unity

Game Design Document

Rapid Gameplay Prototyping

Introduction

Here is the email that Steve, the designer, sent to kick-off the development of the Infinite Platformer.

This is less detailed and formal than a Game Design Document (GDD) might typically turn out to be, but at the rapid prototyping stage this may be all the detail you would have to get started on a project.

Gameplay Overview

The **Infinite Platformer** will focus on navigating platforms in a procedurally-generated 2D world. The **main challenge to the player** is jumping from platform to platform, with the jumps becoming increasingly more difficult as the player progresses.

A **secondary challenge will be dodging and using enemy NPCs to your advantage**. The NPCs will look very similar, if not identical, but will have different colors the player must use to figure out how to avoid or use.

The player can jump on enemies to kill them, but more importantly jump off of them to reach taller or more distant platforms. At the most difficult levels, the player will have to carefully time jumping off an enemy if they wish to make the next platform.

The Player

The **player character control** will be physics-based, though precise horizontal control during jumps is essential. The screen needs to give the player a good view ahead of them. This means the player can't go back - only forward. They can stop if they need to. Like enemies, the player aesthetic is still under the concept phase, but currently looks to be headed towards a fantasy-based design.

Player death shouldn't be a big deal, and we plan on giving them large number of lives so that one bad jump doesn't mean that they need to immediately restart. If the player falls, they come right down from above as if they fell through the world. We may introduce a mechanic to gain additional lives if we feel the need.

Enemies

To begin, the following enemies have been defined. We'll brainstorm more later. Current designs for the enemies include gremlin-like creatures, but Art is still concepting visuals.

- **Walker:** Simplest ground enemy. Walks along a platform, changing direction if it encounters a cliff or bumps into a wall.
- **Jumper/Bouncer:** Ground enemy that jumps up when it lands on the ground. We'll use physics to give it a nice arc. It doesn't move side-to-side unless pushed.
- **Charger:** This is a ground enemy identical to the Walker, but will speed up if it gets close to the player and is moving towards the player. Will attempt to push the player, and will run off a cliff to do so.
- **Bobber:** Air enemy that moves up and down between a high and low point. Not physics-based, it just bobs at a constant velocity. It doesn't move side-to-side unless pushed.

- **Flapper:** Air enemy identical to the Bobber, but uses physics to jump up when it reaches a low point.
- **Orbiter:** Air enemy that orbits around a point in a circular pattern.

Level Design

The **levels need to be randomly generated** from smaller platform sections we'll design. The same platform section can't appear next to itself.

Enemies should randomly appear on these platforms, with flying enemies filling gaps. Gaps farther than the player can jump will always have a flying enemy for them to jump off of.

Score Points

Player scores points for how far they get, and bonus points for enemies killed. Distance is more important than how many enemies the player has killed.

We'll have to play with the scoring math to find a good balance. Lives and score need to be shown to the player at all times, but should not distract gameplay.