Matthew Bradon

Design Document for:

A Dash Away From Death

Randomly Generated Rage Platformer

Written by Matthew Bradon

Version # 1.00

Friday, March 3rd, 2023

Table of Contents

Design History

This document describes the history of the development of A Dash Away From Death and the design choices I made in developing the game. The aim of this document is that you, the reader, understand what and why I made the design choices I made.

Version 1.0.0
An overview of the vision for the game.

Game Overview

Philosophy

Random Generated Maps

This game from the outset was trying to achieve a randomly generated map of premade rooms that a player character would traverse while an enemy would chase. This came into the form of a dungeon crawler which the player is trying to escape death.

Object Oriented Programming

When designing the systems for this game, object-oriented programming concepts such as abstraction and encapsulation came to the forefront. The State Manager system was built with polymorphism in mind.

Common Questions

What is the game?

This game is a procedurally generated platform game where the player is being chased by the reaper who mimics the players movements exactly. The player must traverse the ever-expanding maze with spikes around every corner. The player must do this all while dodging the reaper.

Why create this game?
I created this game as I wanted to create an interesting spin on the endless runner genre by having it be a random dungeon crawler platformer while keeping the integral element of the endless running.

Where does the game take place?

The game takes place in a dark blue dungeon where death looms in every corner. The opening spells a tale of a rumored dungeon with many riches but is said to be plagued with death. You the player are to explore this dungeon in hopes of fame and riches

What do I control?

You control a dark knight with a golden cape that is well suited to traverse the dungeon with precision and skill.

What is the main focus?

The goal of the game is to survive as long as possible against the reaper and the dungeon's challenging platforming. The player also traverses deeper and deeper into the dungeon with an increasing count of how many levels they have explored.

What is different?

This game is not your standard endless runner, it is a fusion of rogue-like dungeon platformers with the endless running genre. On top of being a hybrid it is also a rage-game which the player must narrowly avoid being blindsided by the deadly dungeon.

Feature Set

General Features

2D Pixel Art Graphics
Diverse movements
Keyboard and Controller support

Gameplay

Challenging platforming
Ever changing levels which the player must traverse
Maneuvering around the reaper that chases and pressure you

The Game World

Overview

The world of this game is the dungeon which is both challenging and rewarding to traverse your way through. There is a path from the start room to the end room and it's the players job to find that path all without dying/

Level Generation

This is the essence of the game, the randomly generated levels of premade rooms. As the player plays the game, they learn the ins and outs of each room but each playthrough feels fresh as it is put together differently each time. It strikes a balance between familiarity and new experience.

Spikes

The levels of the dungeon are designed to be deadly so one must balance caution as well as speed. The spikes facilitate the need for skill. They are a hurdle to get over by learning how to dodge them fastly, efficiently and to feel cool when doing so.

The Physical World

Overview

The dungeon has a critical path that connects the start room to the end room. Along with the critical path random rooms are placed around the map that can lead to dead ends which can prove to be fatal when backtracking.

Key Locations

The 2 key locations are the beginning of the level where the player spawns and the bottom of the level where there is another door to lead to the next level. The entire goal of the game is to traverse until you reach the end door

Travel

The player traverses the dungeon by running, jumping, climbing, and dashing through the deadly spikes and hard platforms to reach.

Objects

Spikes and Doors are the two objects in the game. The door is the goal to reach, and the spikes are the obstacle between.

Time

Time will be used as a measure of the players' skill. The game will have a timer of how long the player has survived, the longer the better.

Rendering System

Overview

Rendering will all be handled by Godot's GLES2. This is Godots built in renderer which handles all the rendering.

2D/3D Rendering

GLES2 is Godot's OpenGL-based rendering backend

Camera

Overview

The camera will be a simple camera that is centered with the player and moves with it. The camera will be slightly zoomed out to show the players peripherals.

Game Engine

Overview

Godot is an open-source game engine that can handle both 2d/3d games. It is rising in popularity over its counterparts such as the robust Unity game engine. While not as powerful as Unity or Unreal engine it is quite small, only being around 100mb. It offers a wide plethora of languages to use such as C#, C++ and its own GDScript.

Godot Scene Tree

Godot has an intuitive node system where each node can have children or parents that all exist on the large root tree. This makes it extremely easy to access instances of objects and interact with them.

Nodes

Godot has many nodes which handle just as many things. It has nodes for both 2d and 3d and neither. There are nodes for audio, sprites, raycasts, lighting etc.

Signals

Signals are a feature of Godot that allow you to signal other objects to run code after a certain event has occurred. These are useful for all aspects of game design. I.e. signaling a button has been pressed to change scene

Collision Detection

Godot handles its collision through the CollisionShape2D node. This node will have a shape to choose from such as a capsule, rectangle, or a custom polygon. Each CollisionShape has a Collision Mask which means it will look out on that mask for other CollisionShapes to collide with.

Game Characters

Player

The player will be a dark knight character that will jump across the screen. The player will have no attacks but will be able to move across the dungeon is a satisfying way.

Enemies

The one and only enemy is the reaper which will mimic the players movements by an offset of x amount I.e. It will 2 seconds behind the player. There will be no way to “kill” the reaper, but you will be able to defeat it by smartly out maneuvering the reaper by using its path against it

User Interface

Overview

The user interface consists of a main menu, options menu, pause menu and game over screen. Each of these will be interactive you will be able to traverse to each menu. When in the main scene you will see a timer counting up showing how long you have lasted and a level counter showing the current level.

Main Menu
The main menu will hold the ability to start the game and traverse to the options menu. This menu will contain a nice looking graphic of the dungeon setting the scene for the game.

Options Menu
The options menu will contain the information regarding the controls for keyboard and controller. When in the menu you will have the option to return to the main menu.

Pause Menu
This menu pauses the game freezing both the player and the reaper. This gives the user the ability to leave without the risk of dying. When in the pause menu the user will be greeted with two options to resume the game or quit the game entirely.

Game Over Menu
It will display the time the player has lasted in the game up until the point of death and will present the number of levels the user was able to complete. It will give the options of either returning to the main menu or beginning another game.

Main Interface
This will include the survival timer which will be counting up from when the player first starts the game and a level count for every level the player has completed

Movement

Overview

The movement system of a dash away from death inlcudes 5 main movements (walking, jumping, dashing, climbing and wall jumping). Each of these moves come together to form a diverse moveset which the player can use to navigate the dungeon in separate ways.

Jumping
The jump is broken down into 3 steps the time it takes to reach the peak the peak and descent. Based on these timings the jump velocity and gravity is calculated. This is to create a nicer feeling jump with a large degree of control. The jump also has a feature called coyote time which allows for a jump input when the player momentarily falls off the floor. This is to create a more responsive jump.

Dashing
Dashing gives the player an instant boost to speed in a direction they are choosing. The player can choose from any direction except down. However, the play only has one dash per airborne moment. Managing the dash counter is an element of gameplay the player must face.

Climbing
The player can climb against vertical walls going either up or down or stationary. The climb is limited by how long the limit is set meaning the player cannot indefinitely stay stuck to the wall. This feature adds urgency to the players actions.

Wall Jumping
When the player is against the wall and jumps they perform a wall jump. This is a special jump which has both horizontal and vertical velocity aware from the wall they are facing.

Musical Scores and Sound Effects

Overview

The music will be designed with the central premise of a mysterious blue dungeon. The sound effects will be created using my voice with several effects.

Music Design

The score will be made using a program called bosca ceoil. The song will be a synth song designed around the fact it will be looped constantly. The idea of the song is to be a mysterious and calm song so it fits the dungeon atmosphere and does not becoming grating

Sound APIs

Godot has built in audio handlers called AudioStreamPlayers they can play a single sound of your choosing with several options such as looping. This will be used for both sound effects and music

Sound Design

Sound effects will be created using the mouth and voice along with audacity to change speed and pitch. Audacity will also be used to layer mutiple sounds to create the desired sound effect. The sound effects to be created are jump sounds, fall sound and a death sound.

Single-Player Game

Overview
The game involves running from the start door to the end door of the dungeon while dodging the reaper and traps within the dungeon.

Rage Game
The levels are designed to kill the play quickly. There can be little window to avoid certain doom. I want this game to feel fair when it comes to the player controls, but the levels feel unfair with the small window of time you have before certain doom. The player is motivated to beat said levels as the payoff of making a jump that you were stuck on feels great

World Editing

Overview

There is no world editing tool in the game, however in Godot engine there is the tilemap which makes up all the rooms that make up the map.

Room Details
There are 7 types of rooms (left-right, left-right-top, left-right-bottom, left-right-top-bottom, fill, end, start). You can have any number of these and each one is decided by random integer generators.

Map Size Control
You can control the height and width of the map inside the class responsible for map generation

"Programming Appendix"
List of programming concepts used in this document.
Object-Oriented Programming: Programming based on the concepts of objects which contain data and code

Encapsulation: The bundling of data and code into single entities.

Abstraction: Concept of hiding the internal implementation of an object only interacting with that objects interfaces.

Polymorphism: Is the ability for an object to take on many forms I.e a state machine being for both players and enemies.

GLES2: A graphics renderer

“Character Appendix”

Dark Knight: The player that the user controls.
Reaper: The main enemy that chases and kills the player on contact.

“Objects Appendix”

List of objects mentioned in this document.
Spikes: Will instantly kill you if you touch them.
Door: The start and end point of a level and the goal.

“User Interface Appendix”

List of UI elements in this document.

Surival Timer:	The timer that counts how long that player has been playing for.
Level Count:	The count of how many levels the player has beating.
Options Menu:	Displays the controls.
Main Menu:	Starts the game.
Pause Menu:	Pauses the game.