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

**Public document**

|  |  |
| --- | --- |
| **N1545** | 2023/12/20 |
| **Source** | Requirements (SPG) |
| **Title** | Technical Report: Guidelines for mitigation of data loss and cheating in online multiplayer gaming WD0.4 |
| **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**

|  |
| --- |
| **WD0.4** |

**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**

**(Under development)**

Contents

[1 Introduction 3](#_Toc153968328)

[2 Scope 3](#_Toc153968329)

[3 Terms and Definitions 3](#_Toc153968330)

[4 References 4](#_Toc153968331)

[5 Context 4](#_Toc153968332)

[6 Process Steps Description 4](#_Toc153968333)

[7 An Exemplary Application of the Process 5](#_Toc153968334)

[7.1 Game Description 5](#_Toc153968335)

[7.2 Process Implementation 5](#_Toc153968336)

# 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

Spatial attitude: position, orientation, velocity, angular velocity, acceleration

# 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 Steps Description

The process is defined as a series of steps to follow.

Describe the key steps needed to design and implement an MPAI-SPG model. The first 4 steps are required to describe the game to take more informed decisions for the implementation of SPG.

1. Select the game
2. Define the entities (to enable parameters identification):
   1. Environment
   2. Human-controlled players (HPC) and Non-player characters (NPC)
3. Define the elements of the game state
4. How they affect the game state
5. Train AI agent players to simulate HPCs. Also describe why are we doing this: because we need a large dataset to train the network that will be used to predict elements of the game state.
6. Define the game state elements that will be predicted
7. Define types of training data
8. Collect data for training the prediction network
9. Train the prediction network:
   1. Define viable architectures
   2. Define the training parameters
   3. Compare training results of different architectures
10. Select a pool of the best trained network
11. Implement the selected prediction networks into the game server logic
12. Test the effects of the prediction networks on the game:
    1. Define objective and subjective metrics
    2. Collect and analyse data
13. Define the elements affecting the system:
    1. Data loss
    2. Cheating: which game state variables are affectable by the clients
14. Design and Implement the software modules which use the predicted state for their specific purposes:
    1. Data loss mitigation
    2. Cheating
15. Implement modules which simulate the disturbances:
    1. Data loss, possibly universal to all game types
    2. Cheating, it should imitate the cheating opportunities a player would exploit
16. 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 6 using a car racing game as an example.

## Game Description

**Step 1**

We developed a simple car racing game [REFERENCE TO THE GAME]

Describe the game in detail so that all the elements below are comprehensible to the reader who knows nothing about the game.

**Step 2**

Here describe which are the entities of the game:

* Environment: describe the racing track and how are thy composed (composed of a series of elementary blocks: straight, curve, ....). Also describe the concept of a checkpoint in the track.
* HPC/NPC: the cars. Describe how are they controlled and by which parameters. User input control affecting speed and direction. Describe also the physical parameters used (mass, drag, ....). Describe also the relations with:
  + CAR vs CAR: collision
  + CAR vs ENVIRONMENT: the environment has boundaries against which cars collide

**Step 3**

The game state is composed of:

* the individual cars’ state (position and orientation on the track, velocity and angular velocity, acceleration and completed checkpoints).
* The current ranking of the cars in the game and the number of laps completed

**Step 4**

When a car overtakes another car the ranking is updated

## Process Implementation

**Step 5**

Describe what we did and how to train the agents simulating HPC behavior for the creation of the dataset: ML-Agents with reinforcement learning.

**Step 6**

In our context we wanted to predict the car’s position, orientation, velocity and angular velocity (spatial attitude)

**Step 7**

We selected the following game state elements: car state (velocity, acceleration, track tile type, local position in track tile).

Describe why we chose these parameters

Data to be shared: analytics, the trained network