# Final Report

*Danshal (VH2)*

## 1 Introduction

Tygron is the end-user of the product. They were looking for an Artificial Intelligence, a Virtual Human so to speak, to participate in their serious game. Their 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. Together, in a both competitive and cooperative spirit, they try to make a plan for the future a certain geographical area.

The AI that is designed can participate in a session regarding the planning of the TU-campus area as the municipality. Tygron wants an Artificial Intelligence that can participate in a session with other humans or other AI. The AI should represent the municipality realistically and provide and as such prove an interesting sparring partner for the other parties.

It is very important that the AI has its own plan concerning the development of the TU-campus. This vision is expressed as certain indicators which each have a score to measure how well we are doing. The AI must have this vision reflected in its own behaviour in two ways. First, the AI 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.

It should be reiterated that the goal of the AI is to be an interesting a 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 AI must be properly documented to allow the end-user to flexibly add to the AI. Furthermore, it would be great if the agent could make actual concessions and propose alternatives to other stakeholders.

Lastly, the AI must communicate to their serious game via their SDK. This will happen through the connector. The connector needs further development to support our product in its entirety. The development of the connector is in cooperation with all other Context-vh groups and requires intensive communication between a lot of parties.

## 2 Overview of the product

In the introduction, it became clear what exactly is expected of the AI 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 AI, 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 AI needs to have. 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 AI 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 by placing parks at locations where the amount of green is still lacking. The AZC module builds AZC units in suitable locations in order to fulfil the corresponding AZC indicator. The negotiation module is by far the most complex module and deals with either accepting or rejecting requests made by the other stakeholders.

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

The connector, which uses the SDK to have the AI connect to a game, has indeed been upgraded in order to accommodate our product.

**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.

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 without also having to accommodate a changing SDK. This did not work out like that either; the SDK was updated.

Third, the relationship between the Student Assistants (SAs) 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 SAs. Especially over the course of one particular week, quite a lot happened. The students decided with Tygron to restructure the project without informing the SAs. However, the SAs 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 the someone from Tygron urging us to change back to the previously discussed structure, before the students asked him to consult with the SAs.

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.

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 AI have only been run in the same session during week nine.

Furthermore, the Product Planning also mentions more complex negotiation and even the AI 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 & structure. 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. However, the relatively low complexity is one of the reasons for the clear structure of the product. Different elements in which the AI 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.

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 AI and connector can be found in the Architecture Design.

**4. Description of functionality**

Our bot consists of a main module, a knowledge file, an event module, an init module and a set of logic modules. We will look at these modules in more detail and give a clear description of how they work.

The main module “tygron.mod2g” is the module that performs all actions within the environment. This happens based on the beliefs that logic modules adopted. Those goals should than be achieved and deleted.

The knowledge file “dummy.pl” contains a list of all supported percepts, when those percepts or resulting beliefs are used they have to have the number of parameters given in this list. This prevents misuse that can cause bugs. The knowledge file also reasoning rules to obtain some of the more useful information from our belief base, like how much money we have or the value of specific indicators. Lastly the knowledge file contains possible goals and how these can be achieved.

The init module “tygronInit.mod2g” is the first module that runs, it makes sure that all necessary beliefs are in the belief base so that even when some things are not percepted in the first cycle the bot can run and doesn’t crash.

The event module “tygronEvents.mod2g” is responsible for percepting the environment, it has the rules to percept and add entities to the belief base. It percepts buildings, functions, indicators, land, logs, stakeholders, requests and zones. When done with all percepts the event module calls our 3 logic modules.

The azc module “azc.mod2g” is a logic module that reasons about building azc units. First it checks how much budget we have. If we have enough it proceeds with selecting a zone where we own land and queries the environment with a custom action. This action finds all buildable land in this zone and returns up to 5 plots of land that have a size that is not bigger than necessary but also not to small. Next the module will adopt a goal to build on a returned plot of land. Lastly the module sets a timeout to make sure the indicators are updated the next time it runs.

The green module “green.mod2g” is a logic module that  tries to satisfy our target for a greener city. First it checks how much budget we have. If we have enough it proceeds with selecting a zone where the target is not yet reached and we own land. Next is uses a custom action to receive a small plot of land. Then the module will adopt a goal to build a park on that plot of land.

The negotiation module “negotiation.mod2g” is a logic module that makes our bot work with other bots. It handles building permits and the buying and selling of land. The bot will accept a request if the results of the action will result in an increase of its indicators or if the requesting stakeholder has a good reputation. Stakeholders can increase their reputation by building structures that increase the bot’s indicators or by buying land at a decent price.

The functionality of these modules is tested by the test module “test.test2g”, this module uses assets to test if functions perform a certain way when given a certain input.

**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 AI compares to a real person. It will focus on whether the AI is a realistic representation of a human. Does it play the serious game in the same manner as its human counterpart does? Is the AI capable of negotiating properly with the other stakeholders? Does the AI 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 AI 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?

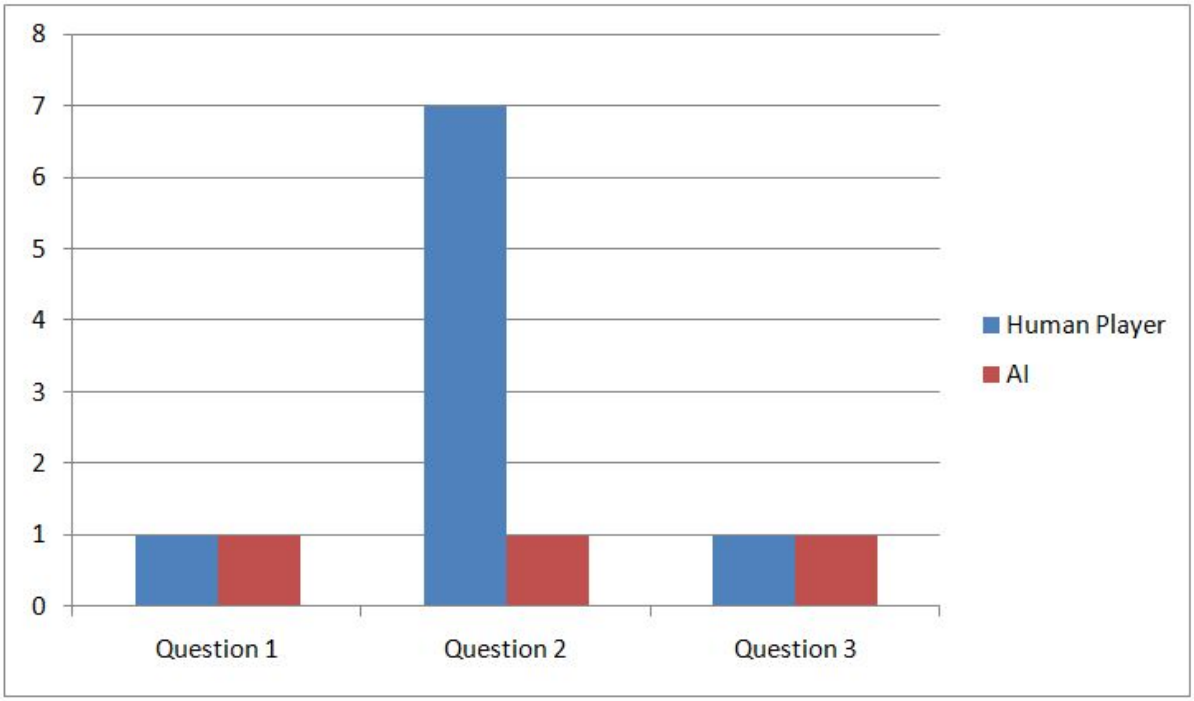
Questions 4 through 6 only require a comment from the test person.

4. Can you tell who is the real human and who is our AI?
5. What criticism of our AI 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 AI was capable of communicating their goals clearly. Both were also not very pleasant to negotiate with. The human performs better as the AI 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 AI 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 AI 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 AI. The building times would have been a giveaway, but the focus of the test person was request processing. Both the human and the AI could handle requests at relatively similar speeds. Another remark was that the AI 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 AI. That is, without communicating properly. The difference is that the human could have easily communicated while the AI does not have this feature. The lack of communication obscures any vision that the AI might have had. This also makes it so the AI 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 without any special knowledge of the topic of city planning. This contributes to how untainted the result are. The recommendations for improve are clear: have the AI 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 goal. 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.

The functional modules in the bot were described in chapter 4. There it was described that the bot consisted of multiple modules. Let us look at the most important modules regarding the aforementioned goals.

The negotiation module handles all the requests from the other bots towards the municipality bot. The requests are handled differently for each kind of request that requires interaction from the municipality. The general manner in how the requests are handled is by looking if the request has a good influence or no influence on the municipality.This can create an interesting simulation given that the municipality will not deny too many requests of the other bots.

However there is still the problem that the bots can come into a deadlock. This means that another bot keeps trying to build a building, but keeps getting denied permission from the municipality bot. An effort to minimize this is to have a favor system, where another bot can still do bad actions at the cost of a favor. This still can get to the point that there are no favors and we get back in the aforementioned problem.

The AZC module handles creating the AZC units. This module finds a good spot to build an AZC and attempts to build it at that spot. After it has built an AZC, the bot cannot build an AZC for ten cycles. This achieves the goal of creating an AZC. The only problem that could happen is that it keeps failing to build an AZC over and over again. This could happen due to other bots doing actions involving the chosen spot by this module.

The green module handles creating parks. Like the AZC module, this module finds a spot and also attempts to build a park at that spot. It does this only if the green indicator in that specific zone the spot lies in, is below a certain threshold. The same problem of the AZC applies to this module of not being able to build a park.

Finally we look at the entire bot and what it should have been able to achieve. The bot is able to handle requests and achieve his indicators. The bot should be able to create an interesting simulation. 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 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 (AI)
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 too 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 (AI)
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 (AI)
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.

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. The buildings were just there.

Question 5
Both the human player and the AI 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.