Product vision

Team WhySoSerious
Max Groenenboom 4169298
Dennis van Peer 4321138
Marciano Jorden 4203992
Rico Tubbing 4254104

May 4, 2016

Contents

1 Introduction

This document describes what the product should become, what will make it unique and what the most important aspects of the final product are. It describes who will use it and what those people want, and how we reach that, and it describes when it should be done.
We will be working for a company called Tygron.

Tygron, 2016 is the developer of an online 3D multiplayer game engine for urban communities. A community can design and maintain a very detailed simulation of their area. In this simulation different stakeholders are able to interact and change the simulated environments to see how this effects the community.
The local government of a city for example could use this simulation to see if their plan to renew the infrastructure would cause a significant change in noise pollution for the affected area's.

Just like in real life, the simulation contains a set of stakeholders each with their own goals and permissions. When one or more of these stakeholders is not willing or able to participate in Tygron's simulation the other stakeholders are required to fulfill those parts of the simulation.
Since this is a lot of unnecessary work for them, Tygron has decided to simulate the actions of the missing stakeholders with artificial intelligence.

This is where we come in. We are working together with 4 other groups from the TU Delft to create a simulation with different stakeholders that compete and cooperate with each other to fulfill their goals.
Afterwards we will use the goal language to create virtual agents that can simulate these stakeholders. These agents should be able to communicate with each other through the game to trade assets and reach certain compromises.

2 Target audience

The target audience for our product is Tygron. The company will use our agents as an example on which to base future agents. This means that it is essential that the implementation of the agents is of a high quality and is very well documented. We will be working at the Tygron every Tuesday. During this time we can get help from the employees on how to work with their engine. We will also discuss our ideas with them to see if we are still making the product that they expect.

This makes analyzing our target audience a lot easier. We already have a few aspects (besides code quality and documentation) that we need to incorporate into the project: The game in which our virtual humans can run must create an interesting dynamic between the different stakeholders. This means our agents can't be simple and need to communicate.

To accomplish this we have to create a language in co-operation with the other teams from the context in which our agents can send and interpret messages. The agents have to use the API provided by Tygron, which we will access using the connector developed by last year's students. Since the API has changed since then, a few modifications need to be made to the connector to get it working.

3 Customer needs

There are certain requirement the agent will have to meet. We have separated these requirements in functional and non-functional requirements.

3.1 Functional Requirements

The agent should be able to negotiate with other bots and with human players alike. This means we have to implement a system enabling the agents to negotiate with each other, and a second system to negotiate with humans.

For the negotiation with other agents, we first have to conceive a way for the agents to communicate with each other, so we have to create a language the agents will use. This has to be carefully thought out with the other groups, so this is no easy task.

Negotiation with humans is, however, a lot more difficult for a computer. We will be using the theory on Lin and Kraus, 2010 to satisfy this requirement.

3.2 Non-functional requirements

There are also certain background requirements our agent will have to satisfy. These are necessary for smooth usage of the agents, and to ensure the customer can use the bots in a proper way.

Efficiency

It is very important the agents do not take too long with their decisions. Too short is also not an option as the agents should resemble a human, if they can make a decision in less than half a second, that doesn't mean they should.

Code Quality

Another requirement is that the source behind the agents will have to be of a high quality. We will achieve this by using comments to clarify difficult pieces of code, separate the functions of the agent in modules and we will ensure a proper consistency throughout the product. If the product is delivered with a low code quality, it is impossible to maintain and customize the product. It is also necessary that all of the code and functions are properly documented.

Reliability

A third requirement is that the entire product is well tested in order to ensure reliability. We need to use different testing techniques to achieve this, like unit tests, integration tests and regression tests. This is the most effective way to improve the reliability of our program. It is impossible to test GOAL code with usual testing frameworks, so we will use the MAS testing framework (Hindriks, 2014, chapter 9) to make unit tests for GOAL. For all other code we will use the JUnit testing framework (JUnit, 2016).

4 Product attributes

The agents we are going to program will be able to simulate the actions a human would do in their situations. For the purpose of creating these bots, we have defined five different possible stakeholders of which we are going to implement one:

1. Government

   The government will ensure all stakeholders will abide by the rules. It will watch and maintain the green index, the amount of water in the area, the zoning. It might also try to fit in a AZC somewhere in the area.

2. Delft University of Technology

   This stakeholder wants to maintain its campus. The university wants to build a new building for the faculty of applies sciences. Another thing they plan to do is renovating existing faculties (Delft University of Technology, 2013). If possible, it will try to have enough parking for each faculty, and a bit of green and water around and in the campus.

3. DUWO (student housing corporation)

   This stakeholder supplies housing to students. It will want to maintain a certain amount of houses for students, that have to be close to the university. It will search for both short term and long term solutions to reach this goal.

4. Civilian housing corporation

   This stakeholder will build housing for the (non-student) civilians in the area. These civilians rather don't live too close to students, and want the area to be green enough.

5. Companies

   This stakeholder will want to build facilities close to the civilians and the students, like parking, supermarkets and recreation. This stakeholder will focus on making profit, rather than keeping the area livable.

Apart from these requirements, all of the stakeholders will also try to avoid bankruptcy.

We have chosen to implement the DUWO stakeholder, so we will now specify this stakeholder in detail.
The DUWO stakeholder will want to build housing for students (DUWO, 2014). Many students want to live at least their study duration in their student home, and some students will want to sacrifice comfort for speed: They come from over the border so they are not able to commute to the campus, so they need a student home close to the campus before their study starts. Students who live near the campus of their university will probably score better grades than students who do not live near the campus (Grayson, 1997).

5 Comparison with competitors

Currently the Tygron engine has no implementation of artificial intelligence of any kind. This means that there is no direct competitor to our agent. Some other games and simulations however, do have virtual humans managing or at least advising parts of a city. In this chapter we analyze a few of them and compare our product to theirs.

Our first competitor is "an integrated agent for playing real-time strategy games" created by the University of California Santa Cruz (MCCoy & Mateas, 2008). The agent implements a strategy manager, income manager, production manager, tactics manager and recon manager. From these only the strategy manager and income manager are relevant to our virtual human.

The strategy manager will try to build buildings in much the same way as our agent will, except the fact that the manager gets its goals from the other managers instead of the pre-defined goals that our agents will have. The goals themselves are a bit simpler for the rts-agent[1] than they are for our virtual humans, as we have to consider multiple factors when placing a building. These factors include the distance to the TU Delft and the quality of the surrounding area.

Secondly we have the much more similar "Intelligent advisor agents in distributed environments" from the Italian "Universitá Della Calabria" (Agnese Augello & Gaglio, 2010). This advisor is actually a multi-agent system (a collaboration of different agents). Our agent however, will be a singular entity with different methods, beliefs and goals.

Other than this, the advisor (in concept) is very similar to the agent that we are going to make; it uses a lot of information about the environment and considers a lot more factors than the rts-agent from the second paragraph. It also has a linguistic agent which is very similar to our agents ability to communicate with the other stakeholders.

These examples can give us a general idea about other agents with similar concepts. Our agent however has to run in the Tygron engine. This gives us an unique set of obstacles and tools to work with.

However there are also intelligent agents that work in different environments, other than the city planning context, but still share some similarities with our agent. These are often core principles and properties that can apply to different environments.

The Diplomat agent (Kraus & Lehmann, 1995), is an agent that is able to play the game Diplomacy. Diplomacy originally is a board game that is played by multiple people competitively. Except for following the game rules, each player is able to negotiate with other players to put themselves in a better position. Here, negotiation usually involve peace treaties and declarations of war on a common target. It is also possible to make false promises and lie to the competitors.
Our agent will be able to negotiate with other parties, both human and artificial, in order to be in a better position just like Diplomat, however, our agent will stay honest throughout the game. This is because, rather than a competition, the scenario in the Tygron Engine is considered a collaborative game, that should end in multiple parties getting better instead of just one absolute winner.

[1]The real-time strategy agent from (MCCoy & Mateas, 2008)

6 Target time frame and budget

Our target time frame to finish this product is 10 weeks, as long as the current education period of the TU Delft. The first two weeks will be used for orientation within the project and setting up a game environment with the other teams in which our agent will operate. The remaining weeks are reserved for creating and enhancing the DUWO student housing agent. Since we are only programming and no one is being paid we do not have a budget.

References

Tygron. (2016, May 2). Tygron next generation. Retrieved from http://www.tygron.com/

Lin, R. & Kraus, S. (2010). Can automated agents proficiently negotiate with humans? *Communications of the ACM*. Retrieved from http://delivery.acm.org/10.1145/1630000/1629199/p78-lin.pdf

Hindriks, K. V. (2014, March 26). *Programming cognitive agents in goal*. Retrieved from http://ii.tudelft.nl/trac/goal/raw-attachment/wiki/WikiStart/Guide.pdf

JUnit. (2016, May 3). Junit - about. Retrieved from http://junit.org/junit4/

Delft University of Technology. (2013). *Roadmap 2020*. Retrieved from http://www.tudelft.nl/fileadmin/UD/MenC/Support/Internet/TU_Website/TU_Delft_portal/Over_TU_Delft/Strategie/roadmap_2010_defversie.pdf

DUWO. (2014). Gecontroleerde groei. Retrieved from https://www.duwo.nl/over-duwo/publicaties/ondernemingsplan/dl-act/open-download/dl-file/koersdocument-2014-2016/?no_cache=1

Grayson, J. P. (1997). Place of residence, student involvement, and first year marks. Retrieved from http://www.canlitsubmit.ca/index.php/cjhe/article/view/183293/183251

MCCoy, J. & Mateas, M. (2008). An integrated agent for playing real-time strategy games. *Proceedings of the Twenty-Third AAAI Conference on Artificial Intelligence*. Retrieved from https://www.aaai.org/Papers/AAAI/2008/AAAI08-208.pdf

Agnese Augello, G. P. & Gaglio, S. (2010). Intelligent advisor agents in distributed environments. *Soro et al. (Eds.): Inform. Retrieval and Mining in Distrib. Environments*, 109–124. Retrieved from https://www.aaai.org/Papers/AAAI/2008/AAAI08-208.pdf

Kraus, S. & Lehmann, D. (1995). Designing and building a negotiating automated agent. *Computational Intelligence 11*. Retrieved from http://www.umiacs.umd.edu/~sarit/data/articles/13.ps