

GOTO

Aarhus 2022

June 13-16

**Run Automated Tests
on a Unity Game**

Florence Rolland & Marek Szymeczko



Abstract

- Hands-on demo about automated unit tests on a Unity game, using the **Test Runner** to run tests using the **Unity Test Framework (UTF)** on a real video game project.
- This talk does not require any knowledge of Unity or game development in general



- We will demonstrate **how to write tests** using the test framework and **build and run your tests** on either Editmode or Playmode.
- We will also demonstrate **how to run the tests from the command line**, and **set up your own CI** to run Unity tests.



Marek

Senior Software engineer
Unity

- security and automation enthusiast
- software developer with 8+ years of commercial experience
- strong background in distributed systems and stream processing

He works inside Test Platform team → responsible for maintaining the Unified Test Runner, Unity's main test runner.



Florence

Senior Software engineer
Unity

- 10+ years in software development and QA
- Designs internal training programs for Unity infrastructure
- CI/CD and test automation enthusiast

Works in Developer velocity team

→ leverages tooling and improves processes to deliver better software, faster



1. UTF, the UTF Package & the Test Runner

UTF acronym means [Unity Test Framework](#)

- library that allows you to test your Unity code in Editmode and Playmode
- also allows you to target platforms such as Standalone, Android, iOS, etc.

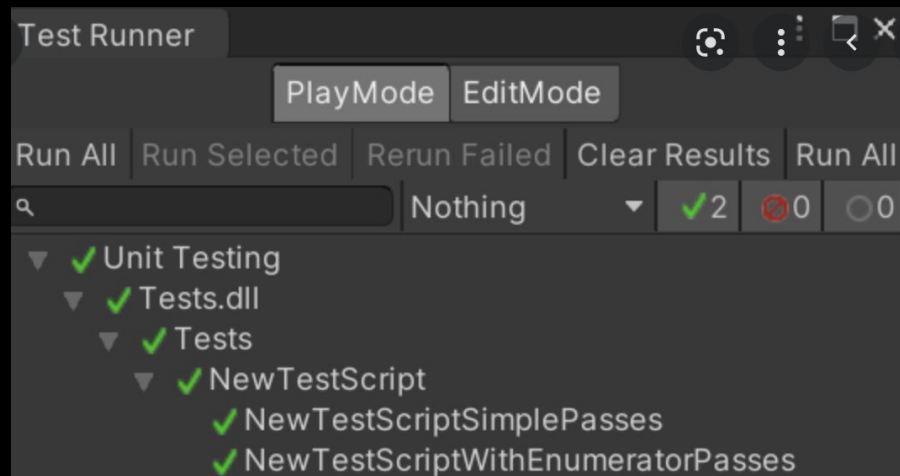
[UTF Package](#) → customer-facing **Unity Package** that improves developer velocity when it comes to writing Unity tests.

→ also provides a UI to run these tests from the Unity Editor → the [Test Runner](#)

Using the Test Runner, you can

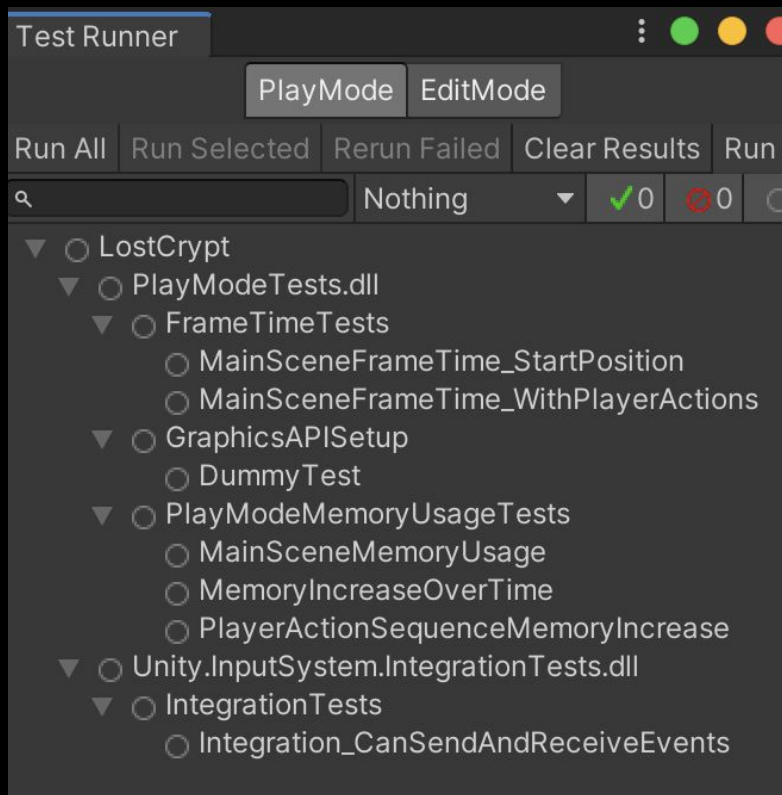
- execute the tests
- gather tests results

Pro tip: Attach the script debugger to the Editor to debug your tests.





1. UTF, the UTF Package & the Test Runner





2. Editmode and Playmode

Editmode

- runs on the Unity editor and has access to Editor code
- Linux, MacOS, Windows
- used to test Editor extensions

Playmode

- runs on a player (standalone) or through the Editor
- requires a player build
- all editor platforms + supported platforms (Mobile, WebGL, Consoles...)
- used to test the game itself

You can trigger automated tests through the UI, or using the command line.



3. Demo project

Lost Crypt

- Unity 2D Sample project
- runs on all Desktop platforms + iOS and Android
- available through the Unity Asset Store and GitHub
- no license or subscription required (free access)

Setup

- Unity LTS 2020.3
- download the project through the Asset Store
- check the automated tests already available for the game





4. What do we test automatically in a game?

Unit Tests

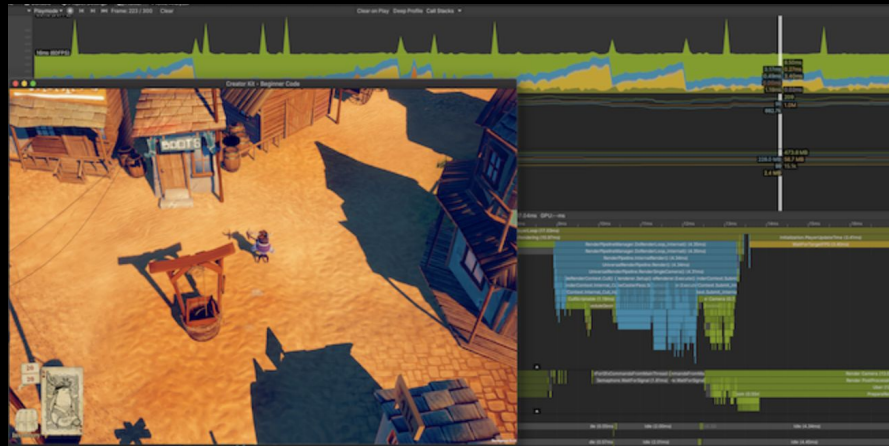
- Feature testing
- Rendering
- Level generation
- Serialization / deserialization

Integration Tests

- validate complex scenarios
- confirm bug fixes
- high coupling, high maintenance

Performance

- CPU / GPU usage
- Physics
- Audio
- Stress Tests





4. What do we test automatically in a game?

Networking

- if multiplayer, runs correctly with several players
- reports data as expected

Compatibility

- builds and runs on all supported platforms
- test on different hardware, graphics configurations, devices...





5. What do we test manually?

Play Tests & Exploratory testing

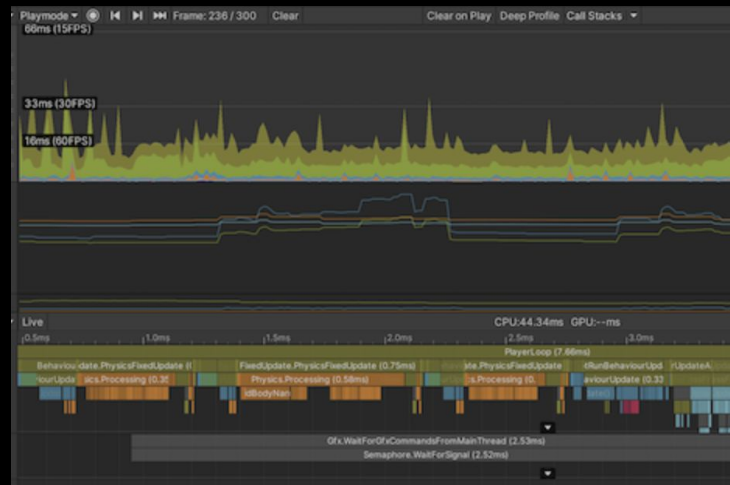
- UI / UX Experience
- game coherence (difficulty, graphics, storytelling)
- advanced network testing (difficult to automate)
- game experience



6. Unity Profiler

The **Unity Profiler** collects data about your game performance and gives you the opportunity to investigate different performance areas

- automated testing through [low level native plugin Profiler API](#)
- manual testing through Unity Profiler window
- CPU / GPU usage
- Network bandwidth
- Memory usage

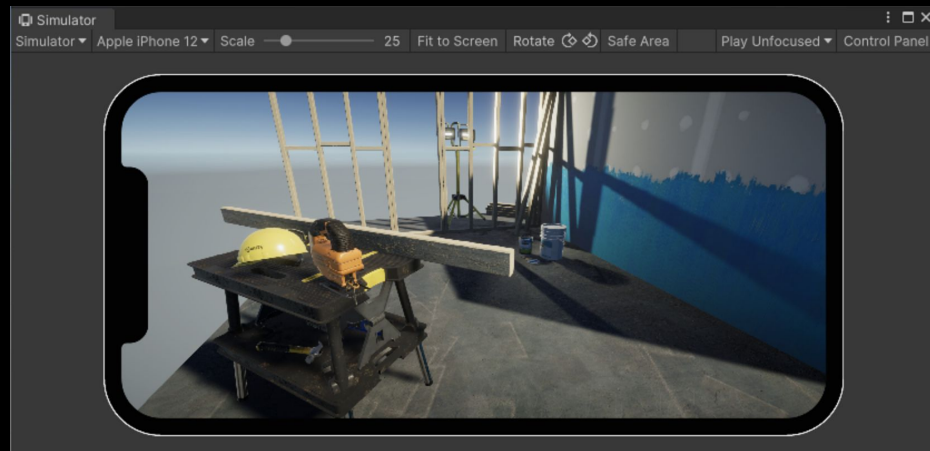




7. Device Simulator

The **Unity Device Simulator** allows you to visualise what your application would look like on different types of devices (iOS, Android)

- no physical device needed
- visualise game on device
- simulate device-specific behaviours
- simulate touch input
- useful for manual testing





8. Automated QA Package

The **Automated QA Package** allows you to record and replay tests for iOS and Android on a Unity game

- codeless automation
- record tests via the UI
- replay tests with UTF through the Test Runner or the command line
- can be set up by manual QAs

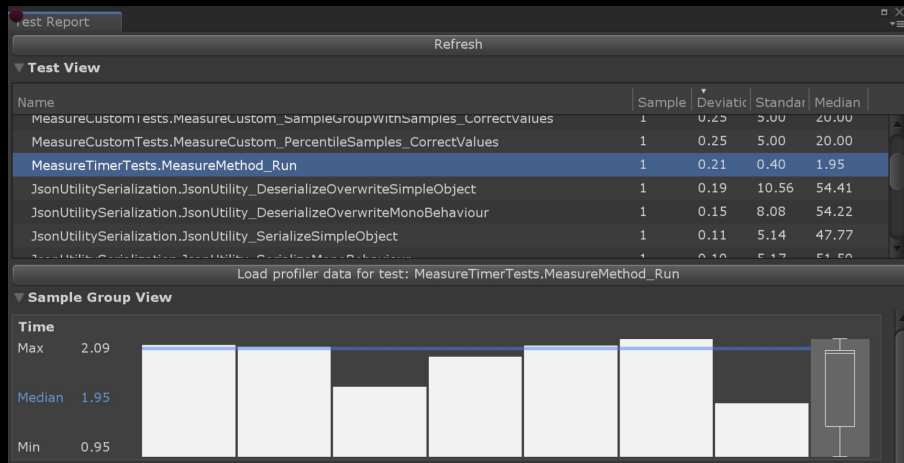




9. UTR Performance Testing extension

Unity package that allows you to investigate different performance areas using the [Unity Profiler](#), and other custom metrics outside of the Profiler.

- measure performance of the Unity Editor or built players
- run tests using the Test Runner
- perfect for collecting performance data on a nightly CI
- generates performance testing report





10. Demo time

→ automated tests
on Lost Crypt project

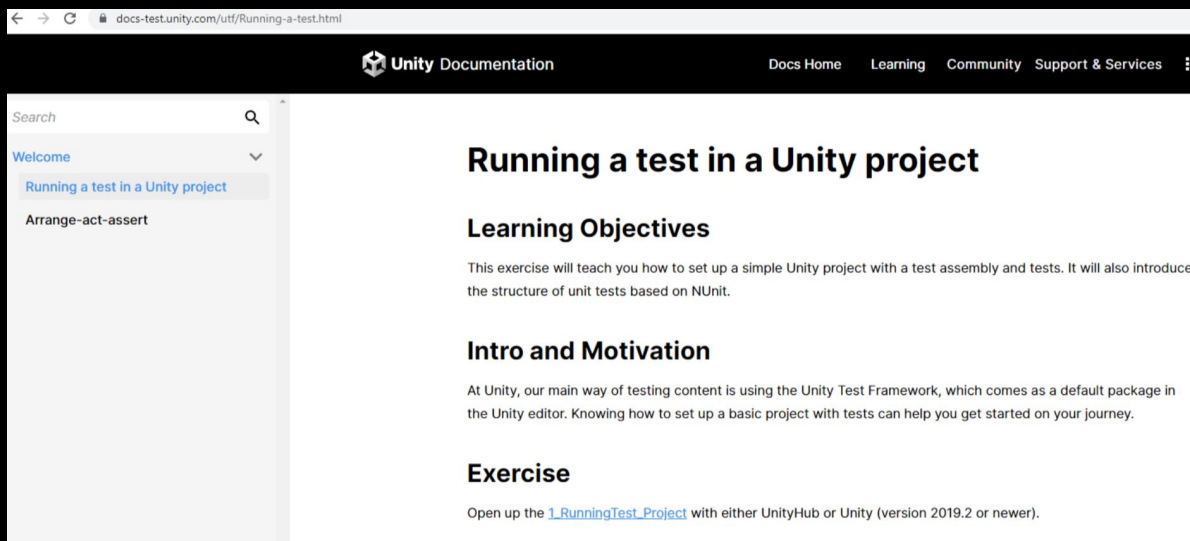




Acknowledgements & Future plans

Special thanks to **Christian Warnecke** for creating training exercises, and to **Giovani Galicia** for reviewing our content and masterclass program.

We hope to make the Unity Test Automation masterclass available through Unity Docs soon!





NightSpirit (4) ☐ Static

Tag **Untagged** Layer **Ground Soft**


Transform ? ⇅ ⋮

Sprite Renderer ? ⇅ ⋮

Visible Only During Night (Script) ? ⇅ ⋮


Material_Spirit_Lit ? ⇅ ⋮

Shader **Shader Graphs/ShaderGraph_Spi**

Sprite Texture  Select

Visual State (Bad to 1

Fresnel Tint and Alpha **HDR** ⇅

Detail Color  ⇅

Wobble Speed 0.55

Wand Position


X Y Z W

Wand Power

Wand Range Multiplier

Render Queue **From Shader** 3000

Double Sided Global Illumination ☐

 MaterialPropertyBlock is used to modify these values

Add Component



A few final words...



Questions?

DON'T FORGET TO **RATE THE SESSIONS**

#GOTOaar

Rate a minimum of **5 sessions** and
claim your **reward** at the
Registration Desk at the Trifork Hall