A Game Engine Layer for the Implementation of Massively Multiplayer On-line Interactive Fiction

Markus Schatten, Tomislav Peharda, Juraj Rasonja
Artificial Intelligence Laboratory
University of Zagreb
Faculty of Organization and Informatics
Pavlinska 2, 42000 Varaždin, Croatia
{markus.schatten,tomislav.peharda,juraj.rasonja}@foi.unizg.hr

Abstract. *In this article we present a game engine layer implemented above the Inform 7 interactive fiction (IF) programming system that allows for the implementation of massively multi-player on-line (MMO) IF games. By using a multiagent system (MAS) approach we show how interaction with dynamic entities like other players, bots or non-player characters (NPCs) can be implemented in IF games allowing for world synchronization over a network. Additionally, we show a proof-of-concept game implemented using the proposed engine and provide guidelines for future research.*

Keywords. game engine, interactive fiction, massively multiplayer on-line games, role playing games, multi agent systems

1 Introduction

Computer games which have a textual user interface and in which players interact with the game using text commands are usually called interactive fiction (IF), text adventures, gamebooks and even in some cases, if an additional graphical interface is present, visual novels (Gómez-Albarrán et al., 2021). Interactive fiction games consist of narrative worlds often organized in so called rooms which players can explore by using various commands. The term "room" here is quite broadly defined and can encompass any type of space an avatar can exist in, including but not limited to actual rooms, houses, caves, open spaces, vehicles, physical dimensions or even states of mind (Aikin, 2009). These rooms are usually connected through doors which represent the gateways between spaces. "Doors" are again very broadly defined and can include various pathways like actual doors, streets, portals, staircases, time machines etc. A lot of IF games fall into the role-playing game (RPG) genre, but there are exceptions.

On the other hand massively multi-player on-line (MMO) games which allow for thousands and sometimes even hundreds of thousands players playing simultaneously, are an interesting contemporary phenomenon due to the possible mutual interactions between players, especially in the form of organizing their behaviour to perform certain game-related tasks (Schatten, Okreša Đurić, Tomičič, et al., 2017). Additionally massively multi-player on-line role-playing games (MMORPGs) which combine the massive multiplayer aspect with the RPG genre, provide us with an opportunity to study the possibility for integrating IF and MMO (Schatten, Okreša Đurić, Tomičić, et al., 2017). IF games in most cases fall into the RPG genre, but there are exceptions.

In this paper we present an initial attempt on creating a game engine for MMO IF games. We base our implementation upon Inform 7[1] IF which is a declarative programming language for the development of IF based on natural language. We have developed a Python interface[2] to the Glulxe IF interpreter in order to be able to extend Inform 7 capabilities with various artificial intelligence (AI) related methods. One application example of this interface was using multiagent system (MAS) inside IF games which has shown very promising characteristics on which we will build upon herein by introducing a game engine layer that allows interaction and synchronization between players inside the same narrative world.

The rest of this paper is organized as follows: firstly in section 2 we provide an overview of related work. Afterwards in section 3 IF games are introduced and described in more detail. In section 4 we give an introduction to MMO games with a special focus on elements that can and should be part of IF games. In section 5 we introduce our game engine layer for MMO IF games and provide insights into implementation specific details. Then in section 6 we provide an example MMO IF game and discuss other possibilities of implementation. In the end in section 7 we draw our conclusions and give guidelines for future research.

[1] Available here: http://inform7.com/

[2] Available here: https://github.com/AILab-FOI/python-glulxe.git

2 Related Work

There are not many papers describing game engines for MMO IF, and up to our best knowledge there are none using an agent based approach in implementing an MMO IF game engine.

For example, in (James Pita, November 2007) a persistent system called True Story that has dynamically generated and contextually linked quests is explored. The implemented game tried to persuade the player to be a maker of its own story by completing quests and taking non-scripted paths trough a game, thus creating a unique user experience. The quest generation engine is based on a number of different constraints such as memories (what the player character did in the past), attributes (what an object is capable of), actions (what an object can do), layer (relationships established between objects) and proximity (area in which objects can interfere). Using these constraints, the system generates quests based on the player character's values of each constraint and state of the world. They achieve a system that can generate numerous quests that are specific for a player and its impact in the game environment. Whilst the implemented game does have elements of the RPG genre and IF, some elements like communication and interaction between players seem not to be addressed and the game cannot be considered an actual MMORPG since network play has been left for future research.

The paper (Jane, 2014) examines the Twine[3] platform which is and open source platform for developing interactive fiction. The author compares games made in Twine with mainstream games with focus on what emotional experience can game produce in player, and among other things, outlines the capability of Twine for implementing multi-player games. One of the first such games was Naked Shades[4], which was developed for the Ludum Dare gamejam in 2013. Sadly, the game isn't available any more, but nevertheless presents some interesting aspects of IF multiplayer games like how the narrative world is influenced by previous and on-line players.

The authors in their article (Crawford and Chen, 2017) compare and analyze visual novels that use virtual reality technology as part of storytelling and enhanced immersion into a story. As they conclude, good graphics are needed for visual novels to be a good game for using virtual reality; otherwise, the player will not experience fully immersed in the story. While the games mentioned in the article do not have MMO elements, some important aspects of MMO IF games are outlined like multi-linear storytelling and player interaction.

[3]Available here https://twinery.org/

[4]Description available here http://ludumdare.com/compo/ludum-dare-26/?action=preview&uid=23541

3 Interactive Fiction

IF games are usually adventure, RPG and other narrative based games written in words in simple scenarios in which players can create stories through interacting with in game characters, items and the environment, usually by issuing defined sets of commands in a textual interface (Buckles, 1985).

IF narrative worlds besides having "rooms" and "doors" connecting them usually describe objects or things that are placed in various rooms which can be examined and interacted with. Such object can for example be:

- non-player characters (NPCs) that the player can communicate with,
- containers that might have other objects within,
- edibles that can be consumed,
- wearables that can be used as clothes or equipment, etc.

As opposed to most computer games focused on graphics, IF is focused on the story and narrative which makes it an interesting and different medium similarly as printed novels differ from movies. While some authors consider it a literary genre (Ziegfeld, 1989) other argue that due to the users interaction and other game elements IF works are mostly computer games (Hausknecht et al., 2020).

4 Massively Multiplayer On-line Role Playing Games

Role-playing video or computer games (commonly referred to as only role-playing games or RPGs) are a game genre in which the player controls the actions of some protagonist (or potentially several party members) in a world which is well defined (Collins and Šabanović, 2021). A massively multi-player on-line game (MMOG) is a (computer) game that supports a great number of players playing on-line simultaneously causing or even fostering interaction among them (Raith et al., 2021). Massively multi-player on-line role playing games are thus a mixture of these two genres allowing players to control the action of their protagonist (avatar) by interacting with a potentially large user-base on-line (Biolcati et al., 2021).

The global market for MMO games is growing rapidly with an estimate of 21.94 Bn€in 2025 (The Insight Partners, 2021). While the economic importance of MMORPGs is obvious, another aspect is of equal importance: it allows us to investigate two aspects of large-scale computing - (1) social interaction of (large numbers of) players through a computing platform as well as (2) the design and implementation large-scale distributed artificial intelligence (in form of non-player

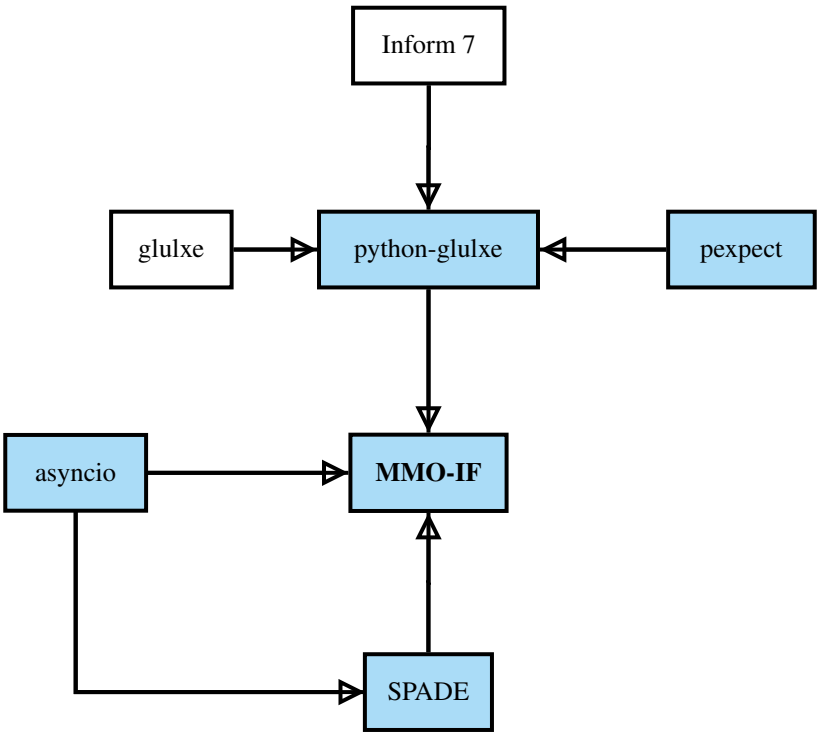

Figure 1: Dependency graph of the implemented game engine layer (blue: Python modules; white: external tools)

characters – NPCs, mobs – various monsters to be fought, as well as AI players – bots).

5 MMO-IF

In the following we shall layout our implementation of the MMO IF game engine layer. The two main characteristics of MMO that the proposed implementation attempts to achieve are enabling communication between players and notifying players when another player enters the current room. Whilst these features are basic for MMO games, using them in an IF environment which is limited to a text only interface, has its particularities.

The communication between players refers to the ability to send a message either privately, to a single player or broadcast to everyone. To make the experience of communication feel more natural, a restriction has been put in place that allows players to send their messages only to players that are located in the same room as the player. To start any type of communication, it is required from a player to execute it by typing in a special command (in our case the command has to start with the `@` symbol). The format is `@player message`, where `player` must be a valid player username to send the message directly, or `everyone` to send it to everyone. Obviously, `message` refers to a string (which might include white spaces) that is to be previewed to the receiver. On the receiving side the agents connected to the MMO server have to monitor communication and react in case a message is sent to the player in question.

The second implemented MMO characteristic relates to notifying a player when another player enters the current room. Contrary to the communication which must be executed by players themselves, notifications about players entering the room are dispatched automatically, every time a player moves between rooms.

We have used a number of technologies to build the described layer[5]. The primary technologies could be narrowed down to Inform 7, Python, and the Extensible Messaging and Presence Protocol (XMPP) through the Smart Python Agent Development Environment (SPADE)[6] framework (Palanca et al., 2020). As the idea is to develop the game engine layer on top of Inform 7, a developer is not required to add any additional code to its IF game, but only to set up a room structure (which is a usual requirement for IF games). Figure 1 shows the various dependencies. The layer is implemented in form of a MAS (Okreša Đurić et al., 2018) in which individual players are represented as agents (game avatars) which are aware of their local environment whilst to anticipate the actions of other players in the game they communicate with a server agents that synchronizes the global state of the game.

To process input or output to the IF game in order to accommodate stated MMO characteristics, a Python

[5]The full code is vailable here: https://github.com/AILab-FOI/MMO-IF

[6]Available here: https://spade-mas.readthedocs.io

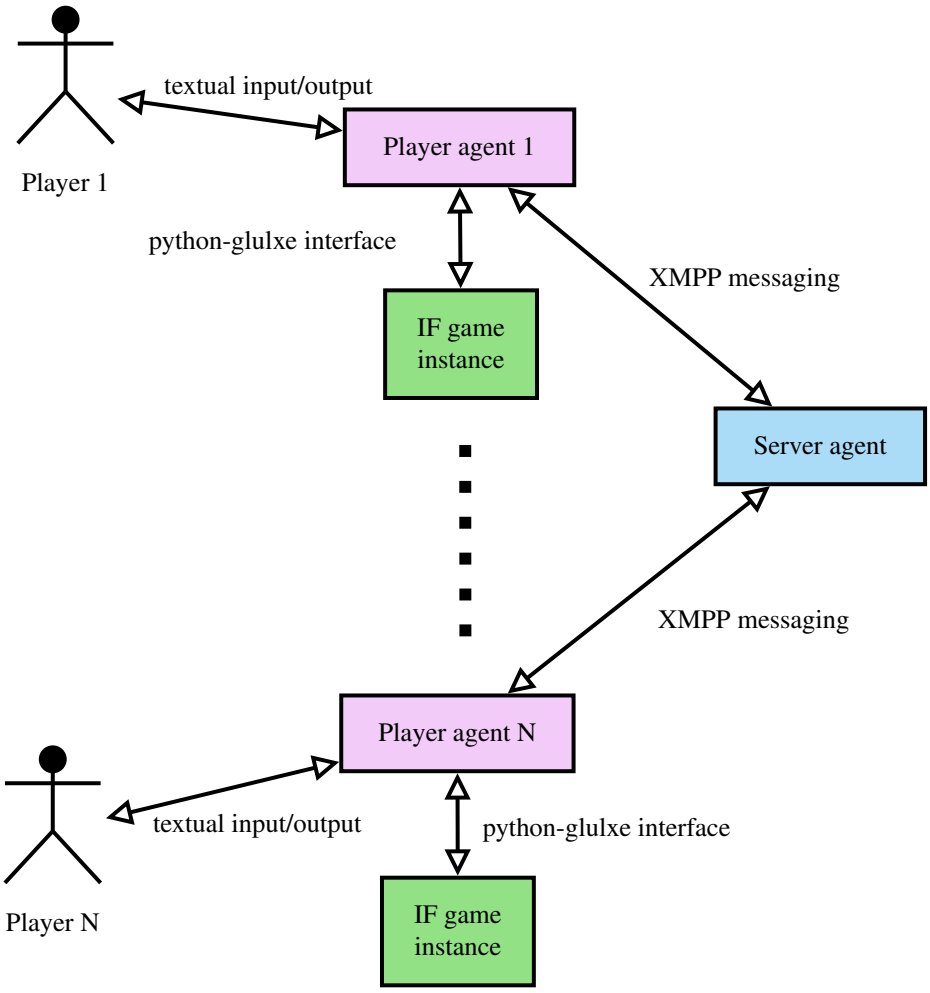

Figure 2: Architecture of interconnected game instances

interface has been put in place serving as a layer between a player and the game. By intercepting the player's input, the interface is used to check if the input is in a format which would indicate that a player is attempting to dispatch a message. On the other hand, by intercepting the game output, the role of the interface is to detect whether a player has changed rooms. On each room change, the game returns an output saying `You entered [name] room` where `name` refers to the name of the room. That way, by parsing the output, the interface keeps track of the player's location.

The XMPP facilitates the communication between players. This is achieved by having a centralized server agent that is aware of the active players and their locations. The two types of the XMPP messages that the server reacts to are when a player changes the location, and when a player wants to send a message. The location change type of the XMPP message is significant as it is involved in the restriction logic that defines who will be the receivers of a message that the server dispatches. Furthermore, when the server receives this type of message, it sends a notification to all players located in the same room as the sender, indicating what player changed its location. The other type of the XMPP message relates to ability to exchange messages between players. On the server end, this includes logic to determine who all shall be receivers of the message. Eventually, the player agent accepts the XMPP messages that the server dispatches, and shows them accordingly in each player's interface.

Figure 2 gives a visual description of how the described system components are connected, as well as how the communication is established.

6 Implementation Example

Below is a basic IF game developed in Inform 7 to support our MMO implementation.

```
"RoOms" by Tomislav Peharda
Release along with an interpreter.
Release along with a website.

When play begins:
        say "You got into the roOm! Try
            ↪ communicating with the
```

```
            ↪ other people in here with @
            ↪ "

The yellow is a room. The description is
    ↪ "You entered the yellow room.".
The blue is a room. It is east of the
    ↪ yellow. The description is "You
    ↪ entered the blue room.".
The green is a room. It is west of the
    ↪ yellow. The description is "You
    ↪ entered the green room.".
The red is a room. It is south of the
    ↪ yellow. The description is "You
    ↪ entered the the red room.".
The orange is a room. It is north of the
    ↪ yellow. The description is "You
    ↪ entered the orange room.".
```

When the IF game is started, a player is positioned in one of the available rooms. In our example, there are five rooms that a player can move between and all of them are named by colors. By default, a player starts in the yellow room. In each direction, there is a single room available as shown on figure 3.

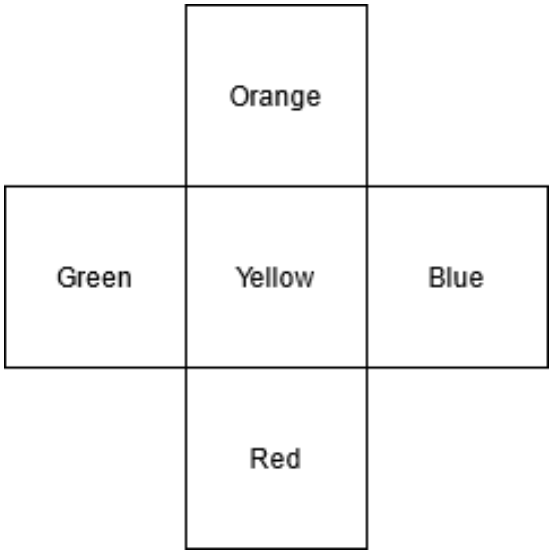

Figure 3: IF example game room structure

The proposed MMO-IF implementation can work along with any Inform 7 game without any adjustments needed. The only requirement that the game must fulfil is that the output format indicating room change also remains unchanged (example: "You entered the yellow room").

In this example we have chosen to showcase three players, but an arbitrary number of players could join the game having the adequate game interface. Below are three blocks showing the output of each player. The players and their corresponding output are in this order tpeharda_agent_alpha, tpeharda_agent_beta, and tpeharda_agent_omega. The output previews both the use-cases, communication between players, and notifications when a player enters a room.

```
# tpeharda_agent_alpha
You entered the yellow room.

tpeharda_agent_beta has entered the
    ↪ yellow room.
tpeharda_agent_omega has entered the
    ↪ yellow room.

> go west
You entered the green room.

tpeharda_agent_beta has entered the green
    ↪  room.
tpeharda_agent_omega has entered the
    ↪ green room.

> @everyone hey everyone!

> @tpeharda_agent_beta hey beta!
```

```
# tpeharda_agent_beta
You entered the yellow room.

tpeharda_agent_omega has entered yellow
    ↪ room.

> go west
You entered the green room.

tpeharda_agent_omega has entered the
    ↪ green room.

tpeharda_agent_alpha said: hey everyone!
tpeharda_agent_alpha said: hey beta!
```

```
# tpeharda_agent_omega
You entered the yellow room.

> go west
You entered the green room.

tpeharda_agent_alpha said: hey everyone!
```

Player tpeharda_agent_alpha has joined the game before everyone else hence it is notified when tpeharda_agent_beta and tpeharda_agent_omega joined the same room. The player in question also dispatches two messages, one directed to everyone with content "hey everyone!", and another one directed precisely to tpeharda_agent_beta with content "hey beta!". Thus, the other two players receive a message depending if they are the specified receiver.

7 Conclusion

In this paper we have provided a novel game engine layer that introduces MMO to IF games using a MASs

approach. It can be used in any existing IF game developed in Inform 7 without any (or sporadically minor) adjustments. We have provided an initial implementation of a MMO IF game to test the implemented layer by using the Inform 7 IF programming language, Python to create an interface and particularly SPADE for the implementation of agents. Whilst the proof-of-concept game is simplistic, it shows a promising social interaction and user experience not usually present in IF games. The MASs approach taken herein allows us to implement numerous AI techniques in modelling player interaction as outlined in (Schatten, Okreša Đurić, and Peharda, 2021).

In the current implementation only communication and player presence are implemented, but there are many other important features that should be addressed mostly related to synchronization of the game state, items and in game characters. Such and similar features will be subject to our future research.

Acknowledgement

This work has been fully supported by the Croatian Science Foundation under the project number IP-2019-04-5824.

References

Aikin, J. (2009). The inform 7 handbook.

Biolcati, R., Pupi, V., & Mancini, G. (2021). Massively multiplayer online role-playing game (mmorpg) player profiles: Exploring player's motives predicting internet addiction disorder. *International Journal of High Risk Behaviors and Addiction*, *10*(1).

Buckles, M. A. (1985). *Interactive fiction: The computer storygame" adventure"* (Doctoral dissertation). University of California, San Diego.

Collins, S., & Šabanović, S. (2021). " what does your robot do?" a tabletop role-playing game to support robot design. *2021 30th IEEE International Conference on Robot & Human Interactive Communication (RO-MAN)*, 1097–1102.

Crawford, R., & Chen, Y. (2017). From hypertext to hyperdimension neptunia: The future of vr visual novels: The potentials of new technologies for branching-path narrative games. *2017 23rd International Conference on Virtual System Multimedia (VSMM)*, 1–7. https://doi.org/10.1109/VSMM.2017.8346298

Gómez-Albarrán, M., Sarasa-Cabezuelo, A., Sierra-Rodríguez, J.-L., & Temprado-Battad, B. (2021). Authoring and playing interactive fiction with conventional web technologies. *Multimedia Tools and Applications*, 1–43.

Hausknecht, M., Ammanabrolu, P., Côté, M.-A., & Yuan, X. (2020). Interactive fiction games: A colossal adventure. *Proceedings of the AAAI Conference on Artificial Intelligence*, *34*(05), 7903–7910.

James Pita, S. B., Brian Magerko. (November 2007). True story: Dynamically generated, contextually linked quests in persistent systems. *Future Play '07: Proceedings of the 2007 conference on Future Play*, 145–151.

Jane, F. (2014). Untangling twine: A platform study. *DiGRA '13 - Proceedings of the 2013 DiGRA International Conference: DeFragging Game Studies*.

Okreša Đurić, B., Rincon, J., Carrascosa, C., Schatten, M., & Julian, V. (2018). Mambo5: A new ontology approach for modelling and managing intelligent virtual environments based on multi-agent systems. *Journal of Ambient Intelligence and Humanized Computing*.

Palanca, J., Terrasa, A., Julian, V., & Carrascosa, C. (2020). Spade 3: Supporting the new generation of multi-agent systems. *IEEE Access*, *8*, 182537–182549.

Raith, L., Bignill, J., Stavropoulos, V., Millear, P., Allen, A., Stallman, H. M., Mason, J., De Regt, T., Wood, A., & Kannis-Dymand, L. (2021). Massively multiplayer online games and well-being: A systematic literature review. *Frontiers in Psychology*, *12*, 2369.

Schatten, M., Okreša Đurić, B., & Peharda, T. (2021). An agent-based game engine layer for interactive fiction. *International conference on practical applications of agents and multi-agent systems*, (in print).

Schatten, M., Okreša Đurić, B., Tomičić, I., & Ivković, N. (2017). Agents as bots–an initial attempt towards model-driven mmorpg gameplay. *International conference on practical applications of agents and multi-agent systems*, 246–258.

Schatten, M., Okreša Đurić, B., Tomičič, I., & Ivković, N. (2017). Automated mmorpg testing–an agent-based approach. *International conference on practical applications of agents and multi-agent systems*, 359–363.

The Insight Partners. (2021). *Mmo games market to 2025 - global analysis and forecast by genre, platform and revenue model*. https://www.theinsightpartners.com/reports/mmo-games-market

Ziegfeld, R. (1989). Interactive fiction: A new literary genre? *New Literary History*, *20*(2), 341–372.