

УДК 004.054

АВТОМАТИЗАЦИЯ ТЕСТИРОВАНИЯ ИГРОВЫХ ПРИЛОЖЕНИЙ

Чигасов Г.О., студент гр. ПИм-221, II курс

Научный руководитель: Славолюбова Я.В., к.ф.-м.н., доцент

Кузбасский государственный университет

имени Т.Ф. Горбачёва

г. Кемерово

Тестирование – неотъемлемая часть современного процесса разработки приложений. Тестирование предоставляет полезную информацию разработчикам программного обеспечения и помогает выявить потенциальные слабые стороны ПО и документации до релиза. Одним из видов тестирования программного обеспечения, широко использующемся сегодня, является автоматизированное тестирование.

Автоматизация тестирования обладает следующими преимуществами:

1. Высокая повторяемость тест-кейсов.
2. Высокая надёжность, не зависящая от длины тестового цикла.
3. Высокая скорость выполнения тестового набора.
4. Имеется возможность генерации тест-кейсов.
5. Более широкий тестовый охват функций приложения.
6. Обеспечивает согласованность тестовых моделей.
7. Экономит время и деньги.
8. Позволяет выполнять процесс тестирования без вмешательства человека.
9. Сокращает время выхода продукта на рынок.

Стоит отметить, что, на данный момент, полную автоматизацию тестирования ввести не представляется возможным ввиду того, что некоторые области тестирования без человека не существуют (например, тесты пользовательского опыта).

В игровых проектах автоматизация тестирования также достаточно сильно развита, однако в большинстве случаев инди-игр проще использовать методы ручного тестирования, так как затраты на введение автоматизации довольно часто не окупаются.

В процессе разработки компьютерной игры «Homecoming» [1] на этапе тестирования возникла необходимость ввести автоматизацию для ускорения исправления ошибок.

Цель работы: рассмотреть виды автоматизированных тестов, выбрать наиболее необходимые; рассмотреть и выбрать технологии разработки тестов, составить и реализовать пример набора автоматических тестов для компьютерной игры «Homecoming».

Рассмотрим виды автоматизированных тестов (таблица 1).

Таблица 1. Виды автоматизированных тестов

Вид тестирования	Краткая характеристика
Unit-тесты (Unit Tests)	проверяют поведение только одной функции/метода
Компонентные/модульные тесты (Component Tests)	проверяют работоспособность конкретного компонента, без взаимодействия с другими.
Сервисные тесты (Services Tests)	проверяют доступность веб-сервисов, к которым обращается приложение
Тесты пользовательского интерфейса (User Interface Tests, UI Tests)	проверка корректности работы и удобства использования пользовательского интерфейса
Визуальные тесты (Visual Tests)	проверяют отображение элементов интерфейса на экране
Тесты безопасности (Security Tests)	проверка соблюдения мер безопасности и защиты данных
Тесты производительности (Performance Tests)	Проверка скорости работы приложения под определённой нагрузкой
Тесты доступности (Accessibility Tests)	проверка программ на пригодность к использованию людьми с нарушениями слуха, зрения, двигательной активности

Исходя из анализа таблицы принято решение о необходимости автоматизировать часть UI и Unit – тестов, а также необходимость автоматизировать тесты производительности. При этом планируется использовать готовую систему тестирования для обработки результатов.

Во время подготовки к разработке алгоритмов автоматизированного тестирования нами были выбраны следующие технологии:

1. C# – язык программирования.
2. Visual Studio – среда разработки.
3. База данных MySQL.
4. Grafana dashboards.

В качестве основного языка программирования был выбран C# – объектно-ориентированный язык программирования, разработанный корпорацией Microsoft. C# используется для создания различных приложений, веб-разработки, разработки игр, приложений для мобильных устройств, робототехники, научных расчетов и многих других областей. Основой C# является язык C++ [2]. Однако данный язык включает в себя много новых конструкций и синтаксических улучшений, которые делают его более простым в использовании и повышают производительность программ. Данный язык программирования был выбран в силу того, что код самой игры также написан на данном языке, что позволяет достаточно удобно проводить тесты.

Для удобства работы с языком C#, была выбрана среда разработки Visual Studio, которая позволяет работать с кодом проекта в удобном формате. На

текущий момент, актуальной версией является Visual Studio 2022 (17.9.4). От остальных сред разработки отличается использованием таких технологий, как AI IntelliCode и Live Share [3]. Первая позволяет автоматически дописывать строки или фрагменты кода. Вторая – открывает новые возможности для парного программирования и анализа кода. Данные технологии позволяют значительно сэкономить время и распределить его на другие задачи.

На рисунке 1 представлен пример теста, написанного в среде Visual Studio на языке C#:



```
1  using System;
2  using System.Collections;
3  using System.Collections.Generic;
4  using System.ComponentModel;
5  using NUnit.Framework;
6  using UnityEngine;
7  using UnityEngine.TestTools;
8
9  public class PersonTest
10 {
11     [UnityTest]
12     public IEnumerator PersonMove()
13     {
14         var gameObject = new GameObject();
15         var Person = gameObject.GetSpawner<NPC>();
16         float initialYPos = NPC.transform.position.y;
17
18         yield return new WaitForSeconds(0.1f);
19
20         Assert.Less(NPC.transform.position.y, initialYPos);
21     }
22 }
23
24
```

Рисунок 1 – Пример unit-теста

После проведения серии тестов данные необходимо сохранить. Воспользуемся базой данных MySQL. MySQL – одна из наиболее используемых систем управления базами данных. MySQL управляет реляционными базами данных, то есть такими, в которых таблицы связаны между собой.

MySQL работает по принципу клиент-сервер. Компьютер пользователя (клиент) отправляет запрос. Сервер баз данных его обрабатывает и предоставляет ответ. Именно поэтому часто можно услышать понятие MySQL-сервер. Это сервер, на котором хранится база данных [4]. Она используется для создания и управления базами данных для веб-сайтов, приложений и других программных продуктов. MySQL является одной из наиболее распространенных СУБД в мире и распространяется на условиях открытой лицензии. На рисунке 2 представлена структура базы данных.

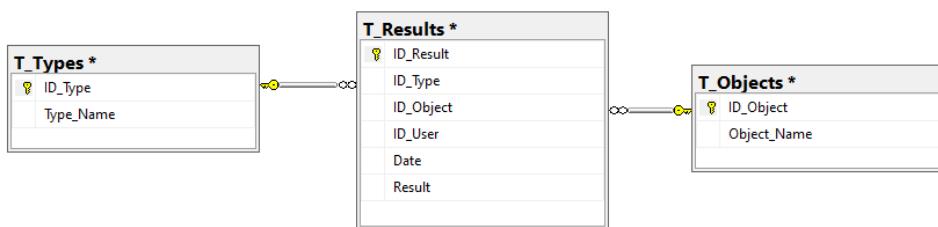


Рисунок 2 – Структура базы данных

После проведения серии тестов и записи в базу, данные о результатах необходимо визуализировать. Для этого будем использовать сервис Grafana. Grafana – это платформа с открытым исходным кодом для визуализации,

мониторинга и анализа данных [5]. Grafana позволяет пользователям создавать дашборды с панелями, каждая из которых отображает определенные показатели в течение установленного периода времени. Каждый дашборд универсален, поэтому его можно настроить для конкретного проекта или с учетом любых потребностей разработки и/или бизнеса.

Таким образом в результате работы были проанализированы и выбраны виды автоматизированных тестов, необходимые к автоматизации, подготовлена база данных для хранения результатов, а также созданы некоторые Unit-тесты, проверяющие работу некоторых механик игры. В планах создание большего числа тестов и подключение сервиса Grafana для визуализации.

Список литературы:

1. Чигасов Г. О. Разработка трёхмерной компьютерной игры «Homecoming» в жанре action-adventure/ Г. О. Чигасов. – Текст: электронный // Россия молодая: Сборник материалов XIV всероссийской научно-практической конференции молодых ученых с международным участием. – Кемерово, 2022. – URL: <https://science.kuzstu.ru/wp-content/Events/Conference/RM/2022/RM22/pages/Articles/031679.pdf> (дата обращения: 28.03.2024).
2. Медиаблог. Язык программирования C#. – URL: <https://www.tgu-dpo.ru/news/2023/04/06/yazyk-programmirovaniya-s/> (дата обращения: 31.03.2024).
3. Хабр: Visual Studio 2022. – URL: <https://habr.com/ru/companies/microsoft/articles/553442/> (дата обращения: 31.03.2024).
4. Сайт дистрибутивов Linux. – URL: https://wiki.gentoo.org/wiki/MySQL/Startup_Guide/ru (дата обращения: 26.03.2024).
5. Хабр: Grafana как еще один инструмент для технического мониторинга создаваемых нами программных продуктов [Электронный ресурс]. – URL: <https://habr.com/ru/companies/slurm/articles/431122/> (дата обращения: 01.04.2024).