High concept Statement

A turn-based strategy game that involves two players who take turns trying to kill each other on a procedurally generated world.

**Game rules**

* Both players will start on the middlemost square.
* The game will take place on a 3\*3 board
* Players will take turns and make 2 moves per turn.
* A Space on the board can be captured b a player but costs 3 moves, and only that player can step on that space.
* If no item, buff or player is on a square the square will become inactive and no player can step on the space for 2 turns and will rotate to a become a new space.
* Spaces will also have effects like range increases where other spaces are affected by up to 2 spaces away
* Players can acquire the following Buffs which will temporary increase certain stats
  + HP bonus
  + Def bonus
  + Atk Bonus
* Player can also acquire the same Bonuses as the buffs, from item but become permanent but only one can be held at a time

**Players Moves**

* Players can move 1 space in any direction.
* If both players are on the same space, A player can attack another player.
* Any player can heal 10% of their HP for one move per turn
* Players can take or leave an item or buff on any space

**Game state representation**

Any game state will be represented with a 2D array of a custom script tiles and players positions, the tiles array will have information on the different tiles and will record if the tile is active, cooldown, captured or has a player on it, they will also have information on what item, buff or effect they have and the next rotation of item, buff or effect.

Along with the board representation the players information will also recorded with the current stats, HP, Def, and Atk, they will record their buffs and durations, and which item they have on them.

Both representations will be stored inside its own object of representation

An example will look like this

**Board representation**

Object boardRep

[0,0] [0,1] [0,2]

[1,0] [1,1] [1,2]

[2,0] [2,1] [2,2]

One of these spaces will look like this with the following information inside

Object Tile: null [0,0]

* Bool Active: true
* Bool Cooldown: false
* Int Duration: null
* Bool Captured: false
* Bool Player: true
* object [] Spot:
  + Buff: Atk
  + Item: null
  + Effect: null
* objects [] next:
  + Buff: null
  + Item: sword
  + Effect: null

The players will be stored in their own array with the following information for both players

[0]

Object playerRep

Int posX: 0

Int posY: 2

Int Hp: 10

Int Def: 0

Int Atk: 1

String item: sword

Object buff

String Buff: atk

Int Duration: 2

**Game state Utility function description**