The lost Empire

Group10: Daniel, Nianzu Ma, Yue Ben
2013 spring

The Lost Empire is a three-dimensional thinking game related to the domain of scuba diving. This game is designed to be virtual reality with a first person view of the environment, which gives the user a nearly authentic diving experience. Also, it adds a treasure hunting as entertainment element to stimulate user's interest to adventure in the underwater world and experience scenarios that can only being seen in films. Since this game is for both scuba divers at different levels and common users without much scuba diving knowledge, sufficient tutorial document, video and mode are provided to inspire the user to learn more diving skills while having fun in the game.

The game will have an adventure mode. The player in the game will start out with nothing but a scuba suit and hope of finding the ultimate treasure. The Player will have a limited gas supply and is very vulnerable to the dangers of diving underwater including drowning, breathing poisonous Oxygen etc. But there is also a high reward for this exploration, as the Player can collect various items that are worth money on the surface. But they do have to make it out alive.

The game is eventually expected to be published by Electronic Arts. The anticipated game player is between a teenager and someone in their mid-thirties. Scuba diving fans may also appreciate the game, regardless of their previous gaming experience. Professor Bell and Michael Angelo Gagliardi are Scuba Diving experts who provided scuba diving information for the game. Their effort ensures the realistic diving experience of player.

System Architecture

The system of the game is an open layered system architecture, briefly decomposed into User Interface subsystem, Game subsystem, User Data subsystem, and Item subsystem.

1. User Interface subsystem

The Lost Empire will be launched from and be run by the User Interface System, which is responsible for the initial game start up, allowing the user to select and load user data file. The User Interface Subsystem contains the world map view, Game view, Store view and the Bag view.

The World Map view World Map view provides overview of the map and game level selection. User can either select level according to description or following the recommended routine for the surfing order among the Levels.

The Game View (within Game/Level) The Game View is the view shown to the user when they are playing the Game. There will be text instructions and panels with lots of parameters to give guidance to user during the exploring.

The Bag view At any time inside or outside the Level, the Player can click Bag and see Equipment/Tools and Items inside.

The Store view The player can go to Store to sell Items outside the Level, and buy or upgrade Equipment, as well as upgrade his own physique such as breathing rate.

2. Game Subsystem

The Game subsystem contains two sub-subsystems - the World Map Subsystem and the Game Play Subsystem, and itself has no real responsibilities.

The World Map Subsystem The World Map Subsystem is mainly responsible for managing all game data, maintaining all game level maps and saved games. During the activities in the Game Play subsystem including creating a new file, loading a file, or appending temporary game to a file and so on–--- all the data involved is provided and stored in the World Map subsystem.

The Game Play Subsystem The Game Play Subsystem is initialized based on the parameters sent from World Map. It is responsible for translating user commands into game actions and also real time calculations of depth, ATA, Narcosis, breathing rate, gas consumption, and time left and so on. The Game Play Subsystem will also save current game if requested, append the user's saved file if the user exits the level successfully and returns to the World Map view, or it will exit the user's saved state and return to the User Interface Subsystem. Its direct cooperator is the Game view in the User Interface subsystem, which is responsible for displaying a viewport, or in some cases a 3D view, in real time.

3. User Data Subsystem

The User Data subsystem is responsible for maintaining user profile and game-related user data, such as the player's breathing rate and so on, to help judging user's ability and adjust game progress.

4. Item Subsystem

The Item subsystem is responsible for Item operations in Store and in Bag – realizing functions called in the Store view and the Bag view, also defining the basic attributes in the abstract class Item (not realized), as well as maintaining data files for Items in Store. Not all items in the game are managed here, player-owned Items (in his bag) are in the player's user data (thus in User Data subsystem), while Treasure Items still inside levels are in Game/Level data files.

The game has a sports-oriented, dynamic operation interface with blue as dominate hue and integrating dynamic lines, pictures and shape for every bar, button and background. It can give user optimal graphic display optimally based on different hardware. The game control is based off of current PC games which facilitate most customers. It can be easily extended for adding extra levels or new series of adventure mode of different themes/stories. The game shall be written in Java and Java OpenGL, using flat files for storage and primarily compatible with recent versions of windows (XP to 8) and Unix-like system with Java Runtime Environment.