An Agent-based Game Engine Layer for Interactive Fiction*

Markus Schatten[1][0000−0001−6910−8675], Bogdan Okreša Đurić[1][0000−0001−5653−6646], and Tomislav Peharda[1][0000−0002−2081−4816]

Artificial Intelligence Laboratory, Faculty of Organization and Informatics, University of Zagreb, Pavlinska 2, 42000 Varaždin, Croatia
(markus.schatten, dokresa, tomislav.peharda)@foi.unizg.hr
http://ai.foi.hr

Abstract. Interactive fiction (IF) is a type of computer game in which players use text commands inside a literary narrative in order to influence the environment, the story and/or characters. We have developed an agent based game engine layer that allows us to introduce intelligent agents into such games including but not limited to chatbots, autonomous agents, expert systems as well as implement ontology based content generation. We provide demo implementations of games which include the implemented methods and show the benefits of using them.

Keywords: interactive fiction · artificial intelligence · computer games · multiagent systems.

1 Introduction

Interactive fiction (IF), text adventures, gamebooks and even in some cases visual novels comprise computer games in which players interact with the game using text commands. These narrative worlds usually consist of a number of rooms (whereby the term "room" is very broadly defined and can include any kind of imaginable space or even states of mind) connected by doors (again very broadly defined), and in which objects or things can be placed that can be examined and interacted with. Such things can, for example include non-playing 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.

2 Main Purpose

An important aspect of game engine design is the integration of artificial intelligence (AI) [2]. Integrating AI into IF seemingly presents an interesting challenge due to specifics of the medium. Most games in industry often use very limited capabilities of (especially modern) AI which is why we decided

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

to introduce a game engine layer that allows for the implementation of fairly advanced concepts.

We have developed our game engine layer above Inform 7,[1] a declarative programming language for the development of IF based on natural language syntax. The following listing shows a way to describe a world in Inform 7:

```
"The Dungeon" by "Markus Schatten, Bogdan Okreša Đurić & Tomislav Peharda".
When play begins:
      say "You find yourself in a dungeon surrounded by darkness. " ;
The pit is a room. The description is "This is where you woke up."
A torch is here. The description is "You can see a dim light flickering a few
    ↪ steps away from you." ...
```

An example interpreter session is shown in the following listing:

```
You find yourself in a dungeon surrounded by darkness.
The Dungeon
An Interactive Fiction by Markus Schatten,Bogdan Okreša Đurić & Tomislav Peharda
The pit
This is the place where you woke up.
You can see a torch and a Chest (closed) here.
--> open chest
You open the Chest, revealing an old smelly cheese.
--> take torch and cheese
torch: Taken.
old smelly cheese: Taken. ...
```

Whilst such an interface allows a player to use numerous commands based on natural language processing (NLP), the implementation of game actors like NPCs or mobs and the interaction with them is fairly limited to predefined mechanics.

3 Demonstration

We have developed an agent-based Python interface[2] to the glulxe IF interpreter shell that can execute a number of IF formats in terminal sessions. The developed interface allows us to place filters in front of the IF shell and thus interact with the player on one side and control the game on the other (see Fig. 1). For the implementation we have used the Smart Python Agent Development Environment (SPADE) [1]. In the following, we will show proof-of-concept games building upon this interface.

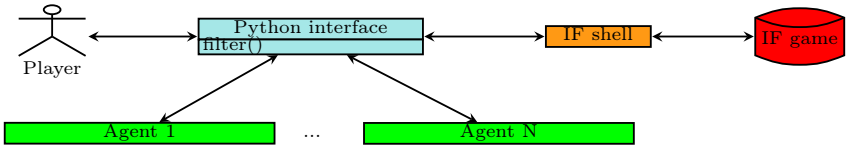

Fig. 1. Python interface to an IF shell

3.1 Chatbot Agents

Whilst NLP has since its beginning been a part of IF, the textual interface to the player has always been constrained to a certain number of commands. Also, it does not recognize any possible synonyms of objects or artifacts defined in the

[1] Available here: `http://inform7.com/`
[2] Available here: `https://github.com/AILab-FOI/python-glulxe.git`

game (if they are not explicitly encoded into the game) nor does it recognize common phrases. To implement a chatbot agent we have used Chatterbot[3] as can be seen in the following code excerpt.

```
def train( bot ):
   bot.set_trainer( ListTrainer )
   bot.train( [ 'where am i', 'look' ] )
   bot.train( [ 'what is this place', 'look' ] )
   bot.train( [ 'give me that torch', 'take torch' ] )
   bot.train( [ 'i want that torch', 'take torch' ] )
   bot.train( [ 'what is in that chest', 'open chest' ] )
   bot.train( [ 'let me open that chest', 'open chest' ] )
   bot.train( [ 'yay cheese', 'take cheese' ] )
   bot.train( [ 'take the cheddar', 'take cheese' ] ) ...
```

In this way we were able to train the chatbot to understand a number of common phrases that may be used by the player and turn them into the previously mentioned predefined commands, as shown in the following game session:

```
--> where am i
This is the place you woke up.
You can see a torch and a Chest (closed) here.
--> gimme the torch
Taken.
--> what's in that chest
You open the Chest, revealing an old smelly cheese.
--> take the cheddar
Taken.
```

Besides using chatbots as a means of achieving user friendliness of the interface, we could have used it to add additional personality traits to in-game NPCs. For example, we could train one chatbot for each NPC including various special types of conversations that can be understood and performed by each of them to boost player experience.

3.2 Autonomous (Background) Agents

Autonomous background agents can provide us with additional dynamics in IF environments. In the following example we show how IF games can be manipulated by an external agent that randomly generates actions thus directly impacting the game-play regardless of the player's actions. In our example, there is an agent that generates actions at random times. We have built in two actions that can teleport or disarm a player. The following listing shows the implementation of these two actions in Inform 7:

```
Disarming is an action applying to nothing. Understand "disarm" as disarming.
Instead of disarming:
      if the player carries anything:
            say "The elf disarmed you";
            now everything carried by the player is in the location;
      otherwise:
            say "The elf tried to disarm you, but you carry nothing"
Teleporting is an action applying to nothing. Understand "teleport" as
    ↪ teleporting.
Instead of teleporting:
      say "Elf teleported you to a different room...";
      move the player to a random room
```

[3] Available at: `https://chatterbot.readthedocs.io/`

When the autonomous agent decides to interrupt the game by invoking a command, the agent that communicates with the game receives the command and processes it by sending an appropriate command to the interpreter. This provides interesting dynamics to IF games which are usually static, i.e. can only be changed by user actions or special types of events.

3.3 Expert Systems

Expert systems (ESs) can be of great value to IF game design especially for the implementation of certain "expert" NPCs that can help the player to decide about certain situations. As an example we have developed a very simple decision tree based ES that can recognize four types of cheese. By using our interface we can allow the player to interact with the ES agent in-game when they for example ask some NPC (in our example the orc lady) about cheese as shown in the following listing:

```
--> ask orc lady about cheese
Orc Lady: Is the cheese soft?
--> no.
Orc Lady: Does is taste very umami?
--> yes.
Orc Lady: Ahh... parmesan, king of all cheeses!
```

3.4 Generating Content

Although a narrative of an IF instance could be considered similar to a book, and therefore unalterable, the digital context of IF encourages the idea of having parts of such a narrative, or indeed narrative as a whole, generated automatically, as opposed to having been written by a human.

The approach herein is based on a developed ontology that consists of concepts that can be used to describe the world that should be generated. The concepts existing in the generated world represent a subset of all the concepts that are modelled as available in the observed world. The following listing shows an example random generated world using the developed ontology:

```
Meduseld is a room. The description of Meduseld is "You are now in the Golden
    ↪ Hall of Meduseld, the seat of power in Rohan." Understand "The Golden
    ↪ Hall" as Meduseld.
A metal throne is a thing in Meduseld. The description of the metal throne is "
    ↪ This is the throne of the ruling House of Rohan."
A large chest is a container in Meduseld. The description of the large chest is
    ↪ "A large chest that can house many items." It is opaque and openable.
```

4 Conclusion

In this paper we have implemented an agent based game engine layer and shown a number of possible use cases. With Python being one of the most popular programming languages for AI with thousands of libraries an modules available this opens a wide set of possibilities for testing various approaches. Our future research will be focused on the implementation of multiplayer features for IF.

5 References

1. Palanca, J., Terrasa, A., Julian, V., Carrascosa, C.: Spade 3: Supporting the new generation of multi-agent systems. IEEE Access **8**, 182537–182549 (2020)
2. Yannakakis, G.N., Togelius, J.: Artificial intelligence and games, vol. 2. Springer (2018)