

ECS
<ul style="list-style-type: none"> <li>- double _time</li> <li>- std::size_t _livingEntities</li> <li>- std::queue&lt; Entity &gt; _freeEntities</li> <li>- std::unordered_map&lt; Entity, Signature &gt; _signatures</li> <li>- std::vector&lt; void * &gt; _componentStorages</li> <li>+ ECS()</li> <li>+ ~ECS()=default</li> <li>+ Entity createEntity()</li> <li>+ void killEntity(Entity e)</li> <li>+ Signature getSignature(Entity e) const</li> <li>+ void setSignature(Entity e, Signature sig)</li> <li>+ Entity addComponent(Entity e, const ComponentType &amp;component)</li> <li>+ bool hasComponent(Entity e) const</li> <li>+ const ComponentType * getConstComponent(Entity e) const</li> <li>+ ComponentType * getComponent(Entity e)</li> <li>+ std::vector&lt; Entity &gt; getEntitiesByComponents()</li> <li>+ void killComponent(Entity e)</li> <li>+ double getTime() const</li> <li>+ void addTime(double dt)</li> <li>- void verifyComponentStorage(ComponentTypeID TypeID)</li> </ul>

#\_ecs



ISystem
<ul style="list-style-type: none"> <li>+ ISystem(ECS &amp;ecs)</li> <li>+ virtual ~ISystem() =default</li> <li>+ virtual void update(double dt)=0</li> </ul>



CollisionSystem
<ul style="list-style-type: none"> <li>+ CollisionSystem(ECS &amp;ecs)</li> <li>+ void update(double dt) override</li> <li>- bool checkAABB(const Components::Position &amp;pos1, const Components::Collider &amp;coll1, const Components::Position &amp;pos2, const Components::Collider &amp;coll2) const</li> <li>- void resolveCollision(Components::Position &amp;mobilePos, const Components::Collider &amp;mobileColl, const Components::Position &amp;staticPos, const Components::Collider &amp;staticColl) const</li> </ul>