Communication between two digital agents in Geometry Friends

Akane Yamada   s1190095

Supervised by Maxim Mozgovoy

1 Abstract

This study make digital agents communicate in Geometry Friends [1]. Geometry Friends is a cooperative puzzle game. There are two agents and many diamonds in the game. Agents cooperate together to collect all the diamonds. Some diamonds cannot be collected without cooperation. For making agents cooperate, this researcher caused them to communicate with one another. Because if they cannot communicate, they do not realize that when they aim at diamonds cooperation is required. To make agents communicate is an effective way of making games seem more realistic. In order to implement communication, a blackboard model is used. The model has three components. There are a blackboard, knowledge sources, and an arbiter. This study shows that this model can be used successfully to cause agents to communicate with each other. Moreover, this model can successfully facilitate Agent to Agent Communication(AAC).

2 Introduction

Digital games have role of story teller. If agents who are in the world of digital game move effectively, a power as story teller is more higher. The method to show agents more effectively is wide-ranging. This researcher elected to make agents look like human. To implement the method, this study simulated communication between two agents in Geometry Friends [1].

Geometry Friends is a cooperative puzzle game developed by the GAIPS INESC-ID laboratory [2]. There are two agents and many diamonds in the game. Agents cooperate together to collect all the diamonds. If there are no diamonds remaining, the game is over. There are very few examples of implementation of communication between agents in games. Consequently, this researcher created a generic model for this purpose. In this research, this researcher used a blackboard model [3] as a base to implement communication.
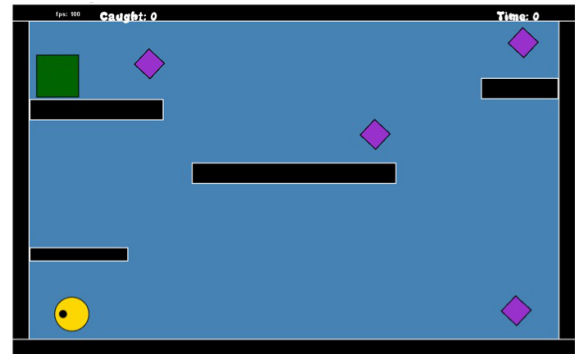

Figure 1: Screen of Geometry Friends

3 Review of past studies

To date, few studies have been conducted in this area. Of those, two of the most significant are a study of Hearsay II research on C4 Architecture. Hearsay-II is speech-understanding system developed by Carnegie-Mellon University between 1971 and 1976 [4]. They proposed a blackboard model framework and improved it as distributed blackboard model.

The blackboard approach has been applied in numerous areas, including the following:

- sensory interpretation,
- design and layout,
- process control,
- planning and scheduling,
- computer vision,
- case-based reasoning,
- knowledge-based simulation,
- knowledge-based instruction,
- command and control,
- symbolic learning,
- data fusion [3]

In 2001, Blackboard model is used as agents' internal blackboard by C4 architecture model [5]. After they published that model, the C4 architecture was introduced to the internal character model of Halo (Bungie, 2001) at a Game developers conference. Subsequently, the C4 architecture model was also used in some First

Person Shooter games [6]. The foregoing models are used for the internal decision-making of each individual agent. But in this project they are applied to communication between agents.

4 Method

4.1 Equipment

OS:Windows7
Development environment: Visual Studio 2010 Professional
GeometryFriends Framework: version31

4.2 Blackboard model

Blackboard model is a model in which many knowledge sources cooperate with each other through shared memory. The structure of the blackboard model is made up of the following components:

1.Blackboard

Blackboard is a publicly read/writeable information display.

2.Knowledge Sources(KSs)

In this architecture, knowledge sources collaborate to solve problems. KSs only have very narrow regions of expertise, and so only know what to do in a very narrow set of circumstances. If their preconditions become true, they can control the information of the blackboard.

3.Arbiter

The arbiter manages all KSs to phase the order and timing. The order is related to their strategy. If plural KS attempt to execute to same information, Arbiter have to reject some KSs according to the strategy.
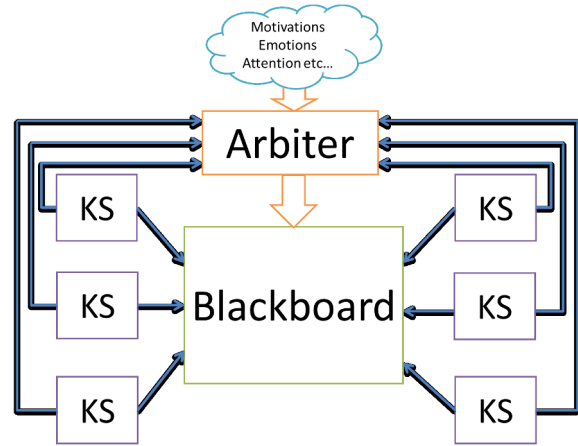
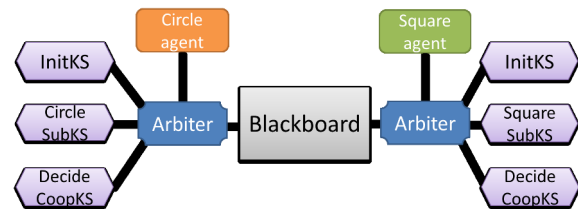The relation of these components is described in Figure 2.

5 Implementation

Figure 3 is relation of components in this project. There is one public blackboard. Agents have an arbiter and the arbiter has knowledge source.

5.1 Blackboard Record

Blackboard record is a form when an arbiter writes to blackboard. Blackboard record has 4 variables:subject, target, which knowledge source, and time.

Figure 2: Relation of components [7]

Figure 3: Relation of components in this project

5.2 KnowledgeSources(KSs)

There are 4 KnowledgeSources(KSs) in this research. Each details are below.

InitKS

InitKS gets sensor information from Agents. Then he makes diamonds information to Blackboard record as targets that anyone can aim at.

CircleSubKS

CircleSubKS is a selector for a Blackboard record that does not currently contain subject information. Based on the distance between the circle agent and the diamond, he chooses diamond's subject. The subject chosen is the one with the minimum length.

SquareSubKS

SquareSubKS is almost the same at CircleSubKS. The only difference is that its shape is a square and not a circle.

DecideCoopKS

There are some diamonds that agents cannot collect alone. In such a situation, they have to cooperate in getting them. This KS decides these diamonds according to the environment of the stage.

Blackboard Record
1.Who try to collect
2.Which diamond
3.Who published
4.When published

Figure 4: Form of blackboard record

6   Results

Implemented model was experimented in a game stage. The image of the stage is figure 5.
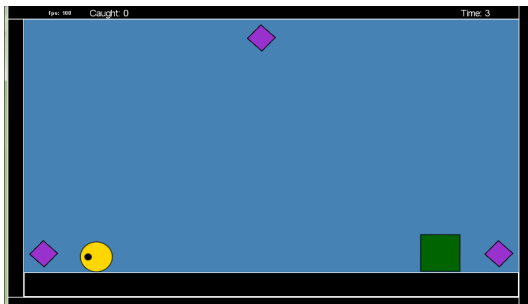

Figure 5: Game stage in this experiment

Figure 6 was written by Arbiter to a blackboard. An Arbiter writes knowledge source information par one second. DecideCoopKS worked at step14 was observed. Since step41, there were no targets. Consequently, agents could collect all diamonds by using this model.

7   Conclusion and Future work

Considering the results, this researcher could make agents communicate each other by using a blackboard model. Owing to this, agents could cooperate and collect diamonds which are needed a power both circle and rectangle.

In future work, there are 3 problems in this research.

1.Graphs

It takes 2 minutes to make navigation graphs. Moreover the graphs sometimes are not made correctly. This research did not consider obstacles arrangement. For these reasons, we have to make a more suitable graph in the future.

2.Movements

In this research, agents cannot stop immediately because they cannot control their velocity. It should be good to give consideration to a velocity of an agent and a distance between agent and target. To get diamonds which are at a high point it is neccesary to cooperate with agents.

3.Dynamic print

In this research, an arbiter wrote knowledge source information at a background. Therefore this researcher cannot see the information dynamically. In a future research have to include some tools which can make information be seen.

8   Acknowledgments

9   References

References

[1] Geometry friends. [Online]. Available: http://gaips.inesc-id.pt/geometryfriends/

[2] Intelligent agents and synthetic charactyers group. [Online]. Available: http://gaips.inesc-id.pt/gaips/index.php

[3] D. D. Corkill, "Blackboard systems," *AI expert*, vol. 6, no. 9, pp. 40–47, 1991.

[4] R. D. Fennell and V. R. Lesser, "Parallelism in artificial intelligence problem solving: A case study

of hearsay ii," *Computers, IEEE Transactions on*, vol. 100, no. 2, pp. 98–111, 1977.

[5] D. Isla, R. Burke, M. Downie, and B. Blumberg, "A layered brain architecture for synthetic creatures," in *International Joint Conference on Artificial Intelligence*, vol. 17, no. 1. Citeseer, 2001, pp. 1051–1058.

[6] J. Orkin, "Simple techniques or coordinated behavior," in *AI Game Programming Wisdom*, S. Rabin, Ed. Charles River Media, Inc., 2003, vol. 2, ch. 3.2, pp. 199–206.

[7] D. Isla and B. Blumberg, "Blackboard architectures," in *AI Game Programming Wisdom*, S. Rabin, Ed. Charles River Media, Inc., 2002, vol. 1, ch. 7.1, pp. 333–344.

```
------ Step:1 ------
From SquareSubKS:rectangle aim to a diamond(1192,648).
From CircleSubKS:circle aim to a diamond(88,648).
From InitKS        :notype aim to a diamond(616,88).
------ Step:2 ------
From SquareSubKS:rectangle aim to a diamond(1192,648).
From CircleSubKS:circle aim to a diamond(88,648).
From InitKS        :notype aim to a diamond(616,88).
------ Step:3 ------
From SquareSubKS:rectangle aim to a diamond(1192,648).
From CircleSubKS:circle aim to a diamond(88,648).
From InitKS        :notype aim to a diamond(616,88).
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
------ Step:11 ------
From SquareSubKS:rectangle aim to a diamond(1192,648).
From CircleSubKS:circle aim to a diamond(88,648).
From InitKS        :notype aim to a diamond(616,88).
------ Step:12 ------
From SquareSubKS:rectangle aim to a diamond(1192,648).
From CircleSubKS:circle aim to a diamond(88,648).
From InitKS        :notype aim to a diamond(616,88).
------ Step:13 ------
From CoopKS        :coop aim to a diamond(616,88).
------ Step:14 ------
From CoopKS        :coop aim to a diamond(616,88).
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
------ Step:39 ------
From CoopKS        :coop aim to a diamond(616,88).
------ Step:40 ------
From CoopKS        :coop aim to a diamond(616,88).
------ Step:41 ------
------ Step:42 ------
------ Step:43 ------
```

Figure 6: Printed information