IEntitiyDecorator # entity: IEntity* # entityList : std::vector<IEntity*>* <<constructor>> + IEntityDecorator(IEntity* wrapped) <<misc>> + virtual void OverridedUpdate(double dt, std::vector<IEntity*> scheduler) = 0 + virtual GetSpeed() const : float {abstract} + virtual GetPosition() const : Vector3 {abstract} + virtual GetDirection() const : Vector3 {abstract} + virtual GetDestination() const : Vector3 {abstract} + virtual GetColor() const : std::string + virtual GetDetails() const : JsonObject {abstract} + virtual GetAvailability() const : bool + virtual SetAvailability() const : bool + Update(double dt, std::vector<IEntity*> scheduler) : void + virtual SetGraph(const IGraph* graph): void + virtual SetPosition(Vector3): void + virtual SetDirection(Vector3) : void + virtual SetDestination(Vector3): void + virtual SetColor(std::string) : void + virtual Rotate(double): void + virtual Jump(double): void + IEntityDecorator(const IEntityDecorator&) = delete + IEntityDecorator& operator=(const IEntityDecorator&) = delete

BatteryDecorator

+ maxCharge : const double + drainRate : const double

- charge : double - recharging : bool

- toRechargeStation: IStrategy* - targetRechargeStation: IEntity*

<<constructor>>

+ BatteryDecorator(wrapped: IEntity*, maxCharge: double, drainRate: double)

+ BatteryDecorator(wrapped: IEntity*, obj: JsonObject&)

+ OverridedUpdate(dt: double, scheduler: std::vector<IEntity*>)

+ recharge(amount: double): bool

+ getCharge() const: double

+ IsOrWillBeMarooned(): bool

IEntity