

## ECS

```
- double _time
- std::size_t _livingEntities
- std::queue< Entity > _freeEntities
- std::unordered_map< Entity, Signature > _signatures
- std::vector< void * > _componentStorages
+ ECS()
+ ~ECS()=default
+ Entity createEntity()
+ void killEntity(Entity e)
+ Signature getSignature(Entity e) const
+ void setSignature(Entity e, Signature sig)
+ Entity addComponent(Entity e, const ComponentType &component)
+ bool hasComponent(Entity e) const
+ const ComponentType * getConstComponent(Entity e) const
+ ComponentType * getComponent(Entity e)
+ std::vector< Entity > getEntitiesByComponents()
+ void killComponent(Entity e)
+ double getTime() const
+ void addTime(double dt)
- void verifyComponentStorage(ComponentTypeID TypelID)
```

#\_ecs

## ISystem

```
+ ISystem(ECS &ecs)
+ virtual ~ISystem() =default
+ virtual void update(double dt)=0
```

## InputSystem

```
- std::map< sf::Keyboard::Key, GameAction > m_keyMappings
- std::map< unsigned int, GameAction > m_joystickButtonMappings
- std::map< GameAction, bool > m_currentActionStates
- std::map< GameAction, bool > m_previousActionStates
- unsigned int m_joystickId
- float m_deadzone
- float m_playerSpeed
+ InputSystem(ECS &ecs)
+ ~InputSystem()=default
+ void update(double dt) override
+ bool isActionActive(GameAction action) const
+ bool wasActionPressed(GameAction action) const
+ bool wasActionReleased(GameAction action) const
+ void setKeyMapping(sf::Keyboard::Key key, GameAction action)
+ void setJoystickButtonMapping(unsigned int button, GameAction action)
+ void setDeadzone(float deadzone)
- void setDefaultMappings()
- void updateJoystickInput()
- void applyInputToPlayers()
```

-inputSystem

## GameEngine

```
- sf::RenderWindow window
- bool isRunning
+ GameEngine()
+ ~GameEngine()
+ void run()
+ GameEngine()
+ ~GameEngine()
+ void run()
- void processInputs()
- void update()
- void render()
- void processInputs()
- void update()
- void render()
```