# Tygron Virtual Humans

## TI2806 Context Project Final Report

## Team


Members

Ronald van Driel
# 4289870

Matthijs Halvemaan
# 4353803

Marco Houtman
# 4307232

Joshua Slik
# 4393007

Lisette Veldkamp
# 4394712

This page is intentionally left blank.

# Contents

This page is intentionally left blank.

# Introduction

This context project was essentially tantamount to a request from the company 'Tygron'. The main request was to substitute human players with artificial intelligence within the 'Tygron Engine'. 'Tygron' is the company that provided the 'Tygron Engine' which is a serious game for urban planning. This means that the game can be used to examine possible solutions for problems in development projects for cities and villages. In this game there are certain stakeholders that can be played by humans. For our project we have developed a scenario, a specific environment, with the following five stakeholders: 'Municipality', 'Private Housing Corporation', 'Facilities', 'DUWO' and 'TUDelft'. These stakeholders all have different interests and goals that they will try to accomplish.

Initially, a total of five teams had to collaboratively come up with an interesting scenario within the 'Tygron Engine'. This scenario had to have enough possibilities for interaction and decision logic for each of the five stakeholders.

Each of these five teams was requested to develop a virtual human to substitute the human player for a single stakeholder. This means that all five teams were essentially responsible for a single virtual human which, in conjunction, were expected to be able to play in our a game session without any human interference.

Our team was responsible for the virtual human to play the 'Private Housing Corporation' stakeholder. Each virtual human is required to be capable of making human-like decisions in the game and accurately conform to their stakeholders' properties.

In this report we will first give an overview of what we have developed. After this we will reflect on the product and process from a software engineering perspective. Then we will give an exact description of all the functionalities that our agent has, followed by an explanation of our interaction design. Eventually we will evaluate our product and describe our outlook on the project.

This page is intentionally left blank.

# 1 Product Overview

In this chapter we will provide an overview of our developed product and implemented software.

The product we developed consists of the virtual human, written in GOAL, an agent programming language for programming rational agents, and the connector that lets the agent connect to the Tygron Environment, written in Java. The connector makes sure that we can provide our agent with continually updated information from the Tygron Engine, which is connected to a session environment. This is needed for the agent to act and react on everything that happens in the environment.

The connector uses the Tygron SDK to return information from the Tygron Engine to the Tygron Environment. There are different kinds of objects and data in the environment which are collected in itemsets called MapLinks, these MapLinks can be added to a listener, which notifies the Tygron Environment when MapLinks have been updated. When the Tygron Engine has been modified it will return an updated itemset to the Tygron Environment. Another way the SDK has been utilized is by creating actions in the environment which call on usable actions in the SDK to manipulate the Tygron Engine. The connector is the interface through which the agent is provided useful information from the environment. The agent will use the relevant information to decide the best course of actions in its current state.

All stakeholders have a set of indicators which are goals for the stakeholder to achieve at the end of a session. Every agent is supposed to try and achieve these indicators. When a session is started in the Tygron Engine an agent can be added to this session. The moment the agent is added to the session it will start collecting all useful data it received from the environment. When all the necessary data has been inserted into the agent it will look at the data and derive its own decisions. If a decision leads to an action in the session then the session's data will change. These changes will be sent to the Tygron Environment which returns the information to the agent. When the agent receives this new data it will update its knowledge and develop new decisions according to its indicators. The agent will always run until the session is finished, because there is no guarantee that achieved indicators will not be changed later on.

This page is intentionally left blank.

# 2 Reflection from a software engineering perspective

The "Virtual Humans"-project was mostly divided into two separate products. First of all there was the agent itself, and secondly there was the connector between the environment and the agents. These two parts will be discussed separately, because the agent was individual effort for each group and the connector was a collaboration between all groups. General software engineering principles will be discussed first, followed by the software engineering aspects that relate to the two different products specifically.

## 2.1 General

We were required to use a single communication platform, Slack, in which all teams and their members as well as the teaching assistants were available.

However, initially we had some start up problems with the project, such as: lack of information, scattered communication and absence of team members. Throughout the project these issues were attempted to be resolved by forcing team members to use Slack as the main communication platform, by making clear appointments and by introducing penalties for absence. Despite efforts being made to resolve these problems we kept struggling with these issues, because several team members had low motivation or had an occupied scheme. Drops in motivation were most likely introduced by multiple issues that kept lingering due to poor communication. Many of these issues were keeping team members behind while they, if directly discussed, could have been resolved before stagnating the working progress. For upcoming projects everyone should ensure themselves to have enough time reserved. Also a single communication channel should already be frequently used at the start of any project, so that it will not delay communication later on.

When we were able to start working on the agent full-time, we had an problem with people that were writing dependencies for other team members without communicating their progress clearly, so the dependent people did not know when they could start working on their part. There were also some issues with improper and dirty branching, but we always managed to resolve these before the next sprint.

We were required to use the SCRUM-methodology. This states that our product should be in a working state at the end of every sprint, with a sprint being defined as one week. According to SCRUM, this is realized with the use of backlog at the beginning of each sprint, daily SCRUM meetings during the sprint, and a reflection at the end of the sprint.

As for our scrum management, the backlog was systematically made at the beginning of each sprint and was overall effectively used to divide the tasks for the upcoming sprint. At the beginning of the project, the prioritization of the tasks was not completely up to par, but this was adjusted accordingly in later sprints. We elected to have the daily SCRUM meetings through voice chat, every day. However the attendance at these meetings was abysmal, often only two or three members of the group were present during these meetings. Reflections were also made systematically at the end of each week, but the overall quality of the reflections varied quite a lot during the project. To our displeasure, we were unable to deliver a working agent at the end of some sprints. At times this was caused by factors outside of our control, but it was also due to bad time management, which caused the need for tasks to be carried over to future sprints.

Additionally a pull-based development model had to be used throughout the development-process. New functionality, bug fixes, documentation etc. were all added through pull requests,

and were supposed to only be merged after at least a majority of the participants have reviewed and approved the request.

Within our group, reviews of pull requests were not always extensive enough, which led to lots of inconsistencies in style, which caused that later on re-factoring had to be done.

Most of the afore mentioned issues would have had significantly less impact if team members showed more dedication at the start of a sprint, because then there would have been more time to correct mistakes and solve issues that arose throughout the project. This would be the best improvement for following projects.

## 2.2 Private Housing Company Agent

We were obliged to utilize continuous integration with proper branch management. For that purpose we also had to make use of static analysis tools such as: 'CheckStyle', 'PMD' and 'FindBugs'. Additionally we had to use test driven development. We had to make use of 'Travis' to check each commit on static analysis and tests.

The agent was developed with the GOAL programming language. The use of GOAL caused a lot of challenges regarding static analysis tools and continuous integration. None of the static analysis tools required for the project worked in conjunction with GOAL. This meant that the 'Travis' configuration used for the agents is very limited.

The test driven development model was completely unused during the project. Lots of functionality was added to the agent without any automatic tests. This was partly caused by bad time management but mostly because of external issues. Tests in GOAL initially did not function properly, which made automatic testing available only past halfway in the project. As soon as the issue was solved, we started adding automatic tests for already existing modules. For multiple modules however, there was either interaction required with other stakeholders or there were multiple connector dependencies, which meant that there was no way to add proper automatic tests for these modules. We managed to get around the lack of tests fairly good as we always had all our code manually tested thoroughly and all encountered issues were usually solved before merging a pull request. Some pull requests however were still improperly tested before the merge, causing team members to spend much time on testing and debugging the code of another member.

At first, we managed to find a way that Travis would verify that all test cases written would run successfully. However we later discovered that GOAL-tests do not run as intended when ran by Travis. Because of this reason, only some specific tests are ran through Travis.

At the end of the project we provided a product which did satisfy our client. We were not able to implement advanced decision logic or a well defined strategy. In sprint three we encountered the largest problem in the project. The connector was embarrassingly less developed than was assumed. The reaction to this was that the following weeks up to week seven was spent mostly to only on working on the connector to get it as far as we required it to be. Despite all efforts being made on the connector, we still ended up with less functionality than we needed to implement our business plan.

Another issue arose with GOAL debugging, which stopped working and had not been fixed since, so we came up with a workaround to still be able to get some debug information while testing the bot. This caused some performance loss, because testing would take considerably longer.

In sprint six one of our members experienced a large problem with GOAL, because of which we practically lost one member for a week worth of work.

## 2.3 Tygron Connector

Since the connector was completely written in Java, static analysis tools and continuous integration did not cause issues, unlike with the agents in GOAL. This meant that all required static analysis tools were used extensively, creating a consistent style and level of quality in the connector.

Test driven development was not used in the development of the connector, this was a decision made with all groups, because for the provided connector it was not feasible to obtain a high enough level of understanding to start test driven development given the time-frame. But unlike with the agent, every feature was thoroughly tested automatically before being added to the final product.

# 3 Description of Developed Functionalities

This chapter describes developed functionalities for the connector and the GOAL agent.

The agent that we have been working on fulfils the role of a Private Housing Corporation. What this corporation wants is to keep up and improve its reputation and obviously, to make profit. In order to achieve these two things we have designed our agent to take all necessary indicators into account. We will now provide an explanation per developed functionality based on the strategy that the agent has.

## 3.1 Connector

The connector was a joint effort by all groups which were working on this context project. Our group has done several contributions to the connector on which we will expand: The custom indicators, used to track progress of the stakeholder, the Upgrade Types percept, which makes it possible for a bot to use Upgrades, and the Extension of Buildings percept, which gave more functionality to our agent.

**Custom Indicators** The custom indicators percept was developed to make sure that every group had the possibility to create and update indicators which they deemed fit to reach their goal in the played session. It enabled the user to choose which information the indicator should send to the agent and it has been expanded throughout the process to accommodate every groups needs.

**Upgrade Types** The Upgrade Types percept is used by the agents to look which buildings they own can be upgraded into a building which would improve their indicators. The Upgrade Types percept was built to return an upgrade identifier with only one set of upgrades which could be called on by the agent, in the following weeks we noticed that some of the Upgrade Types had multiple sets and the percept was updated by project group RAMpestampers.

**Extension of Buildings** The Buildings percept has been extended with extra information regarding the zones in which a building had been built. This makes it possible for the agent to make a better decision on what to do with its owned buildings as he is aware in which zone they are located.

## 3.2 GOAL Agent

The agent was solely developed by our group and has the following functionalities which will be expanded on: Building and demolishing buildings, Upgrading Buildings, Buying and Selling land and communication with other stakeholders.

**Building and Demolishing buildings** The agent will build and demolish building which are available for him to build and demolish. It will also be focussed on building luxury housing and demolishing the amount of social housing. This is because the Private Housing Corporation is looking to improve the overall quality of housing in the zones they have buildings. Also buildings are demolished if they are not useful to the housing corporation and could not be sold to other stakeholders.

**Upgrading buildings** The agent will be looking to renovate as much buildings as possible to improve the overall quality of the housing in its zones if they were not chosen to be demolished for more modern housing. It is also used to keep the Municipality on its side by improving one of the Municipality's indicators, this might mean the Municipality will return a favor later into the session.

**Selling expensive land** The Housing Corporation owns a piece of land which is very expensive to work with because it contains a building it can't use and it is very expensive to demolish the building for a few extra houses. This is why the agent tries to sell this piece of land to other stakeholders, the agent offers this land to the other stakeholders on 3 different prices, the first price is 50 euros above the average price per $m^2$, the second price on the average price and the third price is 50 euros under the average price.

**Communicating with other stakeholders** The agent has communication with other stakeholders to a certain extent, it will always react on permits which he should react on. If another stakeholder tries to sell land to the agent and the agent has been trying to buy this land but the offer was never accepted the agent will look at the price the other stakeholder offers and decide if it wants to buy the land. This works in the same extent for land our agent is selling, if another stakeholder tries to buy land on a location the agent is selling the land it will take the price in consideration.

# 4 Interaction Design

This chapter describes the Human-Computer interaction, abbreviated as HCI, module that was realized for the user interaction with the developed solution. This section reports the method used to evaluate our Agent, the results gathered from conducting the method and a discussion on the way the method was conducted and the method's outcome.

## Method

**Goal** The goal of our contextproject is to make an agent for the Tygron environment which acts as humanlike as possible, this will be tested by asking several test subjects to play a game with the bot in the environment and react on the way the bot behaves in the environment.

**Test subject** The test subjects will be several different people who will get a short introduction on using the tygron environment. The subjects will not have an extended background on the topic regarding urban planning.

**Procedure** The test will be conducted in the following way:

1. The test subject is entering a room with a computer running Tygron.
2. A brief explanation about Tygron is given regarding the actions a user can use and the way the user can improve his indicators.
3. The user gets 10 minutes to use the environment and ask questions regarding its workings.
4. A new environment is started and agent is added to the environment.
5. The user gets about 20 minutes to play with the agent in the environment.
6. A short Q&A is conducted with the Test Subject regarding the way the other stakeholder played.

**Metrics** There is only one metric used: During our Q&A we ask the test subject a few questions about the bot and the last question we ask him is if he believes he was playing against a human or a bot.

## Results

Three test subject have conducted the test, they figured out that they were playing against a bot. This was because they all felt like the bot was acting *too fast*. All of the test subjects also had something to say about the fact that the bot was *not reacting* to offers to buy land from and sell land to them.

## Conclusion

It was very easy for our test subject to find out that the agent indeed was a bot, this was due to his fast actions and that he always declined buying land from the players.

**Suggestions for improvement** The most important suggestion for improvement was implementing a delay for the bot which makes him slower, the bot was *very fast* and *ran a lot of actions* before the subject even thought of what he had to do, the reaction time on buying the land was almost instantly when he requested to buy it which *felt unhuman*, this was also true for the reaction on the bot whom was trying to buy land from our subject. Another suggestion was working on his decision making as the test subject perceived the actions as *random and not very thoughtful*.

# 5 Evaluation

In this chapter we will evaluate the state of the product compared to what we had in mind at the start of the project as well as evaluate what went wrong along the way.

## 5.1 Evaluation of the product

The product is, in it's entirety as it is today, not exactly as we envisioned at the start of the project.

### 5.1.1 Product Plan evaluation

We have implemented all our *must haves* as listed in our Product Plan, though we haven't (fully) implemented some *should haves* and *could haves*, most notably the following:

*The agent should be able to make decisions which have a negative effect on its goals, if it has a great benefit that can later be utilized.* This *should have* may have been too ambitious and, in retrospect, should have been listed as a *could have*.

Another one that was not implemented is *The agent should be able to ask other parties to perform actions which are beneficial to the goals of our agent.* This was not implemented because this level of communication and interaction with other agents is not implemented in both the other bots and the connector at this time. It turned out to be very difficult to achieve, because it requires carefully devised protocol messages or natural language processing.

### 5.1.2 Business Plan evaluation

Our team started on the project by creating a general plan of what we want our agent should be capable of, these objectives were written down in our business plan. The business plan mainly describes strategies and possible actions, the way we expected that it would satisfy our client. During the project, we found out that this business plan was very optimistic concerning the state of the connector. Almost all our objectives written in there need a lot of spatial knowledge, something that the connector is not able to provide yet. Some of our requirements were: *We prefer not to build houses near student houses*, *We want to make sure that our residents have parking lots near their houses*, *We want to build houses on ground that is not too close to roads that create a lot of noise disturbance*, *We want to build houses near enough green so the residents will have a nice environment.* All of which require a lot of spatial information and calculations, which we expected the environment would be able to provide us. After the realisation that the environment was in a very poor state, these requirements were practically given up on.

We have implemented the requirements that *We must obey all local restrictions.* and *We want to renovate or rebuild outdated districts whenever they do not satisfy our conditions.*

## 5.2 Evaluation of implementation

This section will mainly focus on the functionality of the agent. Performance will be evaluated and failures will be analysed for each module implemented for the agent. For each of these modules a brief description of the implemented functionality is provided and is followed up by an evaluation.

### 5.2.1 Upgrade

The construct module uses indicators to determine the type of building that the agent requires more of. When we own buildings that can be upgraded within a zone that has low indicator values for liveability, we will use upgrades to renovate this building. We will prioritize buildings with flat roofs in zones which also need more green, due to the fact that these can also be upgraded with a green roof after renovation. We want to build green roofs, because this helps the municipality accomplish their target for green, in return the municipality is more likely to approve our decisions. Considering the limited available upgrade types in the environment we implemented most of the beneficial possibilities. However it would be nicer to have our agent derive an optimal upgrade strategy considering more factors such as: relative location, budget, facilities and specific area strategies. To prevent buildings from being demolished after upgrading they simply can never be demolished any more. It is also possible that we upgrade a building in an area that we are trying to sell, or will be sold in the future. These interactions might lead to conflicting situation later on when a different strategy is adopted. It would be an improvement if the agent could derive a much less localized strategy instead of determining this for each building, that way conflict scenarios can be avoided. We also never look at the size of a building which we upgrade, while this can be used to improve indicators more efficiently.

### 5.2.2 Construct and Demolish

The construct and demolish module primarily work together. The demolish module will attempt to demolish unwanted buildings to free up land for newer buildings to be constructed. We do so according to our building indicator, which gives the agent a global target amount for all different types of buildings. The types of buildings for which the agent has too many will be demolished if they will not be upgraded. When a buildings is demolish there is available land for constructing. The agent will determine the building that we still require the most of, according to our building indicator, and it will construct this building on the specific location. This implementation of these modules can be considered rather basic, because only global targets have been set. This results in somewhat randomly distribution of locations for demolishing. It would be much better if these indicators were used to determine a strategy for each zone. We were not able to develop such strategies as the connector did not provide us with enough spatial properties to derive reasonable strategies from. Additionally, we never construct a building on land where we did not previously had a building demolished. We should have implemented functionality to build on any type empty land, because we could obtain empty land by buying as well. The construction module could also be extended to choose an optimal floor size with regards to the local restrictions from the municipality.

### 5.2.3 Buy land and Sell land

The buy and sell land modules are used to exchange pieces of land between stakeholders in the environment. Our sell land module finds all buildings our agent owns, which are not much use for our stakeholder but might be valuable to other stakeholders. These buildings along with the land they are located on, will be attempted to be sold what we deem as the best fitting stakeholder. Once the agent attempts to sell land it will decrease the price for this location every time an offer has been turned down, it does so three times after which it will try to sell the land to all other stakeholders, again with decreasing prices. Buy land works in a similar fashion, it also utilizes the same method to increase the price of an offer to buy land, but in this case there are only two stakeholders considered, the land owner and the agent's stakeholder. We also accept offers with a beneficial price to buy or sell land from our stakeholder, but only if we were already attempting to buy or sell this land. Although basic functionality is delivered, there is much more functionality achievable than present in our modules. It would be very nice, when another stakeholder sends us a request to buy or sell land, to evaluate the request based on our indicators and not only look whether we already wanted to buy or sell this land, because it will not occur often that the exact same piece of land will be requested. We also do not have any logical decision making on buying land, we should have used our land indicator to set a target amount of land owned so that we can attempt to expand our properties, this way we can use environmental factors to find land to buy. Our sell land module looks only at building types that we do not generally want for our stakeholder, such as shopping and offices. It would be a great addition if this module also looks

at spatial properties to sell land with bad conditions considering its surroundings. That way we could try to achieve optimal locations for private homes, in such a way that we avoid: loud roads, student noises and green shortage in our surroundings.

This page is intentionally left blank.

# 6 Outlook

If, in the future, we were to work on this project again, all focus would be directed at developing the agent, since the connector is now fully functional. Since a lot of time and effort was put into the development of the connector this time around, not everything we set out to implement was finished. For instance, we think that the agent still lacks the ability to interact with other stakeholders. Our biggest goal for when continuing with this project would be to be able to have a wide variety of outcomes when running our agent in conjunction with the agents for the other stakeholders.

This page is intentionally left blank.

# 7 Appendix A: Question Form

**Q:**

**What was your overall opinion of the game?**

**A:**

**Q:**

**What would you say about the decisions the other stakeholder made?**

**A:**

**Q:**

**What did you think about the way the other stakeholder interacted with you?**

**A:**

**Q:**

**Do you think the other stakeholder was a human or a bot? why?**

**A:**

## Test Subject 1

### Q:

**What was your overall opinion of the game?**

### A:

**An interesting game with a lot of information. It was really nice that you could see what would happen to your score when you built something, it gave you the idea that you could really experiment before you bought anything.**

### Q:

**What would you say about the decisions the other stakeholder made?**

### A:

**I did not really get the feeling that the other stakeholder was playing the game right, it felt like he just wanted to try things just like me and it went really fast!**

### Q:

**What did you think about the way the other stakeholder interacted with you?**

### A:

**He gave me the option to buy one building from him, when I declined he tried it again with a lower price, it just felt like he was trying to use the sell button for the first time, when I tried to sell something to him he instantly declined, even when I put up a very high price, which I found very weird.**

### Q:

**Do you think the other stakeholder was a human or a bot? why?**

### A:

**With how fast he went I think it was a bot, it would explain why he did not really react on what I did and why he was really fast in trying to sell land to me.**

## Test Subject 2

### Q:

**What was your overall opinion of the game?**

### A:

**Not really my game, I never enjoyed games like this I am more into card games.**

### Q:

**What would you say about the decisions the other stakeholder made?**

### A:

**I did not really look at what he was doing but I saw in the block of text on the right that he was always doing something, a lot was happening in the game, I could not follow all of it. I don't even believe you should be able to play the game that fast.**

### Q:

**What did you think about the way the other stakeholder interacted with you?**

### A:

**The only interaction I had with him was when he was trying to buy something from me, I just sold it to him. After that I tried to buy something from him but he did not even buy anything I was trying to sell him, even when it was at a very low price.**

### Q:

**Do you think the other stakeholder was a human or a bot? why?**

### A:

**A bot, clearly, that would explain why he was so fast but I find it weird that he never accepted anything I tried to sell him and he did not sell anything after that first building, was the bot broken? *we told him why the bot would not do this.* That explains it but it was still easy to see that it was a bot because he was too fast.**

## Test Subject 3

### Q:

**What was your overall opinion of the game?**

### A:

**Honestly, I thought it was kinda boring. Nothing really happens except for the buildings that are being demolished or built. I would rather play SimCity then, as that is way more interactive.**

### Q:

**What would you say about the decisions the other stakeholder made?**

### A:

**I did not really pay that much attention to the kind of things that the other stakeholder was doing, but it seemed that he was constantly constructing new buildings and upgrading buildings**

### Q:

**What did you think about the way the other stakeholder interacted with you?**

### A:

**I did not have that much interaction with the other stakeholder. He tried to sell a building to me a few times but I did not want it. After that he was probably done with me because he did not try to sell me anything anymore, or buy anything from me. Also he did not accept any offers that I made (maybe he was still mad because I did not want to buy his building).**

### Q:

**Do you think the other stakeholder was a human or a bot? why?**

### A:

**I think it was a bot since the only thing he was doing was demolishing, building and upgrading things (it was like he kept doing the same things over and over again). Other than that he did not seem to respond to anything that I was doing in the game.**