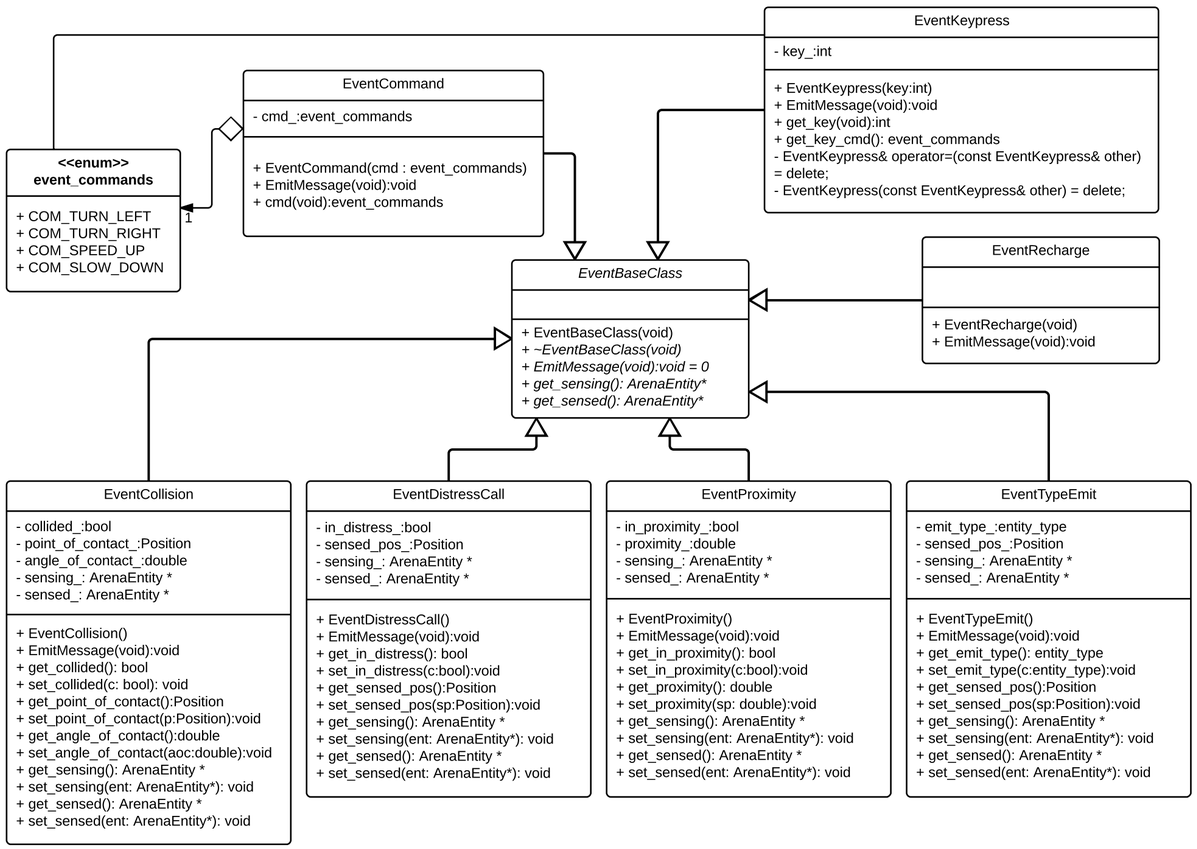
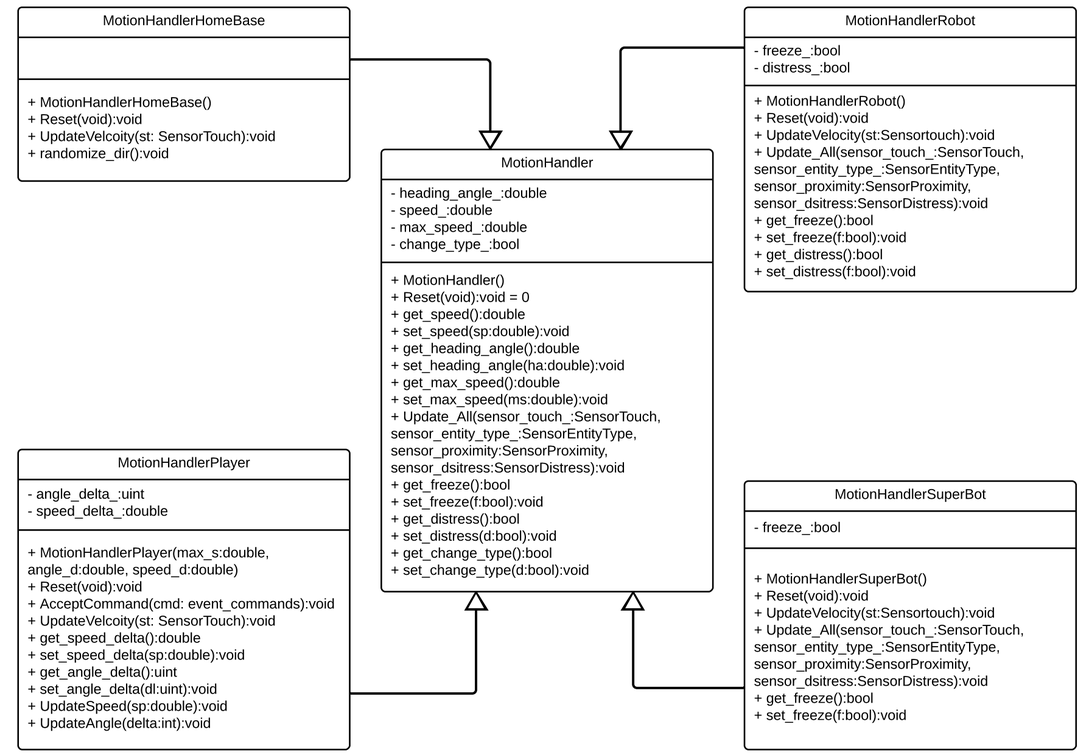
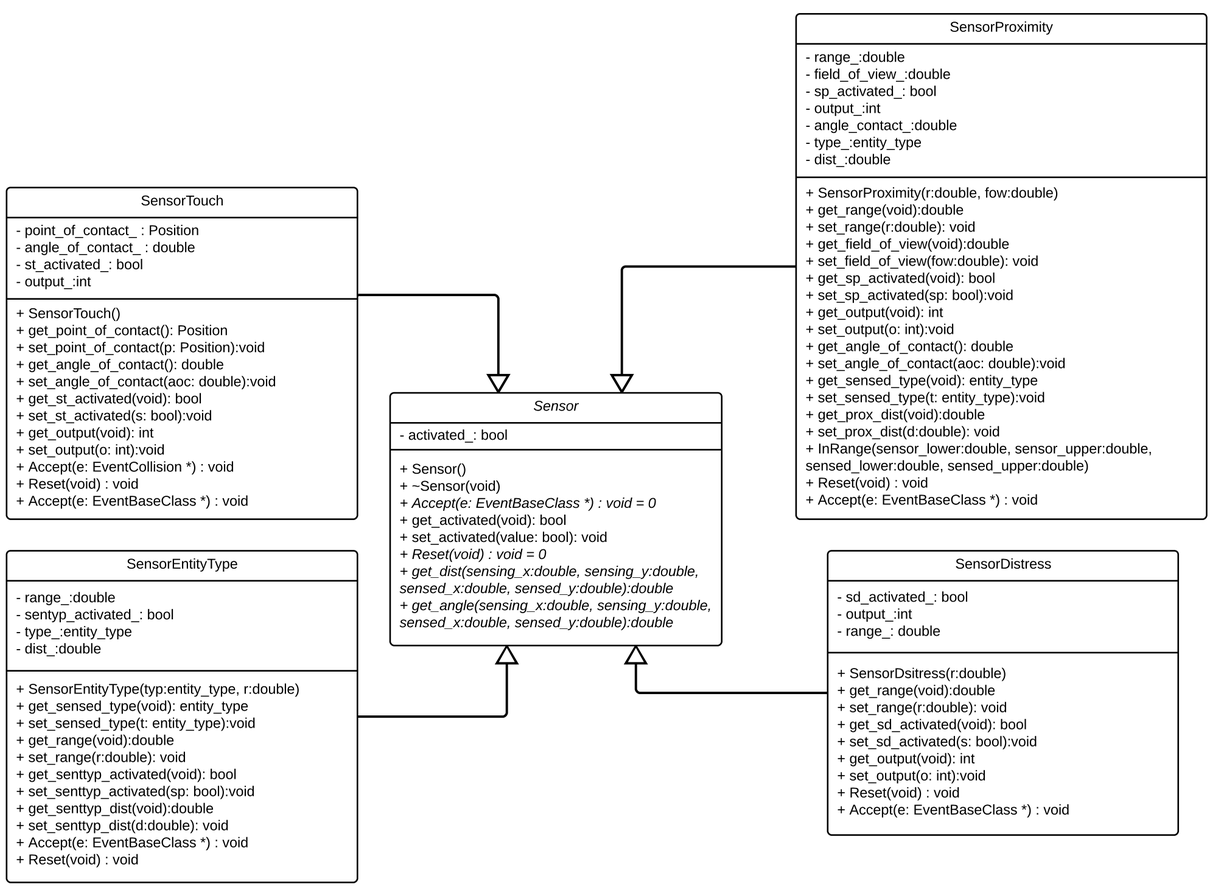
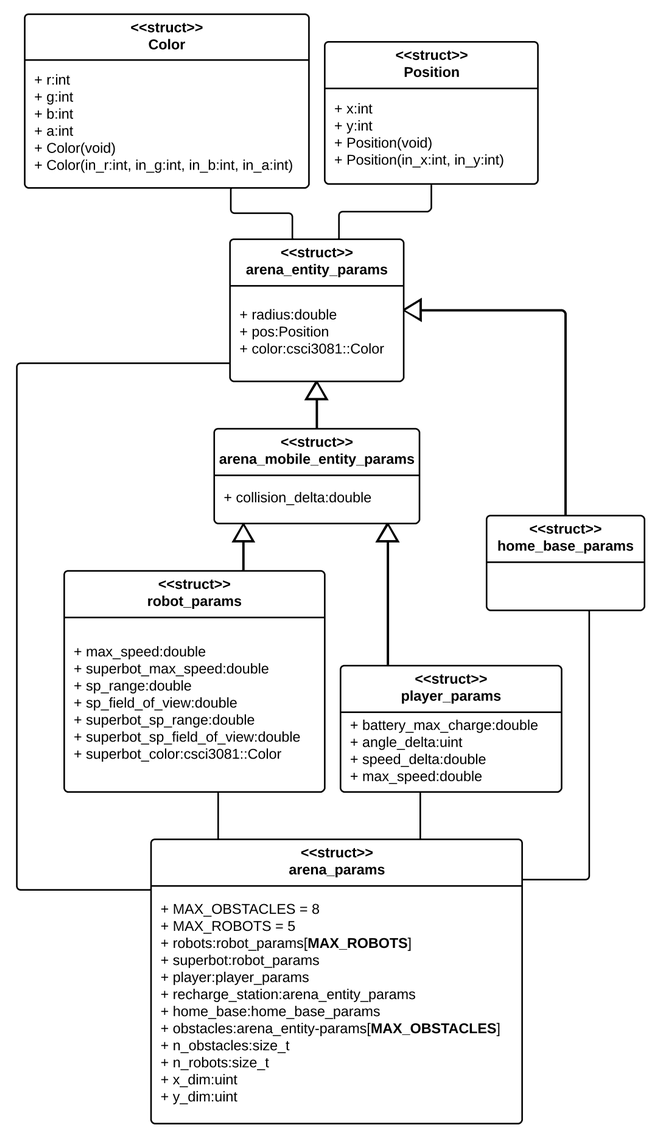
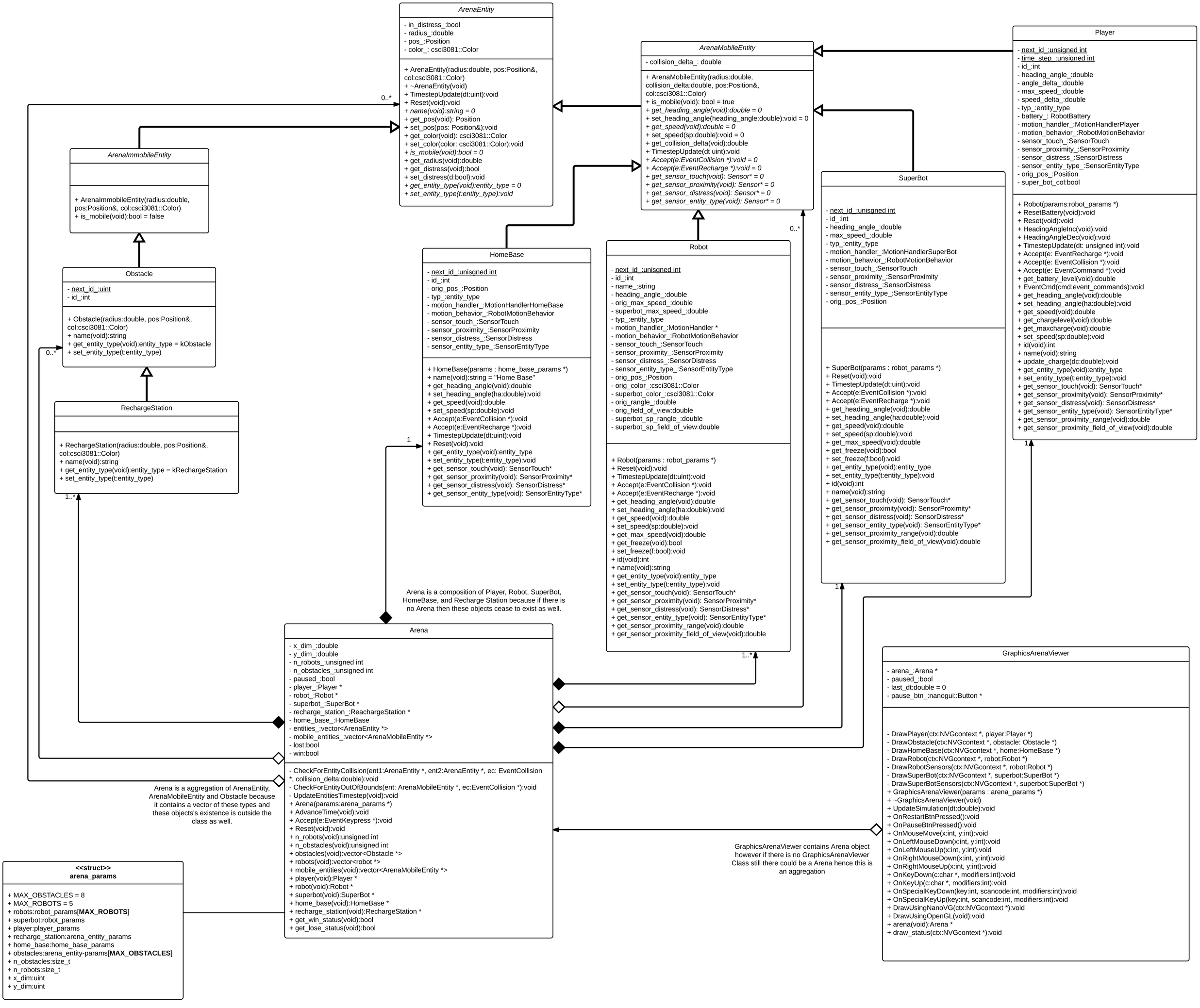
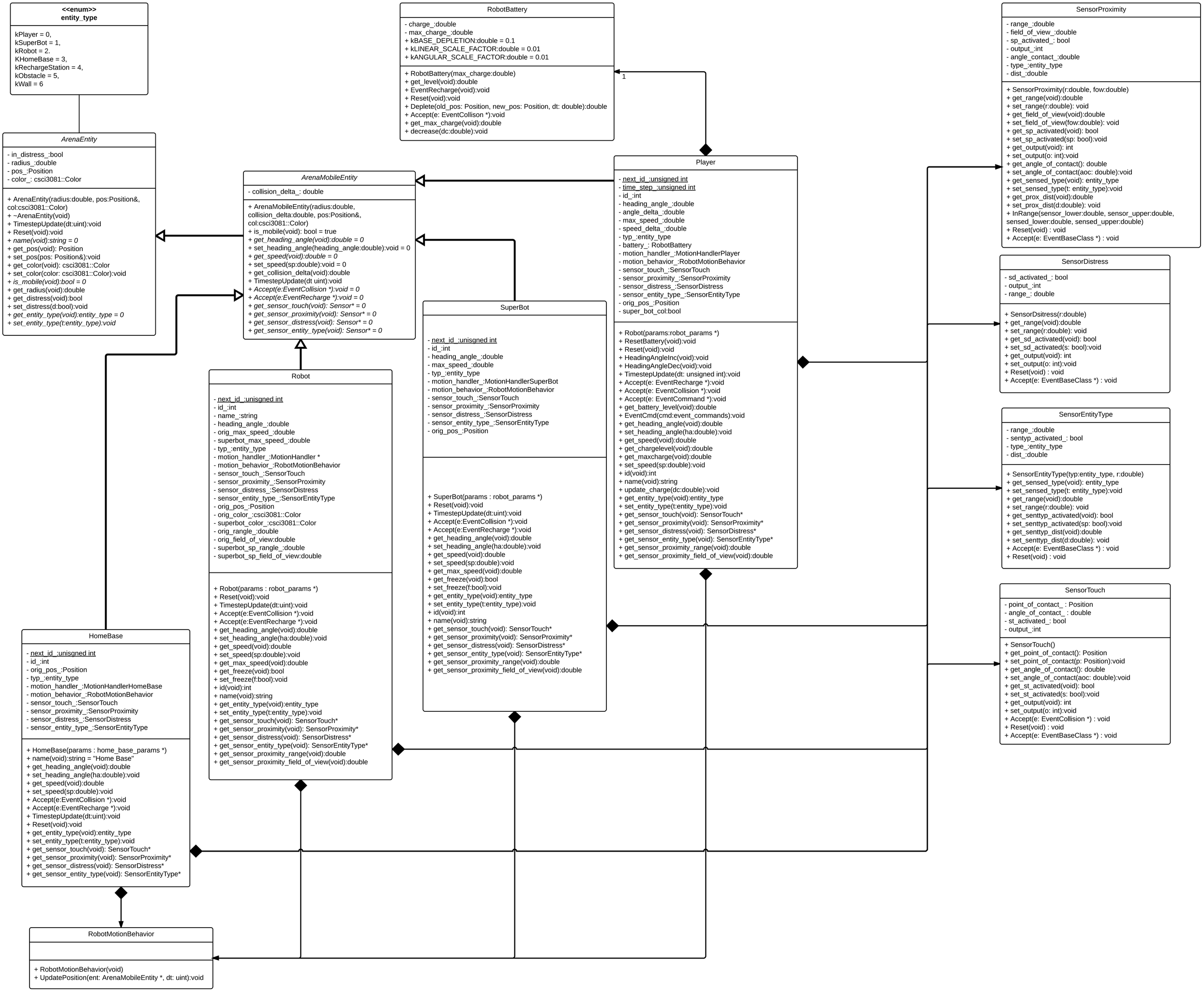

EventKeypress
- key_:int
+ EventKeypress(key:int)
+ EmitMessage(void):void
+ get_key(void):int
+ get_key_cmd(): event_commands
- EventKeypress& operator=(const EventKeypress& other) = delete;
- EventKeypress(const EventKeypress& other) = delete;
EventCommand
- cmd_:event_commands
+ EventCommand(cmd : event_commands)
+ EmitMessage(void):void
+ cmd(void):event_commands
<<enum>>
event_commands
+ COM_TURN_LEFT
+ COM_TURN_RIGHT
+ COM_SPEED_UP
+ COM_SLOW_DOWN
1
EventBaseClass
+ EventBaseClass(void)
+ ~EventBaseClass(void)
+ EmitMessage(void):void = 0
+ get_sensing(): ArenaEntity*
+ get_sensed(): ArenaEntity*
EventRecharge
+ EventRecharge(void)
+ EmitMessage(void):void
EventCollision
- collided_:bool
- point_of_contact_:Position
- angle_of_contact_:double
- sensing_: ArenaEntity *
- sensed_: ArenaEntity *
+ EventCollision()
+ EmitMessage(void):void
+ get_collided(): bool
+ set_collided(c: bool): void
+ get_point_of_contact():Position
+ set_point_of_contact(p:Position):void
+ get_angle_of_contact():double
+ set_angle_of_contact(aoc:double):void
+ get_sensing(): ArenaEntity *
+ set_sensing(ent: ArenaEntity*): void
+ get_sensed(): ArenaEntity *
+ set_sensed(ent: ArenaEntity*): void
EventDistressCall
- in_distress_:bool
- sensed_pos_:Position
- sensing_: ArenaEntity *
- sensed_: ArenaEntity *
+ EventDistressCall()
+ EmitMessage(void):void
+ get_in_distress(): bool
+ set_in_distress(c:bool):void
+ get_sensed_pos():Position
+ set_sensed_pos(sp:Position):void
+ get_sensing(): ArenaEntity *
+ set_sensing(ent: ArenaEntity*): void
+ get_sensed(): ArenaEntity *
+ set_sensed(ent: ArenaEntity*): void
EventProximity
- in_proximity_:bool
- proximity_:double
- sensing_: ArenaEntity *
- sensed_: ArenaEntity *
+ EventProximity()
+ EmitMessage(void):void
+ get_in_proximity(): bool
+ set_in_proximity(c:bool):void
+ get_proximity(): double
+ set_proximity(sp: double):void
+ get_sensing(): ArenaEntity *
+ set_sensing(ent: ArenaEntity*): void
+ get_sensed(): ArenaEntity *
+ set_sensed(ent: ArenaEntity*): void
EventTypeEmit
- emit_type_:entity_type
- sensed_pos_:Position
- sensing_: ArenaEntity *
- sensed_: ArenaEntity *
+ EventTypeEmit()
+ EmitMessage(void):void
+ get_emit_type(): entity_type
+ set_emit_type(c:entity_type):void
+ get_sensed_pos():Position
+ set_sensed_pos(sp:Position):void
+ get_sensing(): ArenaEntity *
+ set_sensing(ent: ArenaEntity*): void
+ get_sensed(): ArenaEntity *
+ set_sensed(ent: ArenaEntity*): void

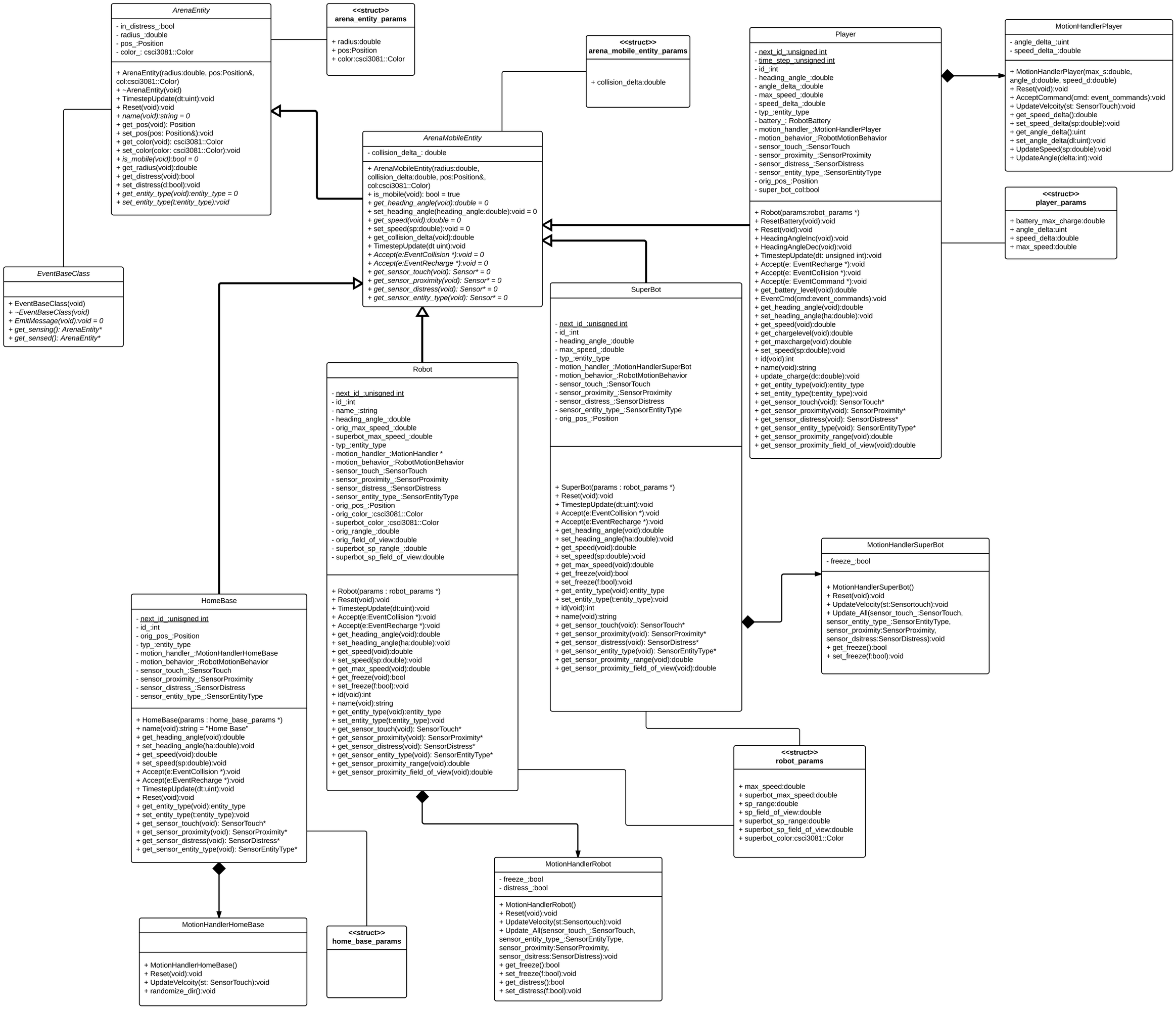

MotionHandlerHomeBase
+ MotionHandlerHomeBase()
+ Reset(void):void
+ UpdateVelcoity(st: SensorTouch):void
+ randomize_dir():void
MotionHandlerRobot
- freeze_:bool
- distress_:bool
+ MotionHandlerRobot()
+ Reset(void):void
+ UpdateVelocity(st:Sensortouch):void
+ Update_All(sensor_touch_:SensorTouch, sensor_entity_type_:SensorEntityType, sensor_proximity:SensorProximity, sensor_dsitress:SensorDistress):void
+ get_freeze():bool
+ set_freeze(f:bool):void
+ get_distress():bool
+ set_distress(f:bool):void
MotionHandler
- heading_angle_:double
- speed_:double
- max_speed_:double
- change_type_:bool
+ MotionHandler()
+ Reset(void):void = 0
+ get_speed():double
+ set_speed(sp:double):void
+ get_heading_angle():double
+ set_heading_angle(ha:double):void
+ get_max_speed():double
+ set_max_speed(ms:double):void
+ Update_All(sensor_touch_:SensorTouch, sensor_entity_type_:SensorEntityType, sensor_proximity:SensorProximity, sensor_dsitress:SensorDistress):void
+ get_freeze():bool
+ set_freeze(f:bool):void
+ get_distress():bool
+ set_distress(d:bool):void
+ get_change_type():bool
+ set_change_type(d:bool):void
MotionHandlerPlayer
- angle_delta_:uint
- speed_delta_:double
+ MotionHandlerPlayer(max_s:double, angle_d:double, speed_d:double)
+ Reset(void):void
+ AcceptCommand(cmd: event_commands):void
+ UpdateVelcoity(st: SensorTouch):void
+ get_speed_delta():double
+ set_speed_delta(sp:double):void
+ get_angle_delta():uint
+ set_angle_delta(dl:uint):void
+ UpdateSpeed(sp:double):void
+ UpdateAngle(delta:int):void
MotionHandlerSuperBot
- freeze_:bool
+ MotionHandlerSuperBot()
+ Reset(void):void
+ UpdateVelocity(st:Sensortouch):void
+ Update_All(sensor_touch_:SensorTouch, sensor_entity_type_:SensorEntityType, sensor_proximity:SensorProximity, sensor_dsitress:SensorDistress):void
+ get_freeze():bool
+ set_freeze(f:bool):void

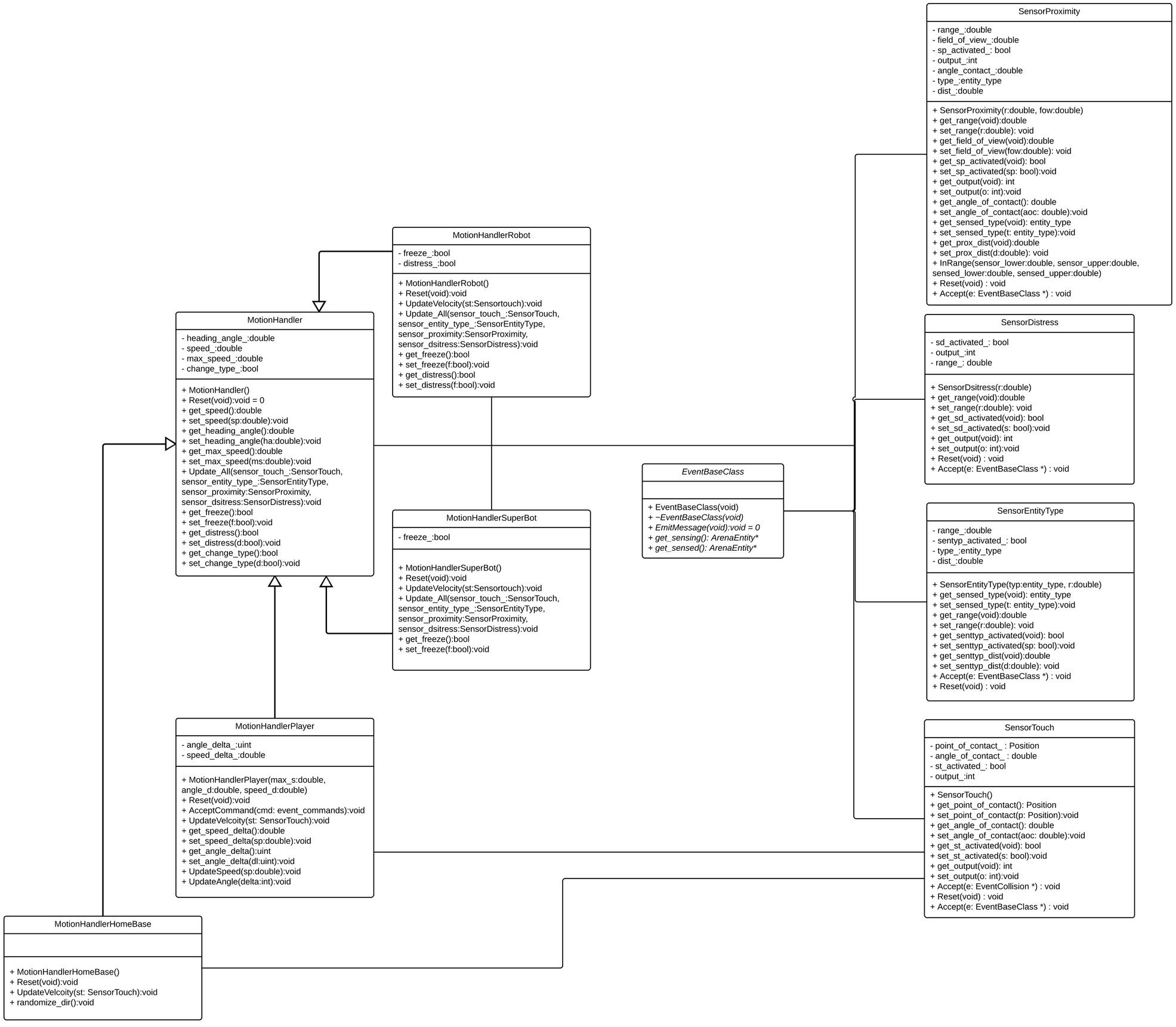

SensorProximity
- range_:double
- field_of_view_:double
- sp_activated_: bool
- output_:int
- angle_contact_:double
- type_:entity_type
- dist_:double
+ SensorProximity(r:double, fow:double)
+ get_range(void):double
+ set_range(r:double): void
+ get_field_of_view(void):double
+ set_field_of_view(fow:double): void
+ get_sp_activated(void): bool
+ set_sp_activated(sp: bool):void
+ get_output(void): int
+ set_output(o: int):void
+ get_angle_of_contact(): double
+ set_angle_of_contact(aoc: double):void
+ get_sensed_type(void): entity_type
+ set_sensed_type(t: entity_type):void
+ get_prox_dist(void):double
+ set_prox_dist(d:double): void
+ InRange(sensor_lower:double, sensor_upper:double, sensed_lower:double, sensed_upper:double)
+ Reset(void) : void
+ Accept(e: EventBaseClass *) : void
SensorTouch
- point_of_contact_ : Position
- angle_of_contact_ : double
- st_activated_: bool
- output_:int
+ SensorTouch()
+ get_point_of_contact(): Position
+ set_point_of_contact(p: Position):void
+ get_angle_of_contact(): double
+ set_angle_of_contact(aoc: double):void
+ get_st_activated(void): bool
+ set_st_activated(s: bool):void
+ get_output(void): int
+ set_output(o: int):void
+ Accept(e: EventCollision *) : void
+ Reset(void) : void
+ Accept(e: EventBaseClass *) : void
Sensor
- activated_: bool
+ Sensor()
+ ~Sensor(void)
+ Accept(e: EventBaseClass *) : void = 0
+ get_activated(void): bool
+ set_activated(value: bool): void
+ Reset(void) : void = 0
+ get_dist(sensing_x:double, sensing_y:double, sensed_x:double, sensed_y:double):double
+ get_angle(sensing_x:double, sensing_y:double, sensed_x:double, sensed_y:double):double
SensorEntityType
- range_:double
- sentyp_activated_: bool
- type_:entity_type
- dist_:double
+ SensorEntityType(typ:entity_type, r:double)
+ get_sensed_type(void): entity_type
+ set_sensed_type(t: entity_type):void
+ get_range(void):double
+ set_range(r:double): void
+ get_senttyp_activated(void): bool
+ set_senttyp_activated(sp: bool):void
+ get_senttyp_dist(void):double
+ set_senttyp_dist(d:double): void
+ Accept(e: EventBaseClass *) : void
+ Reset(void) : void
SensorDistress
- sd_activated_: bool
- output_:int
- range_: double
+ SensorDsitress(r:double)
+ get_range(void):double
+ set_range(r:double): void
+ get_sd_activated(void): bool
+ set_sd_activated(s: bool):void
+ get_output(void): int
+ set_output(o: int):void
+ Reset(void) : void
+ Accept(e: EventBaseClass *) : void


<<struct>>
Color
+ r:int
+ g:int
+ b:int
+ a:int
+ Color(void)
+ Color(in_r:int, in_g:int, in_b:int, in_a:int)
<<struct>>
Position
+ x:int
+ y:int
+ Position(void)
+ Position(in_x:int, in_y:int)
<<struct>>
arena_entity_params
+ radius:double
+ pos:Position
+ color:csci3081::Color
<<struct>>
arena_mobile_entity_params
+ collision_delta:double
<<struct>>
home_base_params
<<struct>>
robot_params
+ max_speed:double
+ superbot_max_speed:double
+ sp_range:double
+ sp_field_of_view:double
+ superbot_sp_range:double
+ superbot_sp_field_of_view:double
+ superbot_color:csci3081::Color
<<struct>>
player_params
+ battery_max_charge:double
+ angle_delta:uint
+ speed_delta:double
+ max_speed:double
<<struct>>
arena_params
+ MAX_OBSTACLES = 8
+ MAX_ROBOTS = 5
+ robots:robot_params[MAX_ROBOTS]
+ superbot:robot_params
+ player:player_params
+ recharge_station:arena_entity_params
+ home_base:home_base_params
+ obstacles:arena_entity-params[MAX_OBSTACLES]
+ n_obstacles:size_t
+ n_robots:size_t
+ x_dim:uint
+ y_dim:uint


ArenaEntity
- in_distress_:bool
- radius_:double
- pos_:Position
- color_: csci3081::Color
+ ArenaEntity(radius:double, pos:Position&, col:csci3081::Color)
+ ~ArenaEntity(void)
+ TimestepUpdate(dt:uint):void
+ Reset(void):void
+ name(void):string = 0
+ get_pos(void): Position
+ set_pos(pos: Position&):void
+ get_color(void): csci3081::Color
+ set_color(color: csci3081::Color):void
+ is_mobile(void):bool = 0
+ get_radius(void):double
+ get_distress(void):bool
+ set_distress(d:bool):void
+ get_entity_type(void):entity_type = 0
+ set_entity_type(t:entity_type):void
ArenaMobileEntity
- collision_delta_: double
+ ArenaMobileEntity(radius:double, collision_delta:double, pos:Position&, col:csci3081::Color)
+ is_mobile(void): bool = true
+ get_heading_angle(void):double = 0
+ set_heading_angle(heading_angle:double):void = 0
+ get_speed(void):double = 0
+ set_speed(sp:double):void = 0
+ get_collision_delta(void):double
+ TimestepUpdate(dt uint):void
+ Accept(e:EventCollision *):void = 0
+ Accept(e:EventRecharge *):void = 0
+ get_sensor_touch(void): Sensor* = 0
+ get_sensor_proximity(void): Sensor* = 0
+ get_sensor_distress(void): Sensor* = 0
+ get_sensor_entity_type(void): Sensor* = 0
Player
- next_id_:unsigned int
- time_step_:unsigned int
- id_:int
- heading_angle_:double
- angle_delta_:double
- max_speed_:double
- speed_delta_:double
- typ_:entity_type
- battery_: RobotBattery
- motion_handler_:MotionHandlerPlayer
- motion_behavior_:RobotMotionBehavior
- sensor_touch_:SensorTouch
- sensor_proximity_:SensorProximity
- sensor_distress_:SensorDistress
- sensor_entity_type_:SensorEntityType
- orig_pos_:Position
- super_bot_col:bool
+ Robot(params:robot_params *)
+ ResetBattery(void):void
+ Reset(void):void
+ HeadingAngleInc(void):void
+ HeadingAngleDec(void):void
+ TimestepUpdate(dt: unsigned int):void
+ Accept(e: EventRecharge *):void
+ Accept(e: EventCollision *):void
+ Accept(e: EventCommand *):void
+ get_battery_level(void):double
+ EventCmd(cmd:event_commands):void
+ get_heading_angle(void):double
+ set_heading_angle(ha:double):void
+ get_speed(void):double
+ get_chargelevel(void):double
+ get_maxcharge(void):double
+ set_speed(sp:double):void
+ id(void):int
+ name(void):string
+ update_charge(dc:double):void
+ get_entity_type(void):entity_type
+ set_entity_type(t:entity_type):void
+ get_sensor_touch(void): SensorTouch*
+ get_sensor_proximity(void): SensorProximity*
+ get_sensor_distress(void): SensorDistress*
+ get_sensor_entity_type(void): SensorEntityType*
+ get_sensor_proximity_range(void):double
+ get_sensor_proximity_field_of_view(void):double
ArenaImmobileEntity
+ ArenaImmobileEntity(radius:double, pos:Position&, col:csci3081::Color)
+ is_mobile(void):bool = false
Obstacle
- next_id_:uint
- id_:int
+ Obstacle(radius:double, pos:Position&, col:csci3081::Color)
+ name(void):string
+ get_entity_type(void):entity_type = kObstacle
+ set_entity_type(t:entity_type)
RechargeStation
+ RechargeStation(radius:double, pos:Position&, col:csci3081::Color)
+ name(void):string
+ get_entity_type(void):entity_type = kRechargeStation
+ set_entity_type(t:entity_type)
HomeBase
- next_id_:unisgned int
- id_:int
- orig_pos_:Position
- typ_:entity_type
- motion_handler_:MotionHandlerHomeBase
- motion_behavior_:RobotMotionBehavior
- sensor_touch_:SensorTouch
- sensor_proximity_:SensorProximity
- sensor_distress_:SensorDistress
- sensor_entity_type_:SensorEntityType
+ HomeBase(params : home_base_params *)
+ name(void):string = "Home Base"
+ get_heading_angle(void):double
+ set_heading_angle(ha:double):void
+ get_speed(void):double
+ set_speed(sp:double):void
+ Accept(e:EventCollision *):void
+ Accept(e:EventRecharge *):void
+ TimestepUpdate(dt:uint):void
+ Reset(void):void
+ get_entity_type(void):entity_type
+ set_entity_type(t:entity_type):void
+ get_sensor_touch(void): SensorTouch*
+ get_sensor_proximity(void): SensorProximity*
+ get_sensor_distress(void): SensorDistress*
+ get_sensor_entity_type(void): SensorEntityType*
Robot
- next_id_:unisgned int
- id_:int
- name_:string
- heading_angle_:double
- orig_max_speed_:double
- superbot_max_speed_:double
- typ_:entity_type
- motion_handler_:MotionHandler *
- motion_behavior_:RobotMotionBehavior
- sensor_touch_:SensorTouch
- sensor_proximity_:SensorProximity
- sensor_distress_:SensorDistress
- sensor_entity_type_:SensorEntityType
- orig_pos_:Position
- orig_color_:csci3081::Color
- superbot_color_:csci3081::Color
- orig_rangle_:double
- orig_field_of_view:double
- superbot_sp_rangle_:double
- superbot_sp_field_of_view:double
+ Robot(params : robot_params *)
+ Reset(void):void
+ TimestepUpdate(dt:uint):void
+ Accept(e:EventCollision *):void
+ Accept(e:EventRecharge *):void
+ get_heading_angle(void):double
+ set_heading_angle(ha:double):void
+ get_speed(void):double
+ set_speed(sp:double):void
+ get_max_speed(void):double
+ get_freeze(void):bool
+ set_freeze(f:bool):void
+ id(void):int
+ name(void):string
+ get_entity_type(void):entity_type
+ set_entity_type(t:entity_type):void
+ get_sensor_touch(void): SensorTouch*
+ get_sensor_proximity(void): SensorProximity*
+ get_sensor_distress(void): SensorDistress*
+ get_sensor_entity_type(void): SensorEntityType*
+ get_sensor_proximity_range(void):double
+ get_sensor_proximity_field_of_view(void):double
SuperBot
- next_id_:unisgned int
- id_:int
- heading_angle_:double
- max_speed_:double
- typ_:entity_type
- motion_handler_:MotionHandlerSuperBot
- motion_behavior_:RobotMotionBehavior
- sensor_touch_:SensorTouch
- sensor_proximity_:SensorProximity
- sensor_distress_:SensorDistress
- sensor_entity_type_:SensorEntityType
- orig_pos_:Position
+ SuperBot(params : robot_params *)
+ Reset(void):void
+ TimestepUpdate(dt:uint):void
+ Accept(e:EventCollision *):void
+ Accept(e:EventRecharge *):void
+ get_heading_angle(void):double
+ set_heading_angle(ha:double):void
+ get_speed(void):double
+ set_speed(sp:double):void
+ get_max_speed(void):double
+ get_freeze(void):bool
+ set_freeze(f:bool):void
+ get_entity_type(void):entity_type
+ set_entity_type(t:entity_type):void
+ id(void):int
+ name(void):string
+ get_sensor_touch(void): SensorTouch*
+ get_sensor_proximity(void): SensorProximity*
+ get_sensor_distress(void): SensorDistress*
+ get_sensor_entity_type(void): SensorEntityType*
+ get_sensor_proximity_range(void):double
+ get_sensor_proximity_field_of_view(void):double
Arena is a composition of Player, Robot, SuperBot, HomeBase, and Recharge Station because if there is no Arena then these objects cease to exist as well.
Arena
- x_dim_:double
- y_dim_:double
- n_robots_:unsigned int
- n_obstacles_:unsigned int
- paused_:bool
- player_:Player *
- robot_:Robot *
- superbot_:SuperBot *
- recharge_station_:ReachargeStation *
- home_base_:HomeBase
- entities_:vector<ArenaEntity *>
- mobile_entities_:vector<ArenaMobileEntity *>
- lost:bool
- win:bool
- CheckForEntityCollision(ent1:ArenaEntity *, ent2:ArenaEntity *, ec: EventCollision *, collision_delta:double):void
- CheckForEntityOutOfBounds(ent: ArenaMobileEntity *, ec:EventCollision *):void
- UpdateEntitiesTimestep(void):void
+ Arena(params:arena_params *)
+ AdvanceTime(void):void
+ Accept(e:EventKeypress *):void
+ Reset(void):void
+ n_robots(void):unsigned int
+ n_obstacles(void):unsigned int
+ obstacles(void):vector<Obstacle *>
+ robots(void):vector<robot *>
+ mobile_entities(void):vector<ArenaMobileEntity *>
+ player(void):Player *
+ robot(void):Robot *
+ superbot(void):SuperBot *
+ home_base(void):HomeBase *
+ recharge_station(void):RechargeStation *
+ get_win_status(void):bool
+ get_lose_status(void):bool
Arena is a aggregation of ArenaEntity, ArenaMobileEntity and Obstacle because it contains a vector of these types and these objects's existence is outside the class as well.
GraphicsArenaViewer
- arena_:Arena *
- paused_:bool
- last_dt:double = 0
- pause_btn_:nanogui::Button *
- DrawPlayer(ctx:NVGcontext *, player:Player *)
- DrawObstacle(ctx:NVGcontext *, obstacle: Obstacle *)
- DrawHomeBase(ctx:NVGcontext *, home:HomeBase *)
- DrawRobot(ctx:NVGcontext *, robot:Robot *)
- DrawRobotSensors(ctx:NVGcontext *, robot:Robot *)
- DrawSuperBot(ctx:NVGcontext *, superbot:SuperBot *)
- DrawSuperBotSensors(ctx:NVGcontext *, superbot:SuperBot *)
+ GraphicsArenaViewer(params : arena_params *)
+ ~GraphicsArenaViewer(void)
+ UpdateSimulation(dt:double):void
+ OnRestartBtnPressed():void
+ OnPauseBtnPressed():void
+ OnMouseMove(x:int, y:int):void
+ OnLeftMouseDown(x:int, y:int):void
+ OnLeftMouseUp(x:int, y:int):void
+ OnRightMouseDown(x:int, y:int):void
+ OnRightMouseUp(x:int, y:int):void
+ OnKeyDown(c:char *, modifiers:int):void
+ OnKeyUp(c:char *, modifiers:int):void
+ OnSpecialKeyDown(key:int, scancode:int, modifiers:int):void
+ OnSpecialKeyUp(key:int, scancode:int, modifiers:int):void
+ DrawUsingNanoVG(ctx:NVGcontext *):void
+ DrawUsingOpenGL(void):void
+ arena(void):Arena *
+ draw_status(ctx:NVGcontext *):void
GraphicsArenaViewer contains Arena object however if there is no GraphicsArenaViewer Class still there could be a Arena hence this is an aggregation
<<struct>>
arena_params
+ MAX_OBSTACLES = 8
+ MAX_ROBOTS = 5
+ robots:robot_params[MAX_ROBOTS]
+ superbot:robot_params
+ player:player_params
+ recharge_station:arena_entity_params
+ home_base:home_base_params
+ obstacles:arena_entity-params[MAX_OBSTACLES]
+ n_obstacles:size_t
+ n_robots:size_t
+ x_dim:uint
+ y_dim:uint
0..*
0..*
1..*
1
1..*
0..*
1
1


<<enum>>
entity_type
kPlayer = 0,
kSuperBot = 1,
kRobot = 2.
KHomeBase = 3,
kRechargeStation = 4,
kObstacle = 5,
kWall = 6
ArenaEntity
- in_distress_:bool
- radius_:double
- pos_:Position
- color_: csci3081::Color
+ ArenaEntity(radius:double, pos:Position&, col:csci3081::Color)
+ ~ArenaEntity(void)
+ TimestepUpdate(dt:uint):void
+ Reset(void):void
+ name(void):string = 0
+ get_pos(void): Position
+ set_pos(pos: Position&):void
+ get_color(void): csci3081::Color
+ set_color(color: csci3081::Color):void
+ is_mobile(void):bool = 0
+ get_radius(void):double
+ get_distress(void):bool
+ set_distress(d:bool):void
+ get_entity_type(void):entity_type = 0
+ set_entity_type(t:entity_type):void
RobotBattery
- charge_:double
- max_charge_:double
+ kBASE_DEPLETION:double = 0.1
+ kLINEAR_SCALE_FACTOR:double = 0.01
+ kANGULAR_SCALE_FACTOR:double = 0.01
+ RobotBattery(max_charge:double)
+ get_level(void):double
+ EventRecharge(void):void
+ Reset(void):void
+ Deplete(old_pos: Position, new_pos: Position, dt: double):double
+ Accept(e: EventCollison *):void
+ get_max_charge(void):double
+ decrease(dc:double):void
1
SensorProximity
- range_:double
- field_of_view_:double
- sp_activated_: bool
- output_:int
- angle_contact_:double
- type_:entity_type
- dist_:double
+ SensorProximity(r:double, fow:double)
+ get_range(void):double
+ set_range(r:double): void
+ get_field_of_view(void):double
+ set_field_of_view(fow:double): void
+ get_sp_activated(void): bool
+ set_sp_activated(sp: bool):void
+ get_output(void): int
+ set_output(o: int):void
+ get_angle_of_contact(): double
+ set_angle_of_contact(aoc: double):void
+ get_sensed_type(void): entity_type
+ set_sensed_type(t: entity_type):void
+ get_prox_dist(void):double
+ set_prox_dist(d:double): void
+ InRange(sensor_lower:double, sensor_upper:double, sensed_lower:double, sensed_upper:double)
+ Reset(void) : void
+ Accept(e: EventBaseClass *) : void
ArenaMobileEntity
- collision_delta_: double
+ ArenaMobileEntity(radius:double, collision_delta:double, pos:Position&, col:csci3081::Color)
+ is_mobile(void): bool = true
+ get_heading_angle(void):double = 0
+ set_heading_angle(heading_angle:double):void = 0
+ get_speed(void):double = 0
+ set_speed(sp:double):void = 0
+ get_collision_delta(void):double
+ TimestepUpdate(dt uint):void
+ Accept(e:EventCollision *):void = 0
+ Accept(e:EventRecharge *):void = 0
+ get_sensor_touch(void): Sensor* = 0
+ get_sensor_proximity(void): Sensor* = 0
+ get_sensor_distress(void): Sensor* = 0
+ get_sensor_entity_type(void): Sensor* = 0
Player
- next_id_:unsigned int
- time_step_:unsigned int
- id_:int
- heading_angle_:double
- angle_delta_:double
- max_speed_:double
- speed_delta_:double
- typ_:entity_type
- battery_: RobotBattery
- motion_handler_:MotionHandlerPlayer
- motion_behavior_:RobotMotionBehavior
- sensor_touch_:SensorTouch
- sensor_proximity_:SensorProximity
- sensor_distress_:SensorDistress
- sensor_entity_type_:SensorEntityType
- orig_pos_:Position
- super_bot_col:bool
+ Robot(params:robot_params *)
+ ResetBattery(void):void
+ Reset(void):void
+ HeadingAngleInc(void):void
+ HeadingAngleDec(void):void
+ TimestepUpdate(dt: unsigned int):void
+ Accept(e: EventRecharge *):void
+ Accept(e: EventCollision *):void
+ Accept(e: EventCommand *):void
+ get_battery_level(void):double
+ EventCmd(cmd:event_commands):void
+ get_heading_angle(void):double
+ set_heading_angle(ha:double):void
+ get_speed(void):double
+ get_chargelevel(void):double
+ get_maxcharge(void):double
+ set_speed(sp:double):void
+ id(void):int
+ name(void):string
+ update_charge(dc:double):void
+ get_entity_type(void):entity_type
+ set_entity_type(t:entity_type):void
+ get_sensor_touch(void): SensorTouch*
+ get_sensor_proximity(void): SensorProximity*
+ get_sensor_distress(void): SensorDistress*
+ get_sensor_entity_type(void): SensorEntityType*
+ get_sensor_proximity_range(void):double
+ get_sensor_proximity_field_of_view(void):double
SensorDistress
- sd_activated_: bool
- output_:int
- range_: double
+ SensorDsitress(r:double)
+ get_range(void):double
+ set_range(r:double): void
+ get_sd_activated(void): bool
+ set_sd_activated(s: bool):void
+ get_output(void): int
+ set_output(o: int):void
+ Reset(void) : void
+ Accept(e: EventBaseClass *) : void
SuperBot
- next_id_:unisgned int
- id_:int
- heading_angle_:double
- max_speed_:double
- typ_:entity_type
- motion_handler_:MotionHandlerSuperBot
- motion_behavior_:RobotMotionBehavior
- sensor_touch_:SensorTouch
- sensor_proximity_:SensorProximity
- sensor_distress_:SensorDistress
- sensor_entity_type_:SensorEntityType
- orig_pos_:Position
+ SuperBot(params : robot_params *)
+ Reset(void):void
+ TimestepUpdate(dt:uint):void
+ Accept(e:EventCollision *):void
+ Accept(e:EventRecharge *):void
+ get_heading_angle(void):double
+ set_heading_angle(ha:double):void
+ get_speed(void):double
+ set_speed(sp:double):void
+ get_max_speed(void):double
+ get_freeze(void):bool
+ set_freeze(f:bool):void
+ get_entity_type(void):entity_type
+ set_entity_type(t:entity_type):void
+ id(void):int
+ name(void):string
+ get_sensor_touch(void): SensorTouch*
+ get_sensor_proximity(void): SensorProximity*
+ get_sensor_distress(void): SensorDistress*
+ get_sensor_entity_type(void): SensorEntityType*
+ get_sensor_proximity_range(void):double
+ get_sensor_proximity_field_of_view(void):double
SensorEntityType
- range_:double
- sentyp_activated_: bool
- type_:entity_type
- dist_:double
+ SensorEntityType(typ:entity_type, r:double)
+ get_sensed_type(void): entity_type
+ set_sensed_type(t: entity_type):void
+ get_range(void):double
+ set_range(r:double): void
+ get_senttyp_activated(void): bool
+ set_senttyp_activated(sp: bool):void
+ get_senttyp_dist(void):double
+ set_senttyp_dist(d:double): void
+ Accept(e: EventBaseClass *) : void
+ Reset(void) : void
Robot
- next_id_:unisgned int
- id_:int
- name_:string
- heading_angle_:double
- orig_max_speed_:double
- superbot_max_speed_:double
- typ_:entity_type
- motion_handler_:MotionHandler *
- motion_behavior_:RobotMotionBehavior
- sensor_touch_:SensorTouch
- sensor_proximity_:SensorProximity
- sensor_distress_:SensorDistress
- sensor_entity_type_:SensorEntityType
- orig_pos_:Position
- orig_color_:csci3081::Color
- superbot_color_:csci3081::Color
- orig_rangle_:double
- orig_field_of_view:double
- superbot_sp_rangle_:double
- superbot_sp_field_of_view:double
+ Robot(params : robot_params *)
+ Reset(void):void
+ TimestepUpdate(dt:uint):void
+ Accept(e:EventCollision *):void
+ Accept(e:EventRecharge *):void
+ get_heading_angle(void):double
+ set_heading_angle(ha:double):void
+ get_speed(void):double
+ set_speed(sp:double):void
+ get_max_speed(void):double
+ get_freeze(void):bool
+ set_freeze(f:bool):void
+ id(void):int
+ name(void):string
+ get_entity_type(void):entity_type
+ set_entity_type(t:entity_type):void
+ get_sensor_touch(void): SensorTouch*
+ get_sensor_proximity(void): SensorProximity*
+ get_sensor_distress(void): SensorDistress*
+ get_sensor_entity_type(void): SensorEntityType*
+ get_sensor_proximity_range(void):double
+ get_sensor_proximity_field_of_view(void):double
SensorTouch
- point_of_contact_ : Position
- angle_of_contact_ : double
- st_activated_: bool
- output_:int
+ SensorTouch()
+ get_point_of_contact(): Position
+ set_point_of_contact(p: Position):void
+ get_angle_of_contact(): double
+ set_angle_of_contact(aoc: double):void
+ get_st_activated(void): bool
+ set_st_activated(s: bool):void
+ get_output(void): int
+ set_output(o: int):void
+ Accept(e: EventCollision *) : void
+ Reset(void) : void
+ Accept(e: EventBaseClass *) : void
HomeBase
- next_id_:unisgned int
- id_:int
- orig_pos_:Position
- typ_:entity_type
- motion_handler_:MotionHandlerHomeBase
- motion_behavior_:RobotMotionBehavior
- sensor_touch_:SensorTouch
- sensor_proximity_:SensorProximity
- sensor_distress_:SensorDistress
- sensor_entity_type_:SensorEntityType
+ HomeBase(params : home_base_params *)
+ name(void):string = "Home Base"
+ get_heading_angle(void):double
+ set_heading_angle(ha:double):void
+ get_speed(void):double
+ set_speed(sp:double):void
+ Accept(e:EventCollision *):void
+ Accept(e:EventRecharge *):void
+ TimestepUpdate(dt:uint):void
+ Reset(void):void
+ get_entity_type(void):entity_type
+ set_entity_type(t:entity_type):void
+ get_sensor_touch(void): SensorTouch*
+ get_sensor_proximity(void): SensorProximity*
+ get_sensor_distress(void): SensorDistress*
+ get_sensor_entity_type(void): SensorEntityType*
RobotMotionBehavior
+ RobotMotionBehavior(void)
+ UpdatePosition(ent: ArenaMobileEntity *, dt: uint):void


ArenaEntity
- in_distress_:bool
- radius_:double
- pos_:Position
- color_: csci3081::Color
+ ArenaEntity(radius:double, pos:Position&, col:csci3081::Color)
+ ~ArenaEntity(void)
+ TimestepUpdate(dt:uint):void
+ Reset(void):void
+ name(void):string = 0
+ get_pos(void): Position
+ set_pos(pos: Position&):void
+ get_color(void): csci3081::Color
+ set_color(color: csci3081::Color):void
+ is_mobile(void):bool = 0
+ get_radius(void):double
+ get_distress(void):bool
+ set_distress(d:bool):void
+ get_entity_type(void):entity_type = 0
+ set_entity_type(t:entity_type):void
<<struct>>
arena_entity_params
+ radius:double
+ pos:Position
+ color:csci3081::Color
<<struct>>
arena_mobile_entity_params
+ collision_delta:double
ArenaMobileEntity
- collision_delta_: double
+ ArenaMobileEntity(radius:double, collision_delta:double, pos:Position&, col:csci3081::Color)
+ is_mobile(void): bool = true
+ get_heading_angle(void):double = 0
+ set_heading_angle(heading_angle:double):void = 0
+ get_speed(void):double = 0
+ set_speed(sp:double):void = 0
+ get_collision_delta(void):double
+ TimestepUpdate(dt uint):void
+ Accept(e:EventCollision *):void = 0
+ Accept(e:EventRecharge *):void = 0
+ get_sensor_touch(void): Sensor* = 0
+ get_sensor_proximity(void): Sensor* = 0
+ get_sensor_distress(void): Sensor* = 0
+ get_sensor_entity_type(void): Sensor* = 0
Player
- next_id_:unsigned int
- time_step_:unsigned int
- id_:int
- heading_angle_:double
- angle_delta_:double
- max_speed_:double
- speed_delta_:double
- typ_:entity_type
- battery_: RobotBattery
- motion_handler_:MotionHandlerPlayer
- motion_behavior_:RobotMotionBehavior
- sensor_touch_:SensorTouch
- sensor_proximity_:SensorProximity
- sensor_distress_:SensorDistress
- sensor_entity_type_:SensorEntityType
- orig_pos_:Position
- super_bot_col:bool
+ Robot(params:robot_params *)
+ ResetBattery(void):void
+ Reset(void):void
+ HeadingAngleInc(void):void
+ HeadingAngleDec(void):void
+ TimestepUpdate(dt: unsigned int):void
+ Accept(e: EventRecharge *):void
+ Accept(e: EventCollision *):void
+ Accept(e: EventCommand *):void
+ get_battery_level(void):double
+ EventCmd(cmd:event_commands):void
+ get_heading_angle(void):double
+ set_heading_angle(ha:double):void
+ get_speed(void):double
+ get_chargelevel(void):double
+ get_maxcharge(void):double
+ set_speed(sp:double):void
+ id(void):int
+ name(void):string
+ update_charge(dc:double):void
+ get_entity_type(void):entity_type
+ set_entity_type(t:entity_type):void
+ get_sensor_touch(void): SensorTouch*
+ get_sensor_proximity(void): SensorProximity*
+ get_sensor_distress(void): SensorDistress*
+ get_sensor_entity_type(void): SensorEntityType*
+ get_sensor_proximity_range(void):double
+ get_sensor_proximity_field_of_view(void):double
MotionHandlerPlayer
- angle_delta_:uint
- speed_delta_:double
+ MotionHandlerPlayer(max_s:double, angle_d:double, speed_d:double)
+ Reset(void):void
+ AcceptCommand(cmd: event_commands):void
+ UpdateVelcoity(st: SensorTouch):void
+ get_speed_delta():double
+ set_speed_delta(sp:double):void
+ get_angle_delta():uint
+ set_angle_delta(dl:uint):void
+ UpdateSpeed(sp:double):void
+ UpdateAngle(delta:int):void
<<struct>>
player_params
+ battery_max_charge:double
+ angle_delta:uint
+ speed_delta:double
+ max_speed:double
EventBaseClass
+ EventBaseClass(void)
+ ~EventBaseClass(void)
+ EmitMessage(void):void = 0
+ get_sensing(): ArenaEntity*
+ get_sensed(): ArenaEntity*
SuperBot
- next_id_:unisgned int
- id_:int
- heading_angle_:double
- max_speed_:double
- typ_:entity_type
- motion_handler_:MotionHandlerSuperBot
- motion_behavior_:RobotMotionBehavior
- sensor_touch_:SensorTouch
- sensor_proximity_:SensorProximity
- sensor_distress_:SensorDistress
- sensor_entity_type_:SensorEntityType
- orig_pos_:Position
+ SuperBot(params : robot_params *)
+ Reset(void):void
+ TimestepUpdate(dt:uint):void
+ Accept(e:EventCollision *):void
+ Accept(e:EventRecharge *):void
+ get_heading_angle(void):double
+ set_heading_angle(ha:double):void
+ get_speed(void):double
+ set_speed(sp:double):void
+ get_max_speed(void):double
+ get_freeze(void):bool
+ set_freeze(f:bool):void
+ get_entity_type(void):entity_type
+ set_entity_type(t:entity_type):void
+ id(void):int
+ name(void):string
+ get_sensor_touch(void): SensorTouch*
+ get_sensor_proximity(void): SensorProximity*
+ get_sensor_distress(void): SensorDistress*
+ get_sensor_entity_type(void): SensorEntityType*
+ get_sensor_proximity_range(void):double
+ get_sensor_proximity_field_of_view(void):double
MotionHandlerSuperBot
- freeze_:bool
+ MotionHandlerSuperBot()
+ Reset(void):void
+ UpdateVelocity(st:Sensortouch):void
+ Update_All(sensor_touch_:SensorTouch, sensor_entity_type_:SensorEntityType, sensor_proximity:SensorProximity, sensor_dsitress:SensorDistress):void
+ get_freeze():bool
+ set_freeze(f:bool):void
Robot
- next_id_:unisgned int
- id_:int
- name_:string
- heading_angle_:double
- orig_max_speed_:double
- superbot_max_speed_:double
- typ_:entity_type
- motion_handler_:MotionHandler *
- motion_behavior_:RobotMotionBehavior
- sensor_touch_:SensorTouch
- sensor_proximity_:SensorProximity
- sensor_distress_:SensorDistress
- sensor_entity_type_:SensorEntityType
- orig_pos_:Position
- orig_color_:csci3081::Color
- superbot_color_:csci3081::Color
- orig_rangle_:double
- orig_field_of_view:double
- superbot_sp_rangle_:double
- superbot_sp_field_of_view:double
+ Robot(params : robot_params *)
+ Reset(void):void
+ TimestepUpdate(dt:uint):void
+ Accept(e:EventCollision *):void
+ Accept(e:EventRecharge *):void
+ get_heading_angle(void):double
+ set_heading_angle(ha:double):void
+ get_speed(void):double
+ set_speed(sp:double):void
+ get_max_speed(void):double
+ get_freeze(void):bool
+ set_freeze(f:bool):void
+ id(void):int
+ name(void):string
+ get_entity_type(void):entity_type
+ set_entity_type(t:entity_type):void
+ get_sensor_touch(void): SensorTouch*
+ get_sensor_proximity(void): SensorProximity*
+ get_sensor_distress(void): SensorDistress*
+ get_sensor_entity_type(void): SensorEntityType*
+ get_sensor_proximity_range(void):double
+ get_sensor_proximity_field_of_view(void):double
<<struct>>
robot_params
+ max_speed:double
+ superbot_max_speed:double
+ sp_range:double
+ sp_field_of_view:double
+ superbot_sp_range:double
+ superbot_sp_field_of_view:double
+ superbot_color:csci3081::Color
HomeBase
- next_id_:unisgned int
- id_:int
- orig_pos_:Position
- typ_:entity_type
- motion_handler_:MotionHandlerHomeBase
- motion_behavior_:RobotMotionBehavior
- sensor_touch_:SensorTouch
- sensor_proximity_:SensorProximity
- sensor_distress_:SensorDistress
- sensor_entity_type_:SensorEntityType
+ HomeBase(params : home_base_params *)
+ name(void):string = "Home Base"
+ get_heading_angle(void):double
+ set_heading_angle(ha:double):void
+ get_speed(void):double
+ set_speed(sp:double):void
+ Accept(e:EventCollision *):void
+ Accept(e:EventRecharge *):void
+ TimestepUpdate(dt:uint):void
+ Reset(void):void
+ get_entity_type(void):entity_type
+ set_entity_type(t:entity_type):void
+ get_sensor_touch(void): SensorTouch*
+ get_sensor_proximity(void): SensorProximity*
+ get_sensor_distress(void): SensorDistress*
+ get_sensor_entity_type(void): SensorEntityType*
MotionHandlerHomeBase
+ MotionHandlerHomeBase()
+ Reset(void):void
+ UpdateVelcoity(st: SensorTouch):void
+ randomize_dir():void
<<struct>>
home_base_params
MotionHandlerRobot
- freeze_:bool
- distress_:bool
+ MotionHandlerRobot()
+ Reset(void):void
+ UpdateVelocity(st:Sensortouch):void
+ Update_All(sensor_touch_:SensorTouch, sensor_entity_type_:SensorEntityType, sensor_proximity:SensorProximity, sensor_dsitress:SensorDistress):void
+ get_freeze():bool
+ set_freeze(f:bool):void
+ get_distress():bool
+ set_distress(f:bool):void


SensorProximity
- range_:double
- field_of_view_:double
- sp_activated_: bool
- output_:int
- angle_contact_:double
- type_:entity_type
- dist_:double
+ SensorProximity(r:double, fow:double)
+ get_range(void):double
+ set_range(r:double): void
+ get_field_of_view(void):double
+ set_field_of_view(fow:double): void
+ get_sp_activated(void): bool
+ set_sp_activated(sp: bool):void
+ get_output(void): int
+ set_output(o: int):void
+ get_angle_of_contact(): double
+ set_angle_of_contact(aoc: double):void
+ get_sensed_type(void): entity_type
+ set_sensed_type(t: entity_type):void
+ get_prox_dist(void):double
+ set_prox_dist(d:double): void
+ InRange(sensor_lower:double, sensor_upper:double, sensed_lower:double, sensed_upper:double)
+ Reset(void) : void
+ Accept(e: EventBaseClass *) : void
MotionHandlerRobot
- freeze_:bool
- distress_:bool
+ MotionHandlerRobot()
+ Reset(void):void
+ UpdateVelocity(st:Sensortouch):void
+ Update_All(sensor_touch_:SensorTouch, sensor_entity_type_:SensorEntityType, sensor_proximity:SensorProximity, sensor_dsitress:SensorDistress):void
+ get_freeze():bool
+ set_freeze(f:bool):void
+ get_distress():bool
+ set_distress(f:bool):void
MotionHandler
- heading_angle_:double
- speed_:double
- max_speed_:double
- change_type_:bool
+ MotionHandler()
+ Reset(void):void = 0
+ get_speed():double
+ set_speed(sp:double):void
+ get_heading_angle():double
+ set_heading_angle(ha:double):void
+ get_max_speed():double
+ set_max_speed(ms:double):void
+ Update_All(sensor_touch_:SensorTouch, sensor_entity_type_:SensorEntityType, sensor_proximity:SensorProximity, sensor_dsitress:SensorDistress):void
+ get_freeze():bool
+ set_freeze(f:bool):void
+ get_distress():bool
+ set_distress(d:bool):void
+ get_change_type():bool
+ set_change_type(d:bool):void
SensorDistress
- sd_activated_: bool
- output_:int
- range_: double
+ SensorDsitress(r:double)
+ get_range(void):double
+ set_range(r:double): void
+ get_sd_activated(void): bool
+ set_sd_activated(s: bool):void
+ get_output(void): int
+ set_output(o: int):void
+ Reset(void) : void
+ Accept(e: EventBaseClass *) : void
EventBaseClass
+ EventBaseClass(void)
+ ~EventBaseClass(void)
+ EmitMessage(void):void = 0
+ get_sensing(): ArenaEntity*
+ get_sensed(): ArenaEntity*
MotionHandlerSuperBot
- freeze_:bool
+ MotionHandlerSuperBot()
+ Reset(void):void
+ UpdateVelocity(st:Sensortouch):void
+ Update_All(sensor_touch_:SensorTouch, sensor_entity_type_:SensorEntityType, sensor_proximity:SensorProximity, sensor_dsitress:SensorDistress):void
+ get_freeze():bool
+ set_freeze(f:bool):void
SensorEntityType
- range_:double
- sentyp_activated_: bool
- type_:entity_type
- dist_:double
+ SensorEntityType(typ:entity_type, r:double)
+ get_sensed_type(void): entity_type
+ set_sensed_type(t: entity_type):void
+ get_range(void):double
+ set_range(r:double): void
+ get_senttyp_activated(void): bool
+ set_senttyp_activated(sp: bool):void
+ get_senttyp_dist(void):double
+ set_senttyp_dist(d:double): void
+ Accept(e: EventBaseClass *) : void
+ Reset(void) : void
MotionHandlerPlayer
- angle_delta_:uint
- speed_delta_:double
+ MotionHandlerPlayer(max_s:double, angle_d:double, speed_d:double)
+ Reset(void):void
+ AcceptCommand(cmd: event_commands):void
+ UpdateVelcoity(st: SensorTouch):void
+ get_speed_delta():double
+ set_speed_delta(sp:double):void
+ get_angle_delta():uint
+ set_angle_delta(dl:uint):void
+ UpdateSpeed(sp:double):void
+ UpdateAngle(delta:int):void
SensorTouch
- point_of_contact_ : Position
- angle_of_contact_ : double
- st_activated_: bool
- output_:int
+ SensorTouch()
+ get_point_of_contact(): Position
+ set_point_of_contact(p: Position):void
+ get_angle_of_contact(): double
+ set_angle_of_contact(aoc: double):void
+ get_st_activated(void): bool
+ set_st_activated(s: bool):void
+ get_output(void): int
+ set_output(o: int):void
+ Accept(e: EventCollision *) : void
+ Reset(void) : void
+ Accept(e: EventBaseClass *) : void
MotionHandlerHomeBase
+ MotionHandlerHomeBase()
+ Reset(void):void
+ UpdateVelcoity(st: SensorTouch):void
+ randomize_dir():void