Available online at www.sciencedirect.com

ScienceDirect

Transportation Research Procedia 2 (2014) 452 – 455

Transportation Research Procedia

www.elsevier.com/locate/procedia

The Conference on Pedestrian and Evacuation Dynamics 2014 (PED2014)

A multi-agent system based on Unity 4 for virtual perception and wayfinding

Christian Becker-Asano [a,*], Felix Ruzzoli [a], Christoph Hölscher [b], Bernhard Nebel [a]

[a]Department of Computer Science, Albert-Ludwigs-Universitt Freiburg, 79110 Freiburg, Germany
[b]Department of Humanities, Social and Political Sciences, ETH Zrich, Clausiusstrasse 59, 8092 Zuerich, Country

Abstract

We developed a multi-agent system based on the game engine Unity 4 that allows simulating three-dimensional (3D) way-finding behavior of several hundreds of airport passengers on an average gaming PC. The agents dynamically check their surroundings for visible signs using 3D perception algorithms. Each sign is annotated with the direction of one or more exits and with meta-information such as its readability. If a sign is perceived correctly, the agent interprets its directional information relative to its own location. In combination with the head-mounted display "Oculus Rift" experiment participants can be tested in the same virtual airport terminal.

Peer-review under responsibility of Department of Transport & Planning Faculty of Civil Engineering and Geosciences Delft University of Technology

Keywords: virtual reality; multi-agent system; airport passenger perception; signage perception; game engine; wayfinding

1. Introduction and motivation

Game engines such as the "Unreal Engine" by Epic Games, Inc. (2014), or "Unity 4" by Unity Technologies (2014) provide game developers with efficient tools to rapidly implement and test new ideas. Developing state-of-the-art games requires both programming as well as design skills and, accordingly, these tools need to be easy to use for both programmers as well as designers. In contrast, multi-agent systems such as FreeWalk by Nakanishi (2004), buildingEXODUS by Pelechano and Malkawi (2008) or "CAST Terminal" by Airport Research Center GmbH (2014) are based on completely different premises; cf. Zhan et al. (2008) for an overview.

In the aim to most directly compare wayfinding behavior of virtual agents with that of humans we created a novel multi-agent system (MAS) that efficiently manages dynamic, high level agent perception utilizing the functionality of the game engine Unity 4. Due to its well-designed interface and its open architecture this game engine is accessible to both computer and cognitive scientists. In addition, it is easily combined with the affordable head-mounted display "Oculus Rift", so that the same simulation environment can be used to empirically investigate real human wayfinding behavior as well.

* Corresponding author. Tel.: +49-761-203-8251 ; fax: +49-761-203-8222.
E-mail address: basano@informatik.uni-freiburg.de

Peer-review under responsibility of Department of Transport & Planning Faculty of Civil Engineering and Geosciences Delft University of Technology
doi:10.1016/j.trpro.2014.09.059

This paper presents our ongoing work to create an MAS that faithfully simulates the perception and interpretation of signage information for crowds of autonomous agents. Therefore, we combine the expertise of cognitive psychology researchers with that of computer scientists. With respect to crowd simulation, we focus first on crowd management and public space design, thereby creating a "virtual environment" as described by Zhan et al. (2008) that will enable us to integrate and investigate models of higher cognitive functions from an Artificial Intelligence perspective.

The following section outlines the design goals for our framework, before in Section 3 implementation details are sketched. Preliminary performance results are reported on in Section 4. In Section 5 general conclusions are drawn and directions for future research are outlined.

2. Design goals

The properties of the virtual passengers in this multi-agent system were developed based on established theoretical models in human wayfinding cognition, cf. Carlson et al. (2010), as well as on the results of a research collaboration with Frankfurt Airport. In a series of laboratory-based wayfinding decision-making and eye-tracking experiments by Büchner et al. (2012) different versions of directions signage were compared for a major arrival area. The behavioral studies revealed systematic differences between sign variants with respect to correctness of wayfinding decisions among other factors.

Our multi-agent system focuses on first-person perception of signs taking dynamically changing occlusions into account. In doing so, we implement the concept of "Visibility Catchment Areas" proposed by Filippidis et al. (2006), but in contrast to their offline approach, we implement an online recalculation of sign visibility from the perspective of each agent once per simulation frame. The performance and high level functionality provided by Unity 4 allows to simulate more than 600 airport passengers at a frame rate of 60 Hz.

We aim to evaluate and improve the simulation accuracy of passenger wayfinding behavior, first. Then we will use the resulting MAS to systematically optimize the signage configuration so that the average time it takes an agent to reach its designated exit is minimal.

3. Implementation

The real-world signage properties were translated into differences in readability and text size parameters of the virtual sings. Each sign itself also has a three dimensional location, orientation, and size. Airport passengers are graphically represented as cubes with a collision geometry. In addition, a virtual camera is attached to each agent so that an experimenter can switch to a first-person view during runtime.

When an agent is close enough to a sign, a visibility check is performed casting five rays, one to the sign's center and another four to its resp. corners. Depending on how many of the rays hit their intended targets (indicated by blue and green lines in Fig. 1) or not (red lines in Fig. 1), a visibility value of the sign between zero and one is calculated. Taking the sign's overall readability (a custom value between zero and one) and the reading angle between the agent and the sign into account as well, the agent might misinterpret the sign's content and falsely judge the information irrelevant. If interpreted correctly, however, the exit information is checked for a match with the agent's goal and, in case of success, the direction information is extrapolated relatively to the sign's location and orientation. From this, the agent's next waypoint area is determined.

In our example scenario presented in Fig. 1, agents are spawned randomly at one of the four entrances in regular intervals and they are randomly assigned to reach one of the three exits. These entrances and exits also serve as waypoint areas for the agents.

A level designer manually sets up waypoint areas by simply dragging them into the scene and adjusting their dimensions. Each waypoint area contains a list of connected waypoints, which has to be filled by the designer with appropriate values. This allows the simulation of a human's tendency to proceed to large areas that can be detected from the current location when searching for an exit in a large building such as an airport terminal.

Accordingly, our agents use the rectangular waypoint areas (cp. the five light grey areas plus the five entrances/exits in Fig. 1) together with the associated connectivity graph as high level targets for navigation. When an agent enters a waypoint area and does not directly perceive a new sign, it is programmed to navigate to a randomly chosen point within the area to avoid getting stuck. When this random walk behavior was repeated a customizable number of

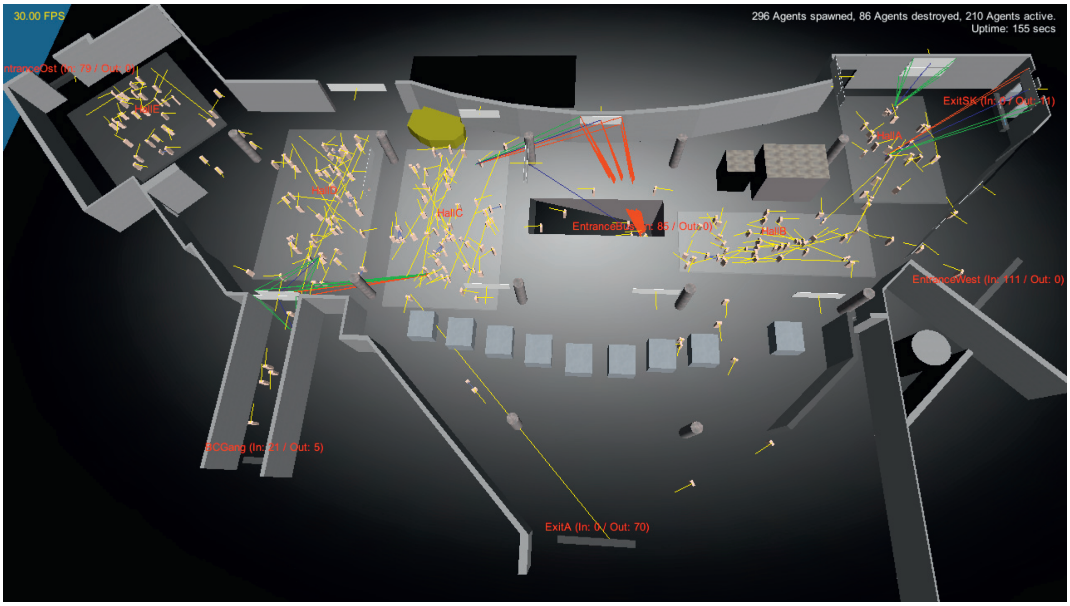

Fig. 1. A screen capture of the Multi-Agent System after 155 seconds simulation time has passed. 296 agents were spawned, of which 86 already found their corresponding exits and were destroyed, thus, leaving 210 active agents in the scene (see upper right corner). The green and blue lines indicated successful raycasts, whereas those represented by red lines were unsuccessful due to obstacles in the line of sight.

times and no goal-relevant sign was detected, the agent chooses a new waypoint area based on the connectivity graph. Concurrently, low level routing and dynamic obstacle avoidance are achieved by Unity's inbuilt functions[1].

At any time, the system designer can select an agent and switch into the first-person viewing mode as presented in Fig. 2. Thereby, an individual agent's state can be examined.

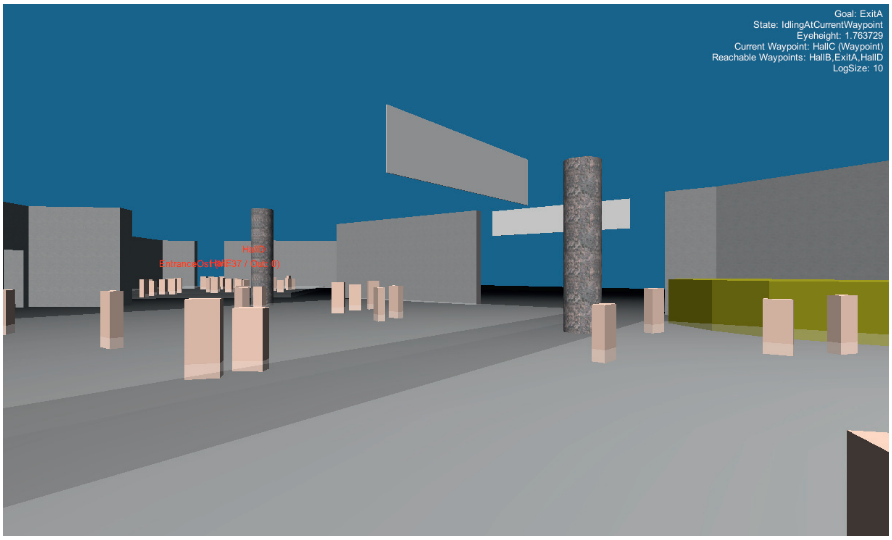

Fig. 2. The first-person viewing mode with status information of the selected agent in the upper right corner of the viewport

[1] Demo video at: `https://www.becker-asano.de/index.php/videos/vr-games/91-multiagent-systems-games-videos`.

4. Preliminary results

We performed preliminary tests regarding the general system performance and the simulation accuracy as compared to real world data. The desired simulation rate of 60Hz can be kept on a Core i5 3.4 GHz CPU with 8GB of RAM and an AMD Radeon HD 7800 Series GPU (4GB) with up to a maximum of 600 active agents. Then the simulation rate starts to drop and approx. 30Hz is reached with 950 active agents. With more than 1700 active agents the simulation rate drops slightly below 12Hz but the system is still running, although the simulated space provided to the agents seems too small for this number of agents.

Furthermore, a comparison with empirical data derived from eye-tracking studies shows promising similarities concerning the navigation success rate of the virtual agents. In the empirical trials the signage was varied with respect to its precision and ambiguity regarding a number of target locations (airport gates). When the virtual signage is systematically varied in a way corresponding to the empirical trials, then the virtual agents' performance changes in a similar fashion as that of the human participants

5. Conclusions

In comparison to other airport simulation systems, e.g. the one by Pelechano and Malkawi (2008), our system more accurately simulates 3D perception and interpretation of signage information. In addition, our system is ready to be used in experiments that employ a first-person perspective as similarly provided by FreeWalk (Nakanishi, 2004).

However, many features that influence human behavior on the cognitive and emotional level are still missing in our agent simulation. For example, the agents do not line up to pass checkpoints, ask airport personnel for help, or get distracted by advertisements and shops. A high-quality model of the airport terminal is currently under development that integrates 3D computer models of the corresponding areas. At present, the WASABI affect simulation architecture by Becker-Asano and Wachsmuth (2010), is being integrated in Unity 4 to simulate the emotion dynamics of the agents. Sign readability and crowd density will then influence each agent's dynamic emotional state, which, in turn, will change its movement parameters.

Finally, virtual reality experiments are currently being conducted using the head-mounted display "Oculus Rift". The acquired data will be compared with trajectory data derived from mobile eye-tracking data that is currently collected in the real airport terminal by Schwarzkopf et al. (2013). This will hopefully allow us to evaluate and improve the accuracy of our airport passenger simulation heuristics in the future.

References

Airport Research Center GmbH, June 2014. CAST Terminal. http://www.airport-consultants.com.

Becker-Asano, C., Wachsmuth, I., January 2010. Affective computing with primary and secondary emotions in a virtual human. Autonomous Agents and Multi-Agent Systems 20 (1), 32–49.

Büchner, S. J., Wiener, J., Hölscher, C., 2012. Methodological triangulation to assess sign placement. In: Proc. of the Symp. on Eye Tracking Research and Applications. ACM, pp. 185–188.

Carlson, L. A., Hölscher, C., Shipley, T. F., Dalton, R. C., 2010. Getting lost in buildings. Current Directions in Psychological Science 19 (5), 284–289.

Epic Games, Inc., June 2014. Unreal Engine. http://www.unrealengine.com/.

Filippidis, L., Galea, E. R., Gwynne, S., Lawrence, P. J., 2006. Representing the influence of signage on evacuation behavior within an evacuation model. Journal of Fire Protection Engineering 16 (1), 37–73.

Nakanishi, H., 2004. Freewalk: a social interaction platform for group behaviour in a virtual space. International Journal of Human-Computer Studies 60 (4), 421–454.

Pelechano, N., Malkawi, A., 2008. Evacuation simulation models: Challenges in modeling high rise building evacuation with cellular automata approaches. Automation in Construction 17 (4), 377 – 385.
URL http://www.sciencedirect.com/science/article/pii/S0926580507000908

Schwarzkopf, S., von Stülpnagel, R., Büchner, S., Konieczny, L., Kallert, G., Hölscher, C., 2013. What lab eye-tracking tells us about wayfinding. a comparison of stationary and mobile eye-tracking in a large building scenario. In: 1st Intl. workshop on eye tracking for spatial research, ET4S 2013 (in conj. with COSIT 2013). Scarborough.

Unity Technologies, June 2014. Unity 4 game engine. http://unity3d.com/unity.

Zhan, B., Monekosso, D. N., Remagnino, P., Velastin, S. A., Xu, L.-Q., 2008. Crowd analysis: a survey. Machine Vision and Applications 19 (5-6), 345–357.