# Technical Document

**Project:** Ocean Ring Toss

**Author:** GameLoft Internship Team A

**Instructor:** Nguyen Thao Duong

## Project Overview

### Introduction

To apply knowledge about C++ and 3D which we have learned in the internship course at Gameloft, our team decide to make and develop Ocean Ring Toss, a 3D game, by using OpenGL ES 2.0 and Training Framework which is provided by the company Gameloft.

Ocean Ring Toss is a game developed for Android operating system. The game is based on Water Ring Toss game, which is usually played by children. The game’s rule is similar to the original game. To win the game, player needs to put all the given rings into the pillars which have the form of anchor.

### System requirements

* OS: Android 4.x and above
* Minimum Hardware:
* 1050 MHz Processor
* 256MB RAM
* 93MB storage available.
* Recommended Hardware:
* GHz Processor
* 128MB RAM
* OpenGL support, 96MB onboard video ram, pixel shader 1.1 supports, vertex shader 1.1 supports.
* 100MB storage available.

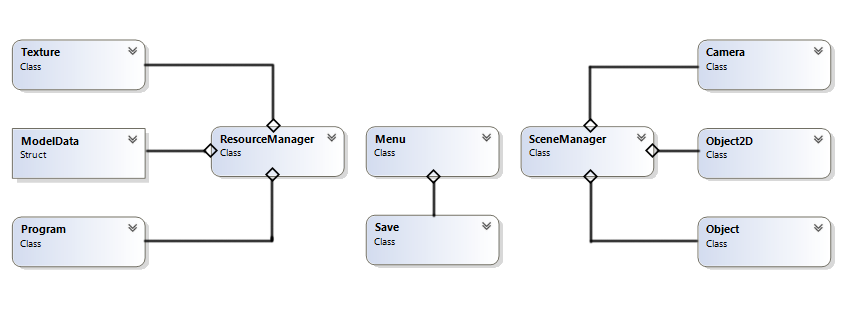
### Technical Issues

* Game could be slow on some phones.
* There are some random crashes in game.
* Graphics are blurry or crumble on some phones.
* Point checking sometime doesn’t work properly.
* Game might crash if users do not clear app properly.

### Third Party Tools

* Blender: a free 3D design tool.
* Pixlr: a free online tool for editing images.
* Visual Studio Community 2013: a free IDE tool for developing C++ program.

## Class Diagram



## Game Engine

The purpose of the game engine is to gather the resource of the game and render models on the game. There are 2 main parts in the game engine: Resource Manager and Scene Manager.

### Resource Manager

Resource Manager’s task is to load game’s resource from available files which is accessed by following the links provided in the RM.txt file. The Resource Manager also stores shader’s information from Fragment Shader file and Vertex Shader file. The game’s resource consists of 2 maim parts, which are .obj files and .tga files.

The .obj files store information of a model, which includes number, position and order of points which are needed to make the model.

The .tga files stores information of a texture.

### Scene Manager

The Scene Manager is responsible for rendering models from available resources loaded through Resource Manager, which creates the scenes in the game. Information about position, size and some other information of models is stored in SM.txt file.

By using Resource Manager and Scene Manager, scene is rendered with just the two mentioned files RM.txt and SM.txt.

## Technical

### Physics

The main technique used in the game is Physic system.

Bullet Physics is used in the game to make some physical effects, especially the collision effect, in the game.

#### Physical properties in the game

btDynamicsWorld is used to create physical properties and set gravity value for objects in the game with the function setGravity.

RigidBody is applied for models in the game.

Bullet Physics provides physical body (btRigidBody), mass and physical shape. There are many kinds of physical shape such as Sphere, Cylinder, Box, TriangleMesh and many other shapes.

getOpenGLMatrix is used in order to get information about position, location of rigidBodies, then the information is used to render respective models.

#### Point Checking

Firstly, CallBackFunc function is created to determine and deal with collisions between pillars and rings in the game.

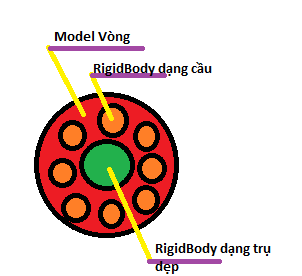
A rigidBody having a circular shape is created to represent for “hole” in a ring, and two rigidBodies which have a cylindrical shape and are on the axis of a pillar are also made. One is at the middle and the other is at the top of the pillar. When “hole” collides with the two cylindrical rigidBodies, the game will check whether the ring goes out or goes into the pillar.

Vòng: consist of

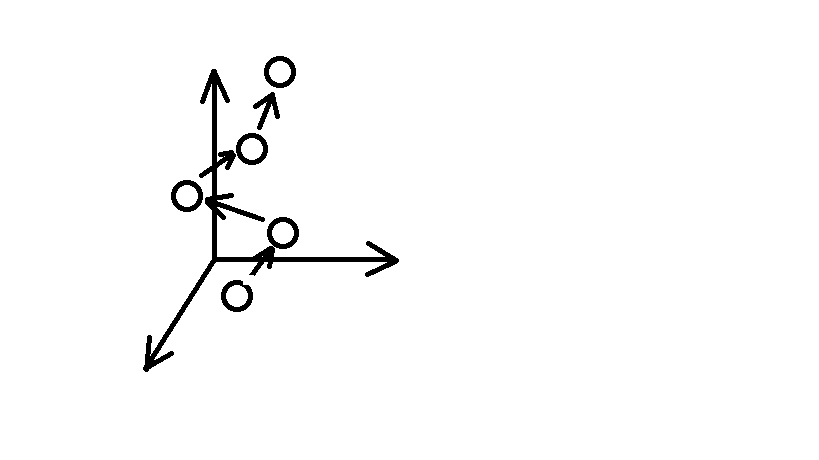
-Ring model.

-36 rigid bodies circular sphere surround ring model.

- A rigid body having a flat cylindrical shape fill in the “hole” on the ring model. This rigid model is invisible and can’t have collisions with other objects except the pillars in the game.



### Bubble

* A global variable vector named bubbles is created to keep object Bubble.
* AddBubble function: is called in function AirGun. When the function is called, the function adds a number of object bubbles to the vector bubbles. Each created bubbles will randomly appear in a set area.
* UpdateBubble function: is called in each game loop to change position of each bubble Object. Bubbles will continuous move up along the Y axis; however, bubbles will also vibrate randomly along the X and Z axes, which make the movement of bubbles more realistic.
* 

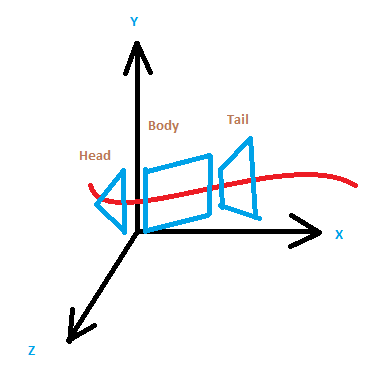
### Fish’s movement

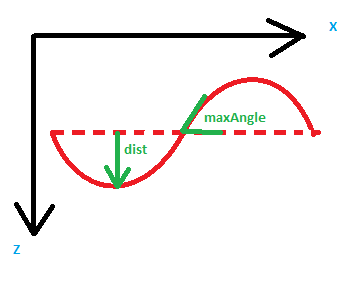
-Wanting to reduce game’s data as many as possible, animation is not created with Colada but by using code instead in the project.

-A fish model is divided into three separate parts: head, body and tail. The information of these parts is stored in a variable, struck swimmingFish. This variable is stored in a global vector, which is similar to the way bubble variable is stored.

-AddSwimminFishWithoutFuckingAnimation function allows swimmingFish to be created with desired model and texture, and puts the variable into the vector.

-UpdateSwimminFishWithoutFuckingAnimation function: is used to move head, body and tail parts of each swimmingFishes in a sin shaped orbit on a plane parallel to the Oxz plane.

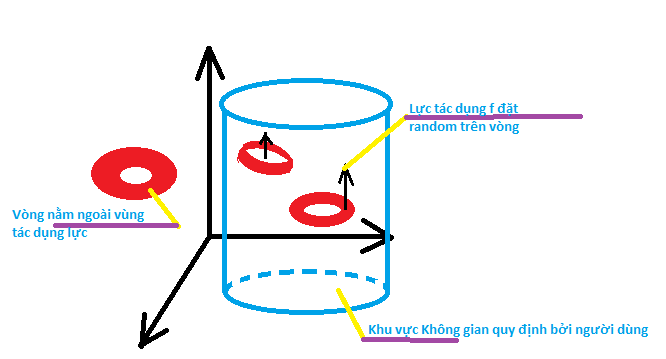




-Parameters: dist, maxAngle and movement velocity along the Ox axis can be adjusted by users.

### AirGun

- AirGun functions: When the function is called, the function will set a force f at a random position on all rings in a fixed area selected by users.

- The higher position a ring is at, the stronger force acts on the ring.

### Sound

A class is designed in Java that can call functions from MediaPlayer and SoundPool, which are two classes in Java and have function to process sound signal. This created class has function to use and process the sound resource.

### Accelerometer

Technique to get event from device with classes supported by Java is used in the game. In common devices, there exist many types of sensor: Type Accelerometer, Rotation Matrix, Proximity, and many other types of sensor. Type Accelerometer, which is used in most smart devices, is used to make gameplay for the game so that every time player tilts his or her phone while playing, a force will act on the rings in the game and make them move.

### Porting

Because the game is initially made in C++, while Android uses Java language, the game must be ported into Android. After the game is ported into Android, Jni will package C++ files into a library which has the file extension, \*.so, then the library will be read with called functions in Java.

### Water Caustic

To make water caustic effect, 2 textures will overlap each other. One texture will make sparkling effect, and the other will be water-background. Texture creating the effect will slide on the water-background texture, which make the whole water background look realistic and lively

### Picking

The Ray Casting technique is used to determine buttons and trigger the AirGuns.