University of Toronto
Faculty of Applied Science and Engineering
ECE419 - Design Document

Date	March 14, 2016
Prepared By (Names and Student #s of Team Members)	-- 0xdeadbeef -- Rushab Ramesh Kumar (1000117919) Ismail Hossain (998340175) -- The Null Terminators -- Shafaaf Khaled Hossain(998891515) Sai Kiran Varikooty (998440307)

The distributed algorithm will be implemented by maintaining sorted queues for each client using logical clocks, broadcasting and acknowledgements. When a client makes a move, the event generated from the move is added to a queue for the client along with a logical clock value. The client then broadcasts the event at the head of the queue to all clients. For each other client in the game, when the client receives an event it will place the event in it's queue in the correct position based on the logical clock value and then broadcast the acknowledgement for the event to all other clients in the game. The client will also wait for acknowledgements from all other clients for this event. When every client has received an acknowledgment from every other client in the game for the head element of the queue, it will dequeue the event and execute the event.

To maintain the correct order of packets, clients will use a logical clock value to assign to every event. A client will examine the logical clock value of an event to determine where to place the event in it's queue. If a client receives an event with a logical clock value that is smaller than the event at the head of its queue, it will place this newly received event at the head of the queue. If the logical clock value of the event it receives is greater than the event at the head of its queue, it will place the event in a later position in the queue. The use of logical clocks allows events to be totally ordered and the use of sorted queues for clients will allow events to be executed in the proper order.

A player locates the Mazewar game by querying a well known address of a naming server. The naming server will contain information about clients currently playing in the game. Particularly, the naming server will have all the clients' ip addresses and ports to be connected to. When a new client tries to join the game, it will send a packet to the naming server which will contain

the client's ip address and port. The naming server stores the client details and replies back with the list of stored clients and their ip addresses and port numbers. The new client will then proceed to connect with every client from that list and joins the game.

The communication protocol for the game will consist of three categories of packets a JOIN and a MOVE packet. The MOVE category consists of the 6 packet types for moves in different directions by clients, firing, and projectile movement. The JOIN packet is sent to the name server when a client initially requests to join the game.

Each client will also use a timer to determine if a rebroadcast of an event is needed. When a client broadcasts its event, it will start the timer. If an acknowledgement from each client is not received within the timer interval, it will retransmit the event to all clients.