Deterministic Replay: Performance Report

Complex Game Systems

Connor Mills

2023

Table of Contents

[Overview 2](#_Toc136859444)

[Demo Application 2](#_Toc136859445)

[Issues Encountered 2](#_Toc136859446)

[Required Changes 4](#_Toc136859447)

[Performance 5](#_Toc136859448)

[Optimizations Included 5](#_Toc136859449)

[Future Improvements 5](#_Toc136859450)

# Overview

The modular system that I have created is a Deterministic Replay System. A deterministic replay is a technique that provides the user with deterministic executions of computer programs while nondeterministic factors are still present. That is to say that it provides the user with the exact same execution of computer programs irrelevant of any factors.

My modular system contains a variety of scripts that the player can use to create recordings of their game or application. The user can choose to add a recorder script onto a game object to capture that specific object’s movement or they can add a tag to a group of game objects and have a recording manager handle recording all the tagged objects.

## Demo Application

I have used my replay system to create a demo racing game where the player can race against ghosts of their previous runs. I used a recorder scripts to capture; the car’s main body’s position and rotation, each of the wheel’s position and rotation, the two skid mark activation and deactivation state, and the drift smoke activation. This allows the game to record the players run through the tack and play it back to them and make the players feel like they are versing an actual player.



# Issues Encountered

* Identify any issues you encountered when creating and integrating the Modular Complex Game System identified in your Design Stage Brief:
  + Account for how you either overcame, fixed, bypassed, or avoided each issue identified.

A number of issues were encountered during the creation and integration of the Modular Replay System. Such as when I was creating the modular system, I was using a Unity project that was using Universal Render Pipeline and when I tested the system in a brand new Unity project I got errors from the materials. The issues was that since I was using URP the materials were converted but since the new project was not using URP it broke the materials. This was an easy fix as I just converted the materials back to using the unity default render pipeline and left it for the user to decide if they want to convert them to URP.

During the creation of the replay system, I encountered difficulty in making the system as modular as possible and as simple for the user to use.

# Required Changes

* Explain any required changes you had to implement into the Modular Complex Game System for it to function for your outlined objective.
* Addition of the tag system

# Performance

* Illustrate the performance of the implemented system:
  + The method of doing this will change on a system-by-system basis but will typically require you to benchmark your system against similar implementations. For example:
    - For an AI algorithm implemented on the GPU, compare and contrast against the same algorithm running on the CPU.
    - For a networking application, measure latency and performance across a range of network conditions.
    - For a procedural generation algorithm, an analysis of the algorithm’s complexity or performance (with reference made to Big O notation) may be sufficient.

## Optimizations Included

* Outline any optimizations included in your implementation.

## Future Improvements

* Outline any areas where improvements could be made.