id: int # graph: const IGraph* + GetSpeed(): float + GetPosition(): Vector3 + GetDirection(): Vector3 + GetDestination(): Vector3 + GetDetails(): JsonObject + GetAvailability(): bool + GetId(): int + Rotate(angle : double) : void + SetPosition(pos : Vector3) : void + SetDirection(dir_ : Vector3) : void + SetDestination(des : Vector3) : void + SetGraph(graph : IGraph*) : void Robot Drone UtilityDecorator - details : JsonObject - myDrone : IEntity* - position : Vector3 - repairStations : vector<IEntity*> - details : JsonObject - direction : Vector3 - durability : float - position : Vector3 - destination : Vector3 - direction : Vector3 - speed : float + UtilityDecorator(drone : IEntity*) - destination : Vector3 - durability: float + GetDurability(): float - speed : float - available : bool + GetSpeed(): float - available : bool - nearestEntity : IEntity* + GetPosition(): Vector3 - nearestEntity: IEntity* + GetDirection(): Vector3 + GetDurability() : float + GetDestination(): Vector3 + GetSpeed(): float + GetDetails(): JsonObject + GetPosition(): Vector3 + GetAvailability(): bool + GetDirection(): Vector3 + GetNearestEntity(scheduler: + GetSpeed(): float + GetDestination(): Vector3 std::vector<IEntity*>): void + GetPosition(): Vector3 + GetDetails(): JsonObject + Update(dt : double, scheduler : + GetDirection(): Vector3 + GetAvailability(): bool std::vector<IEntity*>): void + GetDestination(): Vector3 + Rotate(angle : double) : void + GetNearestEntity(scheduler : std::vector<IEntity*>) : void + GetDetails() : JsonObject + Update(dt : double, scheduler : std::vector<IEntity*>) : void + SetDurability(durability : double) : void + GetAvailability(): bool + Rotate(angle : double) : void + SetPosition(pos : Vector3) : void + GetNearestEntity(scheduler : std::vector<IEntity*>) : void + SetDurability(durability_: float): void + SetDirection(dir_: Vector3): void + Update(dt : double, scheduler : std::vector<IEntity*>) : void + SetPosition(pos_ : Vector3) : void + SetDestination(des : Vector3) : void + Rotate(angle : double) : void + SetDirection(dir : Vector3) : void + SetPosition(pos_: Vector3): void + SetDestination(des : Vector3) : void + SetDirection(dir : Vector3) : void + SetDestination(des : Vector3) : void DurabilityDecorator + DurabilityDecorator(drone : IEntity*, repairStations : vector<IEntity*>) + Update(dt : double, scheduler : std::vector<IEntity*>): void

<<interface>>
IEntity