount of items and can hold base items, equipment and other resources.

### 3.7 /FR50/ Attributes

All entities using this functionality are influenced by 8 attributes called *health, mana, damage, defence, vitality, wisdom, speed* and *agility*. Their values are used as variables in the entities actions and can also be modified. The number of attributes can also be increased during development. Attributes in the core game have to be accessible but not yet interactable for an AI.

### 3.8 /FR60/ Player

A single client is able to control at least one character's actions and manage its resources.

## 4 Usability Specifications

### 4.1 /US10/ Minimalist User Interface

While providing a huge amount of complexity the game also provides clarity for the user using a minimalist user interface. This improves the usability a lot and makes the game more enjoyable for the player.

## 5 Technical Specifications

### 5.1 /TS10/ Cross-Platform

The game runs on multiple platforms including all major desktop operating systems and popular consoles using the cross-platform features of the Unity game engine.

### 5.2 /TS20/ Extendable

The core game will be highly extendable through abstraction, decoupling and maintainability.

### 5.3 /TS30/ Multiplayer Ready

Because the actual game is thought to be played on a server with multiple clients the core is already developed for this purpose. A real multiplayer is not necessary in the core game but definitely nice to have.

## 6 Quality

Extendability and functionality are high priority for the core game because of their huge influence on further development. High quality graphics are only required for presentation and are therefore considered low priority.