BaseAI
The BaseAI class contains all of the basic functions for an AI, as well as the lists of all of the game objects. The AI class inherits all of the BaseAI functions and lists of game objects.

Bases
A base is the source of your virus production. Each turn, you can command each of your bases to spawn a virus. Spawned viruses spawn on top of the base. If any virus tries to return any base, they are "recycled", and die.

Spawn
Spawn is a function of a base. To spawn a virus, a player must have enough cycles to afford a virus of that level. A base can only spawn one virus per turn, and it cannot spawn a virus if there is already a virus on the base.

Virus Cost
To spawn a virus, you must spend cycles. The cost uses the following function:

$$\lfloor baseCost \cdot scaleCost^{\,level} \rfloor$$

baseCost and *scaleCost* are functions in the BaseAI class, which the AI class inherits, and the level is the level of the virus. As of writing this, baseCost is 10 and scaleCost is 1.8. This means that a virus of level 0 costs 10 cycles, level 1 costs 18 cycles, level 2 costs 32 cycles.

Players
The Player object stores all of the general information about a player, such as their team name, their player ID, etc.

Cycles
At the beginning of each turn, your are awarded cycles using the following function:

$$\lceil (possible - worth) \cdot returnAmount + cyclesPerTurn \rceil$$

Where *possible* is the maximum amount you could have this turn, *worth* is your current team worth, *returnAmount* is 10%, and *cyclesPerTurn* is 4. *worth* is calculated as the sum of the cost of all of your living units + your current number of cycles. *possible* is equal to:

$$(\frac{turnNumber}{2} * self.cyclesPerTurn) + startingCycles$$

startingCycles is 20. This function is designed to give you a set amount per turn, as well as refund lost cycles over time.

Tiles
The map is represented as a list of tiles. Each tile has an owner, which can be:
0: Owned by the player with ID 0 (first player)
1: Owned by the player with ID 1 (second player)
2: Neither player owns this tile
3: This tile is blocked. No virus can enter (wall)

To gain control of a tile, move a virus onto that tile. You will maintain control of that tile until a virus of your opponent walks onto that tile.

Maps
Maps are procedurally generated and stored as files. These maps are then played back at random. All maps are symmetric, contain 3 bases for each team, and contain walls. All spaces on the map are reachable from at least one of your bases.

ByteDollars
ByteDollars are your score. Whichever player gets more ByteDollars, wins. On every even numbered turn, a ByteDollar is awarded to the team with the largest area of fully connected territory. What we mean by connected is that starting at a tile you control, how many tiles can you get to walking only on tiles you control. Whichever team has the largest connected section gets 1 ByteDollar.

Viruses
A virus is your unit to move. Viruses have levels, locations, owners, etc. Viruses are spawned at bases, and can be moved 1 space at a time to adjacent spaces. Adjacent is defined as up, down, left, and right. Not diagonal. If a virus attempts to move onto a base, it will die, freeing up its cycles to be refunded. When a virus moves to a tile, it captures that tile. Viruses cannot occupy the same space. As such, virus collisions are handled as follows:

- If two viruses of the same team collide
    - If they are the same level, they will combine, making a single virus of 1 higher level. Ex: Two level 1's collide, make a 2. Two level 2's collide, make a 3
        - The moving virus is destroyed, the stationary levels up and is frozen until the end of your turn
    - If they are different levels, they block movement
- If two viruses of different teams collide
    - Same level, both die
    - Moving level > Stationary level, stationary dies, moving takes its place
    - Moving level < Stationary level, moving dies

Win Conditions
All games end after a set number of turns (Currently 500 total, 250 per player). Whichever player has the most ByteDollars wins. If players are tied in ByteDollars, whichever team has a higher worth (cycles plus sum of living unit worth) wins. If players are STILL tied, first player is declared the winner. This prevents mirrored play.

If a player has mathematically won (IE, have a lead larger than the number of ByteDollars left to be awarded), the game ends early and declares the winner.