

ГОСУДАРСТВЕННОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ
ВЫСШЕГО ПРОФЕССИОНАЛЬНОГО ОБРАЗОВАНИЯ
«ДОНЕЦКИЙ НАЦИОНАЛЬНЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»

Факультет ИСП

Кафедра ПИ

Лабораторная работа №4
по курсу: «Профессиональная практика программной инженерии»
по теме: «Создание самодокументирующегося кода»

Выполнил:

ст. гр. ПИ-19в

Покровский Н.Ю.

Проверил:

Филипишин Д.А.

Дмитрюк Т.Г.

ДОНЕЦК – 2023

Поскольку проект пишется на Lua, то для создания автоматической документации можно использовать специальную утилиту – luadoc.

```
-- Конструктор сущности
function entity:new(...)
    object.new(self, ...)
    -- Animation state
    self.state = "idle"
    self.animations = {}
    for i = 1, #possible_states do
        local state = possible_states[i]
        self.animations[state] = animation(assets.frames[self.name][state])
    end
    self.look = 1
    -- Bind attributes
    self.speed = self.speed
    self.health = self.health
    self.shield = 0
    self.timer = nil
    self.smacked_on_wall = false
    self.smacked_on_wall_x = false
    self.smacked_on_wall_y = false
end

-- Метод обновления сущности
function entity:update(dt)
    -- Reset state, if needed
    if self.state == "moving" and self.velocity:len2() == 0 then
        self.state = "idle"
    end
end
```

Рисунок 1 – Пример использования автоматической документации

LuaDoc	
Index	Files
Files	
animation.lua	animation.lua
assets.lua	assets.lua
background_effect.lua	background_effect.lua
camera.lua	camera.lua
class.lua	class.lua
conf.lua	conf.lua
entity.lua	entity.lua
level.lua	level.lua
main.lua	main.lua
object.lua	object.lua
scene.lua	scene.lua
timer.lua	timer.lua
vec2.lua	vec2.lua

Рисунок 2 – Структура документации

File `entity.lua`

Functions

<code>entity:draw</code> (sb)	Метод отрисовки сущности
<code>entity:heal</code> (v)	Вылечить сущность
<code>entity:hurt</code> (v)	Нанести урон сущности
<code>entity:kill</code> ()	Уничтожить сущность
<code>entity:move</code> (x, y)	Передвинуть сущность
<code>entity:new</code> (...)	Конструктор сущности
<code>entity:safe_set_state</code> (state)	Установка состояния со сбросом таймера
<code>entity:stun</code> (t)	Оглушить сущность
<code>entity:throw</code> (t, velocity)	Бросить сущность в направлении
<code>entity:update</code> (dt)	Метод обновления сущности

Рисунок 3 – Пример раздела документации