Final Report

Michiel van den Berg 4391039 michielvandenb
Stefan Breetveld 4374657 sbreetveld
Timo van Leest 4423798 timovanleest
Daan van den Werf 4369556 djvanderwerf
Job Zoon 4393899 jzoon

June 16, 2016

Contents

1 Introduction

Nowadays, automatic decision making is very important (Li & Zheng, 2010). An example of this is a Multi-agentsystem, which consists of multiple intelligent agents, computer-controlled players in an environment. These agents are controlled with automatic decision making. The purpose of this project is to develop an agent for the company Tygron. Tygron is a company that builds serious games for urban planning. Communities can generate and maintain realistic games to find solutions for city design and development projects. A game consists of a scenario in which several players play different roles, these players are real people. The goal of the virtual humans for serious gaming context project is to develop virtual humans that simulate the people in the Tygron game. Virtual humans can ensure that a serious game can be played even when not all players in the scenario are present. Each team in the project will develop one of these agents, so that we can run a simulation with only agents in the end.

The structure of this report is as follows: In chapter 2, we give an overview of the developed and implemented software product. Chapter 3 consists of reflection on the product and process from a software engineering perspective. In chapter 4 the developed functionalities are described. Chapter 5 contains a special section on interaction design. In chapter 6 evaluation of the functional modules and the product in its entirety, including the failure analysis is given. In chapter 7 an outlook is given.

2 Overview of the software product

In this chapter we will give a brief overview of the product design and implementation. We will briefly outline how it works and some of its limitations.

The developed product consists of two parts, a runtime environment and a GOAL project. The environment is used to connect with the Tygron engine and handles everything we need to make our GOAL project talk with the Tygron engine. It handles translating the xml objects, retrieved through Tygron's backend, to parameters we can use with GOAL and vice versa. It also handles the sendings of actions to the Tygron engine, like buying ground or constructing a building, this can be as simple as translating the parameters, but in certain cases this also handles some of the more complex arithmetic operations like retrieving a list of Multipolygons, a collection of points that form a 2D shape, that can be used to construct buildings. The environment is built on the Tygron sdk, for communication with the engine, and Eishub eis[1], for the communication with GOAL, and is derived from the tygronenv[2]. The functionality of the environment is limited by the actions and data available in the sdk.

The GOAL project does all the decision making, it runs in a loop until all its goals have been achieved or its manually shutdown. It is separated in modules so every distinct function has its own module, i.e. constructing or demolishing are placed in separate modules. Every cycle the agent will call the event module first which will update the stored data with data it gets from the environment, for example a list of building. It has one main module which decides what module should be called each cycle, based on its current goals. Sometimes a module is called that cannot do anything because there is no space to build a construction, it will then adopt a goal to make some space so that it can do something in that module. The goal project runs mainly on swi-prolog which is why it cannot easily handle complex arithmetic operations, this is why we need to handle these in the environment as mentioned before.

[1] `https://github.com/eishub/eis`

[2] `https://github.com/eishub/tygron`

3 Reflection on the product and process

In this section we will reflect on the product and the process of creating it from a software engineering perspective.

3.1 Reflection on the product

The product in its entirely as we have now, is not exactly as we had planned at the beginning of the project. There are some positive and negative outcomes of what we had planned to achieve. The negative outcome is that we are missing some decision logic that we had planned in the beginning of the project. The main reason for this is that we had more focus on the connector in the first few weeks. At the start of the project, we were not aware of the fact that the connector was missing some features that we needed for a good functioning agent. After we had a decent understanding of the available actions and percepts we started working on the GOAL project/agent, this is where we could actually see things change on the Tygron project. The positive outcomes of the product is that we have implemented some features that were not planned. Selling land is a feature that we implemented, but was not planned beforehand. With selling land we earn money with land we do not use.

3.2 Reflection from the software engineering perspective

We have experienced the sprints and communication in our own group as adequate. We had some starting problems, but with the grades of the first weeks, we realized that this was not the way to do a good context project. After sprint two, we made clear arrangements about the project and the collaboration as a group. In our experience, this worked out well, which also can be seen in our rising grades of the sprints as well as the CEQ.

But the biggest issue we had in this project was the communication between groups. This have caused some delay. In the first few weeks, we had only one meeting with all the groups together. The rest of the time, we worked with our own group together and communication between groups was done without face to face communication. This resulted in a very late response time for small issues, which caused some process loss. After sprint 5, we had two meetings every week and made some clear arrangements about responding to pull requests and issues about the connector.

In retrospect we should have immediately started with figuring out how the environment worked and what its limitations were. If we had this in mind, it was easier to accomplish our final result that we had planned. We should

also have asked more questions to other groups and people with more knowledge about the connector from the beginning of the project. After week 4, this was improved and we started to be more productive.

4 Description of the developed functionalities

In this section the functionalities of our agent will be described. The functionalities that we have now are based on the strategy and vision we had in mind at the beginning of the project. The features we did not implement and the reasons why we did not implement them will be described in the next chapters.

4.1 Building functionality

The agent has indicators that represents our vision of the TU Delft. The indicators that are used for building constructions are building high constructions, extending the education space and building green. The developed agent is able to achieve these indicators. It has the functionality to build high buildings, this functionality has the highest priority as it will help us reach our other indicators as well. When building high buildings, the education space will be extended as well, so the agent is able to combine these two indicators with one functionality. When the agent has a piece of land that is not suitable for a building, it will build green to achieve the green indicator goal.

4.2 Demolish functionality

When there is no land available to build a construction, the agent needs to demolish a building to create land for a new construction. It does so by demolishing old buildings, which is another indicator of the agent. This indicator will get a better score when old buildings are demolished.

4.3 Sell land functionality

If the agent has land that is not suitable for any construction, it is not necessary to keep that land. When this is the case, the agent will try to sell that piece of land. This is beneficial for the budget.

4.4 Request functionality

The agent is able to answer requests from other stakeholders. Requests are either accepted or declined. A request is declined when another agent wants to buy a piece of land that is planned for building a construction. Other requests that are beneficial for the agent's indicators are accepted.

5 Development of the HCI module

In this section a brief introduction will be given about the interaction between the user and the agent. When a user wants to play a session in the Tygron engine, but not every stakeholder is able to play their role at that time, the agent can take his place and play the role of the TU Delft stakeholder. The agent responds to the actions that the user makes in a logical way. To achieve this, multiple interaction design techniques were used and the interaction was tested by using a Turingtest. All of these are described in detail in appendix A.

6 Evaluation and failure analysis

In this chapter an evaluation is given of the entire product. This chapter also includes the failure analysis.

6.1 Evaluation of the connector

The connector is pretty much where it needs to be for the agent. Some of the Percepts or custom actions might need some fine tuning to increase performance, but that is not really that important since a human player wouldn't respond immediately either. For example the Buildings percept is so large that updating it takes a good minute at the least, which, while not necessarily a problem, is quite annoying.

6.2 Evaluation of the agent

The agent is split up in separate modules as stated in chapter 2. This really helps with keeping the code readable, because it means that we do not have all our code in the same file. The agent can perform almost all the actions that we wanted to implement at the start of this project. We left out the ability to give money to another stakeholder since we noticed that realistically nobody would just give someone else money.

6.3 Failure Analysis

Failures in the agent are usually the result of a failure in the environment, since the environment is what actually interprets the agent's actions. These failures more often than not throw an error which can be read in the agent's console output. Many possible failures are caught in the tests of the connector, but some of the failures only show up once an agent is run using that version of the connector. These failures most often occur when calling an action with arguments as these can be confusing and are quite prone to errors. Failures in the design of a percept can be quite hard to catch as it can be difficult to debug the agent and check if the percept is correct, since a percept can be quite long. In goal failures can be a bit harder to find since it can be difficult to debug. Apart from that figuring out what caused an error in the code can be quite hard to find since the eclipse plugin doesn't highlight this very well.

7 Outlook

In this section we will talk about the future of the product and what are enhancements for the connector and goal code that could improve the product. Also we will talk about the virtual humans context project and what we think could improve the project for future students.

7.1 Product

Our current agent has no way to find out how a certain indicator can be improved, we had to hardcode the names of the indicators. This could be a big improvement to make the agent a lot more dynamic. Changes in the connector and possibly in the SDK are needed for this, but it would make programming an agent a lot more convenient.

Another improvement would be to change the way land works, our agent has to work with multipolygons currently which makes most computations impossible. Because of this a lot of functionality is moved to the connector. Having to make a function call to the environment for every calculation with land does not work very well for an agent that has to do city planning. If we get a request to buy a piece of land for example, we have to know if there are buildings on this land, if it has a good shape to build buildings and where the land is exactly, to make a good decision about this. But all of this information our agent can not figure out by itself. A grid of the landscape where every building is build within cells would be a solution for this, making it a lot easier to see collisions and locations. This would require a lot of changes to the connector though, but would enable our agent with a lot more possibilities.

To get our agent to behave more like a human there are some enhancements we could make as well. Our current agent has a very specific strategy which will execute in exactly the same order most of the time. This is something that could be improved, as this makes the agent predictable, where a human would act more randomly. The agent has also very fast decision making, where a human would take its time to respond, the agent responds immediatly to most things. When playing as a human this seems a bit odd, and a delay should be introduced.

7.2 Project

We did have fun during this project, but there were also some problems which caused a lot of frustration. The biggest issue was probably that we did not have a good understanding about the functionalities of the connector. We expected it to have most basic functionalities, and that we could later implement some more advanced functionalities if needed. But it turned out that we still had to implement most of the functionality. A good improvement for next year would be to start working on the connector in the first week. It would also be very usefull to have a session in which the connector would be explained. Although working with such a big group has its problems, we do think this should still be done in the next years, it was a big learning experience and we learned a lot about communication and working on a software product with a lot of people. An improvement would be to have a weekly meeting with all the groups and the TAs. We did start having these at the end of the project, but these were a bit chaotic sometimes and not every group made good agreements during this meetings. Another improvement would be to have a stable goal version in which tests and the debugger would work consistently, this caused a lot of delays this year. The last few weeks none of the groups were able to use the debugger succesfully in eclipse and this made fixing bugs a lot harder then it has to be.

References

Li, S., & Zheng, J. (2010). Expert systems with applications. *Elsevier Ltd*.

A Interaction Design

A.1 User

This subsection will describe our user and how he/she uses the system. It starts with a description of the persona. After that a few user scenario's will be described. Finally the tasks will be analysed.

A.1.1 Persona

This subsection describes the user by giving a description of the persona.

Hank - TU Delft

Hank is someone who is in the planning phase of a construction project, for the TU Delft, in and around the campus. He wants to use the Tygron engine for detailed information on possible outcomes of scenarios that can happen during his project, because the Tygron engine can provide him with some insights for this decision before he actually has to spend any money on the actual project. Hank will try to satisfy the needs of the TU Delft as good as possible, which include (in this case) demolishing old education buildings, building newer and higher education buildings and having more greenery on the campus.

He can join a session and play the role of the TU Delft. The project will be a virtual version of the area surrounding the TU Delft. In this virtual version a lot of settings have been configured to match the reality. He has a budget and can do certain actions. He can also see indicators that can help him visualize his current goals and the state of these goals, they also provide some sort of score system. This score system will be shown at the end of the planning session and shows you how much of your own goals you completed.

Lisa – Municipality

Lisa works for the municipality of Delft. She wants to make sure that Hank can complete his project, but he can't break any rules regarding the zone with his project otherwise she has to say no. Since Hank and Lisa need to work together, Lisa can also join the virtual environment of Tygron. She also has some indicators that define her score, like the livability of the area and not allowing buildings of certain height. As municipality she has an important role becuase she needs to give permissions when buildings get build or destroyed for example.

A.2 Usage scenario

The Tygron engine is designed to use in scenarios where all parties involved with the project for instance: the user (project lead), municipalities, housing cooperatives, etc. Each party makes decisions about actions that involve their party, for instance municipalities need to decide if you are allowed to construct a building, if the building is to tall, if you can buy a piece of land for your project, etc. But what if you decided that you wanted to try something in the Tygron engine, but not all other stakeholders are available, you then have to either find someone who can fill in for the other parties or you postpone to a later date.

Our project can help with this by replacing a real world party with a computer controlled party. This computer controlled party, from now on referred to as agent, is designed to follow the guidelines that it is given through the engine, but it can also negotiate with other parties, be they human or computer controlled. The actions these agents take aren't necessarily 100 percent accurate to what the party would actually do, but we try to make it as close as possible.

A.3 Contextual Inquiry

In this subsection, the four principles of the contextual inquiry are described. The four principles are context, partnership, interpretation and focus.

A.3.1 Context

In this project, the Tygron Engine is the engine in which the agents should run. The first week was planned to get used to the engine. Knowing how the engine works, is important for the development of the agent. To achieve this knowledge, we visited Tygron in the first week for questions and explanation about the engine.

A.3.2 Partnership

Because Tygron is the expert of the Tygron Engine, a weekly visit with Tygron is planned. This weekly meetings are planned for question, explanations and discussions. Working with people of Tygron is convenient for these questions, explanations and discussions. We prefer real interaction with the people working at Tygron above interaction with mail.

A.3.3 Interpretation

The avoid situation with wrong interpretations, communication is important. When question occur, the questions are asked to another group or Tygron. When we have another opinion about the answers of the question, we can discuss these face to face, which is mostly more beneficial in terms of time.

A.3.4 Focus

Each visit with Tygron has its own focus. The focus for each visit is based on the problems and questions occurring in the week before the meeting. When the problems are discussed or solved, the focus will be replaced by another task of the sprint planning.

A.4 Evaluation

In this subsection the interaction of a user will be evaluated according to various techniques as described in the Interaction Design course. This will be done by using both a cognitive walkthrough and an emprical method in the form of an expiriment.

A.4.1 Cognitive Walkthrough

In the earlier parts of this project the method cognitive walktrough was used to deduce multiple strategies for the agent. After all the agent needs to be able to make logical choices and be able to respond to the other stakeholders (played by one or more users) in a logical way. This means that there are multiple scenarios in which a user communicates with the agent that should have a clear sequence of actions as a result. For instance, when the user asks to buy land fromt the agent while it is still planning to build on that land or if the price offered is not acceptable will result in the following action sequence:

1. Select a piece of land to buy from the agent
2. Choose a price to offer for the land
3. Place request to buy availible land from the agent
4. Wait for a response
5. Click to confirm the popup saying that the agent declines the offer

This sequence is one of many to which the agent will be able to formulate a logical answer. In the ideal case the agent will give responses that correspond 100% to how the person that the agent is replacing would give. In the next part the actual interaction between a human and the agent will be assessed using a Turingtest.

A.4.2 Turingtest

This part of the evaluation will explain the experiment that we have conducted to test the interaction of our agent with a human player. It contains the method that we used to assess the interaction, the results that we have found and the conclusion that we can draw from our results.

Method

The experiment that was put together was quite simple to do. It consists of a test person behind a computer and one of the people conducting the experiment behind a computer as well. The participant joins a session in the

Tygron environment as the municipality stakeholder while the experimenter joins the session as the TU Delft stakeholder. The participant does not know whether the experimenter had joined himself or joined with the agent. The participant then has to do a few simple tasks such as trying to buy ground from the TU Delft, sell land or respond to any of the requests that the TU Delft makes. The participant is asked to think aloud while making decisions and reacting to the actions of either the person or the agent. At the end of the run the participant tells if they think the TU Delft was played by a human or an agent and how confident they are on a scale of 0 to 100. This process is repeated multiple times.

The test persons that were chosen for this experiment were the parents of the experimenters because our product (the agent) would be used to replace a certain stakeholder and playing together mostly with middle-aged people. On top of that, both of these groups will most likely have zero to no experience with playing together with an agent in the Tygron engine. The expiriment was conducted on 3 different people for 5 separate rounds. The order of choosing between joining as human or agent was the same for every participant: human, agent, human, agent, agent.

Results

These are the results that were found:

	Round	Choice	Actual	Confidence
	1	agent	human	20
	2	human	agent	15
Person 1	3	human	human	40
	4	human	agent	40
	5	agent	agent	70

	Round	Choice	Actual	Confidence
	1	agent	human	10
	2	human	agent	30
Person 2	3	agent	human	80
	4	agent	agent	100
	5	agent	agent	100

	Round	Choice	Actual	Confidence
	1	human	human	18
	2	human	agent	25
Person 3	3	agent	human	25
	4	human	agent	40
	5	agent	agent	50

As stated in the method subsection, the participants were asked to think aloud while making decisions. During the experiments we found that the participants had trouble early on to distinguish between the human and the agent, mostly because they were new to the Tygron engine. They were given assistance and tips on how to use the environment and how to do certain actions. As soon as they started to understand the environment better and recognize the things that were done by the other stakeholder that they began to get more confident about guessing whether they were playing with a human or an agent.

Conclusion

In the end the results of the experiment were satisfactory and gave clear insight of how a person with little experience with the Tygron engine in general handle the actions that the agent executes. Initially the agent and the human were hardly distinguishable, mostly because the sequence of things that the agent would do was unknown for the participant. After a while the participants started to see a pattern in the actions of the agent and were capable of separating the agent and the human quite good.

The results of this experiment were mostly corresponding to the results that were expected. For a person that has little to no experience with using the Tygron engine, the actions of another stakeholder may not be sufficient to tell if the stakeholder is played by a human or an agent. This is why in the first few rounds of the experiment the participants did not get the right answer most of the time and had low confidence in their choice. The agent has a clear strategy that it will almost always execute in the same order. Because of this after a few iterations of the experiment some of the participants started to see a pattern in the actions that were executed. This led to the participants being able to easily distinguish between the human and the agent after some time and had more confidence in doing so. On top of that the agent has a slow start in which it collects data about the area. Because of this delay afters some runs the participants found it easier to recognize.

To improve future experiments equal to this one a few improvements can be made. First of all the agent that is tested could be altered to have some randomness in the choices that it makes, while still achieving the same goals. This prevents the participants from recognizing the agent its pat-

tern of actions, making it harder to tell the difference between it or a human. The same goes for the slow start that the agent has. To prevent this from happening some of the early percepts that the agent gets could be hard-coded so it knows that from the start, as the information in the percept does not change per map. This will result in some problems when playing in a different map, so it is only for this experiment that it would be a good idea. A final improvement that could be made is to instruct the participants better initially on how to the Tygron engine works. This saves time and also helps the participants to have a more clear idea of the things that they are seeing and on how to better distinguish between the agent and the human for more accurate results.

B Glossary

Agent
An agent is an autonomous entity which observes through sensors and acts upon an environment using actuators and directs its activity towards achieving goals.

Engine
An engine is a type of software that generates source code or markup and produces elements that begin another process, allowing real-time maintenance of software requirements.

Connector
A software solution that enables the GOAL agents to connect to the SDK of the Tygron environment.

Percept
Information that an agent receives from the environment that it is in.