2D Adventure Game

Senior Project I
Cristino Lamadrid, Karen Salinas, Keith Harryman, Jose Reyes
Introduction

- Understanding the game development process.
- Expanding our knowledge post Software Engineering.
- Reinforcing and learning a new language: C#
- Working with advanced software such as a game development platform.
Choosing a Game Engine

● Using OpenGL to build from scratch
  ○ Too low level and complex.

● Using a Game Development Platform
  ○ Reduces the need to focus on compatibility concerns with multiple platforms.
  ○ Provides a user friendly experience.
Unity vs Others

- Some are too simple
  - Require little to no programming at all.
  - Use their own proprietary coding language.

- Unreal Engine
  - Focuses on game logic
  - Excessive for 2D, usually used for much larger 3D games.

- Unity
  - Works well with smaller projects and 2D projects.
  - Uses C# or Javascript as its main language.
  - Allows compiling to multiple platforms both mobile and desktop.
Illusion of Depth

- Parallax Scrolling
  - Uses multiple background layers.
  - Each of the layers move with the camera at different speeds.
Illusion of Depth

Each individual piece of the background and foreground appears on a different layer.
Shaders

- **Surface Shaders**
  - Used to simulate realistic lighting on materials.

- **Vertex and Fragment Shaders**
  - Used to show simple and complex effects and other postprocessing effects such as the flame itself.
Camera Manipulation

- Controlling how the game camera reacts.
  - A single player vs multiple players.
  - Showing the player enough information.
  - Reacting to certain events such as unlocking a faraway door.
Camera Manipulation

BattleBlock Theater:
2D Method

Collision Detection

- When one collider makes contact with another object’s collider.
- Shooters require precise collision detection.
- Less frustration for the player.
2D Method

Animation

- Can be done by changing the sprite image at constant intervals.
Sound Capability

- Plays an integral part in any game as it affects the mood of the player at a conscious and subconscious level
Base Game Features

Game Manager
- Controls the state of the game and is referenced by other components of the game such as menus, item pickups, or opening doors.
- Opening doors, NPC interactions, item pickups, or puzzles solved.

Text System
- Displays text to the player such as the game's story, warnings, or events.
Base Game Features

Player & Enemies

● Main character’s abilities.
  ○ Jumping, crawling, climbing, or attacking.
  ○ Unlocking new weapons or abilities such as double jumping.

● Enemy AI
  ○ Scripted AI

● Incremented Difficulty
  ○ Determined based on location and abilities unlocked.
Base Game Features

Save System
- Required in order to save the player’s data locally.
- Must correctly save the player’s progress.

Player HUD
- Displays important information at a glance without opening a menu.
- Usually includes a minimap, health, and ammo.