iCafe Automation Tool High Level Architecture Design

RTG CSE iCafe/Cloud Engineering

July 23th, 2022

`Rev 1.0`

Revision History

A summary of changes in each revision should be provided here.

Date	Revision	Author	Summary
2022/7/15	v1.0	Junda Li	v1.0 Archetecture initial draft
Future revision			

Introduction & Scope

This part will introduce the background, target of this project and scope of this document

Background

RTG CSE iCafe team is responsible for GFX gaming validation to improve product quality. There are many test scenarios which not suitable for manually execution:

- `Performance test`: This scenario requires operations can be percisely reprocued in test cycles to reduce random error, manual random error is inevitable.
- `Long-time stress test`: This scenario requires to run a benchmark for many cycles to stress GFX. It will drain human resource if it is executed manually.
- `Game compatibility test`: This scenario's target is covering as many games as

possible to find GFX compatibility issues. An automative tool can significantly improve validation efficiency.

For these scenarios, An automation tool, which can design, edit, run script for target game and analyzing image quality in real time is required to be developed for effiecny improvement.

Project Target

iCafe Automation Tool project is targeting to development a automative tool for game testing. It should has the following features:

- ☑ A GUI based control panel.
- ☑ Script editor: It can load/save script, single step editing, add /delete/insert/remove a step.
- ☑ Input Simulator: It can tranfer script to control commands to contorl the game and bypass anti-cheat system.
- ☑ IQA: Implement a real-time Image quality assessment algorithm to detect graphical defect.

Scope

This document will introduce the high level design of iCafe Automation Tool, each functional blocks and future extensions.
For detailed development handbook, please find another documents.

High Level System Design

This part will introduce the high level system design and functional blocks.

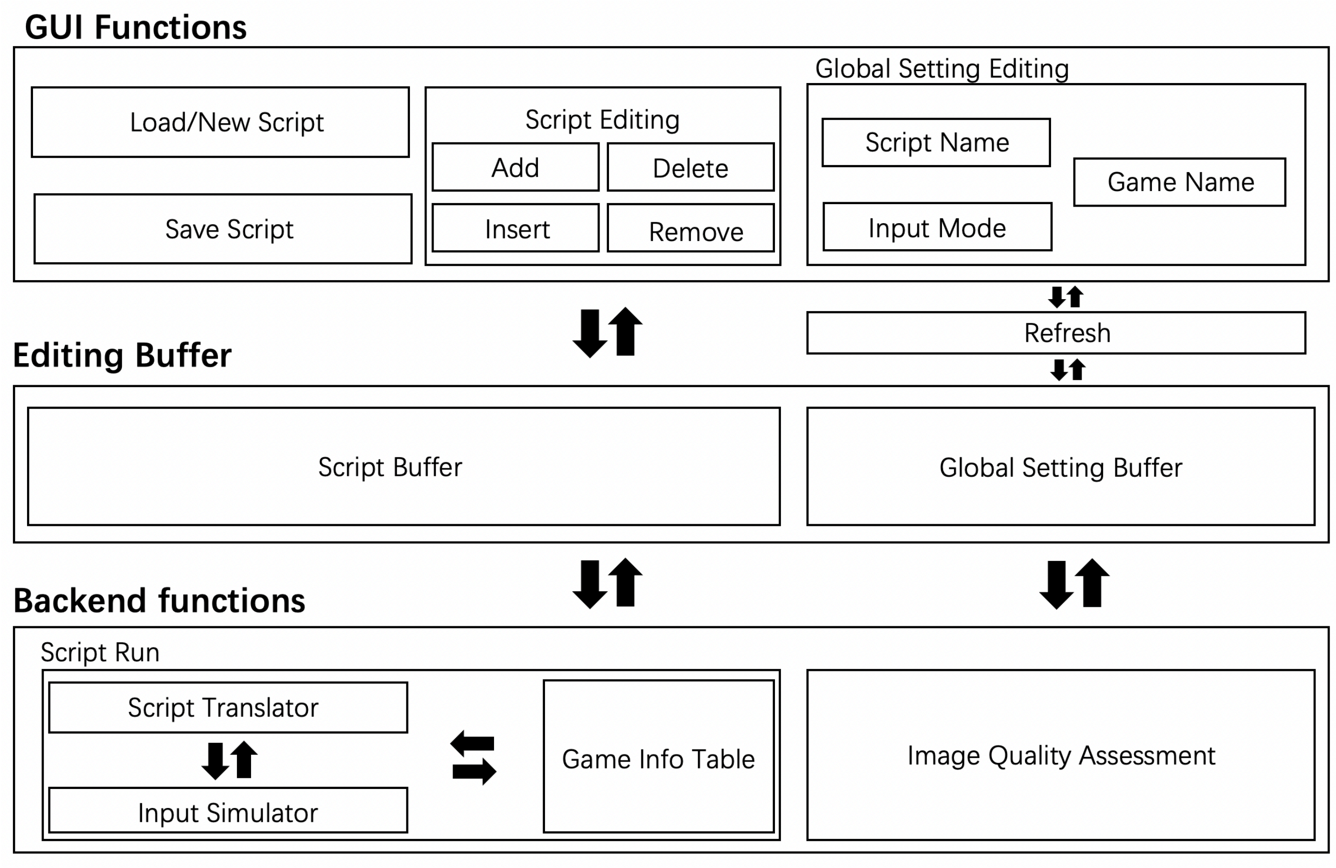

Fig 1: System High-level Architecture

The whole system is mainly divided into 3 layers, GUI Function layer, editing buffer layer and backend function layer.

`GUI Function Layer`: GUI related logical functions are implemented in this layer, these functions are directly changing the items shows on GUI. for script, these changes will be applied to editing buffer immmediately, but for global settings, these changes will bu applied to memory buffer after the user click "refresh" buttom.

`Editing buffer layer`: This buffer saves the scripts and global settings in a memory buffer. All backend functions will only read scripts and setting from this buffer and execute accordingly. It's essensial to keep GUI settings sync with the buffer.

`Backend Functiong layer`: This layer's functions are responsible for script translation, input simulation and image quality assessment.

For detailed functional block introduction, please check the next part.

Functional Blocks

This part will introduce each functional part, mainly focus on its logical function description.

Initialization

Initialization is the first function to be called before all function. The initialization will apply the default values to global settings, scan the default script foldder and add all script file name to loadable script ist and create IQA failed image folder.

Script Load

The load fucntions will load the choosed script from folder. Firstly it will read the settings part and apply them to GUI and global setting part. Then read the script lines one by one and wirte these informations to GUI and editing buffer.

For operation name part, the load function will try to find the name in script file in supported operation list in game info table. If the operation in script file is not found in the table, it will be change to "un-supported operation".

For execution time, it only support 0-9s, for repeat times, it only support 0-9 times.

Script Editing

There are 4 script editing functions, they will directly change the edit buffer but will not write to script file untill the save function is called.

The Add/Delete function will add a new line with default settings at the end/ delete the last line.

The Insert/ Remove function will add a new line with default settings behind the target line/ delete the target line.

Global Setting Editing

The global settings include script name, game name and input mode. These settings will be showed in GUI and user can edit it directly. But these changes will not be efficetive untill the user press "refresh" button to write these settings to editing buffer.

Script Run

When the user click the "Run" button, the operation list generator function will be called firstly to generate opeation list from editing buffer. This process will translate the game operations to exact keys our mouse buttons.

After translation complete, the operation execution function will read the operation list and call input simulator to complete the keyboard/mouse input simulation.

Image Quality Assessment (IQA)

The image quality assessment fucntion will be executed parallely with the input simulator by creating another thread.

This function will compute the edges of each image on screen once a second. If one image's edges have exceeded the boundary, IQA will save the screen snap to the failed image folder.

Script Save

The script save funciton will save the edit buffer's data to a txt file. Firstly it will save the global settings part, then save the each scirpt step in a sinlge line.

If there is a file that has the same name, the save function will cover the old one with new data.