# Central

* Handles what happens on every frame, passes functionality to other handlers (CameraController, Teams etc).
* Need a way to handle sounds, whether that’s a separate class or not I’m not sure.
* Also need handle physics engines, a lot of things can call this. At least a reference to it.

# MapData

* Holds data in each node in the map (defined by an image file).
* Should provide capability for pathfinding including finding the adjacent nodes and calculating the heuristic.
* When querying node data, should return enumerations or sets of.
* Should it return a struct containing relevant information (return by value).
* When getting a point, just take two arguments for x and y coordinates (integers)
* Probably owned by our Central class

# PathFinder

* Needs a reference to MapData to get nodes and adjacent nodes.
* Using an A\* variant
* Find path needs a to and from point. Not really much else. Using integer arguments.

# MapAssets

* Handles the graphical and rendering aspects of the map. Includes the cubes, final model, skybox etc.
  + Generates cubs for walls etc.
  + Plane for ground being used for Ray Query.

# Map

* Contains a MapAssets, MapData and PathFinder to simplify communication.
* Will handle the conversion from real world to grid and vice versa for all the other tools.
* Will include methods for interface such as findpath etc.

# CameraController

* Move when reach edge of screen
* TOP DOWN (90 degrees)
* Zoom in and out

# Team

* **MAYBE WE DON’T NEED A TEAM. MERGE WITH CENTRAL?**
* Should handle the creation and deletion of tanks. Uses a vector of pointers to tanks to handle the deletion.
* Handles score for each team.
* List of units for the team.

# Unit

* Has an isDead state to see if they are dead. If so, other tanks can ignore them.
* Should have a state machine in order to make informed decisions (whether we have multiple is up to implementation).
  + Could be a separate class, based on implementation.
* Need to shoot projectiles
* Stores what team the unit is on.
* Unlimited ammunition, too difficult otherwise.
* Randomly generated names? Yes please!

# Projectile

* Separate to Unit
* Need to decide how it is meant to blow up, but units request a creation instead of creating it themselves.

# UI Class

* Used just to display the user interface such as score and all of that.