Final report

WhySoSerious

June 16, 2016

Contents

1 Introduction

The Tygron Engine is a program created by Tygron, designed for exploratory city planning using a virtual environment. This program makes it possible to plan areas of cities by allowing different stakeholders to log into the same session, and negotiate about plans and measures. Each stakeholder has its own goals, represented by indicators. One of the biggest problems with the usage of the program, is that it is not properly usable if not all parties are present. Tygron wants to overcome this problem by creating virtual humans that can replace specific stakeholders when not all parties are present. It is our job to design these virtual humans.

1.1 Problem description

The goal of the project is to create an example virtual human and a connector which the virtual human can use to communicate with the server. Five different virtual humans with different goals will be created, and will be pitted against each other to achieve their own goals, by negotiating with the other stakeholders. These virtual humans can then be used by Tygron as inspiration to create their own virtual humans.

The virtual humans must be able to think ahead and use an adaptive strategy to reach its goals. Being able to negotiate with other agents would also make its goals a lot more achievable. For keeping track of goals, the Tygron Engine features indicators that automatically update according to what happens in the environment. The virtual humans will have to be able to understand these indicators, as they are the internal representation of their goals.

1.2 End-user requirements

The end-user in this project is the company Tygron. Tygron wants a virtual human that is capable of replacing a real human that controls the stakeholder. To be able to achieve this, Tygron has a few requirements to specify how the virtual human should work.

The virtual human should be able to interact with other stakeholders; it should be able to request permissions for buildings and buy land from and sell land to other stakeholders.

The virtual human should also be able to act in a way that improves its indicators; it should know which action should be done at what moment to get closer to the completion of its goals.

Another requirement for the virtual human is that it should be able to negotiate with other stakeholders. Stakeholders will eventually need things from other stakeholders, like land or some money for building. The virtual human should not just accept any request for any price. It should look at what it needs and whether the price is reasonable before deciding to accept or decline the request. The virtual human should also make requests to other stakeholders and react when a stakeholder declines.

It is important to cooperate within the Tygron game. The outcome of the game should be a situation in which the combined goals of every stakeholder are optimized. The virtual human should try to help other stakeholders achieve their goals, while maintaining its own goals.

The virtual human should be able to operate in a challenging situation. The situation and environment should resemble a fairly realistic setting, instead of an overly simplified testing setting. The virtual human should have enough complexity to be able to be a part of such a setting.

2 Product overview

We were asked to build a virtual human that can run on the Tygron engine. To be able to connect the virtual human to the Tygron servers we had to extend the Tygron EIS-connector. This means that our project consists of both developing the virtual human and extending the functionality of the EIS-connector. This last part was in co-operation with the other groups of the context. In this report we won't cover any changes that our group didn't add.

2.1 Tygron EIS-connector

The first product that has been delivered is an extension of the Tygron EIS-connector. The virtual humans use this to connect to the Tygron server. The connectors receives updates from the server and keeps the virtual human up to date using percepts. The goal agent then uses actions to influence the game. A system has been added to add custom actions, which can be used to generate parameterized data about the game, like places where a virtual human can build.

2.2 Virtual human

The second delivered product is the virtual human itself. The virtual human uses the GOAL language to control a stakeholder (in our case the student housing cooperation DUWO) to fulfill its indicators as defined in the Tygron environment.

3 Reflection

In this section we will reflect both the created product and the problems we encountered during the project.

3.1 Reflection on product

At the beginning of the project there was documentation on what the current Tygron-EIS connector contained. The documentation stated that there were several percepts and actions that we could use for our implementation of the virtual human. This means that we thought that we could focus on the implementation of the virtual human and create an interesting strategy. Unfortunately the documentation was incorrect and changed after about two weeks. This meant that our initial planning was not valid anymore and we needed to change our focus to the Tygron-EIS connector instead of the virtual human.

The virtual human does not contain functionality to negotiate with other virtual humans. Implementing such functionality proved to be very difficult and takes a lot of time as it requires a well-defined language that all agents agree on, as well as the ability to plan an action without it affecting the simulation. Since we needed to add functionality to the Tygron-EIS connector, we had insufficient time.

For the students it is important to live close to Delft University of Technology, so we had the idea to implement a feature for the virtual human that will make it build student housing close to Delft University of Technology buildings. After an internal discussion we decided that this functionality is not important to have because the simulation takes place very close to Delft University of Technology.

Because we had to focus on adding functionality to the Tygron-EIS connector, we had to make some concessions regarding the virtual human. A more detailed description of the strategy we would have wanted can be found in appendix A8

3.2 Problems encountered during the project

At the start of the project we had no idea where we should look when we wanted to add or change the behavior of the Tygron-EIS connector. No one gave us a hint as to where to look or how to add things. Because of this we had to find it out by ourselves. This took us a lot of time. Next time we should ask the TA's or other people who know more about it. Another option is that the TA's could have pointed out where to start.

We were working with 4 other groups on the Tygron-EIS connector, every group consist of 5 people. This means that there were 25 people working on the Tygron-EIS connector. To be able to get a clear view on which groups are doing what on the connector, we had meetings at Tygron on Tuesday and in the Insyght-lab on Wednesday. These meetings took a lot of time for everybody. Next time we could send only two people instead of the whole team. These two people will inform the other team members about what was discussed. This will reduce the overhead of the team.
Another problem of working with so many people is that every pull request to the shared repository needed to be approved by at least three other teams (apart from the team that submitted the pull request). Sometimes confirmation of teams took a lot of time, while occasionally in the meantime other teams had to wait for this pull request.

In week three the teams decided on having a single repository where every team could create their branches and work on the Tygron-EIS connector. However, Sander did not like this at all. For him it was way too complex and time consuming to grade all teams.
Unfortunately Sander was only informed of this at the end of the week, so all our progress of this week had to be redone the next week. We agreed the following times we should ask the TA's first if they agree on such an organizational idea.

Debugging the virtual human in eclipse is extremely difficult because the virtual human can not be resumed after it reaches a breakpoint. Besides this it is not possible to see the virtual humans beliefs, percepts and goals when it has been paused. Pausing and resuming the virtual human is also not possible. The only way to debug in eclipse is by reading the log of the virtual human, which is very time consuming and can be quite confusing to our opinion. When we talked about this problem with Wouter, he told us to use the simpleIDE. Unfortunately simpleIDE had some issues, we could not set breakpoints and we were not able to delete code. We communicated this to Wouter and a fix was online after a week. Unfortunately not all of his were able to use simpleIDE.

In week 7 Wouter decided we should use our own maven module. This module is an extension of Wouters environment. To achieve this we had to rewrite our code and add to the new module, which took a lot of time. We believe that this change should have been done in the first week of the project or even before the start of the project. When this was done before the begin of the project, a lot of merge conflict could have been avoided.

Wouter and Frank are working on a different branch than we are. Next to this they don't pull our changes to their branch. This resulted in a lot of merge conflicts.

In week three the provided sample virtual human was not able to run on the map we specified in the virtual human.

None of us have ever written unit tests for the GOAL language but we were graded on GOAL unit tests. Because of this we had to invest a lot of time to find out how we should use the GOAL unit tests. Besides this, there were some versions of the GOAL runtime where all GOAL tests passed. To find out if we were using the correct version, we first had write tests that fail and find a working version of the GOAL runtime.

The services of Tygron also had quite a few issues. There was a long weekend (from Friday afternoon till Monday morning) in which we could not connect a virtual human to the Tygron Engine. On Friday afternoon of sprint seven, Tygron went down till Saturday afternoon. Next to these big issues there were some days where the Tygron Engine reacted very slow, which made testing of the virtual human quite hard. In all of these cases we have send an email to Frank (from Tygron) directly, with a question if they can do something about our problems.

4 Description

In this section the developed functionality is described. It is split up in two different sections. In section 4.1 the added functionality of the Tygron-EIS connector is described and in section4.2 all functionality of the virtual human.

4.1 Tygron EIS-connector

The following paragraphs describe what we as a group contributed to the connector.

First of all, a custom actions system was developed. Using an interface, `customAction` describing custom actions and a map containing all custom actions, this system made it very easy and straightforward to add new custom actions. A class implementing the interface would have to be added, containing a method that will be called by the virtual human, and a method returning the name the virtual human would have to call. This system did make it necessary to change the way incoming actions are verified, so all actions are approved now.

A custom action has also been added, one for getting all areas a stakeholder can use for a specific goal, which we will call a `relevantAreasAction`. The custom action uses a system similar to the custom actions; it makes it possible to add goals to the action. The interface for these goals extends the custom action interface, thus making it possible to call a goal directly if it is correctly added to the custom actions map.

One `relevantAreasAction` has been added as well, returning all area a stakeholder can build in. This action takes all of the lands of a stakeholder and removes all land it cannot build in. Then it will apply some after-effects, converting the usable land to smaller, better looking shape.

A second custom action was added for disabling certain percepts. This would make it a lot easier to test the virtual human, as it is possible to fake percepts but if the virtual human immediately receives a new percept from the server, the fake data is overwritten, and the test would be inconclusive.

4.2 Virtual human

The following paragraphs describe the inner workings of the virtual human.

The virtual human uses GOAL as its language, and as such, is structured as a GOAL agent is supposed to be structured. It uses six types of modules: A main module, this module is started when the multi-agent system is started, an event module, this module handles all events from the environment, knowledge modules, which contain the general knowledge of the virtual human, a goal module, which states all of the possible goals the virtual human can have, an init module, which contains the initial knowledge of the agent, and regular modules containing the logic the virtual human follows.

The virtual human uses different event modules to control and organize the control flow. The event module uses three modules; one to handle all percepts, then one to process goals and at the end it handles all requests the municipality needs.

For the logic flow, the virtual human uses goal-specific modules. When the goal is to demolish buildings, it will enter a module for all of the demolishing logic. It will adopt the goal for demolishing when it is not possible to build on any piece of owned land. When the virtual human is running out of budget it will use appropriate modules depending on how much money it still has.
If the virtual human is out of budget, it will try to sell land to other stakeholders. It will only try to sell

land that does not influence its indicators. This kind of land are gardens and nature (think of trees). When another stakeholder denies the offer of the virtual human, it will try to sell land to another stakeholder. When all stakeholders deny the virtual human will try to sell a different piece of land.

The virtual human can choose between three different types of student housing: cheap, medium and luxury. The goal of the DUWO stakeholder is defined in such a way that it needs the same amount of houses of each type of student housing. To achieve this the virtual human looks at its indicators and decides which type of student housing to build. If the virtual human is almost out of budget and wants to build medium or luxury housing, it will try to sell land. This is because medium and luxury student housing can be quite expensive.

5 Human Computer Interaction (Interaction design)

5.1 Method

5.1.1 User study

The user study is meant to evaluate the virtual human, based on test interactions with users. It will give insight into how users would act if they were in the same situation as the virtual human. It also makes clear what users think about the virtual human and what could be better. This information can be used to improve the virtual human in a way that it can be more user friendly.

5.1.2 Interaction user with virtual human

—X— people were asked to cooperate in the user study. Each user was given a short explanation of the Tygron Engine Software. After that we let them play around with the stakeholder municipality or facilities for a minute to get them familiar. While they play around they get some information about the stakeholder and its goals. Then we introduce the virtual human to the environment and let it react with the user for three minutes. While the user interacts, their behaviour is studied and written down. Several aspects are noted in particular:

- Do they seem to understand what is happening?
- How long does it take on average to interact with the virtual human?
- Do their actions increase their indicators?

5.1.3 Comparison user with virtual human

The environment is reset and the user now controls the stakeholder DUWO. The virtual human will not get connected to the environment. A study supervisor will control the municipality to accept or decline requests. After a short introduction of the DUWO stakeholder and its goals, the user is able to act as they want for three minutes and their actions are written down. After this, the user is asked about what drived their decisions and what strategy they used to get closer to DUWO's goals.

5.1.4 Evaluation

In the end the user is asked about their experience in general, their thoughts on the virtual human and how they would describe the interaction between them and the virtual human. They are also asked about their expectations and what they think could be improved.

When this is finished the study supervisor compares the strategy and behaviour of the user with the virtual human and writes down the differences.

5.2 Result

5.3 Conclusion

6 Evaluation

This section contains an evaluation of the final delivered product, divided in a subsection for the connector and a subsection for our agent.

6.1 Evaluation of our connector

During the project all groups from the virtual humans context worked together to extend the functionality of the connector between the Tygron-API and the GOAL agent. The added code has been put in a separate modules so that the changes these groups made are easily identifiable. The changes that each individual group has made, can only be seen when you look at the pull requests made in the connector repository. Each pull request has been tagged so that it's easy to see which group made the changes. Furthermore, all changes needed to be tested. Travis was set up to automatically run both the tests and code coverage so that it was easy to see the quality of the changed code. The changes made by all groups to the connector have a code coverage of 81.6%. Our code coverage goal was 75%.

Below you will find a short overview of our contributions to the connector and the impact they have on the connector as a whole in order of importance to the connector.A detailed description of our contributions can be found in the Description section 4 of the report.

- CustomActions system
  The CustomActions system allows a developer to add actions to the environment. A few actions were already implemented but we found that these were not sufficient for working virtual human. We consider this to be our most important contribution to the connector.

- GetRelevantAreas action
  This interface allows a developer to create an extension to the GetReleventAreas action which could send areas owned by our agent which have a certain property (for example: a collection of areas on which the agent can build or which the agent is can sell). Without this action the agent has no sense of location.

- Build part of the GetRelevantAreas action
  Giving the GetRelevantAreas action the build parameter allows an agent to receive all the areas on which it can build buildings. Without this it isn't possible to build a proper building.

- The filterpercepts action
  The filterpercepts action allows an agent to temporally disable certain percepts to make goal testing possible.

6.2 Evaluation of the virtual agent

Unlike the connector every group created their own virtual human. As described in the reflection in section 3, this process depended heavily on the features of the connector. Despite these features being developed a bit slower than expected, we made sure to release a meaningful addition to our agent each week. Every change to the code had to be made in a different branch and pulled to the master using a pull request. This pull request would be automatically tested using Travis-CI configured for goal tests. After at least 2 team members approved the request the new features would be merged into the master branch.

In the beginning of the project we made a high-level product backlog in our product planning. This backlog was ordered according to the MoSCoW method. The must haves from the backlog have all been completed. Except for trading land and inter-agent communication all should haves are also implemented. A more detailed description of how these could be implemented can be found in the outlook section 7.

7 Outlook

Our virtual human simulates the student housing cooperation DUWO. We have created several different strategies that resemble the strategies of DUWO itself. In the end however, the virtual human is restricted by the capabilities of the Tygron-API and EIS-connector.

Building student housing closer to the TU Delft buildings would be a very interesting feature. Students generally want to live as close to their University as possible, incorporating a function that calculates the distance between possible student housing and buildings owned by the TU Delft stakeholder is currently not possible according to Tygron. If this the indicator system was expanded to allow this a lot of interesting goals could be set for both Virtual humans and real humans using the Tygron engine.

Another feature that would make the simulation more interesting is the possibility of upgrading our buildings to be more environment friendly. The Tygron engine allows this for several buildings, but unfortunately student housing is not included. With some more time it would be possible to add this to the environment and incorporate it into the agent.

Finally communication between virtual stakeholders through the Tygron agent could be developed. This improvement is probably the hardest to realize since it requires a well-defined language that all agents agree on, as well as the ability to plan a action without it affecting the simulation. However this would allow our virtual humans to do a lot of interesting things and make more complicated decisions that might not benefit the agent immediately, but help it in the long run.

8 Appendix A: Ideal virtual human strategy for the DUWO stakeholder

This document describes what was the original plan regarding the virtual human strategy for the DUWO student housing stakeholder.

8.1 Goals

In the ideal strategy virtual human has five goals. These are:

- Build enough student housing to house two thousand more students.
- Build houses so each type of student housing is roughly equally represented.
- Build student housing close to the Delft University of Technology.
- Don't exceed the budget of twenty-five million euros and keep ten million euros as a reserve.
- Increase the livability in the zones with student housing.

Each of these goals are paired with an indicator that is visible within the Tygron game.

8.2 Situation

In the starting situation the DUWO stakeholder has some large patches of owned land and a few small ones. Some are very close to the university, but most of them are roughly a kilometer away. All of this land is occupied with buildings or gardens. The stakeholder owns buildings that house roughly two thousand students. Their housing types are not equally represented. Most of the houses are of the cheap and simple type and the luxury type, while the amount of medium type houses are roughly half the size of the amount of cheap type houses.

8.3 Strategy

The virtual human can be in any of the following mental states. Each mental state determines how the agent will act. The first state corresponds to making space by demolishing gardens. Since there is no free space left to build student housing the virtual human should demolish something first. Demolishing student housing can negatively affect the build and variety indicators, while demolishing gardens can not and is relatively cheap, so the virtual human will demolish gardens.

When there is space to build, the virtual human accesses the normal building state. While in this state, the virtual human builds houses on available land. This will increase the build indicator. To also increase the variety indicator, the virtual human builds student houses with the type of which there are the fewest. It selects the places closest to the university.

The third state is activated when the virtual human is close to reaching its reserve budget, this happens when its less than 20% above the budget target. When the building budget is low, the virtual human will build cheap student houses only and try to sell some of the land that does not have student houses on it.

When the virtual human has less than ten million euros, it has used up some of its reserve budget. It has to earn the money back. To accomplish this the virtual human will focus on selling land to other stakeholders, for five hundred euros per square meter, a price somewhat higher than the default price. If they decline it will try to sell it for four hundred euros, and if this is also declined it will try once more for three hundred euros.

The last state to be mentioned corresponds to the livability of zones. When the two thousand extra student houses are realized, the virtual human will look at some luxury houses in zones with the lowest livability and upgrade them to student houses with green roofs. The student containers are also upgraded to cheap student flats. Furthermore, the virtual human will try to sell land in the low livability zones to the municipality so gardens and forests can be built on it.