|  |  |
| --- | --- |
|  | Moving Picture, Audio and Data Coding  by Artificial Intelligence  www.mpai.community |

**Public document**

|  |  |
| --- | --- |
| **N1501** | 2023/11/22 |
| **Source** | Requirements (SPG) |
| **Title** | Technical Report: Guidelines for mitigation of data loss and cheating in online multiplayer gaming WD0.3 |
| **Target** | MPAI Members |

|  |  |
| --- | --- |
|  | Moving Picture, Audio and Data Coding  by Artificial Intelligence  www.mpai.community |

**MPAI Technical Report**

**Server-based Predictive Multiplayer Gaming**

**MPAI-SPG**

|  |
| --- |
| **WD0.3** |

**WARNING**

Use of the technologies described in this Technical Specification may infringe patents, copyrights, or intellectual property rights of MPAI Members or non-members.

MPAI and its Members accept no responsibility whatsoever for damages or liability, direct or consequential, which may result from use of this Technical Report.

Readers are invited to review Annex ? - Notices and Disclaimers.

© Copyright MPAI 2020-2023. All rights reserved.

**Technical Report**

**Server-based Predictive Multiplayer Gaming**

# Introduction

In an online authoritative multiplayer game, each player uses a client to send control data to a server. The server updates the current game state with the data from all clients, and then broadcasts it to all clients. The data originating from a client may be properly or maliciously generated and properly received or not received at all. We do not consider the case when control data are corrupted by the network. In both cases, the game state received from the server does not describe a correct and consistent situation.

Current approaches to address the problem

* Overview
* Limitations

An alternative approach

* Description
* Novelty
* Advantages

# Scope

This document provides guidelines on the design and use of neural networks for the purpose of creating reliable and accurate prediction systems to predict absent or malicious players’ control data in an authoritative server context.

# Terms and Definitions

# References

# Context

Use a neural network to predict and hand over the game state to the server. The server may use the predicted game state in case it is missing some controller data from one or more clients and uses it to detect inconsistent data received by the clients (e.g., due to cheating).

In this section we need to expand the description of the principles on which MPAI-SPG is based.

# Process Description

Describe the key steps needed to design and implement an MPAI-SPG model:

* Select the game
* Define the game state parameters
* Define the entities:
  + Environment
  + Human-controlled players (HPC) and Non-player characters (NPC)
* How they affect the game state
* Train AI agent players to simulate HPCs
* Define types of training data
* Collect data for training the prediction network
* Train the prediction network:
  + Define viable architectures
  + Define the training parameters
  + Compare training results of different architectures
* Select a pool of the best trained network
* Implement the selected prediction networks into the game server logic
* Test the effects of the prediction networks on the game:
  + Define objective and subjective metrics
  + Collect and analyse data
* Define the elements affecting the system:
  + Data loss
  + Cheating: which game state variables are affectable by the clients
* Design and Implement the software modules which use the predicted state for their specific purposes:
  + Data loss mitigation
  + Cheating
* Implement modules which simulate the disturbances:
  + Data loss, possibly universal to all game types
  + Cheating, it should imitate the cheating opportunities a player would exploit
* Evaluate the game experience improved by the SPG prediction system and the modules which harness the predictions.

# An Exemplary Application of the Process

Here we will describe the application of the process outlined in Section 4 using the car racing game as an example.