



**Design Document for:**

# **Raising Hell**

**Operation: Apocalypse**

“There is so much happening right now!”™

All work Copyright ©2018 by The Losers’ Club

Written by Anastasia Miles, Kyle Bartz, Goutham Rajeev, Jeff Pham, Rhian Advincula

Version # 1.00

December 9, 2018

# Table of Contents

<b>DESIGN HISTORY</b>	5
<b>VERSION 0.1</b>	5
<b>VERSION 0.9</b>	5
<b>VERSION 1.0</b>	5
<b>GAME OVERVIEW</b>	6
<b>PHILOSOPHY</b>	6
<b>GETTING “WRONG” JUST RIGHT</b>	6
<b>SIMPLICITY AND CONSISTENCY</b>	6
<b>EXCLUSIVE PC SUPPORT</b>	6
<b>COMMON QUESTIONS</b>	6
<b>WHAT IS THE GAME?</b>	6
<b>WHY CREATE THIS GAME?</b>	7
<b>WHERE DOES THE GAME TAKE PLACE?</b>	7
<b>WHAT DO I CONTROL?</b>	7
<b>HOW MANY CHARACTERS DO I CONTROL?</b>	7
<b>WHAT IS THE MAIN FOCUS?</b>	7
<b>WHAT’S DIFFERENT?</b>	7
<b>FEATURE SET</b>	9
<b>GENERAL FEATURES</b>	9
<b>LEVELS</b>	9
<b>ROOMS</b>	9
<b>2D GRAPHICS</b>	9
<b>HACK-AND-SLASH COMBAT</b>	9
<b>ROGUELIKE</b>	9
<b>HEALING ITEM</b>	9
<b>FIGHTING THE FORCES OF GOOD</b>	10
<b>THE BOSSES: THE HALLOWED</b>	10
<b>MULTIPLAYER FEATURES</b>	10
<b>EDITOR</b>	10
<b>GAMEPLAY</b>	10
<b>FAST-PACED COMBAT &amp; ROGUELIKE</b>	10
<b>INTERESTING OFFENSIVE AND DEFENSIVE DYNAMICS</b>	10
<b>UNCOMMON VISUAL REPRESENTATION OF POWERS</b>	11
<b>UNIQUE AND ORIGINAL SETTING AND QUEST, UNUSUAL POSITION AS AN ANTI-HERO</b>	11
<b>THE GAME WORLD</b>	11
<b>OVERVIEW</b>	12
<b>LEVELS</b>	12
<b>ROOMS</b>	12
<b>PROCEDURAL GENERATION</b>	12
<b>THE PHYSICAL WORLD</b>	13

<b>OVERVIEW</b>	13
<b>KEY LOCATIONS</b>	13
<b>TRAVEL</b>	13
<b>SCALE</b>	13
<b>OBJECTS</b>	14
<b>RENDERING SYSTEM</b>	14
<b>OVERVIEW</b>	14
<b>2D RENDERING</b>	14
<b>PARTICLE SYSTEMS</b>	15
<b>CAMERA</b>	15
<b>OVERVIEW</b>	15
<b>CLIMBING CAMERA</b>	15
<b>PARALLAXING</b>	15
<b>GAME ENGINE</b>	15
<b>OVERVIEW</b>	15
<b>LEVEL GENERATION</b>	16
<b>WATER</b>	16
<b>COLLISION DETECTION</b>	16
<b>INPUT MANAGER</b>	16
<b>ENEMY BEHAVIORS</b>	17
<b>LIGHTING MODELS</b>	17
<b>OVERVIEW</b>	17
<b>THE WORLD LAYOUT</b>	17
<b>OVERVIEW</b>	17
<b>LINEAR LEVELS</b>	17
<b>ROOMS</b>	18
<b>GAME CHARACTERS</b>	18
<b>OVERVIEW</b>	18
<b>CREATING A CHARACTER</b>	18
<b>THE MAIN CHARACTER: THE CULTIST</b>	18
<b>ALLIES: THE DEPRAVED &amp; DEMONS</b>	19
<b>ENEMIES: THE HALLOWED &amp; THE HOLY</b>	19
<b>USER INTERFACE</b>	21
<b>OVERVIEW</b>	21
<b>HEALTH, AMMO, AND MAGIC</b>	21
<b>TRANSITIONS</b>	21
<b>MENUS</b>	21
<b>INTRODUCTION SEQUENCE</b>	21
<b>WEAPONS</b>	22
<b>OVERVIEW</b>	22
<b>MELEE: BEAR CLAWS</b>	22
<b>RANGED: SKULL COMPANION FIREBALLS</b>	22

<b>MAGIC: SKULL COMPANION AREA-OF-EFFECT FIRE</b>	22
<b>MUSICAL SCORES AND SOUND EFFECTS</b>	23
<b>OVERVIEW</b>	23
<b>AUDIO TOOLKIT</b>	23
<b>SOUND DESIGN</b>	23
<b>SINGLE-PLAYER GAME</b>	24
<b>OVERVIEW</b>	24
<b>LINEAR PROGRESSION</b>	24
<b>PLATFORMING</b>	24
<b>HACK-AND-SLASH COMBAT</b>	24
<b>ROGUELIKE</b>	24
<b>STORY</b>	25
<b>HOURS OF GAMEPLAY</b>	25
<b>VICTORY CONDITIONS</b>	25
<b>MULTIPLAYER GAME</b>	27
<b>OVERVIEW</b>	27
<b>CHARACTER RENDERING</b>	28
<b>OVERVIEW</b>	28
<b>RENDERING STATE MACHINE</b>	28
<b>FLASH-ON-HIT SHADER</b>	28
<b>EYE COLOR SHADER</b>	28
<b>WORLD EDITING</b>	29
<b>OVERVIEW</b>	29
<b>FUTURE FEATURES</b>	30
<b>OVERVIEW</b>	30
<b>GENERAL FEATURES</b>	30
<b>ASSET IMPROVEMENTS</b>	30
<b>BACKEND IMPROVEMENTS</b>	30

# Design History

Each section below describes high-level changes made in that version of this document.

## Version 0.1

Version 0.1 is the in-progress first draft of this document. Many aspects of the game are yet undecided, so some information in the document at this stage will not match the final state of the game.

## Version 0.9

Version 0.9 is the second draft of this document. Our design direction was far more solid, and much of the design document was revised to reflect this. Many aspects had to be marked as future plans, as our rate of progress would not permit us to complete every detail we had hoped to see in a final product.

## Version 1.0

Version 1.0 is the fully-revised final draft of the document. It is nearly identical in content to Version 0.9 and changes primarily include grammatical corrections and content clarifications.

# Game Overview

## Philosophy

### Getting “Wrong” Just Right

This game is not driven by narrative and is not intended to discuss any serious issues, philosophical or otherwise. It is not meant to leave an unforgettable mark on the world or to revolutionize the gaming industry. What it is meant to do is entertain. It takes what is known to work--what is known to be fun and enjoyable--and adds a unique twist to it.

We combine some of the most popular genres (roguelike, platformer, hack-and-slash, etc.) to create for the player a fun, challenging, fast-paced, memorable game that takes place in a unique setting. Rather than relying on the classic trope of the player taking on the role of a heroic protagonist out to save the world, we chose to have as our main character a simultaneous protagonist and antagonist out to “save” the world as he sees fit. Though this questionable character and his equally questionable quest are primarily intended to provide a comedic and entertaining setting, there remains the potential for one philosophical discussion: that of right and wrong. Through the character the player may come to understand the significance of perspective in the classification of right and wrong, as well as the subjectivity of the debate as a whole” often what is wrong for one appears right to another.

### Simplicity and Consistency

*Raising Hell* has as one of its core characteristics simplicity. From the setting to the narrative to the combat style, simplicity reigns over all. Upon first glance this may seem like a poor decision and one that would produce boring gameplay. However, the playstyle of this game thrives when the underlying system is simple. Our game, which focuses on combat, is intended to be extremely fast-paced and chaotic. Having too much narrative interference, an overly-complex combat system, or too many mechanics would make the game confusing, artificially challenging, and potentially even unplayable. Thus, we decided to implement a vague, high-level narrative alongside simple combat mechanics with only three weapon types (two of which will be available in the initial demonstration). Combined with the pacing of the combat and the spawn rate of enemies, this should provide fun and challenging gameplay.

While we favor simplicity over complexity, we are extremely careful not to sacrifice consistency. We intend to pay careful attention to things like hitboxes (etc.) to ensure that even though the combat is simple, it is effective and consistent. When one opts for simple controls and mechanics they must work perfectly.

### Exclusive PC Support

Given the time constraints on development, we decided to offer support only for PC. Porting the game to Xbox, PlayStation, or Nintendo platforms may be a possibility in the future, but our first priority is to create a well-functioning, satisfying game. If we stretch ourselves too thin by trying to support too many platforms, the quality of our game will suffer. That said, we do support both keyboard and Xbox controllers for input.

## Common Questions

### What is the game?

*Raising Hell* is a fast-paced, combat-centric game in which the player takes on the role of a cultist who serves the four reigning demon lords known as the Depraved. The cultist has the unique quest of finding and rescuing the Depraved who have been captured by their divine counterparts, the Hallowed. To do so, he must fight through hordes of holy enemies to reach, confront, and defeat the Hallowed. The game thus consists of a series of levels, each filled with enemies and ending with a Hallowed boss. The player

navigates these levels, which implement typical platforming mechanics, and along the way opposes the holy adversaries through the use of special demonic powers bestowed upon him by the Depraved shortly before their capture.

### **Why create this game?**

This game is being developed with the intention of combining a fun, well-known combat genre with an unconventional setting and quest. Most of the developers are avid subscribers to hack-and-slash roguelikes and are well-versed in the genre; thus, this is the perfect type of game for the team to take on. Still, we wanted to set *Raising Hell* apart from the other games of its type by giving it a unique twist. This takes the form of the protagonist and his mission. In contrast to most games in which the user takes on the role of the hero, in *Raising Hell* the player will be an anti-hero--and arguably even a villain. Of course, this is meant to make the player question the role of perspective in the definition of right and wrong. In the end our goal is to create something entertaining and memorable, even if the mechanics behind the game are not particularly revolutionary.

### **Where does the game take place?**

*Raising Hell* takes place during an enormous war between demons and angels over the fate of the world. The former, led by the Depraved, seeks to bring about the Apocalypse and thus the end of mankind, while the latter, led by the Hallowed, desires to enact the Deliverance, or the salvation of man. This war had reached a stalemate, until the five angel lords known as the Hallowed managed to capture their Depraved counterparts and put the demons at a severe disadvantage. Discovering this, the cultist, controlled by the player, takes it upon himself to rescue the Depraved and ensure the Apocalypse. To accomplish this quest, the cultist must fight through a series of levels, each taking place in a setting relevant to a Depraved-Hallowed pair, to find and free his demon masters.

### **What do I control?**

The player controls a robed cultist who serves the Depraved. The cultist has been imbued with special powers by the demon lords, which he uses to aid him in the battle against the Hallowed and their minions. These powers include spawning magic claws with which to rend enemies, summoning a demonic creature to shoot ranged missiles, area-of-effect magic spells (not included in the initial demonstration), and limited shapeshifting.

### **How many characters do I control?**

The player will only ever control the main character, which is the cultist. At some point, we may add limited customization options, such as changing the color of the robes, but this is not a priority and will be considered a stretch goal.

### **What is the main focus?**

The cultist's goal in *Raising Hell* is to find and rescue his Depraved overlords from their holy captors. This involves fighting through levels of enemies who serve the Hallowed to reach and defeat the Hallowed themselves. In doing so, the cultist will free the Depraved one by one while simultaneously weakening the angelic forces, thus giving advantage to the demons in winning the war and achieving the Apocalypse.

### **What's different?**

The general combat mechanics of *Raising Hell* are not particularly new or different, but we add a unique twist to how they appear and are controlled. For instance, having ranged, melee, and magic attacks is a common pattern in hack-and-slash games. However, our presentation of offensive weapons as summons or polymorphisms rather than physical instruments is something that is rarely seen. Further, the restrictions we place on offensive techniques is uncommon. One example is how the player obtains ammo

for his ranged attack, which is through landing melee hits rather than through the collection of physical bullets. Another special aspect of *Raising Hell* is its context, in which the player takes on the role of a questionable anti-hero rather than a righteous protagonist.



# Feature Set

## General Features

*Raising Hell* takes place in a mid-Apocalyptic world in which the demons and angels are each trying to bring their own end to the world. Described below are some of the general features of the game and its setting.

### Levels

The game is organized into five levels, with “level zero” being a shorter tutorial level. Each level is themed around one Depraved-Hallowed pair; for instance, the first level is based on Famine and his Hallowed counterpart Bounty. The level design, enemies, and boss fight all conform to this theme. The second level contains Pestilence and Vim & Vigor, the third has War and Peace, and the last has Death and Life. Levels proceed linearly and end in a boss fight; that is, the player fights his way through a horde of enemies to face a Hallowed boss and free a Depraved master. Only the tutorial and Bounty levels will be present for the initial demonstration.

### Rooms

Each level is split into rooms of varying sizes. The minimum size is equal to the size of the player's screen, but some rooms are larger and allow the camera to follow the player around. A player is trapped inside a room and cannot proceed until all enemies within the room are defeated.

### 2D Graphics

The game is a two-dimensional side scroller. All assets are low-resolution pixelated entities. For instance, the player entity in his default stance is 13px wide and 32px tall. We use 32-bit coloring. All of the assets are original and custom-made.

### Hack-and-Slash Combat

*Raising Hell* implements fast-paced, hack-and-slash combat. The cultist has the ability to polymorph his arm into vicious claws for melee attacks, to summon a flying skull that shoots ranged missiles, and to draw upon the dark energy of his Depraved masters to unleash a magical area-of-effect attack. The main form of offense will be the melee attack, as both ranged attacks and magic require the player to land melee hits or to kill enemies to recharge. The player is also able to crouch and dodge, which are both actions that cause him to polymorph into animals. Note that all of these mechanics will be present for the initial demonstration save the magic attack, which will be implemented for the final release of the game at a later date.

### Roguelike

A typical roguelike will require the player to restart the entire game upon death. We decided that this might be too harsh or tedious due to the expected frequency of death in this style of gameplay; instead, we opted to have the player restart only the current level upon death. This should be a sufficient but not overly-harsh penalty, and still fits the roguelike genre.

### Healing Item

The player will be equipped with a three-use healing item. Uses will be regenerated after a threshold number of kills. The healing item will restore the health of the player so he can take more damage before dying. This feature will be present in the final rendition of the game, but will not exist for the initial demonstration.

## **Fighting the Forces of Good**

The player takes on the role of a cultist supporting the demons and their Depraved leaders in a war against the angels over the fate of the world. Thus, the player will be facing the forces of good which includes clerics, paladins, angels, saints (unnamed), various creatures and animals, and more.

## **The Bosses: The Hallowed**

The cultist's ultimate goal is to free the Depraved by defeating the Hallowed, thus ensuring the victory of the demons in the war over the fate of the world. As such, the boss of each level (excluding the tutorial) is one of the Hallowed. They appear in the following order: Bounty, Vim & Vigor, Peace, Life. Each boss has powers relevant to the domain they represent.

## **Multiplayer Features**

This game will not support multiplayer functionality. In the distant future, after the initial release of the game, we may consider adding multiplayer capabilities.

## **Editor**

We do not intend to support a level editor or enemy editor for *Raising Hell*. All levels and enemy spawns will eventually be procedurally and randomly generated, which will add sufficient variety to the gameplay. As with multiplayer support, we may later consider releasing downloadable content that will allow users to build their own levels and enemies.

## **Gameplay**

### **Fast-Paced Combat & Roguelike**

We chose to implement extremely fast-paced combat to make the game difficult and stressful. Most of the developers are fans of games like *Dark Souls*, *Diablo 3*, *Hollow Knight*, *Salt & Sanctuary*, and other games where combat is high-pressure. As such, we wanted to capture this stressful, difficult style of gameplay while setting our game apart from those listed by making it far more fast-paced. The player and enemies move and attack fast, and entities (including the player but excluding bosses) can sustain only a few hits before dying. This setup, paired with having to restart the level if you die, makes the combat challenging and stressful, but in a fun way. The intention is to create a high-pressure environment where the player truly feels the immensity of what they are up against; after all, it is no small endeavor to take on the entire angelic army on your own.

### **Interesting Offensive and Defensive Dynamics**

Though no one aspect of our combat mechanics is unique or new, our combination of them is relatively so. The player has the following offensive abilities: melee, ranged, and magic. The first is a short-range attack with claws, the second is a long-range missile, and the last is a medium-range area-of-effect magic attack (which will be absent in the initial demonstration). These attacks are all intrinsically related. Melee is the only infinite offensive method--there is no limit to how many times you can attack, though there is a small cooldown between swings. On the other hand, the player only has a small amount of ammo and must generate bullets for his ranged attack by landing melee hits. Also, the player charges up his magic attack with every enemy kill. This relatedness between attack methods is not seen in many games, nor is the availability of all three offensive tactics in a fast-paced environment. In terms of defense, the player is

able to dash and crouch. Both change his shape and hitbox, and the former gives him invincibility frames so he can effectively dodge.

### **Uncommon Visual Representation of Powers**

Our presentation of the combat mechanics is likewise original. Every offensive and defensive tactic relates to the powers bestowed upon the cultist by his Depraved masters. The powers he was given are based largely on shapeshifting, making him something of a polymorphing warlock. When he crouches he turns into a cat, and when he dashes into a wolf. His melee attack consists of his arm turning into a powerful pitch-black bear arm with long magical claws. The ranged missiles are shot by a summoned magical skull. Finally, the area-of-effect spell (which will be absent in the initial demonstration) will consist of the cultist commanding his skull companion to move in a circle around him, breathing fire in every direction. This presentation of powers as a variety of summons and polymorphisms is original and unique.

### **Unique and Original Setting and Quest, Unusual Position as an Anti-Hero**

Many games pursue an “end of the world” story, where the protagonist has to find a way to save everything. Our story takes the opposite approach: the world is ending, and the player’s quest is to ensure it ends in the “worse” way. Both the demons and angels seek to bring an end to mankind, the former through an Apocalypse that causes extinction, and the latter through a Deliverance that brings salvation. The player takes on the role of a cultist who wishes to help bring about the Apocalypse, which is an unusual position to take. Typically the player will be a protagonist seeking to bring about good--instead, the player is an anti-hero (or even an antagonist) who fights the forces of good with the intention of bringing about destruction. This role reversal is atypical and rare in games, and is a refreshing break from the standard.

## **The Game World**

## Overview

As described elsewhere, *Raising Hell* takes place in a mid-Apocalyptic world in which the demons and angels are warring over the fate of humanity and the world. The game is split into levels, each with its own theme relevant to the Depraved-Hallowed pairs.

## Levels

The game is split into five levels, with “level zero” being a simple tutorial. Each level, with the exception of the tutorial, is themed around a Depraved-Hallowed pair. The player proceeds through the level killing enemies until he reaches a boss room, where he will take on and challenge a Hallowed in order to free the corresponding Depraved. Level zero takes place in an underground temple where the cultist worships the Depraved. Level one is in rural farmlands and is tied to Famine and Bounty. The second level is based on Pestilence and Vim & Vigor and takes place in a city. Next is a stairway to Heaven polluted with demons and angels fighting one another, which is home to War and Peace. The final level occurs in the Kingdom of Heaven, which is related to Death and Life.

## Rooms

Each level is composed of a sequence of connected rooms. Rooms are self-contained rectangles in which waves of enemies spawn. The player is restricted to that room until he defeats all enemies, at which point he may progress into the next room. The number of rooms in a given level differs. Rooms vary in length and height and are populated with a random variety of impassable and passable platforms.

## Boss Rooms

The boss room at the end of each level will be constructed to be best tailored to the mechanics of the boss. Unlike normal enemies, the bosses will have a large number of unique movesets, making them far more challenging. We do not want to add artificial difficulty by having an inconvenient generation of the room. Further, many of the boss mechanics rely on the organization of the structures within a room, so we need to be sure these are always accounted for. The boss rooms will typically be simple relative to the rooms in the rest of the level in order to support the boss mechanics and to put more focus on the boss as opposed to the terrain.

## Procedural Generation

Within each level, the number of rooms, their shape, and their layout are procedurally generated. More specifically, first the number of rooms is randomly selected from a range between a preset minimum and maximum. Then, for each room, we randomly select its dimensions from a preset range. Finally, we randomly populate the room with passable platforms (through which the player can jump up and drop down) and impassable platforms (with which the player will collide) according to an intelligent ruleset that guarantees every platform is reachable. The only rooms that do not undergo procedural generation are the boss rooms, one of which appears at the end of each level (excluding the tutorial). The purpose of employing procedural generation is to add variance between playthroughs of each level, thereby augmenting the replayability of the game. It also makes the game slightly more challenging as the player cannot simply memorize room layouts as a means of making it through the level. On this note, boss rooms are not procedurally generated so as to not add artificial difficulty. Boss fights are designed to be significantly more difficult than any other fight in the game, so the unexpected or at times inconvenient arrangement of platforms may simply become an annoyance rather than a fun challenge. Instead, we

manually construct each boss room with an arrangement of platforms that is optimal for the style of combat presented by the boss.

Due to significant time constraints, we were unable to perfectly implement procedural generation. The logic for generation was completed with the exception of verifying reachability between platforms. In order to ensure a smooth demonstration, we procedurally generated rooms but then manually verified them. This means that the initial demonstration will only showcase a single set of room arrangements, which were procedurally generated and then corrected.

## The Physical World

### Overview

The physical world as it appears in *Raising Hell* is exceedingly simple. There are five unique locations, one for each level (including the tutorial). Each is populated with enemies and visuals appropriate to the setting. Travel between them is implied and automatic, taking place once the player completes a level.

### Key Locations

There are five locations that are visited as one plays through *Raising Hell*. The first is the starting/tutorial area, which takes the form of the cultist's temple. This is an underground stone shrine where the cultist worships and performs sacrifices for his Depraved masters. The second is the first actual level of the game and takes place in rural farmlands. Here, the player will encounter both traditional holy enemies such as angels, and animal enemies that have been recruited and empowered by the angels. At the end of the level he will face the Hallowed boss Bounty, the counter to the Depraved Famine. The third location and the next the player will visit is a city where the Hallowed bosses Vim and Vigor dwell. Before facing these twin bosses, the player will have to fight through angelic forces as well as humans that have been recruited. Following the city level, the player will arrive at the stairway level. This location is different from all the others: the player climbs vertically up toward Heaven in a continuous space instead of moving on the ground horizontally through a sequence of disjoint rooms. The level thus takes place in the sky, and the player must jump between platforms to evade the pursuit of the Hallowed boss Peace. Along the way, he must also fight angelic forces to assist the demons that are mounting an assault. Lastly the player will arrive at the Kingdom of Heaven. Here he will have to fight more righteous forces consisting of angels, saints, and other holy entities. At the end of this level he will face the final Hallowed boss, Life. The initial demonstration will showcase only the tutorial and farmland levels.

### Travel

The player moves the cultist around each level with simple left-right directional movement, in addition to jumping, falling, crouching, and dashing. Using these movement mechanics, he can navigate between the various platforms within a room as well as dodge enemy attacks. After a room has been rid of enemies, the player will be free to continue to the next room. Once all the rooms have been cleared, the player will arrive at the boss room. If he defeats the boss, he will be able to leave the boss room and be automatically transported to the next area/level.

As is apparent, travel is very simple. Because the progression of the game is so linear (room to room, level to level), travel is likewise linear. The player progressively moves right (or, in the case of the stairway level, up) within a level, and travel between levels is automatic and implied. Between rooms we play a quick black transition, and between levels we display a loading screen. After this loading screen, the player will simply walk into the first room of the next level.

### Scale

Scale is a difficult quantity to define in a game in the style of *Raising Hell*. It is a simple, cartoony, pixelated side scroller, so entities and objects are not always perfectly proportional. As a generic definition,

we try to preserve logical scaling; that is, the player is about as big as would be realistic relative to other entities on the map. As an example, there is a bull enemy in the Bounty level. We tried to maintain realistic scaling between the bull and the player. In that sense, everything in the game is scaled to the size of the player.

More technically, we define a standard unit within the game as 16ppu (pixels per unit). Every entity, object, and surface within the game is based on this unit measurement. For instance, one “block” of ground is one unit wide and one unit tall. The player is two units tall and half a unit wide. All entities are sized and their assets constructed according to this unit scale. The camera visibility is a rectangle that is 28.5 units wide and 16 units tall, and the camera will follow the player around the room if the room is larger (but if it hits an edge, it will stop moving).

## Objects

The gameplay style of *Raising Hell* takes the form of fast-paced movement and combat. Things like collectible items or frequent interactable objects do not make sense, and will actually feel quite out of place. There is no exploration to be done or environments to be manipulated in any way besides by clever movement tactics by the player. Thus, the game does not have many objects. The initial demonstration will not have any objects, but further development of the game may yield some breakable items and obstacles to make rooms more interesting and challenging.

## Rendering System

### Overview

We decided to use Unity as our core game engine. Accordingly we make use of Unity's Shading Language for any necessary modifications to the default rendering pipeline. Unity's Shading Language is a variant of the HLSL (High Level Shading Language) that is able to compile into shader programs for DirectX 9.0+, OpenGL 2.0+, and OpenGL ES 2.0+. As such we can support all of Windows, Mac, and Linux systems without needing to modify the pipeline. For a majority of what we need to render to the screen, the default Unity pipeline will be sufficient. For the few complex tasks, we will devise our own shaders or make use of Unity's particle systems.

### 2D Rendering

We will keep the pipeline for rendering our 2D sprites to the screen as simple as possible. To start, we define one unit on the screen as a square of 16x16 pixels; that is, our measurement relies on 16ppu (pixels per unit). The camera has a rectangular viewport of 30x16 units at all times. These characteristics cause the pixels of the visual assets to be large and noticeable, which is the intended artistic style choice. We will handle rendering the sprite assets to the screen through the use of Unity's Sprite Renderer. To keep the edges of the pixelated assets as clear, hard, and crisp as possible, we disable all image compression, anti-aliasing, and point filtering within the Sprite Renderer.

We will also add a few shaders to embellish the game. One is for the player sprite, which changes his eye color based on the attack he is using. When he is not making an attack, his eyes remain the normal yellow. A melee attack changes his eyes to purple, a ranged attack to green, and a magic attack (not implemented for the initial demonstration) to cyan. Another shader that will affect both the player and enemies has the effect of dissolving the entity upon death. Finally, we will have a shader that renders the room and level transitions: a black box the size of the screen will sweep right over the screen, cover the entire screen in black and remain for a moment, and then sweep back left to reveal the new room or level.

All rendering calls will be batched in order to reduce the amount of sparse buffer transfers from the CPU to the GPU, thus improving the performance of the game. The order in which entities are rendered to the screen is as follows: Background, Platforms (passable surfaces through which the player can jump and drop), Player, Enemies, Impassable Units (the tiles the player collides with and cannot jump through), and lastly Visual Effects (like particles).

## **Particle Systems**

We will also include a variety of particle effects to add some details to the game. To do so, we will use Unity's built-in Particle System, which allows the user to design the appearance and behavior of a particle effect. Intended particle effects include dust clouds left behind the player when he runs, dust clouds when he jumps and lands, an obscuring cloud of black smoke when he polymorphs, a black trail left behind as he dashes, and a trail on all projectiles.

## **Camera**

### **Overview**

Four out of five levels use the generic camera system which behaves as follows. The camera's viewport is 30x16 units. The positioning and movement of the camera adapts to the size of the room: if the room is at its minimum size (equal to the viewport of the camera), the camera will remain stationary with the entire room in its view. On the other hand, if the room is larger than the viewport of the camera the camera will follow the player, keeping him at its center. When the camera encounters the border of the room it will remain stationary; that is, it will not display anything outside the current room.

We chose to implement the camera in this style so as to keep the cultist the focus of the player's view as opposed to the surroundings. Given the gameplay is based on clearing rooms of enemies, this camera style will allow the player to see most of what is going on and adapt to it while never losing track of his avatar. We intend to make the rooms very populated and busy with a lot of enemies attacking at once, so it is important to ensure that the cultist is always easily and intuitively distinguishable from everything else that is going on.

### **Climbing Camera**

The outlier level in which the camera style changes is the third level, that which takes place on the stairway to Heaven and which will pit the player against the Hallowed boss Peace. Rather than having a horizontal sequence of rooms, this level is formatted as a continuous space of vertical and upper-diagonal movement. We chose to make this level unique for two reasons: to explain how the player travels from Earth (second level) to Heaven (fourth level), and to add some variety to the gameplay. To adapt to this different style of play, we have to slightly modify the camera. Because there is no room or confining space to restrict the camera to, the camera always follows the player while he climbs or falls, consistently keeping him at the center. Also, the camera will be slightly more zoomed-out than usual so as to allow the player to plan their route and dodging more effectively. Note that though this camera functionality is implemented, this level will not be present in the initial demonstration.

### **Parallaxing**

To make the game more visually appealing we will implement some parallax scrolling to the background of each level. Parallax scrolling is a technique where the background elements move at a different pace than the foreground elements, which has the effect of creating an illusion of depth and movement through a multidimensional world. The background of each level has between two and four layers, all of which scroll at different paces. Typically the frontmost layer of the background moves the fastest and the furthest back layer moves the slowest. Both horizontal and vertical parallax scrolling will be supported, with the latter being far more subtle than the former.

## **Game Engine**

### **Overview**

Unity serves as our game engine. The decision to use Unity was motivated by our desire to avoid writing low-level generic game engine functionality and implementations; we wanted instead to immediately begin writing game-specific code. On top of Unity we will design a custom level generation algorithm, a raycast-based collision system, an in-house input manager, and a modular logic component system that will be used to construct enemy behaviors. Though we will develop such schemes, many of Unity's built-in systems will also be utilized, including its Particle Systems and default rendering pipeline.

## **Level Generation**

A level is composed of a sequence of disjoint rooms, with the final room being home to a Hallowed boss. While boss rooms will be handmade in order to be consistent and compatible with the strategies of the boss, all the normal rooms of a level leading up to the boss will be procedurally generated. We have designed a novel algorithm to build each room, to populate it with passable and impassable platforms, and to ensure that all surfaces are reachable. The algorithm will also spawn enemies throughout the room. In designing this algorithm, we had as our goal creating rooms that are simple so as not to overwhelm or confuse the player, while also not being too empty. There needs to be enough platforms to allow the player to maneuver around the room and to dodge enemy attacks, but not so many as to make movement and dodging inconvenient.

After a room size is randomly selected and a few platforms are randomly placed within it according to some spacing and placement restrictions, there remains the issue of whether all platforms are reachable. To solve this, we designed and intended to implement a custom feasibility AI that stimulates the movement of the player to determine if any structures are unreachable. For anything unreachable by the player, a second pass of placement would be performed to add new platforms that make unreachable surfaces reachable. Over this entire generation process, we wanted to minimize the amount of structures added. We are confident that this method would have produced fair, balanced, and simple levels efficiently. However, due to time constraints we were unable to fully realize this feasibility algorithm. For the initial demonstration we instead procedurally generated a series of rooms and then manually corrected any reachability issues.

When a player enters a room, the enemies are then spawned in waves based on the generated room size and layout of structures. To improve efficiency, the overall level system is designed to only have the active room loaded and to utilize object pooling. This has the effect of avoiding constant memory allocation and deallocation during active gameplay.

## **Water**

There will not be water in this game. This is a simple two-dimensional side-scrolling game, so a feature like complex water graphics is not appropriate. The level designs are rather minimal, and the movement and combat is fast-paced; complex systems such as water effects are not in the style or spirit of the game, and would only serve to distract from the combat.

## **Collision Detection**

The core collision system we will be using is Box2D, which is the built-in 2D collision and physics engine in Unity. Box2D is a well-documented and well-made open-source library that performs general optimizations; for example, it uses quadtree collisions to reduce the amount of collision checks per game tick. On top of Box2D we will design our collisions to be based primarily on raycasts rather than using simple axis-aligned box colliders. This raycast design, though it consumes more resources to compute, more effectively facilitates our fast-paced gameplay. For high-speed objects, our raycast approach will be able to return an accurate and true collision detection whereas a simple box collider might miss it. This raycast approach also allows for the player movement to be much more fluid, making it feel quick and responsive. Simple box colliders and a more realistic physics model would have the effect of making movement feel slower and clunkier.

## **Input Manager**



After some investigation we decided that the built-in input manager for Unity was not sufficient for our requirements. The foremost issue was that the Unity input manager does not have responsive controller support, and we felt our gameplay would best be served on a controller. So, we decided to devise our own input system. Our input manager relies on Microsoft's XInput driver to get controller input. We wrote a simple wrapper library to interface the XInput calls. This library is written in C++, but exports the information in C so that the C# code can access it (Unity requires C#). The input manager is designed as an event-based system: you register an event with a button, and the event will be executed for each tick that the button is being actively pressed. For analog inputs, the listening code will be notified of the state of the input mechanism they are observing each tick; for instance, code that relies on the location of the mouse cursor will be notified of its position every tick. Our custom input manager allows us to achieve a much more responsive input system with a clear and organized input and key binding model.

## Enemy Behaviors

In order to streamline development and maximize code reusability, we devised a special modular system that dictates enemy behaviors. This system is based on components, each of which handles on type of behavior. Components can be added to enemies to give them a complex set of behaviors and moves. There will be a core Enemy Component that contains generic data and behaviors that are universal across all enemies. From there we will design and build new components that depend only on the Enemy Component and which dictate new behaviors. For instance, there will be a Dash Component. An enemy with the Dash Component will have the ability to dash--that is, to move toward the player in small sprints with some cooldown. Other examples include the Melee Attack Component and the Ranged Attack Component. We will combine these components to build unique enemies. In other words, each enemy is an amalgamation of components which together will simulate the desired behavior.

## Lighting Models

### Overview

We are using the simple unlit lighting model that is built-in in Unity. We decided against having complex lighting because we felt that it did not fit the style of the game. Any semblance of lighting or shading is included directly within the sprite assets, which gives the sprites the illusion of depth. Since the focus of the game is fast-paced gameplay that keeps the player on his toes, we believe that any dynamic lighting or darker rooms will be distracting and will slow down the pace of the game, thereby detracting from the overall experience. Also, because our art assets are so simple, complex lighting would be inappropriate and look out of place.

## The World Layout

### Overview

*Raising Hell* is a simple 2D side-scrolling game consisting of minimalist sequential levels. It does not take place in a large open-world environment that the player is meant to explore; rather, it plays more like an arcade game where the player is meant to get from the start of the game to the end as fast and easily as possible. Questions of world layout are thus limited to the arrangement of the levels.

## Linear Levels

The game is composed of five levels which the player travels to and completes in sequence. The first is the tutorial level (level zero) that takes place in the cultist's temple to the Depraved. After this level is completed, the player will be transported to the true first level, which occurs in rural farmlands. This area contains many holy enemies such as angels, but also features animal enemies that have been recruited and

empowered by the Hallowed. The Hallowed Bounty is the boss of this level. Next is the city level which is home to the Hallowed bosses Vim and Vigor. Here the player will face more holy soldiers, in addition to humans that have been recruited by the angels including paladins and clerics. The third level takes place on a stairway to Heaven, which the player must ascend as he evades and battles the Hallowed Peace. Throughout this level the player will encounter more holy soldiers, in addition to demons he can assist in battle. Finally, the last level occurs in the Kingdom of Heaven. The player will fight angels, saints, and other holy figures in this level before facing the final Hallowed boss Life. Only the tutorial level and the farmland level will be available in the initial demonstration; further levels will be under development following the demonstration.

Each level is linear in composition: they consist of a sequence of rooms. The player must clear a room entirely of enemies before being permitted to proceed to the next room. The last room is the boss room, which is specially crafted to suit the boss's attack patterns. The exception to this rule is the third level, the stairway to heaven. Though this will not be prepared for the demonstration, it is planned to consist of continuous vertical movement up a series of platforms rather than of a sequence of disjoint rooms.

## Rooms

Most of the levels in *Raising Hell* are composed of rooms. With the exception of boss rooms, rooms will be randomly generated according to a ruleset that controls structure placement. Rooms can vary in size, but will always be rectangular. Populating the rooms will be a random arrangement of passable and impassable platforms, which the player can jump between in order to reach enemies and dodge attacks. Passable platforms are structures which the player can jump up or drop down through. In contrast, impassable platforms are collidable structures which the player can stand on but not pass through. Enemies will be randomly spawned throughout the room (including on both the ground and on platforms) according to a simple rule set controlling density and distribution. On either side of the room is a blocker, which prevents the player from moving to the previous or next rooms. These disappear only when all enemies have been defeated, at which point the player may progress.

# Game Characters

## Overview

The player takes on the role of a cultist who serves the demon lords known as the Depraved. After his masters are captured by their angelic Hallowed counterparts, the cultist takes it upon himself to find and rescue them by defeating the Hallowed. To do so he must fight his way through hordes of holy enemies.

## Creating a Character

This is a simple role-playing game with a pre-written, linear story and quest. The identity of the player's character is integral to this story and as such the character is preset and immutable. There is no character selection or customization; the cultist is the only option. For now everything about the character is preset, but in the future we may allow minor modifications such as changing the color of the robe.

## The Main Character: The Cultist

The cultist is the main character of *Raising Hell*. He is a nameless mysterious figure shrouded in a deep red robe. The cultist spends most of his time worshipping the demon lords known as the Depraved in his underground temple to them. Like the demons he aligns himself with, the cultist wishes to witness the actualization of the Apocalypse and accordingly the end of mankind. Though it is never explicitly stated, the cultist's motivation for his service is his pervasive disgust with mankind; more specifically, he is repulsed by the virtue and goodness that humans boast, when in reality they are depraved, selfish, and evil. He thinks that humans need to face this and accept their true nature as evil, demonic creatures. The cultist also has faith that his loyal service to the Depraved will ensure him a position as a high-ranking demon upon his mortal death. He wants above all to remain by the side of the Depraved and to assist them in their imminent rule over demons and men.

At the start of the game, the cultist's servitude is recognized and rewarded when the Depraved arrive and imbue him with demonic powers. Their motivation for doing so is twofold: to show their trust and faith in him and his loyalty, and to empower him to assist the demons in the war and to help free the Depraved from their imminent capture by the Hallowed. Upon receiving his powers and watching the Depraved be imprisoned, the cultist understands his responsibilities and embarks on a quest to liberate his masters so they may enact the Apocalypse.

## Allies: The Depraved & Demons

The Depraved are the four lords that rule over all demons. They are the embodiment of evil and darkness, and their goal is to defeat their holy Hallowed counterparts so as to win the ongoing war between angels and demons. In doing so, they would be free to actualize the Apocalypse which entails the end of mankind. Some humans would be killed, others enslaved, and the most base and evil would be promoted to demons.

The Depraved mirror the biblical Four Horsemen of the Apocalypse. The first is Famine, who causes chaos and suffering through the destruction of supplies, livestock, and food. The second is Pestilence, who spreads disease, panic, and even misinformation among the masses. Third is War, who provokes violence and conflict among men. The last and most powerful of the Depraved is Death, whose power and responsibilities are embodied in his name.

Famine, Pestilence, War, and Death command vast armies of demons, whom they direct to fight the angels. It follows that all demons in addition to the Depraved are allies of the cultist. Throughout the game, most noticeably in the third level on the stairway to Heaven, the cultist may encounter demons that will assist him and that he can assist.

## Enemies: The Hallowed & The Holy

In contrast to the Depraved, the Hallowed are the five angel lords that command all heavenly forces. Each of the Hallowed is a counter to one of the Depraved. The first, who opposes Famine, is Bounty. She is responsible for blessing mankind with surplus in all domains. The counter to Pestilence is a pair of Hallowed twins known as Vim and Vigor. While Pestilence is the embodiment of disease and weakness, Vim and Vigor represent health and strength. Vim corresponds to health of the mind and spirit and Vigor is the materialization of physical strength and fitness. The Hallowed counterpart to War is Peace, who encourages compromise and cooperation in place of violence. Finally, the most powerful Hallowed is Life, who opposes Death. She controls the creation of all life and the survival and prosperity of mankind. The goal of the Hallowed is to defeat the Depraved and the demons and to bring about the Deliverance, or the salvation of mankind.

Each of the Hallowed serves as a boss in *Raising Hell*, and each has a unique set of moves and powers. The first level's boss is Bounty. She has powers such as summoning sheep and bees to attack the player, burrowing through the ground to quickly reach the player, and making a melee attack with a farmer's scythe. Bounty is the only boss who appears in the initial demonstration, as only the tutorial level and first level will be available to play. The second level is home to Vim and Vigor, whom the player fights simultaneously. Vim is able to cast a variety of spells and can empower his twin. Vigor performs a number of physical attacks, with some attacks being incredibly fast but dealing little damage and others being

slower but heavy-hitting. Peace occupies the third level. The fight with Peace is very different from the others; she chases the cultist up a stairway to Heaven, performing a variety of moves and dashes along the way. If she is able to touch him, she will purify him and effectively kill him. The player must evade her while also landing attacks to defeat her. The final level takes place in the Kingdom of Heaven and is where Life resides. During the fight with Life, she will heal herself in addition to giving life to large numbers of angels, animals, humans, and plants that will attack the cultist.

Outside of boss fights, the cultist will face the holy forces that the Hallowed command. Levels will consist of a series of rooms populated by such enemies which the player must defeat to progress. Examples of enemies include empowered animals such as sheep, recruited humans such as paladins and clerics, angels, generic saints, and more. Each enemy has a far simpler moveset relative to bosses, having only one or two attacks, and can take a small amount of damage from the player before perishing.

# User Interface

## Overview

There are several aspects of the user interface, which are described in the sections below.

## Health, Ammo, and Magic

During the game, located at the top left of the screen there will be several indicators that describe the status of the player. First, there is a set of upside-down black heart icons. These inform the player how much health he has, with each heart indicating one hit point. For every hit the player takes, at least one hit point will be lost. If the player loses all hit points, he dies. Next, there is a set of up to five green and black ammo symbols. This tells the player how many ranged missiles he has left. If he has no charges, there will be no ammo symbols but instead red crosses. Finally, there will be a small bar that displays how much the player's magic attack has charged. When the bar is full, he is able to cast his magic attack and doing so will deplete the whole bar, which will then progressively fill back up as the player kills enemies. This last indicator will not be present in the initial demonstration as the magic attack will not be included.

## Transitions

The game will play quick transitions between rooms and between levels. When the player exits a room to proceed to the next, a black bar will sweep right across the screen to obscure everything. The screen will remain entirely black for half a second, and then it will sweep back left to reveal the new room. In the case of transitioning between levels, the duration for which the entire screen remains black will be slightly longer to provide time for the procedural generation algorithms to complete.

## Menus

There will be two menus present in the game, both very minimal. The first will be the start menu which will appear when the game is launched. This menu will feature only a "Star Game" button and visually will have a simple sprite of the cultist. There will also be music playing while the start menu is open. The second menu is a pause screen which appears when the player pauses the game. Pressing the pause button will freeze the current state of the game and overlay it with a dim gray shade. At the top left of the screen, the typical symbol for "pause" will flash. Once the player resumes the game, the overlay and pause symbol will disappear and the scene will unfreeze.

## Introduction Sequence

When the player starts the game, he will experience a brief introduction sequence that vaguely explains the story of the game. There will be five illustrations that play in sequence. The first will show the cultist praying at his altar to the Depraved. The next illustration will show the Depraved arriving. The third image will display the Hallowed breaking in to capture the Depraved. Following this, the player will see the Depraved being taken prisoner, but infusing the cultist with power at the last moment. Finally, the last illustration will show the cultist accepting his new quest. Each of these illustrations will be visible for about five seconds, and are meant to provide a brief understanding of what is happening. More of the story will be revealed as the player progresses.

# Weapons

## Overview

The player has access to three types of attacks in *Raising Hell*: melee, ranged, and magic. Each has its own rules and mechanics, and each is sourced in some type of polymorphism or summoning, which are magics bestowed upon the cultist by the Depraved. When the player makes each kind of attack, his eyes will temporarily change a different color: purple for melee, green for ranged, and cyan for magic.

## Melee: Bear Claws

The most frequent type of attack made by the player will be the melee attack. To perform a melee attack, one of the player's arm polymorphs into a black bear arm and grows large magical purple claws. Which arm is used alternates between attacks. The melee attack has a hitbox that is 1.75 units wide, beginning from the closest edge of the player hitbox, and is 1 unit tall. If an enemy collides with the melee hitbox, it will take damage. There is a small cooldown between melee attacks (approximately 0.2 seconds) so that the player cannot excessively spam the attack.

## Ranged: Skull Companion Fireballs

Another method of offense available to the player is firing a ranged missile. The missiles are expelled by a summoned skull companion that rotates around the player in a circle. It is positioned 2 units away from the center of the player. The player can control where the skull is facing and aiming with either the mouse cursor or a controller thumbstick. The cultist can thus command the skull to fire a missile in a specific direction. The missile will travel in a straight line in the given direction until it either collides with an enemy or surface, or until it reaches the end of its lifespan.

The player has a maximum of five ranged charges at any time. Once these are expended, he is unable to fire any more ranged missiles. Missiles are recharged by hitting enemies with the melee attack: every two melee hits will regenerate one ranged missile. The lore reasoning behind this is that the missiles are actually a product of blood magic, meaning that missiles are generated only through the collection of blood.

## Magic: Skull Companion Area-of-Effect Fire

The last weapon is an spell. This will not be present in the initial demonstration, but is planned to be implemented in later development. This skill is a powerful area-of-effect spell in which the skull companion travels in a full circle around the player, breathing a stream of cyan fire as it moves. The fire will damage any entity that comes into contact with it. Because this attack is so powerful and effective, it requires seven enemy kills to recharge, and the charge is fully expended with one cast. It cannot be partially casted.

# Musical Scores and Sound Effects

## Overview

*Raising Hell* is a simple game with limited effects, visuals, and actions. As such, audio effects are not of great concern; there are only a few actions that could reasonably have a sound attached. Of course, music is a necessity. The intention is to have one song playing per level, with a tune that captures the theme of the level.

## Audio Toolkit

We will be making use of the built-in Unity Audio Player and Audio Mixer. This supports a variety of compression formats and encodings, but we will primarily be using the standard MPEG codec. We do not need disc compression such as Redbook Audio as we do not plan on doing a disc release for any platform, console or PC.

## Sound Design

For our initial setup, we intend to use simple royalty-free music as soundtracks for our levels. Each level will have its own soundtrack, with the hope that the music captures the atmosphere of the level or of the gameplay. We will also have audio effects for actions such as making an attack, taking damage, and dying. These will also be derived from royalty-free sources. In the future, we hope to commission or compose our own music and sound effects--but royalty-free choices are acceptable for temporary use.

# Single-Player Game

## Overview

This game is a single-player role-playing game in which the player acts as a cultist who seeks to aid the demons in bringing about the Apocalypse. He travels through the land, fighting holy forces while on a quest to unleash the Depraved demon lords who will lead the end of days. The levels consist of interconnected rooms filled with enemies, where platforming is a key focus. The combat is of a very fast-paced hack-and-slash style, where death results in the player having to restart the level.

## Linear Progression

Both the game and the story progress in a linear fashion. There are five levels that proceed linearly from start to finish: the player moves from room to room through the level, defeating holy forces along the way in order to do so, and at the end encounters a boss room containing one of the Hallowed. After one Hallowed/level is beaten, the player moves onto the next level and repeats this process. Levels are ordered by increasing “severity” of the Depraved-Hallowed pair, with Famine-Bounty being the least severe and Death-Life being the most. Thus, the actual movement in the game progresses linearly from the start of the level to the end, and the story progresses linearly in that the player takes on bosses of increasing severity one at a time until he reaches the end of his quest.

## Platforming

Structures within a room are of one of two types: passable platforms or impassable chunks. Much like any platformer, the player will jump around, between, and through these structures as a means of travelling, dodging, and fighting. He is able to jump and drop through passable platforms, but cannot pass through impassable chunks or the ground.

## Hack-and-Slash Combat

Combat in this game takes the form of fast-paced hack-and-slash fighting. The player has three means of attacking: melee, ranged, and magic. When he makes a melee attack, his arm polymorphs into a black bear arm with large magical claws. He can attack left, right, and up. If he is airborne, he can also attack downwards, and if he strikes an enemy he will bounce. Ranged missiles can be fired from the cultist's skull companion, which will rotate around the cultist in the direction the player is aiming. This companion can launch fireballs at enemies. More ammo for fireballs is obtained by hitting enemies with melee attacks. Finally, the player can perform an area-of-effect magic attack in which the skull companion does a full rotation around the player, spewing fire as he does so. Magic attacks are charged up by killing enemies. Note that this last magical skill will not be present in the initial demonstration, but is planned for the final rendition of the game.

## Roguelike

This game is a roguelike, meaning that if the player dies he will suffer consequences. Rather than forcing the player to restart the entire game, we chose to have the player restart the current level upon death.



## Story

*Raising Hell* takes place in a modern mid-Apocalyptic era. The lore of the world describes an ongoing war between the demons and the angels over the fate of humanity. The former, led by four lords known as the Depraved, seeks to bring about the Apocalypse, or the extinction of humanity. On the other hand, the angels, led by five lords known as the Hallowed, desire to enact the Deliverance, or the salvation of mankind. Each of the four Depraved has a Hallowed counterpart: Famine vs Bounty, Pestilence vs Vim and Vigor, War vs Peace, and Death vs Life.

The player takes the role of a cultist who worships and serves the Depraved. Like his masters, he desires above all to see the actualization of the Apocalypse. Why this is so is unclear to the player, but in truth the cultist has faith that he will become a high-ranking, powerful demon upon the coming of the Apocalypse and his own mortal death. With this position he would be able to remain close to and better serve the Depraved.

One day as the cultist is performing his standard rituals in his temple to the Depraved, he is visited by the demon lords themselves. They share that their stronghold has been infiltrated by the angel forces, and they were forced to retreat--though the Hallowed are following close behind. The Depraved instruct the cultist that he is being trusted with magical demonic powers and must help bring about the Apocalypse. Just after they imbue him with power, the Hallowed arrive and capture the Depraved. They carry the demon lords away and imprison them. The cultist takes it upon himself to find and free the Depraved, defeating as many holy enemies as he can along the way, so that they may continue their mission to enact the Apocalypse and wipe out mankind.

The cultist begins by exiting his temple, which is now infested with holy forces. He must fight his way through to escape. After the temple level (the tutorial level), he will proceed to travel to each location where a Hallowed holds a Depraved, fighting through hordes of holy entities in order to reach the Hallowed. First, he visits rural farmlands to find Bounty, who holds imprisoned Famine. Then, he travels to a city to find Vim and Vigor and to free Pestilence. Next he encounters a stairway to Heaven, throughout which demons and angels are fighting. There he encounters Peace who holds War. Finally, in the Kingdom of Heaven, the cultist finds Life who guards the most powerful Depraved, Death. Each time the cultist defeats a Hallowed, the corresponding Depraved is freed and is able to return to leading the demons in the war. Upon the fall of Life and the liberation of Death, the holy forces are thoroughly weakened and without leadership, and the demons are finally able to crush their foes and enact the Apocalypse. What happens to the cultist at this point is not revealed to the player--all that is known is that his quest was completed. The purpose of hiding this is to allow for deliberation and conjecture among players.

Note that the initial demonstration of *Raising Hell* will only briefly explore this story. Only the first two levels--the tutorial and Bounty levels--are planned to be fully developed. Thus, the player will only be exposed to the fairly vague introduction sequence, the tutorial, and the first Hallowed. The full story will not be available in the demonstration.

## Hours of Gameplay

*Raising Hell* has only four levels plus a tutorial, and if the player performs perfectly each level will last only about ten to fifteen minutes. This would make a flawless run take about 75 minutes. However, this game is also a roguelike, meaning that every time the player dies he will have to restart the level. We also expect a very high frequency of death given the pace and difficulty of the game. As such, a realistic playthrough of the game would take eight to twelve hours. In terms of hours of gameplay, *Raising Hell* is similar to *The Binding of Isaac*: the levels aren't terribly long, but the difficulty of the game is in surviving as any death will cause you to restart.

## Victory Conditions

The player wins the game if he is able to reach and defeat the final boss, Life. This is a simple victory condition, but the journey to attain it is incredibly difficult. The player must defeat the tutorial level, the Bounty level and boss, the Vim and Vigor level and bosses, the Peace level and boss, and the Life level

to even reach the Life boss--and he must achieve victory against this challenging boss. Upon beating Life, the player will have successfully defeated all of the Hallowed and freed all of the Depraved. The demons are now able to win the war against the angels and unleash the Apocalypse, thus fulfilling the cultist's goal.

Note that this victory condition will not be present in the initial demonstration as only the tutorial and Bounty levels will be available to play.

# Multiplayer Game

## Overview

We do not intend to provide support for multiplayer as it is not particularly well-suited to the story. However, depending on how well-received the game is and how much players request it, we will consider adding downloadable content that allows people to play together. This would probably take the form of either an arcade-style endless level, or something more like *Diablo 3* where the players can run instances of procedurally generated levels of increasing difficulty.

# Character Rendering

## Overview

As stated elsewhere, everything in the game, including the player and all enemies, is rendered with the Unity built-in Sprite Renderer. Per our design specification all sprites conform to the 16 ppu (pixels per unit) standard. This ensures they will all have the same crisp visual quality. Further, on top of the existing Unity rendering pipeline we will devise and integrate two shaders for special effects.

## Rendering State Machine

The player, enemies, and bosses each have a variety of animations. To select which sprite animation to display we will develop a rendering system based on a state machine. The main idea behind this system is that sprites are linked to states that describe actions. We then define a strict rule set of when and how the state and thus the animations transition. For instance, the player has an “idle” state, a “locomotion” state, and an “airborne” state. When the first is active his idle animation plays, when the second is active his run animation plays, and when the third is active his jump animation plays. If the player is standing still and then presses the jump button, the state transitions from “idle” to “airborne” and the animation will thus switch from idle to jumping. With a strict enough rule set, this allows for fluid transitions between states and between animations. One example of a restriction is that the player cannot transition directly from being airborne to crouching; he must first land and enter either the idle or locomotive states before being able to transition to the crouch state. This state system extends to all enemies and bosses as well.

## Flash-on-Hit Shader

To make it obvious when an entity has taken damage, we will implement a shader that affects the entity when it receives a hit. This shader affects both the player and all enemies. It causes a recently-damaged entity to flash a specific color for a set duration. By default this color will be set to red. This generic shader will be integrated directly into the rendering pipeline for the sake of efficiency.

## Eye Color Shader

We will also use a shader to add an extra detail to the player. His eye color is yellow by default, but we want it to change based on the type of attack he is using. When he makes a melee attack his eyes will flash purple (the same color as the claws in the melee animation). A ranged attack is associated with green eyes, and a magic attack--though not present for the initial demonstration--would appear alongside cyan eyes. Rather than creating copies of the player sprite with the desired eye colors, we will instead design an efficient shader that dynamically changes the eye color during runtime. This shader will be built off the flash-on-hit-shader, so that the player will still flash if he is hit while he is also attacking.

# World Editing

## Overview

We will not provide support for a world editor. Our levels will be procedurally and randomly generated, and enemy spawns likewise so. Depending on the reception of the game and the demand for it, we may in the future consider offering downloadable content that includes a world and enemy editor.

# Future Features

## Overview

There are many small optimizations, features and details we want to include but will not have time to. These will not really affect the quality of the game; we were careful to implement everything vital to the game before working on minor embellishments. This section describes such additions.

## General Features

- Items - healing and buffs
- Weapon and ability improvements
- New abilities
- Weapon options and diversity

## Asset Improvements

- Improve the shading and appearance of the player idle and run animations
- Add dragon wings to the player jump animation
- Diversify and animate the particle entities
- Add an actual explosion for ranged missile on-hit animations
- Create idle animations for entities stuck on a platform that cannot reach the player, or stuck behind a wall

## Backend Improvements

- Complete procedural generation
- Move all the sprites to a single sheet to reduce GPU calls
- General code revision and improvement