**WHEN FISH GO HOME**

**TECHNICAL DESIGN DOCUMENT**

Document Version: 1.0

August 30, 2015

Author: GameLoft Internship Team B

Instructor: Nguyen Thao Duong

1. **Project Overview**
2. **Game Concept**

The player is a Fish named Aki who is currently on his way to find his children, bring them back and protect them against the Shark clan Fuun. With his courage and wit, Aki will make use of anything he finds on his way and try to shake off those sharks pursuing him and his children.

## **Technical Goals**

* Generate level’s map from file.
* Generate level’s information from file.
* Acceptable A.I for sharks and fishes using Jump Point Search.
* Collision detection against wall and objects.
* 3rd person camera view.
* Item’s effect.
* Player’s control: button touching, player’s movement using touch and move gesture.
* Menu scene.
* Level choosing scene.
* Game’s over scene
* Game’s UI control: Play, tutorial, credit, return, pause, resume, replay buttons
* Water caustic.

1. **System Requirements**

* OS: Android 4.x and above
* Minimum Hardware:
* 1200 MHz Processor
* 256MB RAM
* 5MB storage available.
* Recommended Hardware:
  + 2500 GHz Processor
  + 512MB RAM
  + OpenGL support
  + 10MB storage available.

1. **Technical Issues**

* Game could be slow on some phones.
* Minimap might not work properly.
* Game might crash if users do not clear app properly.

1. **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.

# **Game Play**

## **Game objects**

List of logical objects used in the game:

* Block: A cube object
* Wall: Consist of many blocks
* Gate: Prison’s gate, require key to unlock
* Shark: The enemy
* Big Fish: The main character that the player controls
* Small fish: The prisoner, waiting for rescue
* Starfish: Item – boost player’s speed
* Poison: Item – inverse player’s control
* Turtle: Item – slow enemy down when used
* Shellfish : Item – stun enemy when used
* Key: Quest’s item

## **Player actions**

Player will take on the body of the big fish – Aki and is capable of the following action:

* Moving around.
* Pick up item.
* Use item.
* Take small fish.

## **Victory condition**

When player and fish reach goal’s block.

# **Code Overview**

## **Framework**

Using the NewTrainingFramework provided by GameLoft.

## **Class diagram**

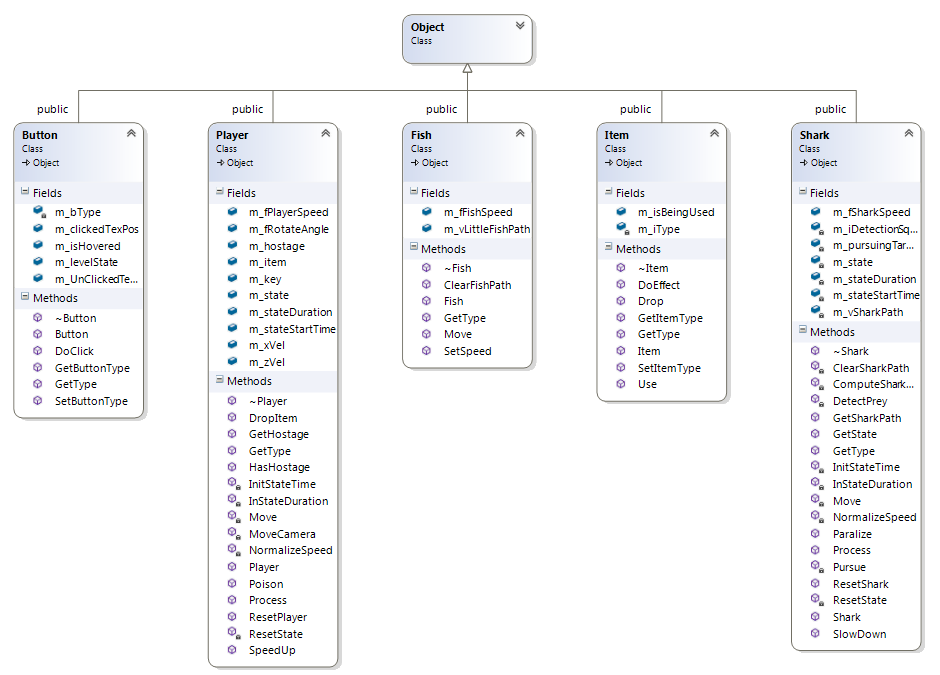


Figure III.1 Object and subclass



Figure III.2 State class and subclass



Figure III.3 The remaining classes



Figure III.4 All struct



III.5 All Enum type

## **Flow**

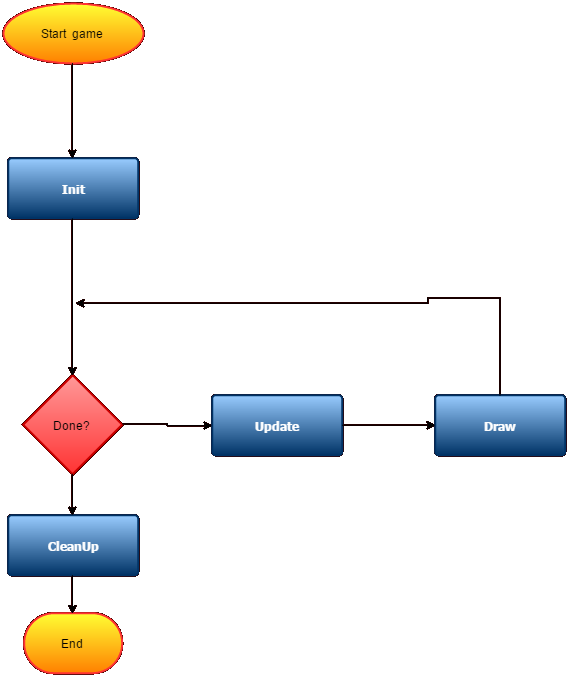


Figure III.1 Main game loop

## **Engine**

The purpose of game engine is to gather the resources of the game and render game’s models.

### Files:

* All codes are contained in standard C++ source files (.cpp) and standard C/C++ headers(.h).
* Resources path are contained within RM file.
* Menu Scene Info is contained within MM file.
* Stage Scenes info s contained within SM file.
* Map info is contained within MAP file.
* Model and texture information is defined in .obj 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.

### Codes:

Class:

* ResourceManager: 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. ResourceManager also stores shader’s information from Fragment Shader file and Vertex Shader file.
* GameManager: Manage game app, switching between state.
* State: Base class for all state, with pause and resume function already implemented.
* MenuManager: Inherit State class. Manage menu scene by reading MM file, get menu object info, process menu button: Play, Tutorial, Credit.
* SceneManager: Inherit State class. 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 objects are stored in SM.txt file.
* Object: Base class for all objects, store object’s vertices and indices buffer, texture, matrix, shader info.
* Shark: Inherit Object class, process shark’s behavior include random movement, detect and pursue enemy, item’s effect.
* Fish: Inherit Object class, process fish’s behavior include following player.
* Player: Inherit Object class, process main character’s behavior: Movement, pick, drop item...
* Item: Inherit Object class, process item’s effect.
* Button: Inherit Object class, process button’s behavior.

## **Techinical**

### Collision

Simple collision detection:

* Each object has its own collision rectangle with which its center coincide object’s center.
* Collision detection using simple AABB algorithm.

### Artificial Intelligent

Using Jump point search (JPS) algorithm:

* Require: a graph which is our map, a start and end position.
* Speed: extremely fast – less than 2 milisecond for every start and end position on map.
* Only find horizontal and vertical path, no diagonal path.

Shark’s AI:

* Can detect fish in a radius of 3 blocks.
* Randomly moving around when nothing is detected by choosing randomly a destination in a range of 5 blocks horizontally and vertically.

Fish’s AI: always follow player when freed.

### 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.

### 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.