96
All the best for your finals!!

Lithos - Final Summary

Group 22 - Aravind Achanta, Yonathon Gordon, Naga Kameswari Spoorthi Pendyala, Sai Priya Jyothula, Sumanth Reddy Pandugula, Vakkalanka V S S Dilip Raju

A Brief Game Description

Lithos is an interactive multiplayer strategy game in which the user plays the role of a tribal leader who tries to overcome various obstacles in Prehistoric Age. Unlike many other games that have been developed in this genre, **Lithos** provides the user with knowledge about Prehistory in an exciting gaming environment without sacrificing either of entertainment or education for the sake of the other.

At the beginning of the game, the player leads an uncivilized tribe with a primitive mindset that is solely focused on survival. The timeline of Prehistory is evenly spread as levels starting from Paleolithic Age, continuing through Iron Age and finally emerging as a civilization. At the start of each level, the goals that are related to the particular level (era) are mentioned. The goals may consist of tribe survival, developing skills, building houses, making implements and technological advances (like discovering fire or inventing wheel). After the completion of the goals, the player is prompted to answer an end of the level quiz which dwells on the events of the current level - if the player answers the quiz satisfactorily, he/she can move on to further levels. The game finally ends with the tribe successfully replicating the events of prehistoric age and settling down as a sophisticated civilization.

The overall concept of the game is that the player has to employ strategies to deal with various situations and crises like hunger, wildlife, climatic conditions that force them to find a shelter, relations with other tribes and progress technologically according to the chronology of Prehistory. The competitiveness of the game is ensured as the player goes up AI which ceaselessly engages the player by tossing unpredictabilities at the tribe whenever there is a period of safety. As the tribe evolves and the game advances, the tribe gains a skill set - the matchmaking app uses this info to hunt down other tribes led by external players with compatible skill set. Then these tribes can go against each other or collaborate to go against some other tribe, basically empowering the game to go into multiplayer mode.

Requirements

Functional Requirements:

The system should be able to load a pre-existing game or new game.
The system should provide a means to view skillset and choose tribe members, implement and assign tasks. The AI should create simulation reports and add challenges.
The system should show information screens displaying goals, clues, meters and skillset.
The system should provide a means to answer quiz and compute score by checking answers.
The system should search for compatible external players, send invites, monitor responses and provide a means to choose between collaborate and fight.

Non-functional Requirements:

Performance: When the user is playing the game, the frame rate in high resolution should be 24 FPS for 90% of the time and should not fall below 18 FPS under any circumstances. If the user bandwidth falls below 1 MBPS, then the graphics should be compromised instead of lowering the frame rate.

Accuracy: All information used in the game should be historically accurate.

Dependability: In the unlikely event of a system crash, data recovery mechanism must be triggered. The game server should be available 99.9% of the time and in case of server maintenance, the server should resume functionality within 15 minutes.

Maintainability and Supportability: The updates should be released every 6 months or in case of severe problems. The plugins and additions should be easily added. The game should run on all major operating systems.

Security: The system should maintain end-to-end encryption and must be free of potential exploits. It should have a secure network to reduce the possibility of counterfeit data.

Usability: The system should be easy to use with minimal computer literacy requirement. The system should support GUI and language customization by the user. The system should also have features like audio cues, closed captioning to aid visual or hearing impaired users.

Look and Feel: The game should have intuitive user elements for an engaging game play with 3D realistic characters. The font styles, colors, sounds and menus should correspond to prehistorical age (such as ancient gothic, manuscript style)

Legal and Standard: The game should avoid copyright infringement and should comply with Milspec, ESRB and FTC standards.

Test Plans and Testing Schedule

The application will require thorough testing for the single player as well as the multiplayer mode. The testing methodology should start with unit testing with approaches of white box, black box and full configuration testing. Unit testing is performed on the components as and when they become available. The individual units are compiled into systems based on functionalities. There will be test cases for the login functionality of the game, for every level in the game, for single and multiplayer mode and for validating the download and installation of the game. After a system has passed all previous tests, integration testing will be done prior to that system's actual integration into the game. Since the game is developed using the UNITY 3D engine, testing should be done within the UNITY framework.

Once the software is developed using the agile methodology and tested, there has to be a build made for each of the platforms such as Windows, Mac OS X and Linux. Each build has to be tested on the platform on which it is meant to run. The test schedule specifies that the project will be given about three years to develop, and so the testing should be done once a month. There will be extensive testing done after a year within the UNITY platform. The last 2 weeks before the final release will be spent on testing the build on various platforms and hardware.