Deterministic Replay

Complex Game Systems

Connor Mills

2023

Table of Contents

[Overview 2](#_Toc133929858)

[The Objective 2](#_Toc133929859)

[Third Party Libraries 2](#_Toc133929860)

[Mathematical Operations and Advance Algorithms 3](#_Toc133929861)

[Operations 3](#_Toc133929862)

[Algorithms 3](#_Toc133929863)

[Integration 3](#_Toc133929864)

[Modular Design 3](#_Toc133929865)

[References 4](#_Toc133929866)

# Overview

The modular system that I will be creating is a deterministic replay system. Deterministic replay is a technique that provides the user with deterministic executions of computer programs while nondeterministic factors are still present.

The system that I will create will record the position, rotation, animation states, and other relevant information of the player-controlled character and any other moving object. That information can then at a later date be used to replay the player’s movement and other objects, displaying a ghost of them that follows the exact same path as they took when they were recorded.

## The Objective

I am designing it to allow development teams access to an easy to use and accessible tool that they can use to add an extra feature to their game. Deterministic replay systems are a great way to enhance a game and to also add a new competitive aspect to single player games. It can be used in multiple different applications for lots of different games. Some examples are:

In a single player racing game, each of the player’s attempt at the course could be recorded and then the fastest time recorded could then be played when the player starts next. This allows the player to compete against an opponent pushing themselves to get better. Or the player could view the replay and see where they did well and where they did well, allowing them to perfect their gameplay.

Another application is in a platformer game where all attempts are recorded and then when the player beats the level all the attempts are then played together showing the player each time they died. This can be seen in the game Super Meat Boy made by Team Meat where at the end of a level a replay of all attempts are played simultaneously.



## Third Party Libraries

The system will be created as a Unity package and will use Unity’s libraries. I will be using MemoryStream and the read and write functions from System.IO to store the game data used in the replays.

# Mathematical Operations and Advance Algorithms

## Operations

The system is very light in mathematical operations. The only mathematical operations that occur is when the scripts that are recording get the difference between the current frame and the previous frame as save it to the queue and when the scripts that are reading from the queue get the saved change and add it to the ghost object.

When the system is recording, the scripts that handles the collection of the object’s data holds onto the previous frame’s data and checks it against the current frame’s data. The script gets the difference between the two frame then stores that into the queue.

Once the system has finished recording and is now playing, the scripts that controls the ghost objects will pull from the queue in order and will get the difference that is stored in the queue and then apply it the ghost object.

## Algorithms

Explain what advance algorithm/s you will be implementing (diagram/s could be used to help support your explanation).

Talk about, how the data is saved and how it is read, what is used to record the data a MemoryStream

# Integration

Illustrate how your system should be integrated into an application.

# Modular Design

Prove how you will design your complex system to be modular. (diagram/s could be used to help support your explanation).

The system I will create will be designed to be modular by having presets the user can use as well as making sure custom recording scripts are easy to create and use.

I will create presets that are needed in most games that have a ghost replay feature. These presets will include the recording and reading scripts that are needed to create the ghosts. The presets will account for what type of information needs to be recorded. Such as position, rotation, scale, animation states, and other relevant data.

To make sure users can create their own recording and reading scripts, I will create well document and fully fleshed out derivative scripts that can be used.

# References

*C# - Queue* (no date) *TutorialsTeacher.com*. Tutorials Teacher. Available at: <https://www.tutorialsteacher.com/csharp/csharp-queue> (Accessed: May 1, 2023).

*How to create a Replay System like in Super Meat Boy using Unity* (2022) YouTube video. added by Trever Mock [Online]. Available at: <https://www.youtube.com/watch?v=ilOQstDnX2I&ab_channel=TreverMock> [Accessed: May 1, 2023].

Engel, T. (no date) *Creating a replay system in Unity*, *kodeco.com*. Edited by A. Rames, M. Moser, and B. MacKinnon. Kodeco. Available at: <https://www.kodeco.com/7728186-creating-a-replay-system-in-unity> [Accessed: May 1, 2023].

*System.IO Namespace* (no date) *Microsoft Learn*. Microsoft. Available at: <https://learn.microsoft.com/en-us/dotnet/api/system.io?redirectedfrom=MSDN&view=net-7.0> [Accessed: May 1, 2023].

Morgan, M. (2011) *Super meat boy is your ticket to Classic 8-bit gaming*, *Wired*. Conde Nast. Available at: <https://www.wired.com/2011/01/super-meat-boy-is-your-ticket-to-classic-8-bit-gaming/> [Accessed: May 2, 2023].

McMillen, E., Refenes, T. and McEntee, J. (n.d.) “Super meat boy.” Asheville, North Carolina, United States: Team Meat.