Chamber's Labyrinth

Specification

Version 0.1.1

VERSION	DATE	TITLE
0.1.1	1/17/2015	General Game Concept

Table of Contents

PAGE	CONTENT
1	Introduction
2.1	Game Summary
2.2	Developer Tools
3	Technical Overview

1 - Introduction

We're here to make a game we will enjoy playing. Sure we'll gain skills along the way, but in the end, gamers just want to game. Our goal is to get a MVP, minimum viable product, done by the end of the quarter. Every **main feature** should be coded, whether it looks the best, or is optimized, doesn't matter as much.

We don't want to program any "accessory" features along the way, just the real deal. An example of an accessory feature would be being able to use different weapons, unless it impacts game progression, or our game's concept is focused on the many different weapons it has-which it's not currently, then we can leave that for later.

To get this done we'll all have to be held accountable and stay focused, weakness is not an option. Just like how our game is going to be hardcore, so will our teamwork. Get ready for a quarter to remember, because we're going to start and end it strong, with an awesome game to match our efforts. We are the pride of DG, lets show them how it's done. Now lets cover the essentials to get this game started.

2.1 - Game Summary

We are making a 2D Roguelike Side-Scroller RPG written entirely in C++. There will be a central character that can move left to right and jump. They will be restricted by the box (environment) they are in and will be able to stand on platforms.

Rooms and their features will be generated by progression through the game. Progression will occur when the central character reaches a generated (after a certain point, within a specified scope) checkpoint.

Environmental theme will associate light with good, darkness will bad. More accessories will be added to the central character, including his sprite, to support the environmental theme.

2.2 - Developer Tools

Git - The technology we will be using to send and retrieve code for our game through the website github. Here is a tutorial.

GitHub - A website used to store our game online, track its technical progress, collaborate online, and keep development on track. Sign up if you aren't already at https://github.com/. If you're signed up, visit our repository.

WhatsApp - A mobile app used as primary group communication.

Google Drive - A cloud service we will be using to share notes and informal documents. Here is our link, request access from an officer if you are not authorized to access the drive.

SFML - SFML is multi-platform and multi-language. It provides a simple interface to the various components of your PC, to ease the development of games and multimedia applications. It is composed of five modules: system, window, graphics, audio and network.

3 - Technical Overview

1. System
    1.1. Resolution
        1.1.1. 1280 x 720 (Unless someone has a lower resolution)
    1.2. Viewport
        1.2.1. Focused on main sprite
2. Level Design
    2.1. Size
        2.1.1. Rooms will vary in size
    2.2. Shape
        2.2.1. Rectangular
    2.3. Artwork
        2.3.1. Background image
3. Player Sprite
    3.1. Spritesheet
        3.1.1. Animations for main sprite
            3.1.1.1. Move right & left
            3.1.1.2. Jump & fall
            3.1.1.3. Stand right & left
4. Platform Sprite
    4.1. Artwork
        4.1.1. Simple platform
5. Engine
    5.1. Gravity
        5.1.1. All units have a downward vector force acting upon them
    5.2. Collision
        5.2.1. Collision detection
        5.2.2. Collision response
    5.3. Movement
        5.3.1. Read Input from keyboard
            5.3.1.1. Arrows
                5.3.1.1.1. Left right to move
            5.3.1.2. Space
                5.3.1.2.1. Jump