UDC 004.8

Yamada A., Mozgovoy M.

communication between two digital agents in geometry friends.

1. Introduction. Digital games take a role of a story teller. Efficient, realistic storytelling requires realistic behavior of agents in a game world. AI technologies that contribute to the realism of digital agents behavior have a strong impact on the quality of computer games [1]. This research elected to make agents look like human. To implement the method, this study simulated communication between two agents in Geometry Friends [2].

Geometry Friends (see Figure 1) is a cooperative puzzle game developed by the GAIPS INESC-ID laboratory [3]. There are two agents (circle and cube) 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 research created a generic model for this purpose. This research used a blackboard model [4] as a base to implement communication.

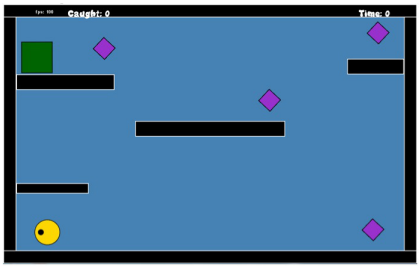

Fig. 1. Screenshot of Geometry Friends

Yamada Akane – Graduate student, University of Aizu;
e-mail: s1190095@gmail.com, tel.: +81-242-37-2664

Mozgovoy Maxim – Senior associate professor, University of Aizu;
e-mail: mozgovoy@u-aizu.ac.jp, tel.: +81-242-37-2664

2. Review of past studies. To date, few studies have been conducted in the area of game AI. Of those, two of the most significant are devoted to the study of C4 Architecture. In 2001, Blackboard model was used as agents' internal blackboard in C4 architecture model [5]. Later, 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 [7].

These aforementioned models are used for the internal decision-making of each individual agent. But in our project they are applied to communication between agents. For our experiments, we used a Windows 7 based PC machine, Visual Studio 2010 Professional environment, and GeometryFriends framework ver. 31.

3. Overview of 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 several KS attempt to trigger conflicting actions, Arbiter have to reject some KSs according to the strategy.

4. Implementation. Figure 2 shows the relation of the components in this project. There is one public blackboard. Each agent communicates with a separate arbiter, and each arbiter connects to knowledge sources.Arbiters store blackboard data in blackboard records. These structures contain four variables: subject, target, which knowledge source, and time.

We used four Knowledge Sources in our experiments:

1. **InitKS.** It gets sensor information from Agents, and transfers diamonds data to the Blackboard record as targets that anyone can aim at.

2. **CircleSubKS.** It is a selector for a Blackboard record that does not yet contain subject information. It chooses the closest diamond for the circle agent.
3. **SquareSubKS.** Same as CircleSubKS, but for the cube agent.
4. **DecideCoopKS.** There are some diamonds that agents cannot collect alone. In such a situation, they have to cooperate to get them. This KS identifies these diamonds according to the environment of the stage.

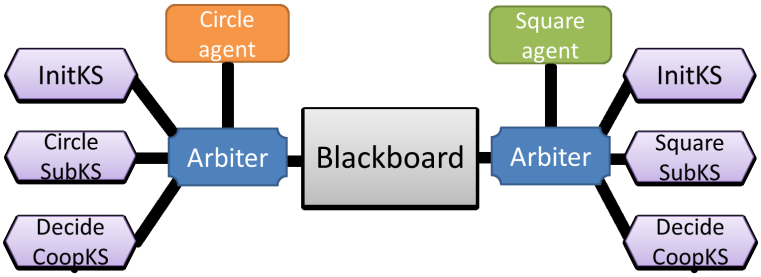

Fig. 2. Relation of components in the project

5. Results. We experimented with the blackboard model using the game stage, shown in Figure 3. The log of the Arbiter actions is provided in Figure 4. Each Arbiter writes knowledge source information once per second. The log shows that DecideCoopKS source first appeared at step 14. Since step 41, there were no more diamond targets. Consequently, the agents could collect all the diamonds by using this model.

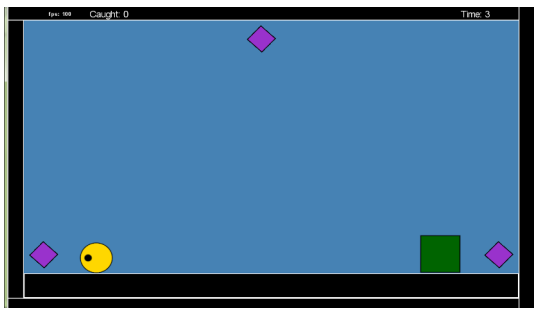

Fig. 3. Experimental game stage

6. Conclusion and Future work. Considering the results, this research could make agents communicate with each other by using the blackboard model. Agents could cooperate and collect the diamonds that required the cooperation of both circle and rectangle agents.
We see three basic problems for the possible future work.

1. **Graphs.** It takes around two minutes on typical equipment to build navigation graphs. Moreover, the graphs sometimes are not generated correctly. Currently we do not consider obstacles arrangement, so we will have to improve graph-based navigation in the future.
2. **Movements.** In our experiments, the agents cannot stop immediately because they cannot control their velocity. It should be good to give consideration to velocities the agents and to the distances between agent and targets.
3. **Dynamic print.** In the current version of the software, an arbiter is unable to provide knowledge source information in real time. In the future this function should be implemented.

Acknowledgments. We thank the GAIPS INESC-ID laboratory for permission to use Geometry Friends. We have also had the support and encouragement of our lab members. Finally, we would also like to express our gratitude to our family members for their moral support and warm encouragements.

```
------ 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 ------
```

Fig. 4. Arbiter log data

References

1. Umarov I., Mozgovoy M. Creating Believable and Effective AI Agents for Games and Simulations: Reviews and Case Study. // Contemporary Advancements in Information Technology Development in Dynamic Environments. IGI Global. 2014. P. 33–57.

2. Geometry Friends [Internet resource]. URL:http://gaips.inesc-id.pt/geometryfriends/

3. Intelligent Agents and Synthetic Characters Group [Internet resource]. URL:http://gaips.inesc-id.pt/gaips/index.php

4. Corkill D. Blackboard systems. // AI expert. 1991. Vol. 6, No 9. P. 40–47.

5. Isla D., Burke R., Downie M., Blumberg B. A layered brain architecture for synthetic creature. // International Joint Conference on Artificial Intelligence. 2001. Vol. 17. No. 1. P. 1051–1058.

6. Halo[Internet resource]:URL:https://www.halowaypoint.com/en-us

7. Orkin J. Simple Techniques or Coordinated Behavior. // AI Game Programming Wisdom. 2003. Vol. 2, P.199–206.