R. Jacob Schonthaler

schon099

**Design Document**

1. **Overview**

This program is one that simulates entities existing in an arena as they interact with each other. The program contains immobile and mobile entities, and the mobile entities can collide with each other and the obstacles. The immobile entities bounce mobile entities off themselves, and the recharge station recharges the player as well if the player collides with it. The mobile entities bounce off each other upon collision, and they also bounce of the walls of the arena upon collision. The special events that happen in the arena with mobile entities are as follows. The player freezes robots upon colliding with them, the player freezes if a superbot collides with it, and a robot becomes a superbot upon colliding with the homebase. The program terminates with a loss if the player losses all its charge, all the robots become superbots, or all remaining robots are frozen by the player.

1. **Class Descriptions**
   1. **Arena**
      1. **Inheritance: N/A**
      2. **Variables**
         1. **player\_**

Accessibility: private

Type: Player\*

Description: A mobile entity controlled by the player.

* + - 1. **recharge\_station\_**

Accessibility: private

Type: RechargeStation\*

Description: An immobile entity which gives player\_ a full charge upon it colliding with recharge\_station\_.

* + - 1. **home\_base\_**

Accessibility: private

Type: HomeBase\*

Description: A mobile entity that turns robots into superbots upon colliding with robots.

* + - 1. **x\_dim\_**

Accessibility: private

Type: double

Description: The max x value in the arena.

* + - 1. **y\_dim\_**

Accessibility: private

Type: double

Description: The max y value in the arena.

* + - 1. **n\_robots\_**

Accessibility: private

Type: unsigned int

Description: The number of robots and superbots in the arena.

* + - 1. **n\_obstacles\_**

Accessibility: private

Type: unsigned int

Description: The number of normal obstacles in the arena

* + - 1. **entities\_**

Accessibility: private

Type: vector<class ArenaEntity\*>

Description: A vector containing all the entities in the arena

* + - 1. **mobile\_entities\_**

Accessibility: private

Type: vector<class ArenaMobileEntity\*>

Description: A vector containing all the mobile entities in the arena.

* + - 1. **game\_over\_**

Accessibility: private

Type: bool

Description: A flag to tell the program that the game is over.

* + 1. **Functions**
       1. **Arena**

Accessibility: public

Return Type: N/A

Description: The constructor to build the arena

Parameters:

* + - * 1. params

Type: const struct arena\_params\* const

Description: A struct containing all the info needed to create the arena and all the entities in it.

* + - 1. **AdvanceTime**

Accessibility: public

Return Type: void

Description: Updates all the entities in the arena

Parameters:

* + - * 1. dt

Type: unsigned int

Description: The amount each entity update is done by

* + - 1. **Accept**

Accessibility: public

Return Type: void

Description: Forwards keypress event to player\_.

Parameters:

* + - * 1. e

Type: EventKeyPress\*

Description: An event containing info on which key was pressed.

* + - 1. **Reset**

Accessibility: public

Return Type: void

Description: Resets the arena and all the entities in it.

Parameters: N/A

* + - 1. **n\_robots**

Accessibility: public

Return Type: unsigned int

Description: Getter for n\_robots\_.

Parameters: N/A

* + - 1. **n\_obstacles**

Accessibility: public

Return Type: unsigned int

Description: Getter for n\_obstacles\_.

Parameters: N/A

* + - 1. **obstacles**

Accessibility: public

Return Type: vector<class Obstacle\*>

Description: Gets all the obstacles in the arena.

Parameters: N/A

* + - 1. **mobile\_entities\_**

Accessibility: public

Return Type: vector<class ArenaMobileEntity\*>

Description: Gets all the mobile entities in the arena

Parameters: N/A

* + - 1. **player**

Accessibility: public

Return Type: Player\*

Description: Getter for player\_.

Parameters: N/A

* + - 1. **home\_base**

Accessibility: public

Return Type: HomeBase\*

Description: Getter for home\_base\_.

Parameters: N/A

* + - 1. **recharge\_station**

Accessibility: public

Return Type: RechargeStation\*

Description: Getter for recharge\_station\_.

Parameters: N/A

* + - 1. **set\_gameover**

Accessibility: public

Return Type: void

Description: Setter for game\_over\_.

Parameters:

* + - * 1. change

Type: bool

Description: The new value for game\_over\_.

* + - 1. **get\_gameover**

Accessibility: public

Return Type: bool

Description: Getter for game\_over\_.

Parameters: N/A

* + - 1. **CheckForEntityCollision**

Accessibility: private

Return Type: void

Description: Determines if two entities have collided.

Parameters:

* + - * 1. ent1

Type: const class ArenaEntity\* const

Description: One of the entities to be checked.

* + - * 1. ent2

Type: const class ArenaEntity\* const

Description: The other entity to be checked.

* + - * 1. ec

Type: EventCollision\*

Description: The collision event for the two entities

* + - * 1. collision\_delta

Type: double

Description: An extra bit for the distance between the two entities when determining if a collision has happened.

* + - 1. **CheckForEntityOutOfBounds**

Accessibility: private

Return Type: void

Description: Determines if a mobile entity has hit a wall

Parameters:

* + - * 1. ent

Type: const class ArenaMobileEntity\* const

Description: The entity being checked.

* + - * 1. ec

Type: EventCollision\*

Description: The collision event for this entity

* + - 1. **UpdateEntitiesTimeStep**

Accessibility: private

Return Type: void

Description: Updates all the entities in the arena.

Parameters: N/A

* + - 1. **operator**

Accessibility: private

Return Type: Arena&

Description: I don’t know???

Parameters:

* + - * 1. other

Type: const Arena&

Description: I don’t know???

* + - 1. **Arena**

Accessibility: private

Return Type: N/A

Description: I don’t know??

Parameters:

* + - * 1. other

Type: const Arena&

Description: I don’t know???

* 1. **GraphicsArenaViewer**
     1. **Inheritance: N/A**
     2. **Variables**
        1. **arena\_**

Accessibility: private

Type: Arena\*

Description: The arena in the program

* + - 1. **paused\_**

Accessibility: private

Type: bool

Description: Paused flag for arena updating.

* + - 1. **pause\_btn\_**

Accessibility: private

Type: nanogui::Button\*

Description: The pause button to set the paused flag.

* + 1. **Functions**
       1. **GraphicsArenaViewer**

Accessibility: public

Return Type: explicit

Description: Constructor

Parameters:

* + - * 1. params

Type: const struct arena\_params\* const

Description: The parameters to create arena\_.

* + - 1. **~GraphicsArenaViewer**

Accessibility: public

Return Type: virtual

Description: I don’t know???

Parameters: N/A

* + - 1. **UpdateSimulation**

Accessibility: public

Return Type: void

Description: Updates the arena

Parameters:

* + - * 1. dt

Type: double

Description: The amount the arena is updated by

* + - 1. **OnRestartBtnPressed**

Accessibility: public

Return Type: void

Description: Resets the arena upon the restart button be clicked

Parameters: N/A

* + - 1. **OnPauseBtnPressed**

Accessibility: public

Return Type: void

Description: Sets the pause flag paused\_

Parameters: N/A

* + - 1. **OnMouseMove**

Accessibility: public

Return Type: void

Description: Gets the x and y coordinates of the mouse when it hovers over the arena.

Parameters:

* + - * 1. x

Type: int

Description: The x coordinate of the mouse

* + - * 1. y

Type: int

Description: The y coordinate of the mouse

* + - 1. **OnLeftMouseDown**

Accessibility: public

Return Type: void

Description: Determines if the left mouse button has been pressed over the arena

Parameters:

* + - * 1. x

Type: int

Description: The x coordinate of the mouse

* + - * 1. y

Type: int

Description: The y coordinate of the mouse

* + - 1. **OnLeftMouseUp**

Accessibility: public

Return Type: void

Description: Determines when the left mouse button has been let go over the arena

Parameters:

* + - * 1. x

Type: int

Description: The x coordinate of the mouse

* + - * 1. y

Type: int

Description: The y coordinate of the mouse

* + - 1. **OnRightMouseDown**

Accessibility: public

Return Type: void

Description: Determines if the right mouse button has been pressed over the arena

Parameters:

* + - * 1. x

Type: int

Description: The x coordinate of the mouse

* + - * 1. y

Type: int

Description: The y coordinate of the mouse

* + - 1. **OnRightMouseUp**

Accessibility: public

Return Type: void

Description: Determines if the right mouse button has been let go over the arena

Parameters:

* + - * 1. x

Type: int

Description: The x coordinate of the mouse

* + - * 1. y

Type: int

Description: The y coordinate of the mouse

* + - 1. **OnKeyDown**

Accessibility: public

Return Type: void

Description: Determines if a key has been pressed

Parameters:

* + - * 1. c

Type: const char\*

Description: The character for the key pressed

* + - * 1. modifiers

Type: int

Description: The keycode for the key pressed

* + - 1. **OnKeyUp**

Accessibility: public

Return Type: void

Description: Determines when a key has been let go

Parameters:

* + - * 1. c

Type: const char\*

Description: The character for the key pressed

* + - * 1. modifiers

Type: int

Description: The keycode for the key pressed

* + - 1. **OnSpecialKeyDown**

Accessibility: public

Return Type: void

Description: Determines if a key that affects the arena has been pressed

Parameters:

* + - * 1. key

Type: int

Description: The key number pressed

* + - * 1. scancode

Type: int

Description: The special number for the key pressed in this program

* + - * 1. modifiers

Type: int

Description: The keycode for the key pressed

* + - 1. **OnSpecialKeyUp**

Accessibility: public

Return Type: void

Description: Determines if a key that affects the arena has been let go

Parameters:

* + - * 1. key

Type: int

Description: The key number released

* + - * 1. scancode

Type: int

Description: The special number for the key released in the program

* + - * 1. modifiers

Type: int

Description: The keycode for the key pressed

* + - 1. **DrawUsingNanoVG**

Accessibility: public

Return Type: void

Description: Draws the arena

Parameters:

* + - * 1. ctx

Type: NVGcontext\*

Description: I don’t know???

* + - 1. **DrawUsingOpenGL**

Accessibility: public

Return Type: void

Description: I don’t know

Parameters: N/A

* + - 1. **get\_arena**

Accessibility: public

Return Type: Arena\*

Description: Getter for arena\_

Parameters: N/A

* + - 1. **DrawRobot**

Accessibility: private

Return Type: void

Description: Draws a robot

Parameters:

* + - * 1. ctx

Type: NVGcontext\*

Description: I don’t know???

* + - * 1. robot

Type: const class Robot\* const

Description: The robot being drawn

* + - 1. **DrawObstacle**

Accessibility: private

Return Type: void

Description: Draws an obstacle

Parameters:

* + - * 1. ctx

Type: NVGcontext\*

Description: I don’t know???

* + - * 1. obstacle

Type: const class Obstacle\* const

Description: The obstacle being drawn

* + - 1. **DrawHomeBase**

Accessibility: private

Return Type: void

Description: Draws the home base

Parameters:

* + - * 1. ctx

Type: NVGcontext\*

Description: I don’t know

* + - * 1. home

Type: const class HomeBase\* const

Description: The home base being drawn

* + - 1. **operator**

Accessibility: private

Return Type: GraphicsArenaViewer&

Description: I don’t know???

Parameters:

* + - * 1. other

Type: const GraphicsArenaViewer&

Description: I don’t know???

* + - 1. **GraphicsArenaViewer**

Accessibility: private

Return Type: N/A

Description: I don’t know???

Parameters:

* + - * 1. other

Type: const GraphicsArenaViewer&

Description: I don’t know???

* 1. **Position**
     1. **Inheritance: N/A**
     2. **Variables**
        1. **x**

Accessibility: public

Type: int

Description: The x coordinate of the position

* + - 1. **y**

Accessibility: public

Type: int

Description: The y coordinate of the position

* + 1. **Functions**
       1. **Position**

Accessibility: public

Return Type: N/A

Description: Constructor

Parameters: N/A

* + - 1. **Position**

Accessibility: public

Return Type: N/A

Description: Constructor with parameters for x and y

Parameters:

* + - * 1. in\_x

Type: int

Description: The value for x

* + - * 1. in\_y

Type: int

Description: The value for y

* 1. **Color**
     1. **Inheritance: N/A**
     2. **Variables**
        1. **r**

Accessibility: public

Type: int

Description: The red value for the color

* + - 1. **g**

Accessibility: public

Type: int

Description: The green value for the color

* + - 1. **b**

Accessibility: public

Type: int

Description: The blue value for the color

* + - 1. **a**

Accessibility: public

Type: int

Description: The opaqueness for the color

* + 1. **Functions**
       1. **Color**

Accessibility: public

Return Type: N/A

Description: Constructor

Parameters: N/A

* + - 1. **Color**

Accessibility: public

Return Type: N/A

Description: Constructor with parameters

Parameters:

* + - * 1. in\_r

Type: int

Description: The value for r

* + - * 1. in\_g

Type: int

Description: The value for g

* + - * 1. in\_b

Type: int

Description: The value for b

* + - * 1. in\_a

Type: int

Description: The value for a

* 1. **arena\_params**
     1. **Inheritance: N/A**
     2. **Variables**
        1. **player**

Accessibility: public

Type: struct player\_params

Description: The parameters for the player in the arena

* + - 1. **recharge\_station**

Accessibility: public

Type: struct ArenaEntityParams

Description: The parameters for the recharge station in the arena

* + - 1. **home\_base**

Accessibility: public

Type: struct home\_base\_params

Description: The parameters for the home base in the arena

* + - 1. **obstacles[MAX\_OBSTACLES]**

Accessibility: public

Type: struct arena\_entity\_params

Description: An array with the parameters for all the obstacles

* + - 1. **n\_obstacles**

Accessibility: public

Type: size\_t

Description: The number of obstacles int the arena

* + - 1. **x\_dim**

Accessibility: public

Type: uint

Description: The max x value in the arena

* + - 1. **y\_dim**

Accessibility: public

Type: uint

Description: The max y value in the arena

* + 1. **Functions: N/A**
  1. **ArenaEntityParams**
     1. **Inheritance: N/A**
     2. **Variables**
        1. **radius**

Accessibility: public

Type: double

Description: The radius of the entity

* + - 1. **pos**

Accessibility: public

Type: Type: Position

Description: The position of the entity

* + - 1. **color**

Accessibility: public

Type: Color

Description: The color of the entity

* + 1. **Functions**
       1. **arena\_entity\_params**

Accessibility: public

Return Type: N/A

Description: Constructor

Parameters: N/A

* 1. **home\_base\_params**
     1. **Inheritance: ArenaEntityParams**
     2. **Variables**
        1. **collision\_delta\_**

Accessibility: private

Type: double

Description: The extra cushion for collisions to make radius bigger

* + 1. **Functions**
       1. **home\_base\_params**

Accessibility: public

Return Type: N/A

Description: Constructor

* 1. **arena\_mobile\_entity\_params**
     1. **Inheritance: ArenaEntityParams**
     2. **Variables**
        1. **collision\_delta\_**

Accessibility: private

Type: double

Description: The extra cushion for collisions to make radius bigger

* + 1. **Functions**
       1. **arena\_mobile\_entity\_params**

Accessibility: public

Type: N/A

Description: Constructor

Parameters: N/A

* 1. **player\_params**
     1. **Inheritance: arena\_mobile\_entity\_params**
     2. **Variables**
        1. **battery\_max\_charge**

Accessibility: public

Type: double

Description: The max charge that the battery can hold

* + - 1. **angle\_delta**

Accessibility: public

Type: unsigned int

Description: The amount that the player’s heading changes when the arrow keys are pressed

* + 1. **Functions**
       1. **player\_params**

Accessibility: public

Return Type: N/A

Description: Constructor

Parameters: N/A

* 1. **robot\_params**
     1. **Inheritance: arena\_mobile\_entity\_params**
     2. **Variables**
        1. **angle\_delta**

Accessibility: public

Type: unsigned int

Description: Not in use

* + 1. **Functions**
       1. **robot\_params**

Accessibility: public

Return Type: N/A

Description: Constructor

Parameters: N/A

* 1. **ArenaEntity**
     1. **Inheritance: N/A**
     2. **Variable**
        1. **radius\_**

Accessibility: private

Type: double

Description: The radius of the entity

* + - 1. **pos\_**

Accessibility: private

Type: Position

Description: The position of the entity in the arena

* + - 1. **color\_**

Accessibility: private

Type: Color

Description: The color value combination for the entity

* + 1. **Functions**
       1. **ArenaEntity**

Accessibility: public

Return Type: virtual void

Description: Constructor with parameters

Parameters:

* + - * 1. radius

Type: double

Description: The value assigned to radius\_

* + - * 1. pos

Type: const Position&

Description: The position assigned to pos\_

* + - * 1. color

Type: Color&

Description: The color values assigned to color\_

* + - 1. **ArenaEntity**

Accessibility: public

Return Type: virtual

Description: Default Constructor

Parameters: N/A

* + - 1. **TimestepUpdate**

Accessibility: public

Return Type: virtual void

Description: Updates the entity

Parameters:

* + - * 1. dt

Type: \_unused uint

Description: The amount the entity is updated by

* + - 1. **Reset**

Accessibility: public

Return Type: virtual void

Description: Resets the entity

Parameters: N/A

* + - 1. **get\_name**

Accessibility: public

Return Type: virtual void const=0

Description: Returns the name of the entity

Parameters: N/A

* + - 1. **is\_mobile**

Accessibility: public

Return Type: virtual bool = 0

Description: A flag that determines if the entity is mobile

Parameters: N/A

* + - 1. **set\_pos**

Accessibility: public

Return Type: void

Description: Setter for pos\_

Parameters:

* + - * 1. pos

Type: const Position&

Description: The position assigned to pos\_

* + - 1. **get\_pos**

Accessibility: public

Return Type: const Position&

Description: Getter for pos\_

Parameters: N/A

* + - 1. **get\_color**

Accessibility: public

Return Type: cons Color&

Description: Getter for color\_

Parameters: N/A

* + - 1. **set\_color**

Accessibility: public

Return Type: void

Description: Setter for color\_

Parameters:

* + - * 1. color

Type: const Color&

Description: Values assigned to color\_

* + - 1. **get\_radius**

Accessibility: public

Return Type: double

Description: Getter for radius\_

Parameters: N/A

* 1. **ArenaImmobileEntity**
     1. **Inheritance: ArenaEntity**
     2. **Variables N/A**
     3. **Functions**
        1. **ArenaImmobileEntity**

Accessibility: N/A

Return Type: N/A

Description: Constructor

Parameters:

* + - * 1. radius

Type: double

Description: Value assigned to radius\_

* + - * 1. pos

Type: const Position&

Description: Value assigned to pos\_

* + - * 1. color

Type const Color&

Description: Values assigned to color\_

* + - 1. **is\_mobile**

Accessibility: public

Return Type: bool

Description: Always returns false

Parameters: N/A

* 1. **Obstacle**
     1. **Inheritance: ArenaImmobileEntity**
     2. **Variables**
        1. **next\_id**

Accessibility: private

Type: static uint

Description: The id\_ for the next obstacle created

* + - 1. **id\_**

Accessibility: private

Type: int

Description: The id for the current obstacle

* + 1. **Functions**
       1. **Obstacle**

Accessibility: public

Return Type: N/A

Description: Constructor

Parameters:

* + - * 1. radius

Type: double

Description: Value assigned to radius\_

* + - * 1. pos

Type: const Position&

Description: Value assigned to pos\_

* + - * 1. color

Type: const Color&

Description: Values assigned to color\_

* + - 1. **get\_name**

Accessibility: public

Return Type: string const

Description: The string Obstacle with the id appended is returned

Parameters: N/A

* 1. **RechargeStation**
     1. **Inheritance: Obstacle**
     2. **Variables: N/A**
     3. **Functions**
        1. **RechargeStation**

Accessibility: public

Return Type: N/A

Description: Construction

Parameters:

* + - * 1. radius

Type: double

Description: Value assigned to radius\_

* + - * 1. pos

Type: const Position&

Description: Value assigned to pos\_

* + - * 1. color

Type: const Color&

Description: Values assigned to color\_

* + - 1. **get\_name**

Accessibility: public

Return Type: const string

Description: The string RechargeStation with id\_ appended is returned

* 1. **EventBaseClass**
     1. **Inheritance: N/A**
     2. **Variable: N/A**
     3. **Functions**
        1. **EventBaseClass**

Accessibility: public

Return Type: N/A

Description: Constructor

Parameters: N/A

* + - 1. **~EventBaseClass**

Accessibilty: public

Return Type: virtual

Parameters: N/A

* + - 1. **EmitMessage**

Accessibility: public

Return Type: virtual = 0

Parameters: N/A

* 1. **EventCollision**
     1. **Inheritance: EventBaseClass**
     2. **Variables**
        1. **collided**

Accessibility: private

Type: bool

Description: Flag to determine if a collision occurred

* + - 1. **point\_of\_contact\_**

Accessibility: private

Type: Position

Description: The point at which the collision occurred

* + - 1. **angle\_of\_contact\_**

Accessibility: private

Type: double

Description: The heading to the entity being collided with

* + 1. **Functions**
       1. **EventCollision**

Accessibility: public

Return Type: N/A

Description: Constructor

Parameters: N/A

* + - 1. **EmitMessage**

Accessibility: public

Return Type: void

Description: Prints “Collision” and the name and values for the variables when a collision has occurred

Parameters: N/A

* + - 1. **get\_collided**

Accessibility: public

Return Type: bool

Description: Getter for collided

Parameters: N/A

* + - 1. **set\_collided**

Accessibility: public

Return Type: void

Description: Setter for collided

Parameters:

* + - * 1. c

Type: bool

Description: Value assigned to collided

* + - 1. **get\_point\_of\_contact**

Accessibility: public

Return Type: Position

Description: Getter for point\_of\_contact\_

Parameters: N/A

* + - 1. **set\_point\_of\_contact**

Accessibility: public

Return Type: void

Description: Setter for point\_of\_contact\_

Parameters:

* + - * 1. p

Type: Position

Description: Value assigned to point\_of\_contact\_

* + - 1. **get\_angle\_of\_contact**

Accessibility: public

Return Type: double

Description: Getter for angle\_of\_contact\_

Parameters: N/A

* + - 1. **set\_angle\_of\_contact**

Accessibility: public

Return Type: void

Description: Setter for angle\_of\_contact\_

Parameters:

* + - * 1. aoc

Type: double

Description: Value assigned to angle\_of\_contact\_

* 1. **EventRecharge**
     1. **Inheritance: EventBaseClass**
     2. **Variables: N/A**
     3. **Functions**
        1. **EventRecharge**

Accessibility: public

Return Type: N/A

Description: Constructor

Parameters: N/A

* + - 1. **EmitMessage**

Accessibility: public

Return Type: void

Description: Prints “Event Recharge” when a recharge event occurs

Parameters: N/A

* 1. **event\_commands**
     1. **Values**
        1. **COM\_TURN\_LEFT**

Type: enum

Description: The value for pressing the left arrow key

* + - 1. **COM\_TURN\_RIGHT**

Type: enum

Description: The value for pressing the right arrow key

* + - 1. **COM\_FORWARD**

Type: enum

Description: The value for pressing the up arrow key

* + - 1. **COM\_SLOW**

Type: enum

Description: The value for pressing the down arrow key

* 1. **EventKeyPress**
     1. **Inheritance: EventBaseClass**
     2. **Variables**
        1. **key\_**

Accessibility: private

Type: int

Description: The number for the key pressed

* + 1. **Functions**
       1. **EventKeyPress**

Accessibility: public

Return Type: explicit

Description: Constructor

Parameters:

* + - * 1. key

Type: int

Description: Value assigned to key\_

* + - 1. **EmitMessage**

Accessibility: public

Return Type: void

Description: Prints “Event Key Press”

Parameters: N/A

* + - 1. **get\_key**

Accessibility: public

Return Type: int

Description: Getter for key\_

Parameters: N/A

* + - 1. **get\_command**

Accessibility: public

Return Type: enum event\_commands

Description: I don’t know???

Parameters: N/A

* + - 1. **keypress\_to\_cmd**

Accessibility: private

Return Type: enum event\_commands

Description: Translates key\_ to enum event\_commands

Parameters:

* + - * 1. key

Type: int

Description: Value translated into enum event\_commands

* + - 1. **operator**

Accessibility: private

Return Type: EventKeyPress&

Description: I don’t know???

Parameters:

* + - * 1. other

Type: const EventKeyPress&

Description: I don’t know???

* + - 1. **EventKeyPress**

Accessibility: private

Return Type: N/A

Description: Constructor

Parameters:

* + - * 1. other

Type: const EventKeyPress&

Description: EventKeyPress assigned

* 1. **EventCommand**
     1. **Inheritance: EventBaseClass**
     2. **Variables**
        1. **cmd\_**

Accessibility: private

Type: enum event\_commands

Description: The key pressed

* + 1. **Functions**
       1. **EventCommand**

Accessibility: public

Return Type: explicit

Description: Constructor

Parameters:

* + - * 1. cmd

Type: enum event\_commands

Description: Value assigned to cmd\_

* + - 1. **EmitMessage**

Accessibility: public

Return Type: void

Description: Prints “Command” followed by the value of cmd\_

Parameters: N/A

* + - 1. **get\_cmd**

Accessibility: public

Return Type: enum event\_commands

Description: Getter for cmd\_

Parameters: N/A

* 1. **entity\_types**
     1. **Values**
        1. **kRobot**

Type: enum

Description: The value for a robot

* + - 1. **kSuperBot**

Type: enum

Description: The value for a superbot

* + - 1. **kPlayer**

Type: enum

Description: The value for the player

* + - 1. **kHomeBase**

Type: emit

Description: The value for the home base

* + - 1. **kRechargeStation**

Type: enum

Description: The value for the recharge station

* + - 1. **kObstacle**

Type: enum

Description: The value for an obstacle

* + - 1. **kWall**

Type: enum

Description: The value for a wall

* 1. **EventTypeEmit**
     1. **Inheritance: EventBaseClass**
     2. **Variables**
        1. **entity\_**

Accessibility: private

Type: ArenaEntity\*

Description: The entity being identified

* + 1. **Functions**
       1. **EventTypeEmit**

Accessibility: public

Return Type: N/A

Description: Constructor

Parameters: N/A

* + - 1. **EmitMessage**

Accessibility: public

Return Type: void

Description: Prints “EventTypeEmit”

Parameters: N/A

* + - 1. **get\_entity**

Accessibility: public

Return Type: ArenaEntity\*

Description: Getter for entity\_

Parameters: N/A

* + - 1. **set\_entity**

Accessibility: public

Return Type: void

Description: Setter for entity\_

Parameters:

* + - * 1. e

Type: ArenaEntity\*

Description: Value assigned to entity\_

* 1. **EventProximity**
     1. **Inheritance: EventBaseClass**
     2. **Variables**
        1. **pos\_**

Accessibility: private

Type: Position

Description: The position of the entity being viewed

* + - 1. **radius\_**

Accessibility: private

Type: double

Description: The radius of the entity being viewed

* + 1. **Functions**
       1. **EventProximity**

Accessibility: public

Return Type: N/A

Description: Constructor

Parameters: N/A

* + - 1. **EmitMessage**

Accessibility: public

Return Type: void

Description: Prints “Event Proximity”

Parameters: N/A

* + - 1. **get\_pos**

Accessibility: public

Return Type: Position

Description: Getter for pos\_

Parameters: N/A

* + - 1. **set\_pos**

Accessibility: public

Return Type: void

Description: Setter for pos\_

Parameters:

* + - * 1. p

Type: Position

Description: Value assigned to pos\_

* + - 1. **get\_radius**

Accessibility: public

Return Type: double

Description: Getter for radius\_

Parameters: N/A

* + - 1. **set\_radius**

Accessibility: public

Return Type: void

Description: Setter for radius\_

Parameters:

* + - * 1. r

Type: double

Description: Value assigned to radius\_

* 1. **EventDistressCall**
     1. **Inheritance: EventBaseClass**
     2. **Variables**
        1. **entity\_**

Accessibility: private

Type: ArenaEntity\*

Description: The entity being evaluated

* + 1. **Functions**
       1. **EventDistressCall**

Accessibility: public

Return Type: N/A

Description: Constructor

Parameters: N/A

* + - 1. **EmitMessage**

Accessibility: public

Return Type: void

Description: Prints “Event Distress Call”

Parameters: N/A

* + - 1. **get\_entity**

Accessibility: public

Return Type: ArenaEntity\*

Description: Getter for entity\_

Parameters: N/A

* + - 1. **set\_entity**

Accessibility: public

Return Type: void

Description: Setter for entity\_

Parameters:

* + - * 1. e

Type: ArenaEntity\*

Description: Value assigned to entity\_

* 1. **Sensor**
     1. **Inheritance: N/A**
     2. **Variables**
        1. **activated**

Accessibility: private

Type: bool

Description: Flag to determine if sensor was triggered

* + 1. **Functions**
       1. **get\_activated**

Accessibility: public

Return Type: bool

Description: Getter for activated

Parameters: N/A

* + - 1. **set\_activated**

Accessibility: public

Return Type: void

Description: Setter for activated

Parameters:

* + - * 1. value

Type: bool

Description: Value assigned to activated

* 1. **SensorTouch**
     1. **Inheritance: Sensor**
     2. **Variables**
        1. **point\_of\_contact\_**

Accessibility: private

Type: Position

Description: The point of contact between the two entities

* + - 1. **angle\_of\_contact\_**

Accessibility: private

Type: double

Description: The heading to the obstacle being collided with

* + 1. **Functions**
       1. **SensorTouch**

Accessibility: public

Return Type: N/A

Description: Constructor

Parameters: N/A

* + - 1. **Accept**

Accessibility: public

Return Type: void

Description: Determines if a collision has occurred and updates entity

Parameters:

* + - * 1. e

Type: EventCollision\*

Description: The info for the collision

* + - 1. **Reset**

Accessibility: public

Return Type: void

Description: Resets the sensor

Parameters: N/A

* + - 1. **get\_point\_of\_contact**

Accessibility: public

Return Type: Position

Description: Getter for point\_of\_contact\_

Parameters: N/A

* + - 1. **set\_point\_of\_contact**

Accessibility: public

Return Type: void

Description: Setter for point\_of\_contact\_

Parameters:

* + - * 1. p

Type: Position

Description: Value assigned top point\_of\_contact\_

* + - 1. **get\_angle\_of\_contact**

Accessibility: public

Return Type: double

Description: Getter for angle\_of\_contact

Parameters: N/A

* + - 1. **set\_angle\_of\_contact**

Accessibility: public

Return Type: void

Description: Setter for angle\_of\_contact\_

Parametes:

* + - * 1. aoc

Type: double

Description: Value assigned to angle\_of\_contact\_

* 1. **SensorEntityType**
     1. **Inheritance: Sensor**
     2. **Variables**
        1. **range\_**

Accessibility: private

Type: double

Description: The range of the sensor

* + 1. **Functions**
       1. **SensorEntityType**

Accessibility: public

Return Type: N/A

Description: Constructor

Parameters: N/A

* + - 1. **Accept**

Accessibility: public

Return Type: enum entity\_types

Description: Determines if viewed entity is in range; and if so, what type it is

Parameters:

* + - * 1. e

Type: EventTypeEmit\*

Description: The entity being viewed

* + - * 1. pos

Type: Position

Description: The sensor’s robot’s position

* + - * 1. radius

Type: double

Description: The sensor’s robot’s radius

* + - 1. **Reset**

Accessibility: public

Return Type; void

Description: Resets the sensor

Parameters: N/A

* + - 1. **get\_range**

Accessibility: public

Return Type: double

Description: Getter for range\_

Parameters: N/A

* + - 1. **set\_range**

Accessibility: public

Return Type: void

Description: Setter for range\_

Parameters:

* + - * 1. r

Type: double

Description: Value assigned to range\_

* 1. **SensorProximity**
     1. **Inheritance: Sensor**
     2. **Variables**
        1. **range\_**

Accessibility: private

Type: double

Description: The distance range of the sensor

* + - 1. **field\_of\_view\_**

Accessibility: private

Type: double

Description: The angle range of the sensor

* + 1. **Functions**
       1. **SensorProximity**

Accessibility: public

Return Type: N/A

Description: Constructor

Parameters: N/A

* + - 1. **Accept**

Accessibility: public

Return Type: double

Description: Determines if the entity being viewed is in the range of the sensor; and if so, returns how far away it is.

Parameters:

* + - * 1. e

Type: EventProximity\*

Description: The entity being viewed

* + - * 1. pos

Type: Position

Description: The sensor’s robot’s position

* + - * 1. radius

Type: double

Description: The sensor’s robot’s radius

* + - 1. **Reset**

Accessibility: public

Return Type: void

Description: Resets the sensor

Parameters: N/A

* + - 1. **get\_range**

Accessibility: public

Return Type: double

Description: Getter for range\_

Parameters: N/A

* + - 1. **set\_range**

Accessibility: public

Return Type: void

Description: Setter for range\_

Parameters:

* + - * 1. r

Type: double

Description: Value assigned to range\_

* + - 1. **get\_field\_of\_view**

Accessibility: public

Return Type: double

Description: Getter for field\_of\_view\_

Parameters: N/A

* + - 1. **set\_field\_of\_view**

Accessibility: public

Return Type: void

Description: Setter for field\_of\_view\_

Parameters:

* + - * 1. a

Type: double

Description: Value assigned to field\_of\_view\_

* 1. **SensorDistress**
     1. **Inheritance: Sensor**
     2. **Variables**
        1. **range\_**

Accessibility: private

Type: double

Description: The range of the sensor

* + 1. **Functions**
       1. **SensorDistress**

Accessibilty: public

Return Type: N/A

Description: Constructor

Parameters: N/A

* + - 1. **Accept**

Accessibility: public

Return Type: int

Description: Determines if the entity being viewed is in distress

Parameters:

* + - * 1. e

Type: EventDistressCall\*

Description: The entity being viewed

* + - * 1. pos

Type: Position

Description: The sensor’s robot’s position

* + - * 1. radius

Type: double

Description: The sensor’s robot’s radius

* + - 1. **Reset**

Accessibility: public

Return Type: void

Description: Resets the sensor

Parameters: N/A

* + - 1. **get\_range**

Accessibility: public

Return Type: double

Description: Getter for range\_

Parameters: N/A

* + - 1. **set\_range**

Accessibility: public

Return Type: void

Description: Setter for range\_

Parameters:

* + - * 1. r

Type: double

Description: Value assigned to range\_

* 1. **ArenaMobileEntity**
     1. **Inheritance: ArenaEntity**
     2. **Variables**
        1. **collision\_delta\_**

Accessibility: public

Type: double

Description: The extra bit to add to the radius for determining collisions

* + 1. **Functions**
       1. **ArenaMobileEntity**

Accessibility: public

Return Type: N/A

DDescription: Constructor

Parameters:

* + - * 1. radius

Type: double

Description: Value assigned to radius\_

* + - * 1. collision\_delta

Type: double

Description: Value assigned to collision\_delta

* + - * 1. pos

Type: const Position&

Description: Value assigned to pos\_

* + - * 1. color

Type: Color

Description: Values assigned to color\_

* + - 1. **is\_mobile**

Accessibility: public

Return Type: bool

Description: Always returns true

Parameters: N/A

* + - 1. **heading\_angle**

Accessibility: public

Return Type: virtual double const = 0

Description: Returns the heading angle of the entity

Parameters: N/A

* + - 1. **speed**

Accessibility: public

Return Type: virtual double = 0

Description: Returns the speed of the entity

Parameters: N/A

* + - 1. **TimestepUpdate**

Accessibility: public

Return Type: void

Description: Updates the entity

Parameters:

* + - * 1. dt

Type: uint

Description: The amount the entity is updated by

* + - 1. **Accept**

Accessibility: public

Return Type: virtual void = 0

Description: Determines if a collision has occurred and updates the entity

Parameters:

* + - * 1. e

Type: EventCollision\*

Description: The information for the collision

* + - 1. **Accept**

Accessibility: public

Return Type: virtual void = 0

Description: Determines if the entity becomes recharged

Parameters:

* + - * 1. e

Type: EventRecharge\*

Description: The information if a recharge occurs

* + - 1. **get\_heading\_angle**

Accessibility: public

Return Type: virtual double const = 0

Description: Getter for heading\_angle\_

Parameters: N/A

* + - 1. **set\_heading\_angle**

Accessibility: public

Return Type: virtual void = 0

Description: Setter for heading\_angle\_

Parameters:

* + - * 1. heading\_angle

Type: double

Description: Value assigned to heading\_angle\_

* + - 1. **get\_speed**

Accessibility: public

Return Type: virtual double = 0

Description: Getter for speed\_

Parameters: N/A

* + - 1. **set\_speed**

Accessibility: public

Return Type: virtual void = 0

Description: Setter for speed\_

Parameters:

* + - * 1. sp

Type: double

Description: Value assigned for speed\_

* + - 1. **get\_collision\_delta**

Accessibility: public

Return Type: double

Description: Getter for collision\_delta\_

Parameters: N/A

* 1. **HomeBase**
     1. **Inheritance: ArenaMobileEntity**
     2. **Variables**
        1. **heading\_angle\_**

Accessibility: private

Type: double

Description: The heading of the home base

* + - 1. **collision\_delta\_**

Accessibility: private

Type: double

Description: The extra added to the radius to determine collisions

* + - 1. **speed\_**

Accessibility: private

Type: int

Description: The speed of the home base

* + - 1. **pos\_**

Accessibility: private

Type: Position

Description: The position of the home base in the area

* + - 1. **sensor\_touch\_**

Accessibility: private

Type: SensorTouch

Description: A sensor to handle collision and edit the heading angle

* + 1. **Functions**
       1. **HomeBase**

Accessibility: public

Return Type: N/A

Description: Constructor

Parameters:

* + - * 1. params

Type: const struct home\_base\_params\* const

Description: The info for setting up the home base

* + - 1. **get\_name**

Accessibility: public

Return Type: string

Description: Returns “Home Base”

Parameters: N/A

* + - 1. **TimestepUpdate**

Accessibility: public

Return Type: void

Description: Updates the home base

Parameters: N/A

* + - 1. **Accept**

Accessibility: public

Return Type: void

Description: Determines if collision has occurred and updates the home base if so

Parameters:

* + - * 1. e

Type: EventCollision\*

Description: The info for the collision

* + - 1. **get\_heading\_angle**

Accessibility: public

Return Type: double

Description: Returns the heading of the home base

Parameters: N/A

* + - 1. **set\_heading\_angle**

Accessibility: public

Return Type: void

Description: Sets the heading of the home base

Parameters:

* + - * 1. ha

Type: double

Description: Value set to the home base’s heading

* + - 1. **get\_speed**

Accessibility: public

Return Type: int

Description: Returns the speed of the home base

Parameters: N/A

* + - 1. **set\_speed**

Accessibility: public

Return Type: void

Description: Sets the home base’s speed

Parameters:

* + - * 1. sp

Type: int

Description: Value set to the home base’s speed

* 1. **RobotMotionBehavior**
     1. **Inheritance: N/A**
     2. **Variables: N/A**
     3. **Functions**
        1. **RobotMotionBehavior**

Accessibility: public

Return Type: N/A  
Description: Constructor

Parameters: N/A

* + - 1. **UpdatePosition**

Accessibility: public

Return Type: N/A

Description: Updates the owner entity’s position

Parameters:

* + - * 1. ent

Type: class ArenaMobileEntity\* const

Description: The owner entity

* + - * 1. dt

Type: uint

Description: The amount the entity is updated by

* 1. **MotionHandler**
     1. **Inheritance: N/A**
     2. **Variables**
        1. **heading\_angle\_**

Accessibility: private

Type: double

Description: The heading of the owner entity

* + - 1. **speed\_**

Accessibility: private

Type: double

Description: The speed of the owner entity

* + - 1. **max\_speed\_**

Accessibility: private

Type: double

Description: The max value that speed\_ can achieve

* + 1. **Functions**
       1. **Reset**

Accessibility: void

Return Type: virtual void = 0

Description: Resets the motion handler

Parameters: N/A

* + - 1. **UpdateVelocity**

Accessibility: public

Return Type: void

Description: Updates the velocity of the owner entity

Parameters:

* + - * 1. st

Type: SensorTouch

Description: The owner entity’s touch sensor

* + - 1. **get\_speed**

Accessibility: public

Return Type: double

Description: Getter for speed\_

Parameters: N/A

* + - 1. **set\_speed**

Accessibility: public

Return Type: void

Description: Setter for speed\_

Parameters:

* + - * 1. sp

Type: double

Description: Value assigned to speed\_

* + - 1. **get\_heading\_angle**

Accessibility: public

Return Type: double

Description: Getter for heading\_angle\_

Parameters: N/A

* + - 1. **set\_heading\_angle**

Accessibility: public

Return Type: void

Description: Setter for heading\_angle\_

Parameters:

* + - * 1. ha

Type: double

Description: Value assigned to heading\_angle\_

* + - 1. **get\_max\_speed**

Accessibility: public

Return Type: double

Description: Getter for max\_speed\_

Parameters: N/A

* + - 1. **set\_max\_speed**

Accessibility: public

Return Type: void

Description: Setter for max\_speed\_

Parameters:

* + - * 1. ms

Type: double

Description: Value assigned to max\_speed\_

* 1. **PlayerMotionHandler**
     1. **Inheritance: MotionHandler**
     2. **Variables: N/A**
     3. **Functions**
        1. **PlayerMotionHandler**

Accessibility: public

Return Type: N/A

Description: Constructor

Parameters: N/A

* + - 1. **Reset**

Accessibility: public

Return Type: void

Description: Reset the player motion handler

Parameters: N/A

* + - 1. **AcceptCommand**

Accessibility: public

Return Type: void

Description: Determines if a command was sent; and if so, updates the player motion handler

Parameters:

* + - * 1. cmd

Type: enum event\_commands

Description: The event containing the command info

* 1. **RobotMotionHandler**
     1. **Inheritance: MotionHandler**
     2. **Variables: N/A**
     3. **Functions**
        1. **RobotMotionHandler**

Accessibility: public

Return Type: N/A

Description: Constructor

Parameters: N/A

* + - 1. **Reset**

Accessibility: public

Return Type: void

Description: Resets the robot motion handler

Parameters: N/A

* 1. **RobotBattery**
     1. **Inheritance: N/A**
     2. **Variables**
        1. **kBASE\_DEPLETION**

Accessibility: public

Type: double

Description: Always equals 0.1

* + - 1. **kLINEAR\_SCALE\_FACTOR**

Accessibility: public

Type: double

Description: Always equals 0.001

* + - 1. **kANGLULAR\_SCALE\_FACTOR**

Accessibility: public

Type: double

Description: Always equals 0.001

* + - 1. **charge\_**

Accessibility: private

Type: double

Description: The amount of charge left

* + - 1. **max\_charge\_**

Accessibility: private

Type: double

Description: The max amount charge can achieve

* + 1. **Functions**
       1. **RobotBattery**

Accessibility: public

Return Type: N/A

Description: Constructor

Parameters: N/A

* + - 1. **get\_level**

Accessibility: public

Return Type: double  
Description: Getter for charge\_

Parameters: N/A

* + - 1. **EventRecharge**

Accessibility: public

Return Type: void

Description: Sets the charge to max\_charge\_

Parameters: N/A

* + - 1. **Reset**

Accessibility: public

Return Type: void

Description: Resets the battery

Parameters: N/A

* + - 1. **Deplete**

Accessibility: public

Return Type: double

Description: Depletes charge\_ based upon the player’s speed

Parameters:

* + - * 1. old\_pos

Type: \_unused Position

Description: The player’s previous position

* + - * 1. new\_position

Type: \_unused Position

Description: The player’s new position

* + - * 1. dt

Type: \_unused double

Description: I don’t know???

* + - 1. **Accept**

Accessibility: public

Return Type: void

Description: Determines if loss of 20 charge by collision

Parameters:

* + - * 1. e

Type: EventCollision\*

Description: The collision info

* 1. **Player**
     1. **Inheritance: ArenaMobileEntity**
     2. **Variables**
        1. **next\_id\_**

Accessibility: private

Type: static unsigned int

Description: The id number for the next player to be instantiated

* + - 1. **id\_**

Accessibility: public

Type: int

Description: The player’s unique id

* + - 1. **heading\_angle\_**

Accessibility: private

Type: double

Description: The heading for the player

* + - 1. **angle\_delta\_**

Accessibility: private  
Type: double

Description: The change of heading when the arrow keys are pressed

* + - 1. **battery\_**

Accessibility: private

Type: RobotBattery

Description: The battery for the player

* + - 1. **motion\_handler\_**

Accessibility: private

Type: PlayerMotionHandler

Description: This handles the player’s heading and speed

* + - 1. **motion\_behavior\_**

Accessibility: private

Type: RobotMotionBehavior

Description: This handles the position updating of the player

* + - 1. **sensor\_touch\_**

Accessibility: private

Type: SensorTouch

Description: The sensor that determines if the player experiences a collision

* + 1. **Functions**
       1. **Player**

Accessibility: public

Return Type: explicit

Description: Constructor

Parameters:

* + - * 1. params

Type: const struct player\_params\* const

Description: The info for setting up the player

* + - 1. **ResetBattery**

Accessibility: public

Return Type: void

Description: Resets battery\_

Parameters: N/A

* + - 1. **Reset**

Accessibility: public

Return Type: void

Description: Resets the player

Parameters: N/A

* + - 1. **TimestepUpdate**

Accessibility: public

Return Type: void

Description: Updates the player

Parameters:

* + - * 1. dt

Type: unsigned int

Description: The amount the player is updated by

* + - 1. **Accept**

Accessibility: public

Return Type: void

Description: Determines if the battery receives a full charge

Parameters:

* + - * 1. e

Type: EventRecharge\*

Description: The info for the recharge event

* + - 1. **Accept**

Accessibility: public

Return Type: void

Description: Determines if a collision has occurred

Parameters:

* + - * 1. e

Type: EventCollision\*

Description: The collision info

* + - 1. **EventCmd**

Accessibility: public

Return Type: void

Description: Sends a command event to the motion\_handler\_

Parameters:

* + - * 1. cmd

Type: enum event\_commands

Description: The info on the command

* + - 1. **battery\_level**

Accessibility: public

Return Type: double

Description: Returns the charge of battery\_

Parameters: N/A

* + - 1. **battery\_loss**

Accessibility: public

Return Type: void

Description: Causes a loss of 20 charge to the battery

Parameters: N/A

* + - 1. **get\_heading\_angle**

Accessibility: public

Return Type: double

Description: Returns the player’s heading

Parameters: N/A

* + - 1. **set\_heading\_angle**

Accessibility: public

Return Type: void

Description: Sets the player’s heading

Parameters:

* + - * 1. ha

Type: double

Description: Value assigned to player’s heading

* + - 1. **get\_speed**

Accessibility: public

Return Type: double

Description: Returns the player’s speed

Parameters: N/A

* + - 1. **set\_speed**

Accessibility: public

Return Type: void

Description: Sets the player’s speed

Parameters:

* + - 1. sp

Type: double

Description: Value assigned to player’s speed

* + - 1. **id**

Accessibility: public

Return Type: int

Description: Getter for id\_

Parameters: N/A

* + - 1. **get\_name**

Accessibility: public

Return Type: string

Description: Returns “Player” with id\_ appended

Parameters: N/A

* + - 1. **string\_battery\_level**

Accessibility: public

Return Type: string

Description: Returns the charge of battery\_ as a string

Parameters: N/A

* 1. **Robot**
     1. **Inheritance: ArenaMobileEntity**
     2. **Variables**
        1. **next\_id\_**

Accessibility: private

Type: static unsigned int

Description: The id for the next robot instantiated

* + - 1. **id\_**

Accessibility: private

Type: int

Description: The unique id for the robot

* + - 1. **heading\_angle\_**

Accessibility: private

Type: double

Description: The heading of the robot

* + - 1. **angle\_delta\_**

Accessibility: private

Type: double

Description: The amount of change in heading of the robot

* + - 1. **motion\_handler\_**

Accessibility: private

Type: RobotMotionHandler

Description: This handles the heading and speed of the robot

* + - 1. **motion\_behavior\_**

Accessibility: private

Type: RobotMotionBehavior

Description: This updates the robot’s position

* + - 1. **sensor\_touch\_**

Accessibility: private

Type: SensorTouch’

Description: Determines if a collision has occurred; and if so, changes the robot’s heading

* + - 1. **sensor\_distress\_**

Accessibility: private

Type: SensorDistressCall

Description: Determines if a distress call is sensed

* + - 1. **sensor\_type\_**

Accessibility: private

Type: SensorEntityType

Description: Determines the type of an entity near the robot

* + - 1. **sensor\_proximity\_**

Accessibility: private

Type: SensorProximity

Description: Determines the distance from an entity near the front of the robot

* + 1. **Functions**
       1. **Robot**

Accessibility: public

Return Type: N/A

Description:

1. **UML Class Diagrams**