AT02 Gamification Document

Chris O’Brien

30060241

**Version Control:**

To maintain the integrity of the project files throughout production, Git will be used for version control. All project files, including code, assets, and documentation, will be stored in a GitHub repository.

<https://github.com/mooza99088/2024_IntroductionToGameDesign_AT02>

**Project Management:**

I will be managing my project tasks and schedule in a word document. This document will contain a schedule to keep me on track throughout the project and a list of all tasks that need to be completed.

**Asset Implementation**

**Types of Game Assets**

1. Graphics
   * 2D Textures and Sprites
   * 3D Models
2. Audio
   * Sound Effects
   * Background Music
3. Animations
4. Levels/Maps

**2. Evaluating File Formats**

Graphics

* 2D Textures and Sprites
  + PNG: Lossless compression, alpha transparency, widely supported, ideal for UI elements and sprites.
  + JPEG: Compressed, suitable for background textures where slight quality loss is acceptable.
  + TIFF: High-quality, lossless, but larger in file size.
  + SVG: Vector graphics, scalable without loss of quality, good for UI icons.
* 3D Models
  + FBX: Supports a wide range of features including animations, widely used in the industry.
  + OBJ: Simple and widely supported but lacks support for animations.
  + GLTF: Efficient, web-friendly, supports animations and PBR (Physically Based Rendering) materials.
  + STL: Primarily used for 3D printing, lacks texture and material support.

Audio

* WAV: Uncompressed, high quality, large file size.
* MP3: Compressed, smaller file size, widely supported, slight quality loss.
* OGG: Compressed, high quality, smaller file size, open-source alternative to MP3.
* FLAC: Lossless compression, high quality, larger than MP3/OGG but smaller than WAV.

Animations

* FBX: Supports complex animations, widely used.
* BVH: Bio vision Hierarchy, used for motion capture data.
* GLTF: Efficient and supports animations, good for web and modern engines.

Levels/Maps

* JSON/XML: For storing level data.
* Tiled (TMX): Popular format for 2D tile-based maps.
* Unity Scene Files (.unity): Specific to Unity.

**3. Potential Integration Issues**

1. **Compatibility:**
   * Ensure the game engine supports the chosen file formats.
   * Check for any limitations in the engine’s ability to read certain formats.
2. **Performance:**
   * Optimize asset file sizes to balance quality and performance.
   * Consider mipmapping for textures to reduce memory usage.
3. **Pipeline Integration:**
   * Establish a consistent pipeline for importing and converting assets.
   * Automate conversion processes where possible to maintain consistency.
4. **Quality Control:**
   * Implement checks for asset quality (e.g., resolution, compression artifacts).
   * Ensure animations and models conform to engine requirements (e.g., bone structures).
5. **Version Control:**
   * Use version control systems to manage asset versions and collaboration.
   * Handle large binary files efficiently to avoid bloating repositories.
6. **Asset Dependencies:**
   * Manage dependencies between assets (e.g., textures linked to models).
   * Ensure all necessary files are included during builds.

**Monitoring Progress**

**1. Milestone-Based Monitoring**

**Implementation:**

* Key Milestones: Identify critical milestones in the timeline.
* Milestone Reviews: Evaluate progress at each milestone.
* Action Plans: Address any delays or issues promptly.

**Benefits:**

* Focused Checkpoints: Assess project health at key points.
* Early Problem Identification: Prevent issues from escalating.
* **Goal Alignment:** Ensure the project meets objectives and deadlines.

**2. Regular Progress Reviews**

**Implementation:**

* Weekly Meetings: Review progress against the schedule.
* Progress Reports: Team members submit weekly updates.
* Adjustments: Modify schedules and resources as needed.

**Benefits:**

* Frequent Check-ins: Early detection of issues.
* Transparency: Clear documentation of progress.
* Responsive Adjustments: Keep the project on track.

Using these strategies, I can effectively monitor and maintain production progress, ensuring timely project completion.

**Game Engine Evaluation**

**Unity**

|  |  |
| --- | --- |
| Pros | Cons |
| Free to use | New licensing policies |
| Can import unity packages | Does not have built in templates |
| Uses C# as native programming language |  |
| Has all tools needed to make a 3D game |  |
| Capabilities for realistic graphics |  |
| Clean and easy to learn interface |  |

**Unreal Engine**

|  |  |
| --- | --- |
| Pros | Cons |
| Free to use | Cannot import unity packages |
| Full toolset for making 3D games | Uses C++ as native coding language |
| Advanced Lighting and graphics systems | Difficult to understand interface |

Based off the comparison and extra research the most suitable game engine for the project is unity.