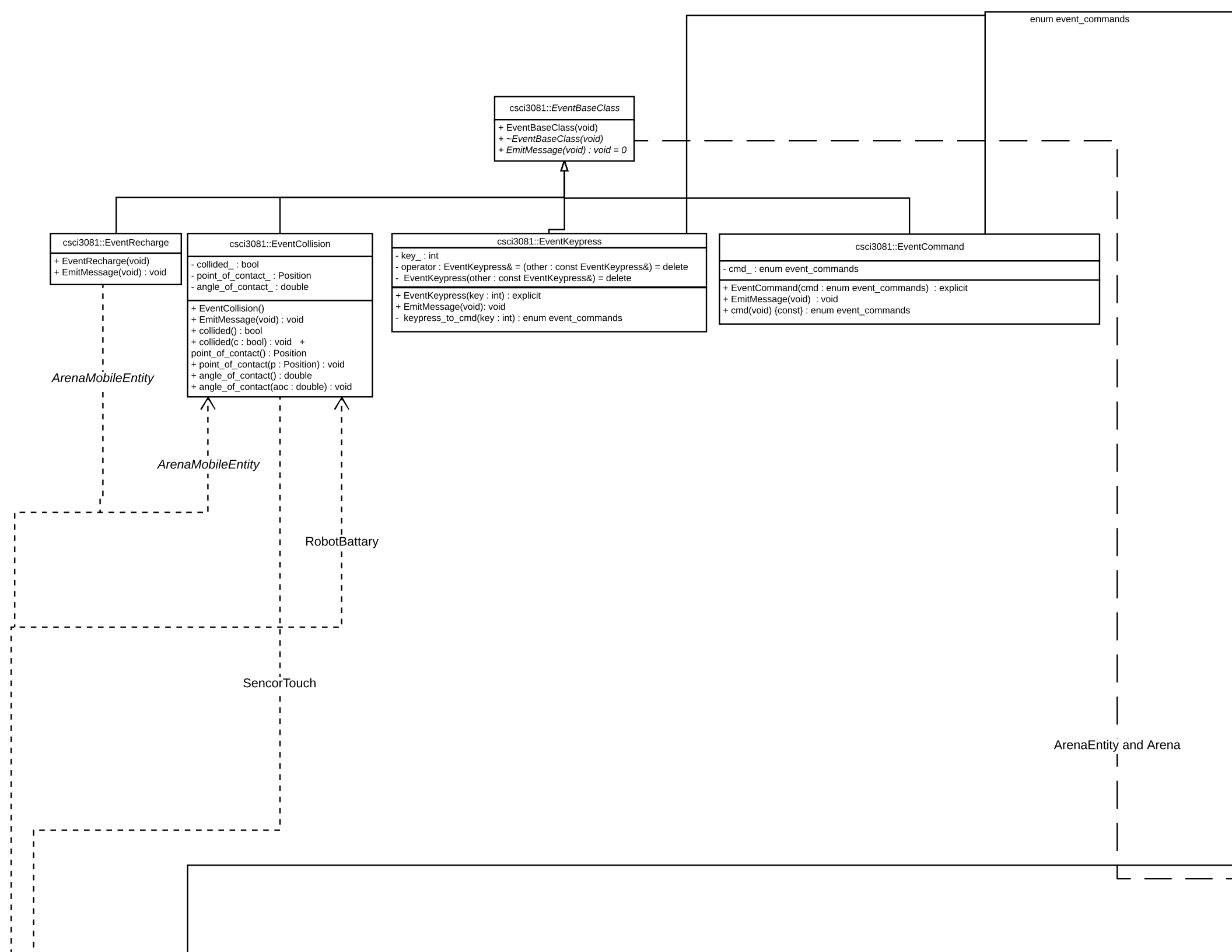
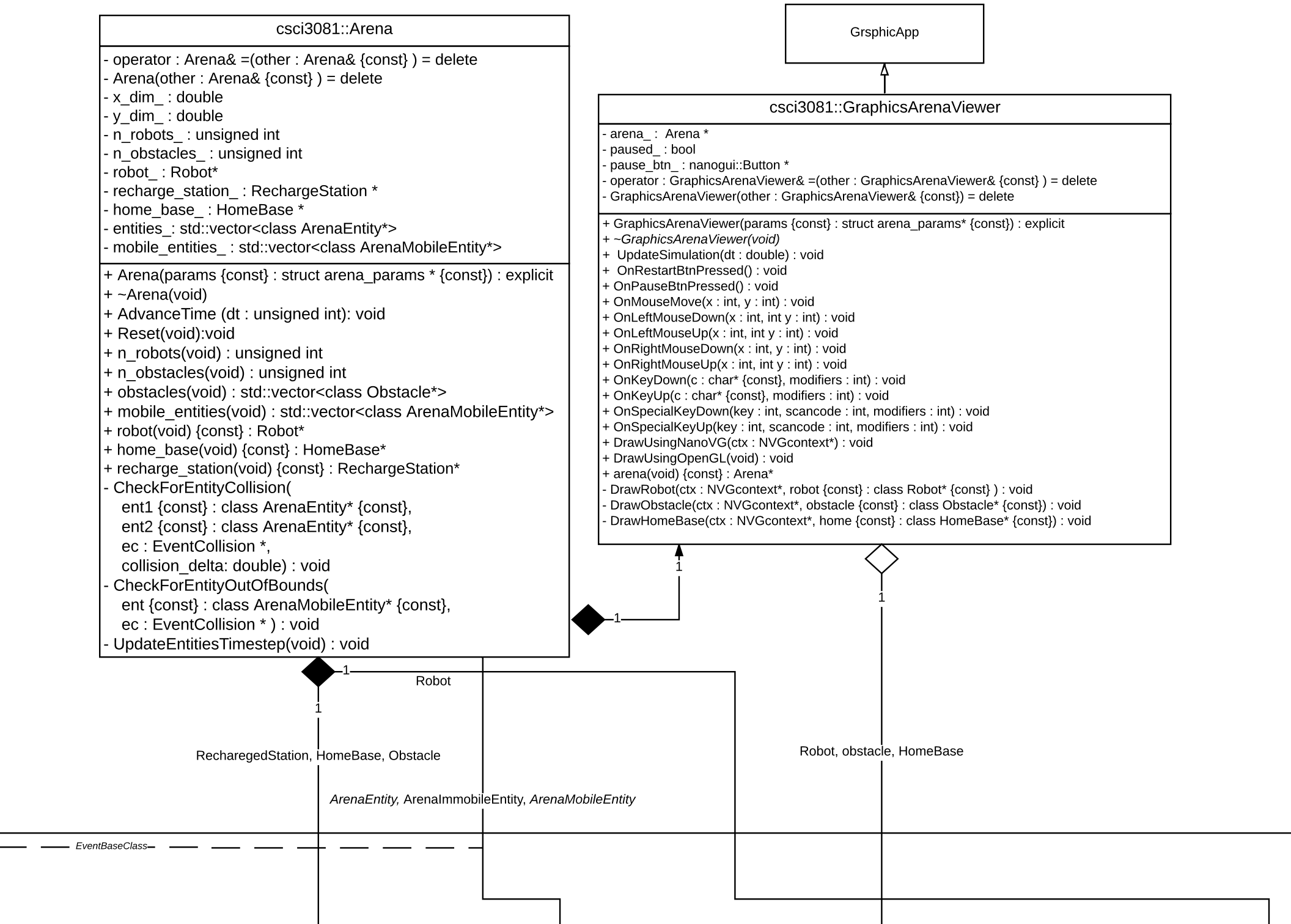
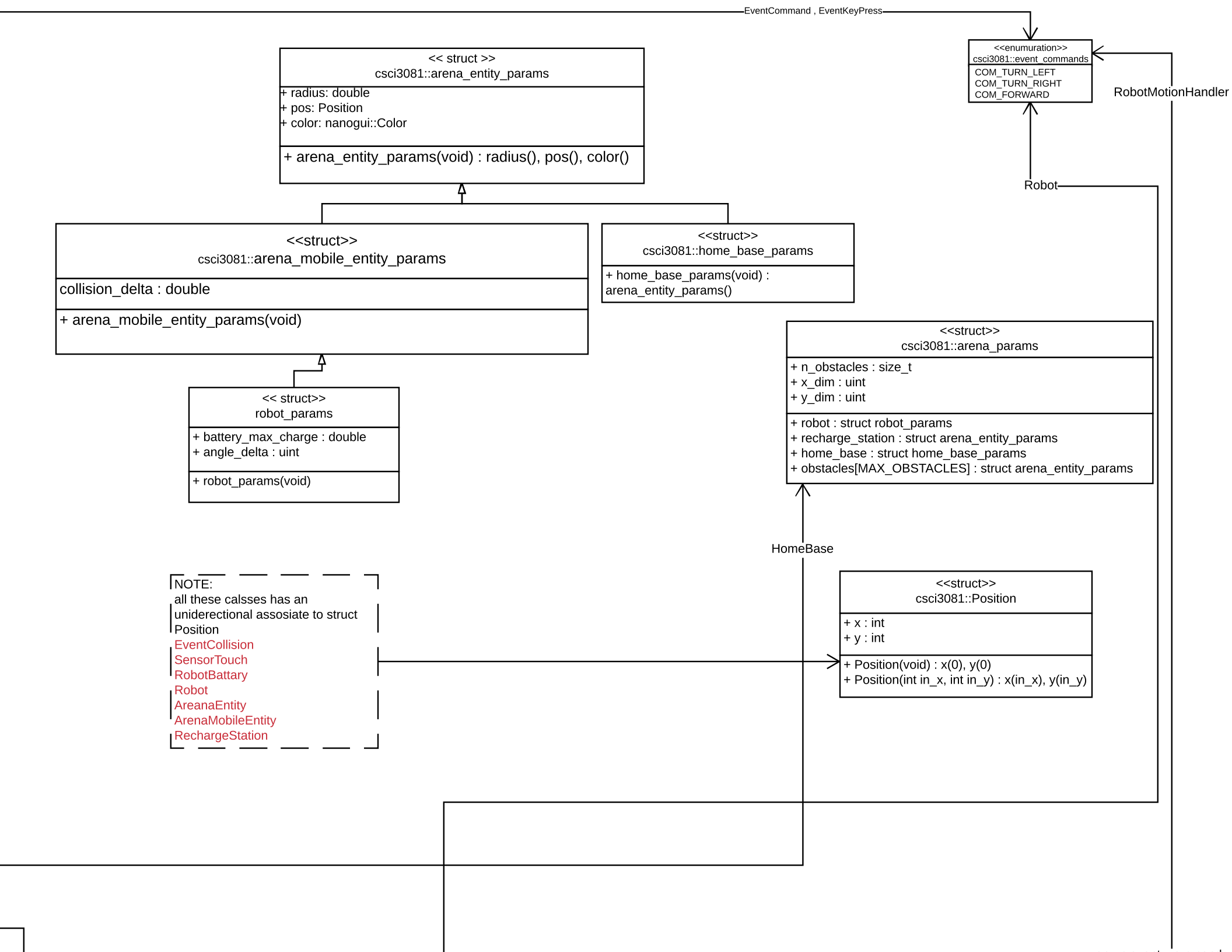
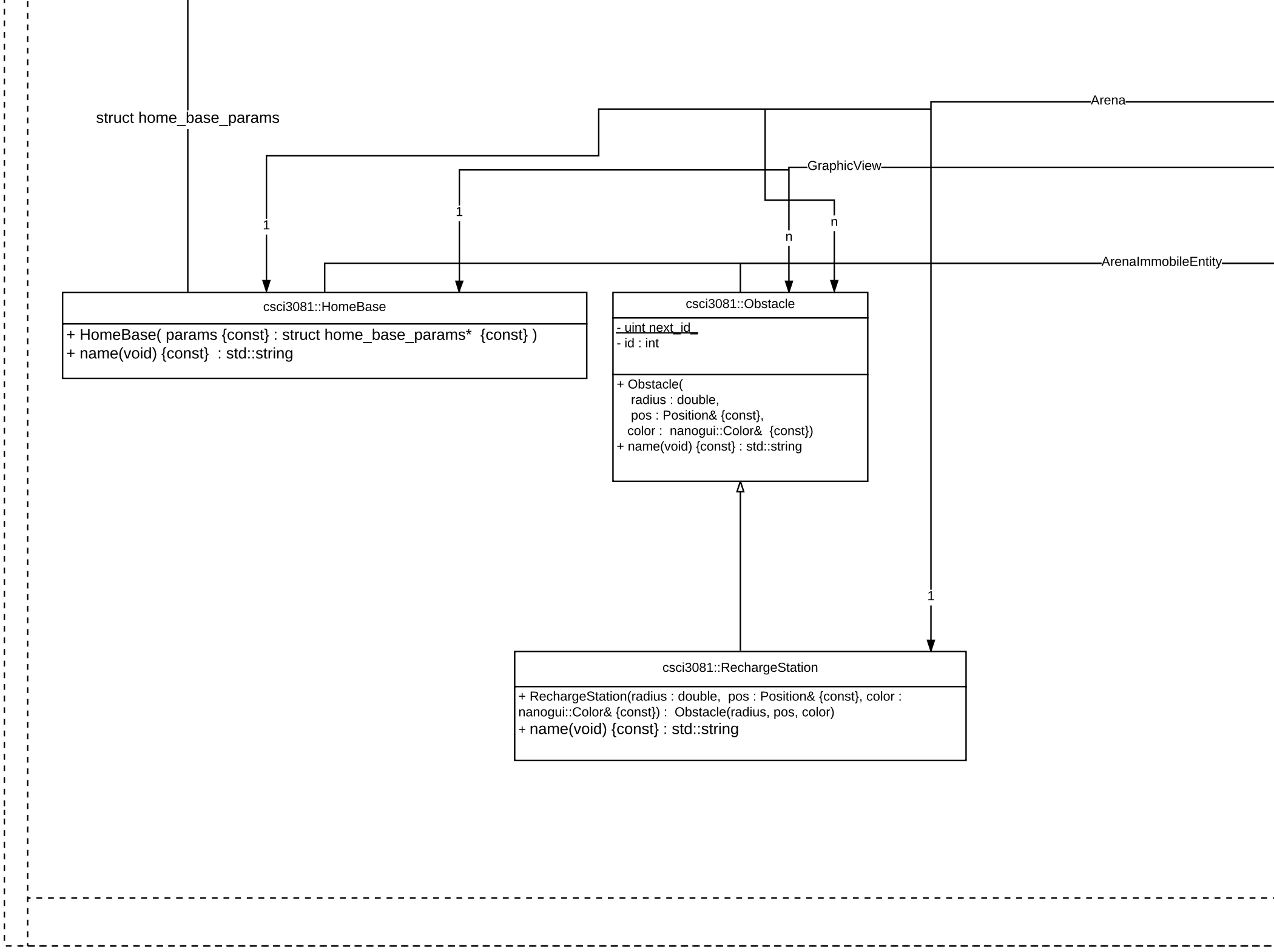
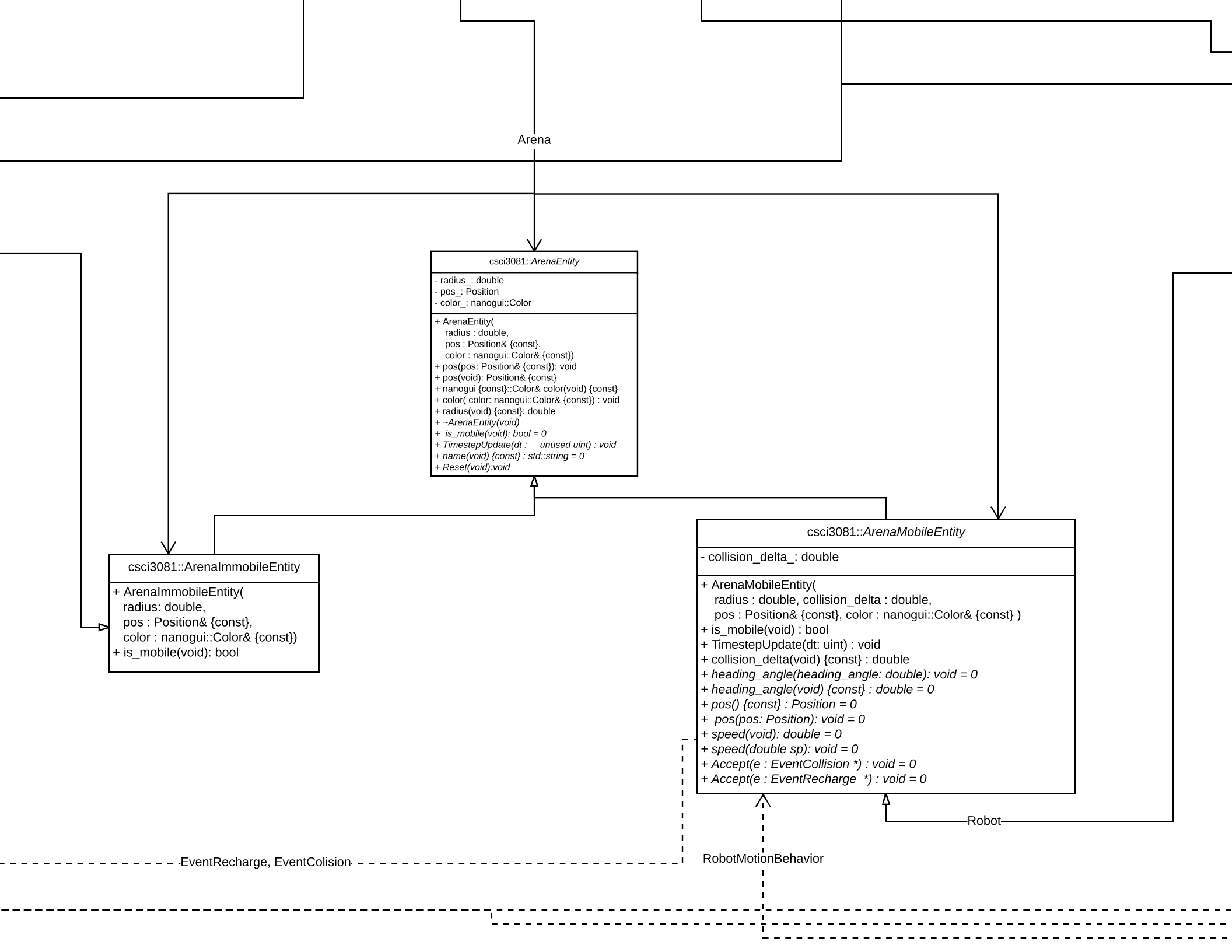
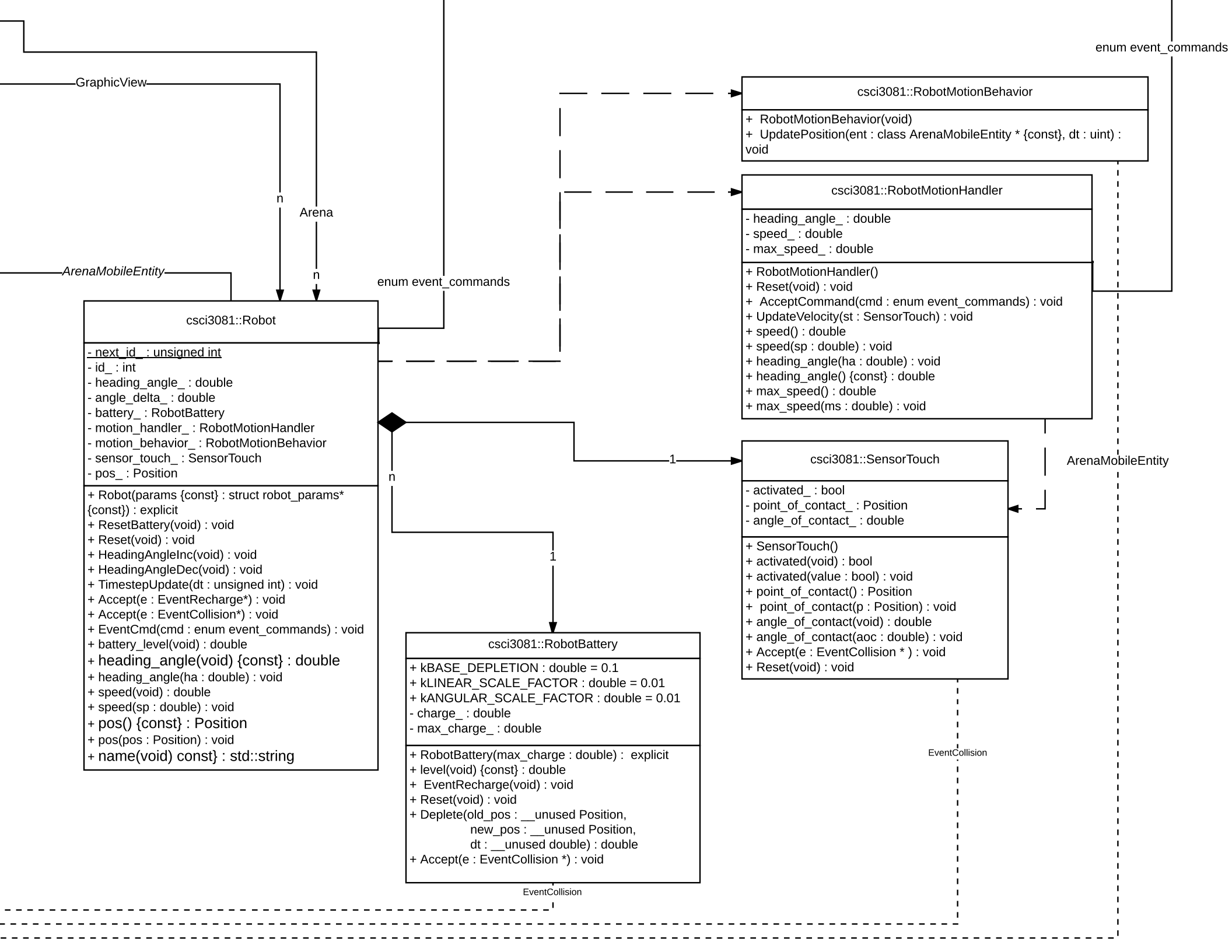

enum event_commands
csci3081::EventBaseClass
+ EventBaseClass(void)
+ ~EventBaseClass(void)
+ EmitMessage(void) : void = 0
csci3081::EventRecharge
+ EventRecharge(void)
+ EmitMessage(void) : void
csci3081::EventCollision
- collided_ : bool
- point_of_contact_ : Position
- angle_of_contact_ : double
+ EventCollision()
+ EmitMessage(void) : void
+ collided() : bool
+ collided(c : bool) : void +
point_of_contact() : Position
+ point_of_contact(p : Position) : void
+ angle_of_contact() : double
+ angle_of_contact(aoc : double) : void
csci3081::EventKeypress
- key_ : int
- operator : EventKeypress& = (other : const EventKeypress&) = delete
- EventKeypress(other : const EventKeypress&) = delete
+ EventKeypress(key : int) : explicit
+ EmitMessage(void): void
- keypress_to_cmd(key : int) : enum event_commands
csci3081::EventCommand
- cmd_ : enum event_commands
+ EventCommand(cmd : enum event_commands) : explicit
+ EmitMessage(void) : void
+ cmd(void) {const} : enum event_commands
ArenaMobileEntity
ArenaMobileEntity
RobotBattary
SencorTouch
ArenaEntity and Arena


csci3081::Arena
- operator : Arena& =(other : Arena& {const}) = delete
- Arena(other : Arena& {const}) = delete
- x_dim_ : double
- y_dim_ : double
- n_robots_ : unsigned int
- n_obstacles_ : unsigned int
- robot_ : Robot*
- recharge_station_ : RechargeStation *
- home_base_ : HomeBase *
- entities_: std::vector<class ArenaEntity*>
- mobile_entities_ : std::vector<class ArenaMobileEntity*>
+ Arena(params {const} : struct arena_params * {const}) : explicit
+ ~Arena(void)
+ AdvanceTime (dt : unsigned int): void
+ Reset(void):void
+ n_robots(void) : unsigned int
+ n_obstacles(void) : unsigned int
+ obstacles(void) : std::vector<class Obstacle*>
+ mobile_entities(void) : std::vector<class ArenaMobileEntity*>
+ robot(void) {const} : Robot*
+ home_base(void) {const} : HomeBase*
+ recharge_station(void) {const} : RechargeStation*
- CheckForEntityCollision(
ent1 {const} : class ArenaEntity* {const},
ent2 {const} : class ArenaEntity* {const},
ec : EventCollision *,
collision_delta: double) : void
- CheckForEntityOutOfBounds(
ent {const} : class ArenaMobileEntity* {const},
ec : EventCollision *) : void
- UpdateEntitiesTimestep(void) : void
GrsphicApp
csci3081::GraphicsArenaViewer
- arena_ : Arena *
- paused_ : bool
- pause_btn_ : nanogui::Button *
- operator : GraphicsArenaViewer& =(other : GraphicsArenaViewer& {const}) = delete
- GraphicsArenaViewer(other : GraphicsArenaViewer& {const}) = delete
+ GraphicsArenaViewer(params {const} : struct arena_params* {const}) : explicit
+ ~GraphicsArenaViewer(void)
+ UpdateSimulation(dt : double) : void
+ OnRestartBtnPressed() : void
+ OnPauseBtnPressed() : void
+ OnMouseMove(x : int, y : int) : void
+ OnLeftMouseDown(x : int, int y : int) : void
+ OnLeftMouseUp(x : int, int y : int) : void
+ OnRightMouseDown(x : int, y : int) : void
+ OnRightMouseUp(x : int, int y : int) : void
+ OnKeyDown(c : char* {const}, modifiers : int) : void
+ OnKeyUp(c : char* {const}, 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) {const} : Arena*
- DrawRobot(ctx : NVGcontext*, robot {const} : class Robot* {const}) : void
- DrawObstacle(ctx : NVGcontext*, obstacle {const} : class Obstacle* {const}) : void
- DrawHomeBase(ctx : NVGcontext*, home {const} : class HomeBase* {const}) : void
1
1
1
1
Robot
1
RecharegedStation, HomeBase, Obstacle
Robot, obstacle, HomeBase
ArenaEntity, ArenaImmobileEntity, ArenaMobileEntity
EventBaseClass


EventCommand , EventKeyPress
<<enumuration>>
csci3081::event_commands
COM_TURN_LEFT
COM_TURN_RIGHT
COM_FORWARD
RobotMotionHandler
<< struct >>
csci3081::arena_entity_params
+ radius: double
+ pos: Position
+ color: nanogui::Color
+ arena_entity_params(void) : radius(), pos(), color()
Robot
<<struct>>
csci3081::arena_mobile_entity_params
collision_delta : double
+ arena_mobile_entity_params(void)
<<struct>>
csci3081::home_base_params
+ home_base_params(void) :
arena_entity_params()
<<struct>>
csci3081::arena_params
+ n_obstacles : size_t
+ x_dim : uint
+ y_dim : uint
+ robot : struct robot_params
+ recharge_station : struct arena_entity_params
+ home_base : struct home_base_params
+ obstacles[MAX_OBSTACLES] : struct arena_entity_params
<< struct>>
robot_params
+ battery_max_charge : double
+ angle_delta : uint
+ robot_params(void)
HomeBase
NOTE:
all these calsses has an
uniderectional assosiate to struct
Position
EventCollision
SensorTouch
RobotBattary
Robot
AreanaEntity
ArenaMobileEntity
RechargeStation
<<struct>>
csci3081::Position
+ x : int
+ y : int
+ Position(void) : x(0), y(0)
+ Position(int in_x, int in_y) : x(in_x), y(in_y)


struct home_base_params
Arena
GraphicView
ArenaImmobileEntity
1
1
n
n
1
csci3081::HomeBase
+ HomeBase(params {const} : struct home_base_params* {const})
+ name(void) {const} : std::string
csci3081::Obstacle
- uint next_id_
- id : int
+ Obstacle(
radius : double,
pos : Position& {const},
color : nanogui::Color& {const})
+ name(void) {const} : std::string
csci3081::RechargeStation
+ RechargeStation(radius : double, pos : Position& {const}, color :
nanogui::Color& {const}) : Obstacle(radius, pos, color)
+ name(void) {const} : std::string


Arena
csci3081::ArenaEntity
- radius_: double
- pos_: Position
- color_: nanogui::Color
+ ArenaEntity(
radius : double,
pos : Position& {const},
color : nanogui::Color& {const})
+ pos(pos: Position& {const}): void
+ pos(void): Position& {const}
+ nanogui {const}::Color& color(void) {const}
+ color(color: nanogui::Color& {const}) : void
+ radius(void) {const}: double
+ ~ArenaEntity(void)
+ is_mobile(void): bool = 0
+ TimestepUpdate(dt : __unused uint) : void
+ name(void) {const} : std::string = 0
+ Reset(void):void
csci3081::ArenaImmobileEntity
+ ArenaImmobileEntity(
radius: double,
pos : Position& {const},
color : nanogui::Color& {const})
+ is_mobile(void): bool
csci3081::ArenaMobileEntity
- collision_delta_: double
+ ArenaMobileEntity(
radius : double, collision_delta : double,
pos : Position& {const}, color : nanogui::Color& {const})
+ is_mobile(void) : bool
+ TimestepUpdate(dt: uint) : void
+ collision_delta(void) {const} : double
+ heading_angle(heading_angle: double): void = 0
+ heading_angle(void) {const} : double = 0
+ pos() {const} : Position = 0
+ pos(pos: Position): void = 0
+ speed(void): double = 0
+ speed(double sp): void = 0
+ Accept(e : EventCollision *) : void = 0
+ Accept(e : EventRecharge *) : void = 0
Robot
EventRecharge, EventColision
RobotMotionBehavior


GraphicView
n
Arena
ArenaMobileEntity
n
enum event_commands
enum event_commands
csci3081::RobotMotionBehavior
+ RobotMotionBehavior(void)
+ UpdatePosition(ent : class ArenaMobileEntity * {const}, dt : uint) : void
csci3081::RobotMotionHandler
- heading_angle_ : double
- speed_ : double
- max_speed_ : double
+ RobotMotionHandler()
+ Reset(void) : void
+ AcceptCommand(cmd : enum event_commands) : void
+ UpdateVelocity(st : SensorTouch) : void
+ speed() : double
+ speed(sp : double) : void
+ heading_angle(ha : double) : void
+ heading_angle() {const} : double
+ max_speed() : double
+ max_speed(ms : double) : void
csci3081::Robot
- next_id_ : unsigned int
- id_ : int
- heading_angle_ : double
- angle_delta_ : double
- battery_ : RobotBattery
- motion_handler_ : RobotMotionHandler
- motion_behavior_ : RobotMotionBehavior
- sensor_touch_ : SensorTouch
- pos_ : Position
+ Robot(params {const} : struct robot_params* {const}) : explicit
+ 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
+ EventCmd(cmd : enum event_commands) : void
+ battery_level(void) : double
+ heading_angle(void) {const} : double
+ heading_angle(ha : double) : void
+ speed(void) : double
+ speed(sp : double) : void
+ pos() {const} : Position
+ pos(pos : Position) : void
+ name(void) const} : std::string
1
csci3081::SensorTouch
- activated_ : bool
- point_of_contact_ : Position
- angle_of_contact_ : double
+ SensorTouch()
+ activated(void) : bool
+ activated(value : bool) : void
+ point_of_contact() : Position
+ point_of_contact(p : Position) : void
+ angle_of_contact(void) : double
+ angle_of_contact(aoc : double) : void
+ Accept(e : EventCollision *) : void
+ Reset(void) : void
ArenaMobileEntity
n
1
csci3081::RobotBattery
+ kBASE_DEPLETION : double = 0.1
+ kLINEAR_SCALE_FACTOR : double = 0.01
+ kANGULAR_SCALE_FACTOR : double = 0.01
- charge_ : double
- max_charge_ : double
+ RobotBattery(max_charge : double) : explicit
+ level(void) {const} : double
+ EventRecharge(void) : void
+ Reset(void) : void
+ Deplete(old_pos : __unused Position,
new_pos : __unused Position,
dt : __unused double) : double
+ Accept(e : EventCollision *) : void
EventCollision
EventCollision

Robot Software

According to the code which programmed by 3081 Staff of Fall 2017, there is an Arena class which is the primary one. If the Arena is gone, so there are no any other classes. the possible actions, which are applied on Arena, are programmed in the GraphicArenaViewer. OnLeftMouseUp, OnPauseBtnPressed or OnRestartBtnPressed are examples of these user-controlled actions. Furthermore, there are many entities are in the arena. Some of these entities are moving like the robots, while the others are steady in their places such as home and recharge station. The robot has sensor and battery which are formed in separated classes, and the robot is moving around. the robot is using the sensor to avoid colliding with the stable obstacles. After a while, the battery of the robot needs to recharge by visiting the recharged station. All in all, the events, which are done on the robot to control, is pressing the keys, recharging the battery and colliding other objects such as obstacles if the sensor does not work effectively.

Syreen Bnabilah