In Search for the Human Factor in Rule Based Game AI: The GrinTu Evaluation and Refinement Approach

Swen Gaudl
Children's Media
Fraunhofer Institute of Digital Media Technology
Hirschlachufer 7
99084 Erfurt, Germany
swen.gaudl@idmt.fraunhofer.de

Klaus P. Jantke
Children's Media
Fraunhofer Institute of Digital Media Technology
Hirschlachufer 7
99084 Erfurt, Germany
klaus.jantke@idmt.fraunhofer.de

Rainer Knauf
Artificial Intelligence Group
Faculty of Computer Science and Automation
Technical University Ilmenau
PO Box 100565
98684 Ilmenau, Germany
rainer.knauf@tu-ilmenau.de

Abstract

What is the biggest difference between playing a game against a human or against a computer generated player? Why do many people believe it is more challenging to play with humans than playing with an artificial player? The big success of massive multiplayer games and the huge number of so-called "LAN parties", where players meet and play with each other, seems to be related to the human demeanor of the players. All this indicates, that the current state of game AI is unsatisfactory compared to the performance of human players. This paper introduces a tool for analyzing basic computer games with incorporated AI modules which store strategies for performing the behavior of artificial players. This sets the stage for a systematic evaluation and refinement of rule based game AI.

Introduction

Imagine a computer game with a programmed adversary NPC (shorthand for "non-player character") which is represented as a rule-based agent and the NPC behaves so well, that it successfully passes an adapted TURING TEST. And now imagine you just delete one rule from the rule base and the resulting NPC fails the TURING TEST.

The resulting consequence out of this is, that before removing this particular rule, the NPC was represented as a minimal version of a rule base including the human factor.

By analyzing the differences between human behavior and intelligent rule-based behavior, we are searching for a way to create artificial human players. In his paper on machine learning (Turing 1950) Alan Turing described an idea to validate a system mimicking a human. There are different kinds of approaches regarding this topic. One of the most popular approaches was WEIZENBAUM'S Eliza program (Weizenbaum 1966).

By evaluation and refinement of rule-based NPCs in digital games it should be possible to see the differences between human and artificial behavior therein.

To reach this goal we created an environment to perform TURING TESTS with NPCs and humans.

JOSTLE 2007

The first game which will be integrated into the GRINTU framework is "JOSTLE 2007", which emphasizes the aspect of jostling stones on a board. JOSTLE was developed by Klaus P. Jantke in the report (Jankte 2007). It was selected because it is easy to learn and understand in terms of game rules and complexity of play. The board is easy to grasp which allows people without any experience on digital games or small children to play the game.

For further information on JOSTLE 2007 please consult (Jankte 2007).

GRINTU-Framework

The GRINTU framework was designed to meet various requirements (GrinTu being an anagram of TURING). The platform is strictly separated into different parts namely the network, the game library, and the analysis package.

To meet the requirement of handling many users, we chose a client server architecture, which gives us the possibility to store the data created by the user in our data base.

The server will use the data base to determine which NPC should play against which proband. After a finished game, the server collects the results of those games.

By using the currently implemented PROLOG NPC module it is possible to run native ISOPROLOG scripts. Prolog offers an easier way to analyze rule-based NPC strategies compared to strategies written in JAVA or C. After getting feedback from the probands we can easily modify the rule base and rerun the TURING TEST for further refinement.

The data mining module is still under development. We constructed the client in a way that offers the possibility to read nearly all user interactions with the game such as mouse movements, pauses and key inputs. For later analysis we are also able to log the course of gaming.

Another part of the data mining module is the questionnaire, which is shown after each game and is used for a survey regarding the behavior of the other players. The data collected by the data mining module is sent to the server and stored in a data base using cross references for all participating players, that means both humans and NPCs.

AI Strategies

There are many strategies which can be applied to an NPC in JOSTLE ranging from easy to sophisticated ones. For reasons of simplicity, we currently use a specific type of strategy which operates without any history or long term goal. In the following listings the predicate worth defines a simple strategy of an NPC which looks for a piece that can be moved a maximum amount of fields.

```
worth([Stone|L],Best):-
    worth1(L,Stone,Best).

worth1([],A,A).
worth1([[X,Y,Z]|R], [Xa,Ya,Za], Best):-
    X > Xa, !, worth1(R,[X,Y,Z],Best).
worth1([_|R],A,Best) :-
    worth1(R,A,Best).
```

The worth predicate receives a list of all gaming pieces and as result returns the one which was best according to the strategy.

In the code above, the core of the strategy can be seen starting at the fifth line. The worth predicate compares the X value of two play pieces and after comparing all pieces it selects the furthest ahead. A change of $X > Xa$ to $X < Xa$ alters the complete behavior of the NPC. With $X < Xa$, the NPC always moves the last piece which results in the fact that all pieces are moved relatively close to each other in so called clusters.

With this knowledge in mind, the implementation of the worth predicate reveals that the implemented predicate does not always move the correct piece. In the upper implementation, the "SWITCH" fields which allow a piece to be transported further ahead were not comprised. With these fields it is possible to bypass long distances.

A smarter implementation would be something like this:

```
worth2(L,Stone):-
    member(Stone,L), die(D),
    [A,_,_] = Stone, B is A+D,
    minTile(B,'switch@').

worth2([S|L],Stone):-
    worth21(L,S,Stone).

worth21([],A,A).
worth21([[X,Y,Z]|R], [Xa,Ya,Za], Best):-
    X > Xa, not(backTile(X)), !,
    worth21(R,[X,Y,Z],Best).
worth21([_|R],A,Best) :- worth21(R,A,Best).
```

The above used base predicates were dropped for reasons of space. In the worth2 implementation, the minTile(B,'switch@') predicate checks if the next possible position is a "SWITCH" field whereas the backTile(X) predicate checks if the next position is a "BACK-3" field, a field which transports a piece three fields towards the "START" field.

Obviously, coding rules that are given informally by natural language is a challenge and bears the risk to lose information in the translation process.

In the previous example, the first implementation does not fulfill its informally given definition, because it does nothing more than moving the furthermost piece.

The second implementation, which comes closer to the natural intention is a little more complex, because it is composed of two different parts. The first part is to find a piece owned by the NPC which can move to a switch field by the rolled number of points. If such a gaming piece is not available, the second clause of the predicate worth2 will be applied. It searches the furthermost piece which will not move onto a back throwing field.

For easy and consistent creation of rule-based NPC behavior, it is quite important to validate changes in existing systems or evaluating new rule sets, which should meet the intent of the NPC's designer.

The term designer was used intentionally, because in modern computer games, most of the people take an artistic perspective for creating content in and around games. This approach only tries to utilize science, but without correct tools there is no utilization.

Evaluation & Refinement

In contrast to verification which is usually understood as thoroughly formal, evaluation is not possible without expert knowledge. Because our research field is in the games domain, for us every gamer is considered an expert.

There is no play without action; there is no game without interaction. Interaction can take place between a player and some opponent, which can be either an NPC or another player. Because of this fact, most of the players have an intuitive understanding of what to expect from an opponent, i.e. a "Theory of Mind", as this is called in Cognitive Science (Baron-Cohen 1991).

It is our goal to extract one required minimal rule set of several valid rule sets, so that the probands think they play against a human entity. We start with a minimal setting using three different game strategies. The NPC strategies are not assigned completely randomly to the probands. In our setting, we want to guarantee that every proband has a chance to play against all different strategies over time.

Our next step is to perfom an evaluation and refinement of such rule bases after obtaining user data along with interaction sequences during a game.

Initial steps have been taken and the results will be part of future publications.

References

Baron-Cohen, S. 1991. Precursors to a theory of mind: Understanding attention in others. In *In A. Whiten (ed) Natural theories of mind*, 233–251.

Jankte, K. P. 2007. Jostle 2007. Technical Report 29.

Turing, A. M. 1950. Computing machinery and intelligence. *Mind* 59:433–460. Reprint:Feigenbaum & Feldman, Computers and Thought, 1963, luger.

Weizenbaum, J. 1966. Eliza—a computer program for the study of natural language communication between man and machine. *Commun. ACM* 9(1):36–45.