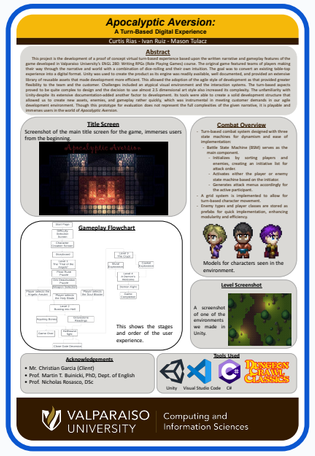
1



Apocalyptic Aversion:

A Turn-Based Digital Experience

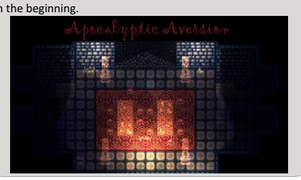
Curtis Rias - Ivan Ruiz - Mason Tulacz

Abstract

This project is the development of a proof of concept virtual turn-based experience based upon the written narrative and gameplay features of the game developed in Valparaiso University's ENGL 280: Writing RPGs (Role Playing Games) course. The original game featured teams of players making their way through the narrative and world with a combination of dice-rolling and their own intuition. The goal was to convert an existing table-top experience into a digital format. Unity was used to create the product as its engine was readily available, well documented, and provided an extensive library of reusable assets that made development more efficient. This allowed the adoption of the agile style of development as that provided greater flexibility to the team and the customer. Challenges included an atypical visual environment and the interaction systems. The turn-based aspects proved to be quite complex to design and the decision to use almost 2.5 dimensional art style also increased its complexity. The unfamiliarity with Unity-despite its extensive documentation-added another factor to development. Its tools were able to create a solid development structure that allowed us to create new assets, enemies, and gameplay rather quickly, which was instrumental in meeting customer demands in our agile development environment. Though this prototype for evaluation does not represent the full complexities of the given narrative, it is playable and immerses users in the world of *Apocalyptic Aversion*.

Title Screen

Screenshot of the main title screen for the game, immerses users from the beginning.

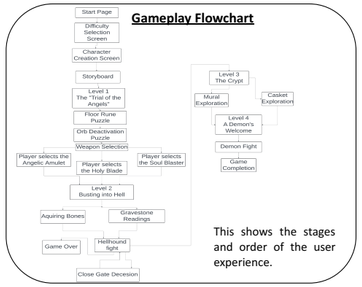


Combat Overview

- Turn-based combat system designed with three state machines for dynamism and ease of implementation:
  - Battle State Machine (BSM) serves as the main component.
    - Initializes by sorting players and enemies, creating an initiative list for attack order.
    - Activates either the player or enemy state machine based on the initiator.
    - Generates attack menus accordingly for the active participant.
- A grid system is implemented to allow for turn-based character movement.
- Enemy types and player classes are stored as prefabs for quick implementation, enhancing modularity and efficiency.

Models for characters seen in the environment.

Gameplay Flowchart



This shows the stages and order of the user experience.

Level Screenshot

A screenshot of one of the environments we made in Unity.

Acknowledgements

- Mr. Christian Garcia (*Client*)
- Prof. Martin T. Buinicki, PhD, Dept. of English
- Prof. Nicholas Rosasco, DSc

Tools Used

Unity Visual Studio Code C#

VALPARAISO UNIVERSITY | Computing and Information Sciences

Click to add speaker notes