Daniel Cecil - Team B3
26 Jan 2011

Software Team Java - First Specification / Proposal

Introduction

The aim of this project is to develop a simple, working, multi-player game based on the Java platform and to include sockets programming. The game chosen to develop is a side-scrolling space shooter similar to the retro games of Asteroid and Space Invaders. The team has already been formed of four members each taking equal responsibility in the program's development. The project will run for 11 weeks in total.

Functionality

- The game will be a side-scrolling space shooter from top to down.
- The user's spaceship will be able to move freely along the x and y axis but remain within the games boundary area (edge of window). The human player will be able to control ones spaceship using the arrow keys or WASD setup along with the space bar to shoot.
- The spaceship will not be able to rotate, instead always facing up which means the shooting will always go vertically upwards.
- The aim of the game is to avoid being hit and eliminate as many opponents as possible. There will be a finite number of spawns for the computer enemy so it is possible for the game to end with the user victorious.
- The computer generated opponents will spawn at the top of the screen and work their way down, the user is able to shoot these opponents. Once they reach the bottom of the screen they will vanish.
- The opponents will have the ability to shoot back at the player, a single shot to a player will 'kill' the user's ship, each user will receive a set of lives at the beginning off the game.
- The opponents will require multiple hits to be 'destroyed', users will receive points from doing so.
- There will be a sense of movement and progression from the user as the background will scroll using a 'camera'.
- The game will implement a home screen to connect to a host, view high scores, play a single player game or quit.
- Extra items that could be implemented depending on available time are:
  - Power ups
  - Boss fights
  - Different weapons
  - Levels with increasing difficulty
  - Advanced movement from enemy players
  - Weapon damage

GUI Design

The GUI will be fairly basic. The program itself will run in a window/frame with a menu bar at the top. The home screen will have buttons that can be selected with the mouse or keyboard. The spaceship will be controlled using keyboard keys only. The game will be able to be scaled to suit different screen sizes and user preferences as well as the option to enter a full screen mode. During the game itself there will be static information shown at the bottom of the screen such as: lives remaining, current score, other player(s) scores, high score and a timer.

Release Plan

Below is a proposed plan of internal releases, building up to a stable, playable game.

v0.1 - Base Code
Simple game window with a single-user spaceship that is able to move around.

v0.2
Basic shooting from the user's spaceship.

v0.3
Static enemies that are able to be 'shot' by the player using collision detection. Score counter implemented at this point.

v0.4
Movement of the enemies.

v0.5
Background scrolling, spawning opponents with shooting back.

v0.6 - Base game
Multi-player capabilities with the user of networking through sockets.

Software Engineering and Testing

As with any project, good software engineering skills are vital to the success of the overall assignment.
The project will see the use of the Incremental software life cycle where the project is broken down into small steps (see release plan) and thoroughly tested, after the unit is stable, work will begin to plan and implement the next step.

Each unit will undergo testing using JUnit within the development environment as well as being tested through the use of user acceptance testing at each step. This feedback will be used to plan for the next stages as to what improvements can be made or extra features the program should include.

As there will be many versions it is important to keep track of these changes - the project will use a subversion repository to organise the versions as well as maintaining progress logs within the system so it is possible to see what each member of the team has done. This will also help with maintenance in case any problems occur in the testing phases.

Networking

Every copy of the game will have two modes; single player or multi-player. The user will be able to select weather they will be the host or the client in the multi-player game. One player will act as the host where the computer generates the opponents and communicates this, as well as the other players movements, shots and scores to the other client. The client will also relay their position and movements.

The client will be able to enter the host’s IP address on the home page to connect. Once a connection has been established the game will begin.

The plans currently are to have only a maximum of two players, if more players are added to the game it may become too crowded to be enjoyable.