

Giants of the Wild Final Report

Group 23: Mo Abdelmajid, Aryan Rogye, Giancarlos Arenas-Barajas, Nikola Kovilic

Project Overview

Giants of the Wild is a 3-D open world game where you take control of Elephants. Players will experience the survival life of an elephant, taking on environmental challenges and interactions, along with dealing with the dangers of poachers, while learning about elephant behavior, diet, and the dangers they face as an endangered species. The game supports both single player and multiplayer modes to help blend education and awareness to elephants.

Purpose of the Project

Giants of the Wild was made to raise awareness for the endangered species on elephants whether it is from climate change, poachers or habitat destruction. The project aims to use a video game system to educate the youth on this topic

Success will be measured of course on how much the game is actually played by our audience, but also the amount of successful playthroughs, since the whole goal is to raise awareness about elephants we want the players to actually successfully live their lives

Scope of the Project

The proposed product focuses on delivering an educational open-world elephant simulation. The single player experience allows the user to explore the different stages of an elephant's life and the challenges they face at each of those times. The single player focuses on the empathetic survival experience, while the multiplayer side adds a more arcade like competitive experience..

Use Case Summary

The requirements for Giants of the Wild come directly from our use cases, with each use case representing a core aspect of the game either being player interaction or system behavior. These helped reinforce the core values of the project, which is to educate and entertain users using the simulator

Requirements Overview

Requirements are parts of the project that need to be met for it to work as intended, these include functional requirements such as player movement, starting the game, AI mechanics and save/progress. Other requirements include data, performance, dependability, security, and accessibility.

Design Goals

The design goals for Giants of the Wild are centered around performance, realism, modularity and educational value. These goals are all important because the game needs to function well at all times since it is a simulator, and modularity allows for easier updates in the future designs of the game

Proposed System Design

Our proposed system design identifies the major classes and structures that make up Giants of the Wild, and model how the core use cases behave, and show the modularity of the subsystems that was talked about previously in the design goals. The dynamic models illustrate the flow of how a single player session would work, as well as how the educational prompts are triggered and displayed in the game.

Additional Design Considerations

There are several design considerations that strengthen the system's robustness, maintainability, and real-world operability. Persistent data management ensures that player progress, settings and world states are reliably saved from the previous game session using the SaveLoadManager. This also defines boundary conditions, which covers startup, shutdown and error recovery which ensures stable behavior across all execution phases. This also shows off the use of the singleton design pattern for the unique global components like GameWorld to enforce consistency and maintainability throughout the system.

Final System Design

Our final system design was a showcase of the key game pieces and what they do. Just to review a few, the AI predators have types and aggression levels, WaterSource has location, refill rate and the current water level. All of these pieces come together into making the GameWorld work and make the game work properly.

Object Designs

Our object designs describe the implementation level structure of the system, and focus on how the software is organized into packages and subsystems to support the games functionality. We have 5 major packages, ClientUI, MessagingService, GameLogic, ControlsLogic, and UserInformation which each group related classes and responsibilities. We also have two subsystems, one for game management and one for application management. Together these all define how the game's internal objects are structured and how responsibilities are divided across modules to support maintainability and scalability