CMPE 491

HIGH-LEVEL DESIGN REPORT

for

"Sanctified Retribution"

by

Metehan TÜTER

Elif Aysu KÜRŞAD

Ceren Bilge OYAR

Erkan SANCAK

Date: 06.06.2023

1. Introduction

1.1 Purpose of the System

The purpose of this system is to develop a 2D Roguelike action game with pixel art style and a captivating storyline. For the system's map design and character/item development, Aseprite, Blender, and the Unity game engine are used. By allowing players to interact with side characters, adversaries, and the terrain itself, the game hopes to deliver compelling gameplay. The existence of an AI system for members means that they will react to player actions and initiate narrative behaviors. Including any adversary kinds suggest that the game will have various difficulties, changing how players approach the game.

Players over the age of 15 are the game's target population, indicating a focus on mature themes and gameplay mechanics. We intend to offer in-game hints where appropriate, such as through interaction with boss characters, to ensure players grasp the game's goals.

The overall goal of this system is to produce a challenging and immersive 2D Roguelike action game for people over the age of 15 that combines interesting mechanics, pixel art aesthetics, and an engrossing plot.

1.2 Design Goals

-Engaging Gameplay: The main goal of this project is to create a game that is immersive, fun, and challenging for the players. This could include creating mechanics that are simple to understand and quick to react to, and beneficial to the players actions.

-Captivating Storyline: An engaging storyline that keeps players interested in the game is another design objective. This entails creating engaging characters, compelling plot roles, and significant player decisions that have an impact on the story.

-Visual Appeal: With a pixel art aesthetic, a design objective is to produce visually pleasing images and animations that improve the game’s ambience. This may mean

creating intricate character and environment designs, making use of vivid color schemes, and maintaining a constant aesthetic direction.

-Exploration and Diverse Content: The game aims to provide a variety of content and promote exploration. This entails creating a map with hidden locations, undiscovered routes, and extra side quests to reward players for thoroughly exploring the game's setting.

-Clear communication and player guidance: One of the design aims is to make sure that players understand the game's mechanics, goals, and growth. To help players progress through the game, this could involve providing simple tutorials, in-game hints, or cues, and clear visual or auditory feedback.

-Player Immersion: The game tries to achieve this by using sound effects, moody music, and sound design that improves gameplay and the story. This can also entail developing an overall gaming universe with recurrent themes and mythology.

1.3 Definitions, Acronyms, and Abbreviations

-2D: Two-dimensional, referring to the game's graphical style and gameplay perspective.

-Roguelike: A category of video games with permadeath, randomly generated levels and an emphasis on strategy and exploration.

-Unity: A well-liked game engine for creating video games is Unity. It is a software framework that makes use of the C# programming language and has libraries and helper packages.

-Aseprite: The project's terrain, character, and object designs were created using the pixel art and animation application Aseprite.

-Blender: For more complex character or environmental elements, the production used a 3D modeling and animation program.

-AI: Artificial intelligence, which in this instance refers to the mechanism in charge of determining how the enemies in the game behave.

-Graphical User Interface (GUI): This term refers to the visual components and controls that players use to engage with the game.

-API (Application Programming Interface): It is a set of guidelines and protocols that enables communication and interaction between various software components.

-DBMS: It is a software tool used to organize and manage databases.

-Quality Assurance: Quality Assurance, or QA, is the process of testing a game to make sure it is reliable, error-free, and operating as intended.

-UI: User Interface, or UI, refers to the visual components and interface controls that let players engage with a game.

1.4 Overview

The project's main goal is to create a 2D Roguelike action game with pixel visuals and an engaging story. The Unity game engine, which offers a software basis for game production and includes necessary libraries and support tools, is used to construct the game. The construction of characters, items, and maps is done with the help of additional programs like Aseprite and Blender. The game includes interactions with enemies, side characters, and the actual game map. The adversaries will include an AI system that causes them to act when the player comes close enough, relating to the game's main storyline. Different enemy types will provide various gameplay concepts, presenting a challenge for the player to modify their approaches. Players above the age of 15 are the game's target population, thus mature themes and gameplay elements are emphasized. When required, in-game hints will be given to make sure players understand the goals and rules of the game.

In brief, the project aims to provide a 2D Roguelike action game that is compelling, challenging, and aesthetically pleasing, with pixel art aesthetics, an alluring story, and a range of gameplay mechanics.

2. Current Software Architecture

As Sanctified Retribution is a new game project, there is no existing software architecture to describe.

3. Proposed Software Architecture

3.1 Overview

The proposed software architecture for Sanctified Retribution is designed with an object-oriented approach, making use of the Unity game engine and the C# programming language. The architecture prioritizes to provide a seamless and immersive 2D roguelike gaming experience while maintaining modularity, reusability, and maintainability to support efficient development and future enhancements. The architecture focuses on separating different aspects of the game's functionality into subsystems, including player controls, game mechanics, level generation, artificial intelligence, graphics rendering, and audio management.

3.2 Subsystem Decomposition

The game can be divided into distinct subsystems to efficiently handle different aspects of gameplay.

-Player Controller: Takes charge of handling player input, controlling the character's movements, managing health and inventory, and facilitating interactions within the game world.

-Enemy AI: Handles enemy behaviors, including movement patterns, decision-making, and combat mechanics of various enemy types.

-Level Generation: Generates procedurally generated levels with various layouts, environmental, and interactive elements.

-Game World: Manages the game's map, objects, and interactive elements.

-User Interface: Manages the user interface, including menus, in-game HUD (heads-up display), notifications, and feedback systems.

-Game Logic: Implements the game's rules, progression, and event handling.

-Animations: Controls the animation states and transitions for characters, objects, and environmental elements.

-Audio Manager: Handles the sound effects, background music, and audio cues to enhance the immersive gaming experience.

-Save System: Handles saving and loading of game progress.

3.3 Hardware/Software Mapping

Sanctified Retribution ensures compatibility and optimal utilization of the available hardware resources on desktop computers. The game is designed to provide an immersive and engaging experience, making use of the monitor screen, keyboard, or optional controller for user interaction. The software products, Aseprite, Blender, Unity, and C#, are integrated into the development process to create visually appealing graphics, implement gameplay mechanics, and achieve the desired functionality on desktop platforms.

- **Aseprite:** Aseprite will be utilized for creating and editing 2D pixel art sprites and animations, which are essential visual assets in the game.
- **Blender:** While Sanctified Retribution is a 2D game, Blender can still be utilized for creating cinematic cutscenes, animated visual effects, or even generating 3D assets that can be used in the game.
- **Unity:** Unity is the game engine used for developing Sanctified Retribution. It provides various tools and features for game development, including physics simulations, asset management, and visual editor.
- **C# Programming Language:** C# is a modern and versatile programming language, to enable seamless scripting of game mechanics, user interfaces, and intricate game logic within the Unity engine.

3.4 Persistent Data Management

Sanctified Retribution will feature a save system that enables players to store and retrieve their game progress. This system will effectively manage the storage of player data, such as completed levels, unlocked items, and character statistics. By implementing this feature, players can easily continue their gameplay from where they left off, ensuring a seamless and enjoyable experience.

3.5 Access Control and Security

Sanctified Retribution prioritizes access control and security to protect user data and maintain a fair gaming experience. The game implements mechanisms to ensure authorized access, prevent cheating, and safeguard player information.

3.6 Global Software Control

System handles the storage of player data, including completed levels, unlocked items, and character statistics. By implementing this functionality, players can effortlessly resume their gameplay from their previous progress so they don't worry about losing their data and ensuring a smooth and enjoyable experience. The system effectively manages and organizes player data, continuation of the game from where players left off.

3.7 Boundary Conditions

The game's features, mechanics, and interactions are limited to the defined specifications and requirements. Any functionalities beyond these boundaries are not within the scope of the game. The following conditions have been established:

- **Offline Mode:** Sanctified Retribution is designed to be played offline, without the need for an internet connection. Players can enjoy the game without relying on online connectivity.
- **Single-Player:** The game is exclusively designed for solo story based gameplay, focusing on individual gameplay story and progression. It does not include multiplayer or cooperative features.
- **Platform Compatibility:** Sanctified Retribution is designed and developed exclusively for the Windows operating system. The game is optimized to run seamlessly on Windows-based desktop computers. While Sanctified Retribution is initially focused on the Windows operating system, there is a possibility of future development for mobile platforms.
- **System Requirements:** The game's simplistic visual style and lightweight processing demands ensure that it can be played on a wide range of computers, including older or less powerful systems.
    - Minimal System Requirements for a Sanctified Retribution:
        - Operating System: Windows 7 or later
        - Processor: Dual-core 2.0 GHz or higher
        - Memory: 2 GB RAM
        - Graphics: Integrated graphics card or dedicated graphics card with at least 512 MB VRAM
        - DirectX: Version 9.0c
        - Storage: 500 MB available space

    - Recommended System Requirements:
        - Operating System: Windows 10 or later
        - Processor: Quad-core 2.5 GHz or higher
        - Memory: 4 GB RAM
        - Graphics: Dedicated graphics card with at least 1 GB VRAM
        - DirectX: Version 11
        - Storage: 1 GB available space

The game will have minimum system requirements to ensure optimal performance. However, please note that these are recommended requirements and may vary depending on the specific details of the game and any future updates.

- **Legal and Copyright Compliance:** Sanctified Retribution will comply with all relevant laws, regulations, and copyright restrictions. It will only use legally obtained assets, respect intellectual property rights, and avoid any infringement of copyrights or licenses.
- **Age Rating:** The game will receive an age rating, indicating the appropriate age group for players.

These boundary conditions provide the framework for the development and deployment of Sanctified Retribution, ensuring that it meets the desired specifications.

4. Subsystem Services

Each subsystem identified in Section 3.2 will provide specific services and functionality essential to the game. These services include, but are not limited to:

-Player movement and combat actions.

-Level generation.

-Enemy AI behavior.

-Game world rendering and management.

-Inventory management and item interaction.

-Quest progression and dialogue interactions.

-User interface display and interaction.

-Audio playback and sound management.

-Save and load game progress functionality.

5. Glossary

-Sanctified Retribution: The title of the game, a 2D pixel-style rogue-like game developed by our team.

-Rogue-like: A genre of video games characterized by procedurally generated levels, permanent death, and a focus on exploration, strategy, and resource management.

-Pixel art: The intentional placement of pixels to create an image. It's a beloved gaming art style for its nostalgia, charm, and accessibility, and it's often used in 2D games to build game sprites and backgrounds.

-Sprites: 2D graphical images or animations used in video games to represent characters, objects, or visual elements within the game world.

-NPCs (Non-Playable Characters): Characters in the game controlled by the computer rather than the player. NPCs in Sanctified Retribution may provide quests, information, or trade services to enhance the player's interaction with the game world.

-**2D Game** 2D games use flat graphics, called sprite, limited to two dimensions: width and height, and don't have three-dimensional geometry

6. References

-https://www.aseprite.org/docs/

-https://www.giantbomb.com/sprite/3015-491/

-https://gamemaker.io/en/blog/make-pixel-art-2d-games

-https://www.blender.org/about/

-https://docs.unity3d.com/Manual/index.html

-https://plarium.com/en/blog/npc-non-player-character/

-https://learn.microsoft.com/en-us/dotnet/csharp/tour-of-csharp/

-https://unity.com/how-to/difference-between-2D-and-3D-games