Final Report

Danshal (VH2)

1 Introduction

Tygron is a company involved in providing city planning software for a number of clients such as the municipality of New York. Tygron is also the end-user of the product. They were looking for an agent using Artificial Intelligence, a Virtual Human so to speak, to participate in their serious game.

The serious game aims to guide the process of city planning. The game is played in sessions. During a session, there are stakeholders - different parties that participate in the session - with different interests. Examples of stakeholder interests are wanting to have a certain amount of a specific type of building, buildings from a certain category or improving the amounts of green in an area. Together, in a both competitive and cooperative spirit, the agents try to make a plan for the future a certain geographical area.

The agent that is designed can participate in a session regarding the planning of the TU-campus area as the municipality. Other agents have been developed to take the roles of TU Delft, DUWO, Services and Private Housing. Tygron wants an agent that can participate in a session with other humans or other agents. The agent should represent the municipality realistically and provide an interesting sparring partner for the other parties.

It is very important that the agent has its own plan concerning the development of the TU-campus. This vision is expressed as certain indicators. Indicators are scores based on queries. An agent tries to fulfill its indicators as much as possible. The agent must have this vision reflected in its own behaviour in two ways. First, the agent can, as municipality, perform its own actions to increase its indicators. Second, it can shape the area by accepting or denying particular kinds of requests made by the other stakeholders. For example, it can deny the permit necessary for another party to construct a building that influences their indicators in a negative way.

It should be reiterated that the goal of the agent is to be an interesting realistic interpretation of the municipality as a stakeholder. This will be the focus of the Human-Computer Interaction Module (HCI Module) of this report.

Other end-user requirements also exist. The agent must be properly documented to allow the end-user to flexibly add to the agent. Furthermore, it would be great if the agent could make actual concessions and propose alternatives to other stakeholders.

Lastly, the agent must communicate to Tygron their serious game via their SDK. This will happen through the connector. The connector needs further development to support our product in it’s entirety. The development of the connector is in cooperation with all other teams that have developed agents. It requires intensive communication between different parties such as the teams themselves, Tygron and the TAs (teaching assistants).

2 Overview of the product

In the introduction, it became clear what exactly is expected of the agent as a product. This section will deal with linking the mentioned end-user requirements to the product by giving a proper overview of the product in question.

In order to build a realistic agent, the GOAL programming language is used. This language allows building a cognitive agent that interacts with its environment. It is an agent that takes action based on their believes and goals, as stated by Hindriks, K.V. (2016).

After initially starting up, the program runs in cycles. In each cycle it percepts information about the environment, updates its beliefs and goals by considering said information and eventually decides on an action to perform. This lends itself very well to the Tygron game environment. The task of processing percepts into certain beliefs is rather straightforward. The heart of the product is its decision making strategy. The decision making strategy pertains to the vision our agent needs to have.

The agent is capable of performing a number of actions. It can construct AZC in suitable locations to facilitate the reception of refugees. It can build parks in areas where the amount of green in not to its liking. It can accept or reject requests from other stakeholders based on the stakeholder, their request history and the influence of the request on its own indicators.

Documentation of the agent is delivered separately in the form of the Architecture Design document. This document is focused on allowing the end-user quick understanding of the agent via different overviews.

The connector, which uses the SDK to have the agent connect to a game, has been upgraded in order to accommodate the agent. The added functionality mainly consists of new actions and percepts. They allow for more targeted information gathering and more targeted performing of actions.

3 Software engineering perspective: a reflection on product and process

There are many aspects concerning software engineering that can and will be discussed in this section. However, the most important and overarching one is the development process. Hence, this will be discussed first.

Regrettably, more negative aspects stand out when it comes to the development process. Most of these, if not all, have to do with communication between the different parties involved in the project.

First is the relation between the different groups. It is fairly challenging to communicate with 25 people. It is difficult to establish a real red thread and get everyone to agree on how to shape the project. All groups needed to use and develop the connector to have their bot working, yet there was no unified approach established from the get to. For example, only after several weeks it was decided the groups would use github issues to communicate further wishes for the connector. This was during a meeting with someone who would be going to guide the groups each week, yet he was seen maybe once more and that was it. The conclusion can be drawn that the startup of the project was very slow. In future projects, ensure all groups together use one approach, one system, as quickly as possible.

Second is the relation between the project and Tygron. The server from Tygron has been shut down multiple times. Once there was an entire weekend wherein the servers were down and the issue was resolved on Monday at noon. Apart from connection errors, the groups were also under the impression that the SDK would be frozen. The plan seemed to be that the groups could work on the connector without also having to accommodate a changing SDK. This did not work out like that either; the SDK was updated. In order to ensure servers are down as little as possible in the future, Tygron should always be informed of such events as quickly as possible. For example, contacting them immediately by mail would be an option.

Third, the relationship between the Teaching Assistants (TAs) and Tygron employees was also less established as we would have liked. In the beginning it seemed as though Tygron was not even aware of the existence of the TAs. Especially over the course of one particular week, quite a lot happened. The students decided with Tygron to restructure the project without informing the TAs. However, the TAs could not grade us correctly by using the new structure and it had to be changed a second time. The next meeting at Tygron included someone from Tygron urging us to change back to the previously discussed structure, before the students asked him to consult with the TAs.

The were other issues besides the examples provided, but these are the ones that stood out the most. Now, armed with knowledge on the development process, it is time to look at the SCRUM documentation and Product Planning.

The were no major problems with the SCRUM documentation. The sprint plan was a significant part of our structural guidance during this project. A lot of time was invested in

constructing each sprint plan. However, it can be seen that there are always changes in planning due to unexpected circumstances. It is important to know this so that there can be accommodated for it in the future.

The Product Planning reveals interesting details concerning the difference in expectation and the actual development process and well as the complexity of our product. For one, the Product Planning mentions close cooperation to establish conflict simulation already within the first couple of weeks of the project. The different agents have only been run in the same session during week nine.

Furthermore, the Product Planning also mentions more complex negotiation and even the agent proposing counter offers during negotiation. As running our bots in the same session happened as late as week nine, these goals have never been achieved.

The comments provided so far are mainly concerning the work process. Now it is time it reflect on the more tangible aspects regarding the product.

The first topic is complexity. Despite large amounts of effort, the product is not extremely complex. There are numerous reasons for this. Most of these reasons can be summed up under the header of problems that have occurred during the development process. The essence of these problems has been described above. All parties could, in hindsight, operated better to have the project results in a more expansive product as was envisioned.

The relatively low complexity is one of the reasons for the clear structure of the product. Different elements in which the agent is interested are present in separate modules. Also, the GOAL programming language has very clear rules pertaining to the structure of the code you write. There is nothing that the team would change about the structure.

Furthermore, we can discuss readability and documentation. These are not wildly interesting when compared to what was previously discussed. There has been an effort to provide the code with good comments throughout the entirety of the project as means of documentation. Higher-level documentation such as overviews on the agent and connector can be found in the Architecture Design. No significant problems have been encountered regarding readability and documentation. The only point of improvement could be updating the Architecture Design more frequently.

4. Description of functionality

Our agent has three main features it uses to reach the goals set in a scenario.
The core of the decision making strategy can be found in three modules: green.mod2g, azc.mod2g and negotiation.mod2g. It is here that the agent can make decisions in order to fulfil its indicators. Each of the three modules has a clearly identifiable task.
The green module deals with improving the green indicator. The AZC module builds AZC units and the negotiation module deals with either accepting or rejecting requests made by the other stakeholders.

The green module will build parks to satisfy the need for green in specific zones if our agent has enough money and available space.

The azc module reasons about building azc units. It will use a custom action to query the environment for buildable land of the right size, then it will build an azc, if this fails it will set a timeout and try again until our indicator is fulfilled.

The negotiation module by far the most complex module. It will reason about the changes that other stakeholders want to make by looking at the indicators. If the indicators of our agents increase we will approve the permit or if the action doesn't require one we will increase the reputation of the stakeholder. If our indicators decrease we will decrease the reputation of the stakeholder, if this reputation gets below a threshold we will no longer approve any of this stakeholders permits. Buying land can also influence stakeholders reputation, buying land for a decent price will increase reputation while buying for cheap will decrease reputation of be denied.

5. Interaction design section

In order to accommodate for interaction design a usability study has been performed. The study will be described below. The details of the project have already been given in the introduction and are not repeated below.

Method

The goal of this study is to find out how our agent compares to a real person. It will focus on whether the agent is a realistic representation of a human. Does it play the serious game in the same manner as its human counterpart does? Is the agent capable of negotiating properly with the other stakeholders? Does the agent have a clear vision on what direction the planning session has to go in?

The procedure will now be explained. First, a test person will be asked to participate in this experiment. It is quite important that the test person is unaffiliated with the project because it allows for a more objective response. Second, what we expect of them will be thoroughly explained. It is very important the test person understands what he/she has to look out for during the experiment.

Afterwards, the experiment itself will be performed. The test person will play two sessions: one against the agent and another one against a human playing as the municipality. The test person itself does not know in what session he/she is playing against who. They will be encouraged to give their opinion all throughout the sessions, and their thoughts will be noted.

The test person will be asked to respond to the following questions:

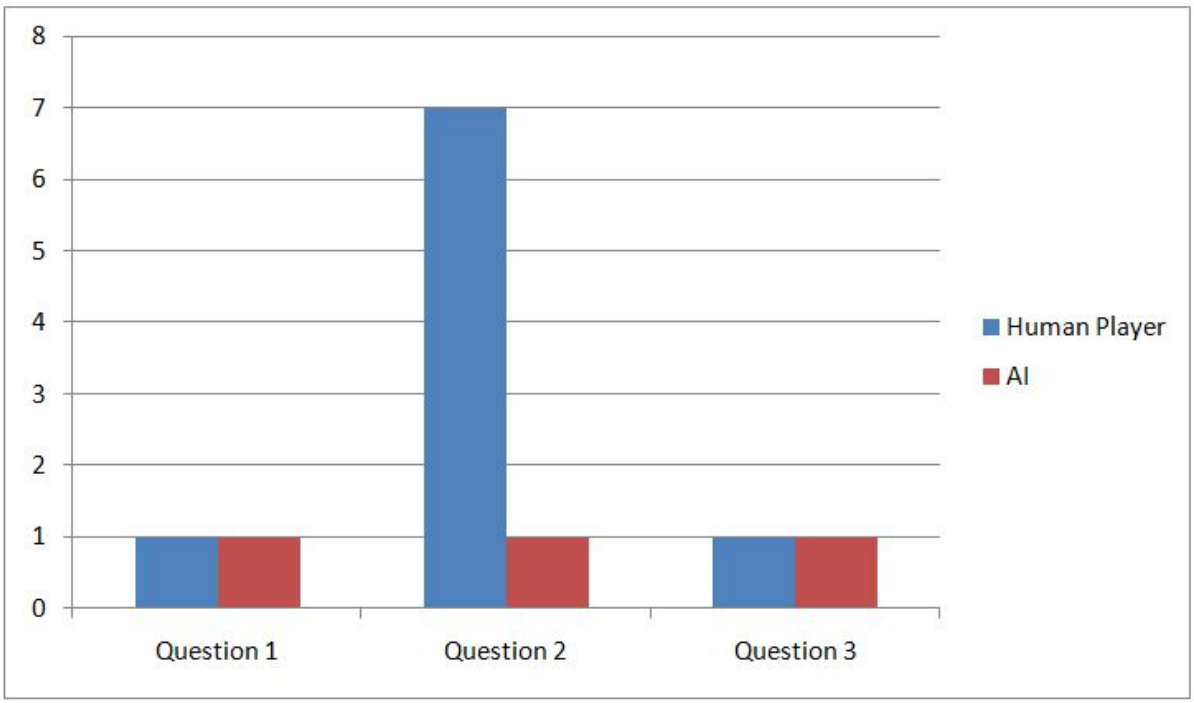
Questions 1 through 3 require both a numerical score (on a scale of 1 to 10 with 10 as the highest) and a brief explanation.
1. How understandable are the goals of the municipality?
2. How noticeable is the presence of the municipality?
3. How reasonable and pleasant is it to negotiate with the municipality?

Questions 4 through 6 only require a comment from the test person.
4. Can you tell who is the real human and who is our AI (agent)?
5. What criticism of our AI (agent) do you have?
6. Do you have any closing remarks regarding this experiment?

Results

The graph on the following page is a representation of the numerical scores gives to questions 1 through 3 in both sessions. The entire answers can be found in Appendix A.

It is evident that neither the team member playing as the municipality nor the agent was capable of communicating their goals clearly. Both were also not very pleasant to negotiate with. The human performs better as the agent when it comes to having a presence during the session.



The explanations given during questions 1 through 3 provide insight into these numerical values. First, neither the human or the agent provided an explanation to justify their actions. Things were being build and requests were either being accepted or denied. However, the logic behind the decision making process and the vision of the municipality was entirely unclear. The test person mentioned that making use of the chat feature or any kind of communication with the municipality would have done a lot to resolve this issue. Second, the presence of the municipality is higher when the role is performed by a human player. The agent builds too fast for a human user to see.

Question 4 through 6 have made clear that the test person could not distinguish between the human and the agent. The building times would have been a giveaway, but the focus of the test person was request processing. Both the human and the agent could handle requests at relatively similar speeds. Another remark was that the agent just accepts or rejects request, it does not negotiate at all.

Conclusion and Discussion

The conclusion can be drawn that the human player has played in the same manner as the agent. That is, without communicating properly. The difference is that the human could have easily communicated while the agent does not have this feature. The lack of communication obscures any vision that the agent might have had. This also makes it so the agent is not able to negotiate properly. What also contributes to the lack of negotiation is the absence of counteroffers.

The test person is unaffiliated with the project and does not possess any special knowledge of the topic of city planning. This contributes to how untainted the result are. The

recommendations that can be made are clear: have the agent communicate more and have it propose counteroffers to the other player.

6. Product Evaluation

The goal of the project is to create a bot that could play the role of the municipality. The bot needs to show an interesting simulation by playing the game with the other bots of the other groups. In order to create this interesting simulation each of the bots have gotten indicators to show how well they are doing. In the case of the municipality, the bot needs to build an AZC. Furthermore it needs to make sure the green in each of the zones is above a certain target value. Next to that the bot needs to make sure the livability score, which is hardcoded for each building, is high enough. Besides achieving all of these goals it also needs to interact with the other bots in the form of requests towards the municipality. In this chapter we will evaluate the functionalities of the agent as well as the bot in its entirety.

The first functionality that we will evaluate is the functionality to handle requests the municipality gets from the other bots. It is good that it can handle all the incoming requests and answer all of them. It even distinguishes different sort of requests and handles them differently.

However it only handles the buy land and permit requests with actual complex behaviour. The remaining requests are always accepted without exception. The municipality would never outright accept a request without thinking about it.

For the permit requests the bot checks if the building would be good for the indicators of the municipality and accepts if it is. The actual municipality would also check if the request has a good influence for them, but they would also check if the concerning stakeholders agree on the permit request. With concerning stakeholders are not only the stakeholders that take part in the request intended, but also the stakeholders that will be affected by this request. So it is not as complex as the real behaviour of the municipality.

For the buy land requests the bot has hardcoded minimal prices per squared meter for each stakeholder. If the price of the request exceeds or is equal to this hardcoded price for the stakeholder, then the bot accepts the request. The municipality would definitely not accept buy land requests like this, because the bot will also sell all the buildings that are on that part of the land that is being sold. Also if a stakeholder would be extremely rich, it could buy up all the land the municipality owns and that is not what should happen.

The next functionality we will evaluate is the favor system, which makes the behaviour of the requests part of the municipality bot complexer. The bot keeps track of all the actions the other stakeholders have performed and checks if those actions have influenced the municipality in a good way. If that is the case, then that stakeholder gains favor. With this favor the stakeholder could get previously rejected permits to be accepted. This is a good way to try to simulate that the real municipality will maintain relationships with stakeholders. The municipality could in those cases also accept requests that are bad for the municipality, but requests something in return from that stakeholder.

The final functionality we will evaluate is the building of AZC’s and parks. The bot just finds spots to build the AZC or park and builds it there. This is definitely not how the real

municipality would do it. They would find an optimal spot to place the AZC or park and check that all other stakeholders do not have major problems with that placement. Otherwise the municipality will try to negotiate or give subsidy to stakeholders to place AZC or parks for them.

Finally we look at the entire bot and what it should have been able to achieve. The bot should be able to create an interesting simulation with the other bots. There will probably not be very much interesting behaviour from all the bots. But the simulation will not end up in a deadlock, because of the favor system of the municipality bot. The bot will just build parks and AZC's and accept or deny requests. So the municipality bot will try to achieve his indicators.

However it is not a replacement for an actual human playing the municipality. There is no real negotiation between the municipality bot and the other bots. The requests are only accepted or denied without making an agreement on how a denied request can still be accepted. Furthermore when building an AZC , the bot does not take other factors into account such as the inhabitants of that area. The same goes for the green in each zone. But the municipality would also plant a single tree every few meters in order to keep the green up. Also the municipality in the real world could give subsidy in order for other bots to create green roofs or build something for the municipality. All of these things the municipality bot lacks.

7. Outlook on the future

Our product is functional but given more time could be developed a lot further. The major improvements we can make lay in three categories: AI, performance and communication.

The AI can be improved in many ways: the selection of land is still far from optimal as we don't get the size of lands back yet, the negotiation module can be improved by better calculating the gains and losses of deals and keeping track of its relations with other stakeholders in a more accurate way.

The performance of the agent is something which can be improved by a lot as currently the connector recalculates a lot of information when the same custom action is called multiple times. Caching can also improve the performance by keeping calculated data in memory.

Lastly we could vastly improve upon the communication with other agent and players. Right now we only work with requests and have no way of telling other stakeholders why we disapproved their requests, also a module or connector feature to handle real conversation could be added.

Bibliography

Hindriks, K.V. (2016). *Programming Cognitive Agents in Goal*. Retrieved from https://bintray.com/artifact/download/goalhub/GOAL/GOALProgrammingGuide.pdf

Appendix A

Session 1 question 1 (The Human)
1. I can only see what is being build. I can try to build things but there is no information on what is acceptable for the municipality or not. The logic behind their decision making and their vision was not clear to me.

Session 2 question 1 (agent)
1. The same applies as during session one. One of my requests was not responded to. I do not know whether I waited long enough for the AI (agent) to respond. This is also a possibility. The building to too fast, I cannot even see what is being build.

Session 1 question 2 (The Human)
7. I saw the municipality doing things. I saw them buildings things and I saw them either approving or rejecting my applications for building permits.

Session 2 question 2 (agent)
1. It went too fast to see anything being build.

Session 1 question 3 (The Human)
1. I understand there is a text box. If would have been a lot better if that was used to explain why or why not certain decisions would have been accepted.

Session 2 question 3 (agent)
1. The remarks from session 1 also apply here. As it were, I only understood yes or no. It can't even be called negotiation. There is no communication whatever, same as last round.

Question 4
There a fast difference in building times. However, the faster builder seems to respond to requests slower. I cannot tell difference between the human and the AI (agent).

Question 4 follow up question: So the difference in building speed... ?
That was not my focus, I was focused on buying land and request processing.
If you focus on the buildings speeds, no human could possibly even get close to the AI (agent). The buildings were just there.

Question 5
Both the human player and the AI (agent) did not communicate at all. This is very important, because otherwise you don't know whether the request is even being processed. When rejected, you don't know why the request was rejected.

Question 6
No.