

<i>Entity.h</i>
hitbox -> rectangular hitbox object velocity -> speed of object set_alive -> boolean damage_type -> String health -> int epipen_cost -> int attacking -> boolean row_based -> boolean
update() -> update current object's position, attacks, and lifetime resolveCollision() -> attack when collision borders are touching getDrawable() -> get entities that need drawn



<i>Entities.h</i>
makeProjectileEntities() -> functions that make projectile entities makePlantEntities() -> functions that make plant entities makeZombieEntities() -> functions that make zombie entities makeStorePlants() -> make the plants that are displayed and are draggable from shop

Greatly simplified,
each group of functions
set the attributes of *all*
entities in that section

So makeZombieEntities()
includes functions that return
entities like SoyBoy,
DiabeticDave, and so on

<i>GameController.h</i>
plants -> list of plants entities zombies -> list of zombie entities projectiles -> list of projectile entities mutex_lock
updateList() -> update list of entites stored in lists processCollisions() -> process entity hitbox collisions update() -> update entites shown on screen swapBuffers() cleanup()

<i>World.h</i>
time -> float epipens -> int
update() -> update the game screen getDrawable() -> get the sprites for entities getMousePosition() -> get position of player's mouse processCollisions() -> process collisions remove_dead() -> remove dead entities from game screen update_list() -> update all entities in a list

<i>ComponentDefinitions.h</i>
makeMovementComponent() -> set an entity's position makeAttackComponent() -> set an entity's attack interval and damage makeGraphicsComponent() -> sets an entity's sprites ClickableMovementComponent() -> for entities that are clickable (epipens)

Also simplified