VG Engine

Tutorial



# handling scenes

# Using GameObjects

GameObject \*object = new GameObject("ship");

# Components

## adding components

Include the appropiate header file.

#include <engine/game/triangleComponent.h>

#include <engine/game/quadrangleComponent.h>

#include <engine/game/transformComponent.h>

TransformComponent \*transform = new TransformComponent(Vector2<int>(64, 64),

Vector2<int>(80, 80), 0.0f, 0u);

object->addComponent(transform);

### making your own components

MyComponent.cpp

#include "MyComponent.h"

#include <engine/game/game.h>

MyComponent::MyComponent() :vg::Component()

{

}

TestComponent::~TestComponent()

{

}

MyComponent.h

#pragma once

#include <engine/game/component.h>

class MyComponent :public vg::Component

{

public:

TestComponent();

~TestComponent();

};

Main.cpp

MyComponent \*myComponent = new MyComponent();

object->addComponent(myComponent);

# Drawing 2D stuff

# Text and fonts

# Handling time

# Input

# Playing sounds and music

Sound\* testSound = new Sound("soundsample.mp3");

testSound->load(Game::getInstance()->getFileManager());

Game::getInstance()->getAudioManager()->addSound(\*testSound);

OR BY ID

Sound\* testSound = new Sound("soundsample.wav");

testSound->load(game->getInstance()->getFileManager());

Game::getInstance()->getAudioManager()->addSound("music1", \*testSound);

Game::getInstance()->getAudioManager()->Play("music1");

# Debugging