Schedule:

- 6/21: Server logic for movement

- 6/27: Server logic for skill usage

- 6/22: Handling arguments

- 6:23: Legal check

- 6/26: Effect handling

- 6/27: Code polish/organization

- 6/29: Game state checks (check for skills at particular locations, check for win condition, updating units and variables properly)

- 6/27: Complete interface for client skill usage

socket.emit('playermove', {unit: 2, type: 'move'});

socket.emit('playermove', {type: 'skill', skill: 'attack', unit: 0, target: 2});

// \* Feel free to modify the structure of anything if it will help you, especially due to exception handling

// \* Once server logic can be handled, we add a system during ready phase (before game starts) to let players select their characters and where to place them in the starting line-up

// Game is won when the opponent's characters are all dead (HP falls to 0) unless otherwise stated

// Move algorithm

// - player sends a move request to the server

// - server checks whether or not the move is legal

// - if legal, apply move and send board state back to players

// - if illegal, ignore move

// Unit attributes:

unit = {

// id tells you what the character is

// id determines every other statistic that is not a 'base attribute'

id: 0, // fighter

// where the unit is located (not a base attribute)

// assigned depending on the location the player chooses on his line-up prior to starting the game

position: [0,0],

// name of character

name: 'fighter',

// maximum health

maxhp: 200,

// current health (not a base attribute)

// when health reaches 0, the character dies and is removed from the map and unable to take actions (unless otherwise stated)

// assigned to be equal to maxhp when game starts (unless otherwise stated)

hp: 200,

// strength, determines attacking power

strength: 40,

// maximum energy cap

maxenergy: 3,

// current energy (not a base attribute)

// energy is reduced for each character when taking an action with an energy cost

// if an action should drop energy below 0, it is not a legal move

// at the start of each player's turn, restore all of the player's character's energy to full (unless otherwise stated)

energy: 3,

// energy cost for movement (movement is up, down, left, right unless otherwise stated)

movecost: 1,

// skills the character has

skills: [

{

// name of the skill is the name of the attribute

name: 'Attack',

// energy cost of the skill

cost: 1,

// valid targets of the skill (determines what arguments are sent to the server)

// valid inputs include [ally/enemy/''/etc] unit, tile, passive

// if target type is passive, it means there is no target and the skill condition will be checked throughout the game (unless otherwise stated)

target: 'enemy unit',

// returns true is the location is valid

// \*\* we may need more properties to list out all of the range coordinates for the client ( rloc: [ [1,0],[-1,0],[0,1],[0,-1] ] )

range: function(dir) {

if (dir[0] === 0 && (dir[1] === 1 || dir[1] === -1)) return true;

else if (dir[1] === 0 && (dir[0] === 1 || dir[0] === -1)) return true;

else return false;

},

// all of the actions that trigger for each skill

action: [

{

// determines type of effect

// valid inputs include damage, stun, heal, weaken, strengthen, custom, etc.

// for a custom skill effect, use the custom property

type: 'damage',

// arguments specific to effect type

// ratio argument means to apply damage equal to strength \* ratio

ratio: 1

// \* these are not needed for the current skill but are listed out for explanation/example

// flat damage (as opposed to ratio)

// damage: 40

// returns true if the effect should be applied

// condition: function() {}

// method is executed for a custom skill effect

// custom: function() {}

}

]

},

{

name: 'Bullet Punch',

cost: 3,

target: 'enemy unit',

range: function(dir) {

if (dir[0] === 0 && (dir[1] === 1 || dir[1] === -1)) return true;

else if (dir[1] === 0 && (dir[0] === 1 || dir[0] === -1)) return true;

else return false;

},

action: [

{

type: 'damage',

ratio: 2.5

}

]

},

]

}

Fighter Class Passive, Ability

Bulldozer – Gains attack for each turn (?) passed. Chargeable(Can dash up to X amount of tiles for X amount of energy) Deal x amount of damage for units Bulldozer ran past. - rhyperior

Immovable Warrior – Stationary (Cannot attack) , Gain immense attack, Attack every enemy around the tile he is in. – glalie

Juggernaut – Gains immense defense for ( 2turns) and able to move twice tiles, as well as attack twice (1 turn)

tank - rhyperior - Able to buff allies raising their defense and giving them a shield. Also able to stun a target for a turn. Weak melee attack

mage - mismagius - Able to cast an AoE spell or a single target spell in a limited range. Stronger ranged attack

rogue - weavile - Able to enter stealth [Untargetable for X turns and melee attack turns into a dash] Medium melee attack

support - cresselia - Enemy target curse: allies deal more damage to cursed target. Heal, Energy heal. Weak ranged attack

fighter - scizor - Able to move twice, attack twice. Strong melee attack