Online Multiplayer Battle Strategy Game

Aim

To develop an online multiplayer strategy game wherein two players pitch their troops against each other aiming to capture the opponent king. The game can be played manually or by writing a JavaScript program.

Utilities Employed

1. KineticJS to develop the game UI on HTML5 canvas.
KineticJS is an HTML5 Canvas JavaScript framework that enables high performance animations, transitions, node nesting, layering, filtering, caching, event handling for desktop and mobile applications, and much more.
http://kineticjs.com/

2. NodeJS in the backend along with ExpressJS for rendering static pages and socket.io for real time updating of the position of troops on the screen.

Node.js is a platform built on Chrome's JavaScript runtime for easily building fast, scalable network applications. Node.js uses an event-driven, non-blocking I/O model that makes it lightweight and efficient, perfect for data-intensive real-time applications that run across distributed devices.
http://nodejs.org/

ExpressJS is a useful JS library based on NodeJS that simplifies many Node tasks like rendering of static web pages. It is a web development framework for NodeJS.
http://expressjs.com/

Socket.io helps to carry out socket I/O operations over the HTTP. **Socket.IO** aims to make real-time apps possible in every browser and mobile device, blurring the differences between the different transport mechanisms. It's care-free real-time 100% in JavaScript.
http://socket.io/

Game Rules

Our game takes you back to the middle Ages, the era of knights and castles.
A player (warlord) must control the units under command and defeat the enemy warlord’s troops. The battle is won by slaying the enemy counterpart.

In ‘Might of Warlords’, we offer two game modes:

1. A click based option, where one must make decisions every turn, and respond using the click interface.
2. An option where one has to upload his code, using the pre-made functions we provide them. There is no control over the game events once the code has been uploaded and executed.

Click Mode:

In this mode, the player has to take his decisions using the graphical interface of the game.
On clicking a unit, a green circle appears indicating the region where the unit is permitted to move.
We identify the unit clicked and accordingly generate the circle with the appropriate radius.
On clicking, it obtains a ‘mini-cell’ location where the unit is placed.
Walls and mountains however cannot be clicked and moved.

Command Mode:

In the command mode, we will provide API’s to the player for their convenience.
They will have to pass arguments to these functions. For example, std_move (king, 2, f) would move the king 2 steps (50px) forward.
The player will have to describe the behavior of his/her units depending on the surroundings and write a suitable code.
Both players have to upload their code. We shall run them alternately and reflect the changes in the graphical interface of both players.

Game Strategy
Implementation

KineticJS has been employed to render the game UI on HTML5 canvas. Each army personnel in the game is a gif image the coordinates of which on the battlefield canvas are changed depending on the moves of the players. In the manual online version, the move made by one player is communicated to the server from where it is routed to the other player. The data transfer employs Socket.io. In the programmable version, both the players upload JavaScript programs to the server. These programs must include calls to the API methods provided for moving different personnel in different manners. The programs are executed on the server and the moves are communicated to each of the clients again employing Socket.io. Static pages like Home, About etc. are rendered using ExpressJS framework.

References

www.nodebeginner.org
http://www.codeschool.com/courses/real-time-web-with-nodejs
http://howtonode.org/
http://nodeguide.com/beginner.html
http://evanhahn.com/understanding-express-js/

http://www.html5canvastutorials.com/kineticjs/html5-canvas-kineticjs-image-tutorial/

Team Mentor: ShivanshuAggarwal

Team Members:

NovnitKashyap
AbhishekAnand
Swathi Krishna
ShubhamTripathi
Ravi Kiran