

## Design Documentation

In this iteration:

Firstly, Arena Viewer is used to create the graphics window and GUI. It is controlling the timing of the screen update by `UpdateSimulation`; performing the drawing by some `DrawObstacle\DrawHomeBase\DrawRobot`, watching for key and mouse events by `On<key/mouse>Event()`, and it is the keeper of the arena. In the key/mouse event, Up arrow increases speed, up to a defined limit; Down arrow decreases speed, down to 0 (no negative speeds); Right arrow changes the heading clockwise by some delta of your choosing; Left arrow changes the heading counter-clockwise by some defined delta. In the GUI, you can pause and restart this graphics window.

Secondly, Arena contains all entities such as Robot, Recharge station, Home base, Circular entities (i.e. Obstacles), and constructor create all of them, then call `AdvanceTime(dt)` to make things happen. This sends message to `UpdateEntitiesTimeStep()`, which in turn for each entity calls `entity->TimeStepUpdate(1)`. After all entities update for the 1 time step, checks for events: robot at charging station; robot at home base; robot has collision with walls and obstacles, after collision, robot reduces velocity and depletes battery; home base has collision with walls and obstacles; Robot changes color as their battery depletion.

Thirdly, for those entities in the Arena, they can be divided into two kinds. One is `MobileArenaEntity`, another is `ImmobileAernaEntity`. Each entities generally have color, position, size, heading and velocity for mobile entity, `is_mobile` return True or False, `reset()`, `TimeStepUpdate`.

(1) Charging station is a `ImmobileArenaEntity`, which has the ability to recharge the robot battery. The robot need only collide with it to get a full charge to its battery.

(2) Circular entity (i.e. Obstacle) also is a `ImmobileArenaEntity`, which get in the way of the robot, and deplete battery and velocity when robot has collision with it.

(3) Home base is a MobileArenaEntity, which is the robot needs to get to to win. When robot has collision with Home base, you win the game. This one is moving, and better it has randomly change directions at unpredictable times to make the game harder. It also will have collision with walls and obstacles, the behavior is similar to the robot as explained below.

(4) Robot, the primary in this arena, is a MobileArenaEntity. Robot has lots of members. For instance,

(I) Motion Handler for mobile entities, which This manages the setting of velocity for the robot (or for any mobile arena entity). Velocity (i.e. heading and speed) are initialized at the start, but will change when either the user inputs a command or the robot collides with something, which will change the direction of the robot.

(II) Motion Behavior for mobile entities, which is the crucial update method for mobile arena objects. It uses the velocity of each mobile entity to calculate a new x,y position using the setter for "this". In iteration 1, the 2 wheels will always have equal velocity, thus the robot will always move in a straight line in the direction of its heading angle.

(III) Robot Battery is depleted as the robot moves, with extra depletion when running into things. You lose when the robot runs out of battery. It is recharged at the charging station with related methods are event\recharge() and deplete().

(IV) Touch Sensor, this one is inherited from sensor base, activate or signal when touched or bumped. Any time the robot collides with something (as determined by the Arena), an event is sent to the sensor to set its activation. Conversely, when collision is no longer detected, an event is sent to the sensor to deactivate it.

reference: ProjectClassOverview.md