Tygron Virtual Humans

TI2806 Context Project Final Report

Team

CodeFox

Members

Ronald van Driel
# 4289870

Matthijs Halvemaan
# 4393007

Marco Houtman
4307232

Joshua Slik
# 4393007

Lisette Veldkamp
4394712

This page is intentionally left blank.

Introduction

In the serious game for urban planning by Tygron, a company that focusses on serious games for urban planning, it is possible to look at possible solutions for problems in development projects in cities and villages. In this game there are certain stakeholders that are played by humans. These stakeholders all have different interests and goals that they will try to accomplish. What we want to achieve by working on this project is to replace human stakeholders by virtual humans that make use of automatic decision making. These virtual humans will be able to make well-informed decisions in the game, making it possible to run sessions of the game without all human stakeholders available.

The agent that we developed is designed to simulate human behaviour as accurately as possible. When there are no humans available at a certain point of time, an agent can just simply fill in the role. Eventually, it might be possible to not have to use humans anymore because agents might even be better at doing the job as they are not biased and can do computations faster.

Each group that is part of this project developed an agent for one of the five stakeholders in the game, so at the end of this project we can run all the agents in a session without the need of any humans.

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.

Contents

This page is intentionally left blank.

1 Overview of the software product

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

The product we developed consists out of the virtual human, written in GOAL, and the connector that lets the agent connect to the environment, written in Java. The connector makes sure that we can provide our agent with continually updated information from the Tygron engine. This is needed for the agent to act and react on everything that happens in the environment.

The connector translates XML objects that are received by the Tygron engine, to parameters usable by the GOAL language and the other way around, so the environment will change when the agent executes actions on it.

The connector uses the Tygron SDK to return information from the tygron engine to the Tygron Environment, the SDK contains MapLinks which contain all the information of every class which can be called on by the environment. These MapLinks can be added to a listener and when a MapLink is being updated by the Tygron Engine the environment gets an update with the itemset the MapLink contains.Another way to utilize the SDK is by creating actions in the environment which call on usable Actions in the SDK to manipulate the engine.

While the connector ensures that the agent will be provided by useful information from the environment, the agent takes all the information in account to decide the best thing to do in his situation.

The agent never terminates unless it is manually done. In the beginning the agent will initialize its beliefs in the tygronInit module. The project in GOAL runs through an endless amount of cycles. Every cycle the agent will handle all the incoming percepts and process them accurately. It will then enter the tygronGoals module where the agent will decide upon which goals to adopt. Following this in the main module the agent will handle the goals by entering the required modules accordingly (the buyLand, sellLand, demolish, construct and upgrade modules). In these modules the agent will perform actions that are defined in the action module.

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. 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

For this project 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. 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 the 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 often it was due to bad time management, and many tasks needed to be carried over to future sprints.

Secondly a pull-based development model was used throughout the development. New functionality, bug fixes, documentation etc. were all added through pull request, and were only merged after at least two members of group (or two other groups in case of the connector) approved of the changes. Unfortunately, reviews were not always very critical, which led to lots of inconsistencies in style, which in turn meant that later on a lot of refactoring needed to be done.

2.2 Private Housing Company Agent

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. Unfortunately, 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. At first, we did manage 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.

The test driven development model was completely unused during the project. Lots of functionality was added to the agent without any tests whatsoever. Manual testing was always performed, but automatic tests were not added until very late in the development process. Also not all modules have automated test cases and because of the many dependencies with the connector, creating these automated tests proved to sometimes be near impossible.

2.3 Tygron Connector

Since the connector was completely written in Java, static analysis tools and continuous integration were no challenge, 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 found in the connector.

Test driven development was, unfortunately, also not used in the development of the connector, but unlike with the agent, every feature was thoroughly tested before being added to the final product.

3 Description of Developed Functionalities

In this chapter we will extensively describe the functionalities that we developed for our 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 profits. In order to achieve these two things we have designed our agent to take all necessary indicators in account. We will now provide an explanation per developed functionality based on the strategy that the agent has.

3.1 Constructing Functionality

When the agent has land without any buildings on it, for example from a previously demolished building, it will construct a new building here. Depending on the indicators for the balance and the environment, it will be luxury, normal or social housing.

3.2 Upgrading Functionality

The agent will adopt a goal to improve a zone whenever this certain zone does not live up to our standards according to the indicator target. We will then find buildings we own in this zone so that we can improve them and also the indicator numbers.

3.3 Buying Land Functionality

Currently the agent is never looking to buy any land. However we will change this in the short future. When the agent gets a request from another stakeholder looking to sell land to us, we will always say no. We will change this later too.

3.4 Selling Land Functionality

The agent will try to sell land that contains buildings on it that have nothing to do with housing, such as offices, leisure and other categories that do not fit our housing image. The agent makes an effort to sell these buildings to other stakeholders, to which they might be useful. It starts by trying to sell it to the Facilities stakeholder, as they are always looking to buy land. The agent will make a maximum of three attempts to sell the land to the potential buyer, each time with a slightly reduced price. Whenever an agent sends us a request to buy land from us, we will decline the offer as we already have construction plans for all empty land that we possess.

3.5 Demolishing Functionality

When our agent attempted to sell land but it did not succeed, the agent will demolish the building so that some other building that will be more useful to us can be constructed. Also the agent will demolish social housing buildings whenever the indicator that gives the number of different kind of housings points out that we would like to have more luxury or normal housing.

This page is intentionally left blank.

4 | Interaction Design

This chapter describes the 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 Q&A is conducted with the Test Subject regarding the way the other stakeholder played.

Metrics There is only one metric used: After the Q&A we look at the way the test subject reacted on the actions of the agent and determine from these results if he felt like he was playing against a human or against a bot.

Results

For now only one test subject has conducted the test, he figured out pretty quickly that it was a bot because he believed him to be *too fast* and his reactions on the subjects action were *too predictable.* He also stated that the agent sometimes made irrational decisions regarding use of his environment and was not very proactive towards the test subject.

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 sometimes irrational thinking.

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 ambitions 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 other bots and the connector at this time, and this is also very difficult to achieve without implementing some natural language processing.

5.1.2 Business Plan evaluation

Our business plan was very optimistic concerning the state of the connector. Almost all our requirements written in there are requirements that 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 these things require a lot of spacial information and calculations, which we thought 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 failures

We had some start up problems at the start of the project, like a lack of information, scattered communication and members not arriving to meetings on time. These were resolved by making Slack the main communication platform and make clear agreements for the meetings.

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.

In the following sprints we encountered problems like people that were writing dependencies for other people did not communicate their progress clearly, so the dependent people did not know when they could start working on their part. There were some issues with improper and dirty branching. GOAL debugging stopped working and has not been fixed since, so we had to come up with a roundabout way 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.

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