Move System

+elapsedTime: double

+step(double elapsedTime): step through all GameObjects with Velocity component and change their position accordingly

Collider System

+elapsedTime: double

+step(double elapsedTime): step through all GameObjects with Collider component and use the Lambda function provided by each GameObject's Collider to check condition and update each GameObject's status

Enemy

+ Position
+ Physics Behavior
+ Health
+ Mass
+ Attack
+ SimpleAI
+ Generator
+ Velocity
+ Collider

Stone

+ Position
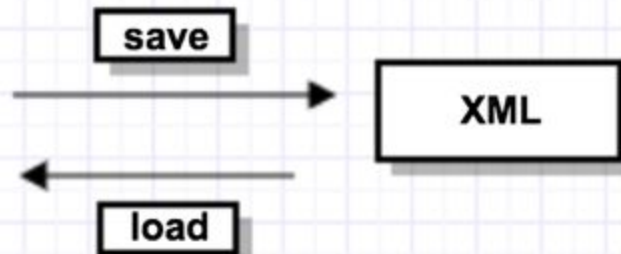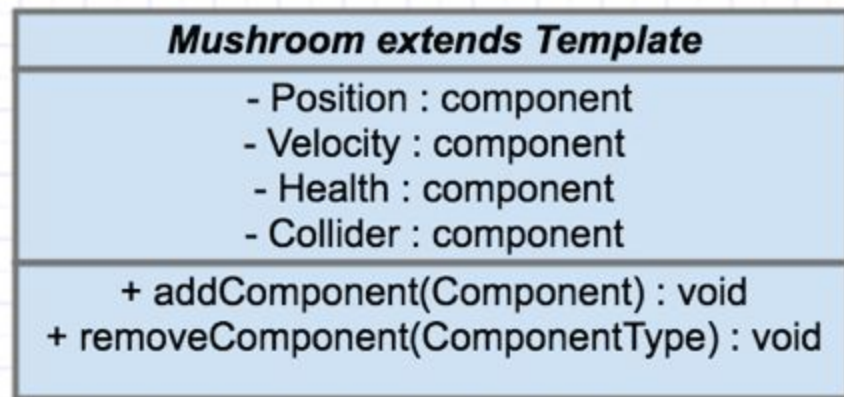+ Physics Behavior
+ Health
+ Mass
+ Collider

gameobject : Goals

- PositionGoal : component
- TimeGoal : component
- BossGoal : component
- MonsterGoal : component
- ScoreGoal : component

system : GoalsSystem

- PositionGoalSystem
- TimeGoalSystem
- BossGoalSystem
- MonsterGoalSystem
- ScoreGoalSystem

Mushroom extends Template

- Position : component
- Velocity : component
- Health : component
- Collider : component

+ addComponent(Component) : void
+ removeComponent(ComponentType) : void

save → **XML**

load ←

add to inventory

Inventory implements Iterable<Template>

+ addTemplate(Template) : void
+ getTemplate() : Template
+ removeTemplate() : void

PhysicsEngine

updateKinematics(IGameobject obj)

getCollisionSurface(IGameobject a, IGameobject b)

getColliders(IGameobject obj)

<<Interface>>
IPhysicsStrategy

+ updatePosition(IGameobject);
+ handleCollision(Type, IGameobject);

Implements

DefaultStrategy

IGameObjects contain predefined (added in authoring environment) strategies

Extends

Extends

ProjectileStrategy

CharacterStrategy