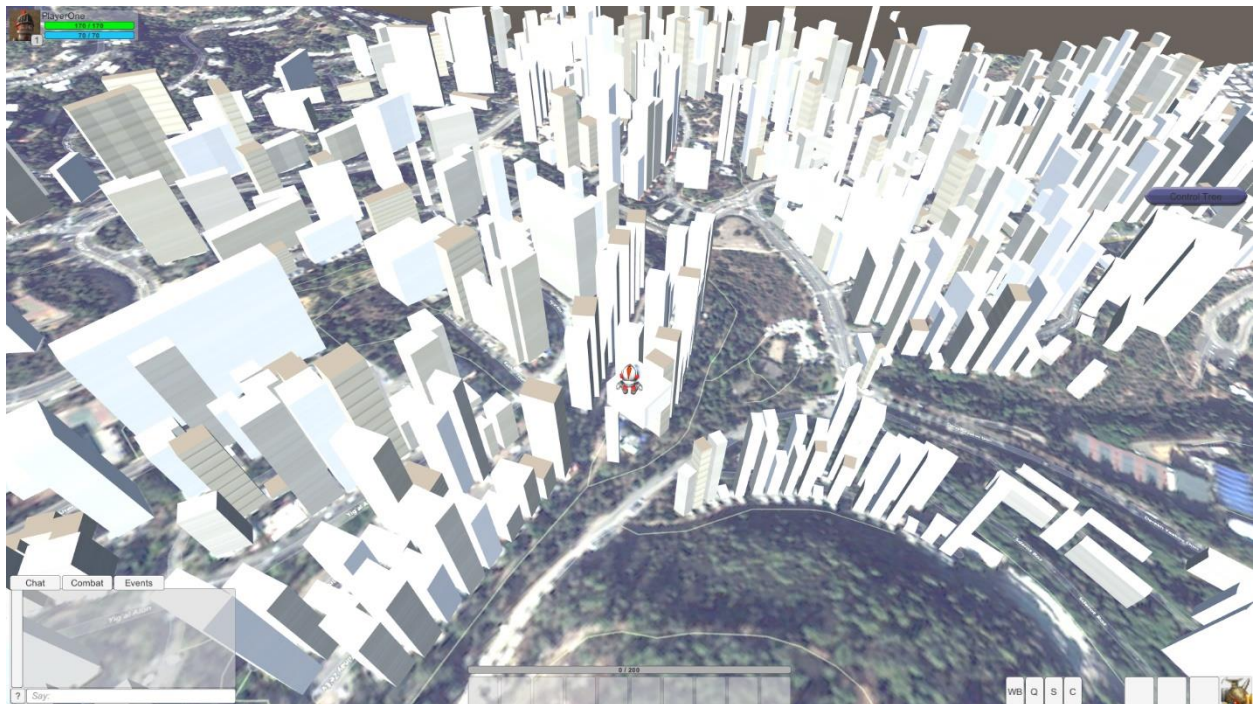
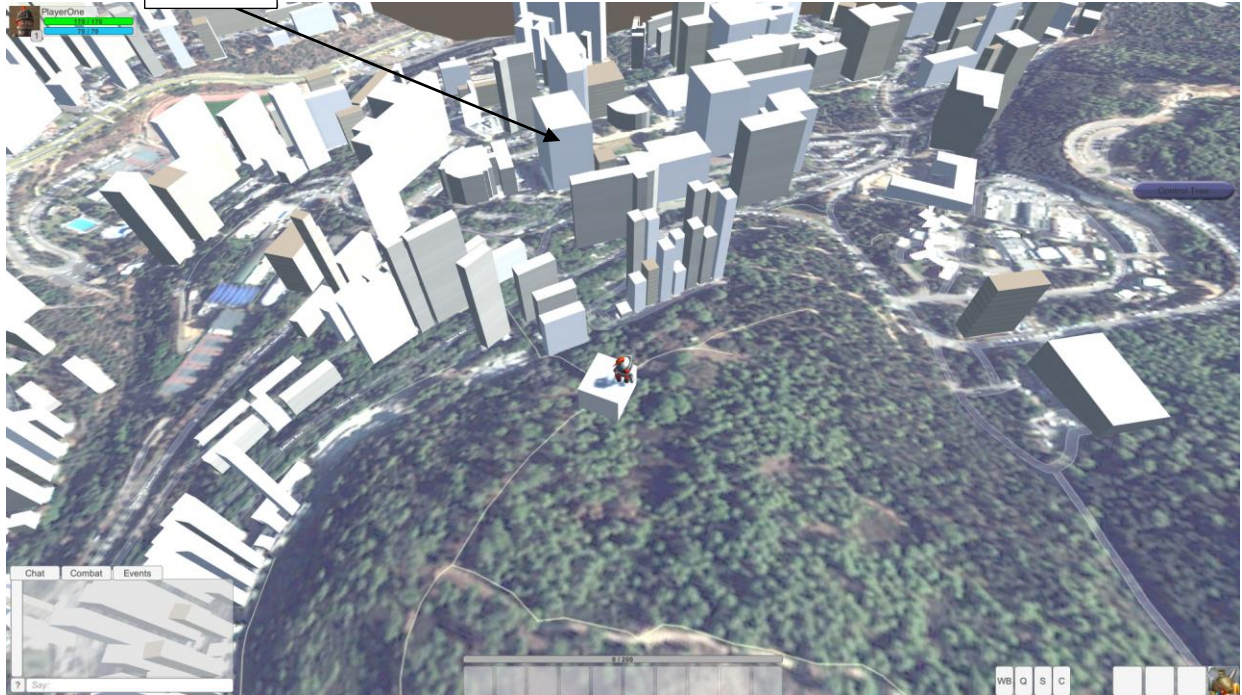


# Graphics Project

## Unity Game: Eyal Maymon

Taub



## Project Description:

This graphics project is a game built using Unity 3D editor with a java backend.

Players in the game can interact with other players and fight each other or with one another and computer AI, in chess-like skirmishes between units they acquire.

The game is of the following genre: Massive Multiplayer Online Turn Based Strategy (MMOTBS).

The terrain was created using satellite images, while the buildings were created using an online database, both of the Technion area.

## Challenges:

- Creating the terrain and aligning the buildings with it.
- Modifying packages.
- Recursive tree algorithms to create a spaced tree, to change players' standing with each other in the case of Control Tree changes, and taxation algorithm that sifts up from the players' nodes in the Control Tree.
- Server Java Backend: A "shallow" battle image had to be kept on the server side for synchronicity.
- Learning Unity from scratch and Creating my own modules both in Unity and on the Java backend, including UI.

## Challenges Accepted:

### Creating the terrain:

- I used a package which queries for satellite data and constructs the terrain. I then used another package to create the buildings in their respective x,y coordinate, but their height was many times not synchronized with the terrain height data. So I used a Raycast, to hit the terrain on a straight down vector(0,-1,0) from it's current x,y,z coordinates. I used that solution because it seemed that whenever the height was off, it was higher than the terrain.

### Modifying packages:

- The Turn Based package had to be redesigned to accommodate multiplayer, having most of its classes as static.  
Solved it by creating a context class that held the no-longer static classes.

### Tree Algorithms:

- Solved by modelling use-cases and synthesizing generic algorithms to fit them.

### Server Java Backend:

- Solved by use of a powerful serialization technique called Flat Buffers (by Google).  
The battle image that is kept on the server side, is first initialized, then mutated to keep updated and synchronized.

### Learning Unity and designing original parts:

- Hard work.

## User Guide:

The game consists of Players, Units, Control Points, Battles, Mobs, and a Control Tree.

Players – Each human player can log in with a character and play the game controlling the character, hence – player.

Units – Units are used by a player to fight battles, either in open field or by garrisoning them in a control point by moving unit items from the inventory to the control point.

A player can also garrison his character in a control point, which will move all the unit items from his inventory to the control point, and will prevent the player from starting battles or being attacked by battle.

Control Points – Control Points appear in the form of buildings taken from an online database, with roughly accurate height – from the Technion area, which shows in the form of satellite imagery terrain and synced with the Technion area buildings.

Control Points generate revenue every fixed time amount (currently 30 min), and revenue generated by them can be used by players to purchase units from merchants.

The generated revenue is paid half to the point's controller and half to the point's enabler, both of which are players, if there are any to the point.

Control Points can be captured, by a player with units in his inventory, only if the current point controller and the player don't have a common ancestor in the Control Tree which isn't its root.

If there are units garrisoned inside a control point, or players' garrisons with armies – then to capture the point a Siege Battle needs to take place, if the sieging armies are victorious the point is captured.

When a player captures the point, if it was previously controlled by a player, that player is demoted to be the point's enabler, and if it is the player's last controlled point, the controlling player gains control of the previous controller and becomes his parent in the Control Tree. In the case, it is the point's enabler last enabled

point, the capturing player gains control of the enabler, and a previous controller if there was one is completely evicted from the point (but the player still gains control of him).

Control Tree – After the revenue is generated by the control points, a taxation phase begins, and sifting from the Control Tree's leaves up – tax is being paid by each player from the income generated by the control points he enables or controls, to his parent in the control tree.

Battles – Battles are started when a player either attacks another player on open field – and then an open field battle starts, or a player lays siege to a control point. Battles are fought using units.

When an open field battle is started, all players in range will be put into battle, and their hostility towards each other will be the same as the player's name coloring (green-friendly, yellow – neutral and still attackable, red – hostile).

If a player is put under another's control in the control tree, the player's name coloring and hostility in battles will be put to green, as in friendly, towards all the players from the new parent up to the root.

When a siege battle is started all players that are friendly and nearby the sieging player will join the battle.

The game's goal is to gain control of as many control points and players as possible.

Note: There are limitations to how many control points a player can capture, but not to how many he can enable, meaning controlling other players can happen indefinitely.