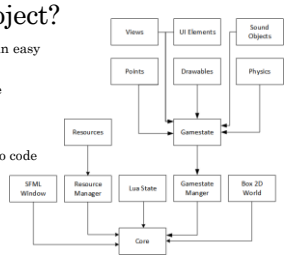
2D Game Engine

By Cyrus Hanlon

What is my project?

- 2D Game engine that provides an easy to use and powerful Lua API
- High performance and low game development times
- Easily expandable
- Complete, users will only need to code in Lua



Major features - 1

Simple to use Lua API – (Peer review proves that)

ECS – (Easily Expandable without much development experience new components can be added to the game engine)

SFML – (Good performance with low development overhead)

Animations – (All handled by the engine so the Lua developer doesn't need in depth knowledge to implement polished games)

Major features - 2

Box2D – (Used in 2D games on all devices so has proven track record)

Resource management – (Lua developers don't need to bother with directly managing assets, they can refer to them by path)

Input Handling - (Takes out all direct hardware access from the Lua developer, could be expanded to handle controllers etc)

Expandability – (All systems are robust and generalised so third party developers can easily add new features)

How it works

```
local box = {}
box.drawable = DrawableComponent.New()
box.drawable:SetTexture("resources/textures/box.png")
box.physics = PhysicsComponent.New(
    b2BodyDef.New({
        active = true,
        type = "dynamic"
    }),
    b2FixtureDef.New({
        density = 1,
        friction= 1,
        restitution = 0.3--math.random( 50 ) / 100
    }),
    1, 1)
box.entity = Entity.New("ball"..#boxes)
box.entity:AddDrawable(box.drawable)
box.entity:AddPhysics(box.physics)
```

- This code is simple in Lua
- Behind the scenes a lot is going on

How it works

```
local box = {}
box.drawable = DrawableComponent.New()
box.drawable:SetTexture("resources/textures/box.png")
box.physics = PhysicsComponent.New(
    b2BodyDef.New({
        active = true,
        type = "dynamic"
    }),
    b2FixtureDef.New({
        density = 1,
        friction= 1,
        restitution = 0.3--math.random( 50 ) / 100
    }),
    1, 1)
box.entity = Entity.New("ball"..#boxes)
box.entity:AddDrawable(box.drawable)
box.entity:AddPhysics(box.physics)
```

- Example of creating a box in Lua
- Create both component types
- Create entity
- Add the components to the entity

How it works

```
local box = {}
box.drawable = DrawableComponent.New()
box.drawable:SetTexture("resources/textures/box.png")
box.physics = PhysicsComponent.New(
    b2BodyDef.New({
        active = true,
        type = "dynamic"
    }),
    b2FixtureDef.New({
        density = 1,
        friction= 1,
        restitution = 0.3--math.random( 50 ) / 100
    }),
    1, 1)
box.entity = Entity.New("ball"..#boxes)
box.entity:AddDrawable(box.drawable)
box.entity:AddPhysics(box.physics)
```

- Create userdata objects
- Insert components into appropriate containers in the gamestate
- Request a resource from the resource manager and set it on the drawable
- Create Box2D physics object and add to the Box2D world
- Create entity and place it in the entity container in the gamestate
- Attach the components to the entity by setting the parent property

Evaluation - Peer Review

- Jim | 27
  - Programmer with 3 years professional experience
  - 5+ years making game modes for Garrysmod in Lua
- Wessel | 20
  - 1 year of education in programming
  - CS Student
- Roman | 20
  - 1 year of education in programming
  - CS Student
- Henry | 23
  - Personal interest in programming
  - Engineering graduate

Examples

- Snake
  - Simple complete game example
- Fall
  - Box2d and SFML camera example
- Quarry
  - Made by Jim
- Zombie Survival
  - Made by Henry