# TI2806 Contextproject - Final Report Draft

Virtual Humans - Group 4

https://github.com/tygron-virtual-humans

Sven van Hal - 4310055

Tom Harting - 4288319

Nordin van Nes - 4270029

Sven Popping - 4289455

Wouter Posdijk - 4317149

June 18, 2015

# Contents

## 1 Introduction

This report is the final project report for Group 4 of the Virtual Humans for Serious Gaming Contextproject. The main goal of this project was to make a virtual human that could replace an actual human in the Tygron (Tygron, 2015) urban planning game. This virtual human should be able to play the game like it is a real human, this includes making rational choices as well as making choices based on emotions. For an in-depth description of our customers and their requirements, please read our Product Vision (Contextgroups, 2015).

The focus of our group was on making the emotion part of the virtual human. In this report you can find how we made this emotional part, you can also find a description of the software engineering aspect as well as the interaction design, the failure analysis and an outlook in what is next.

# 2 Overview

Our final product is a GAMYGDALA (Popescu et al., 2013) plug-in for the GOAL programming language (Hindriks, 2014). The final product can be split into two separate parts: the GAMYGDALA port and the GOAL plug-in.

## 2.1 GAMYGDALA port

“GAMYGDALA is an emotional appraisal engine that enables game developers to easily add emotions to their Non-Player Characters (NPC)." (Popescu et al., 2013).

GAMYGDALA is written in Javascript, but for it to properly work with GOAL, which is written in Java, we needed to port GAMYGDALA to Java. This GAMYGDALA port is our biggest software product. When we finished our initial port, we noticed that the Java code did not follow the Software Engineering principles. This is why we decided to do a total code refactor.

Our final version of the port is far better then the original port on account of the Software Engineering principles and it behaves in the same way as the original GAMYGDALA code. You can read more about this refactor in the Emergent Architecture Design (VH4, 2015) and in next chapter.

## 2.2 GOAL plug-in

The other part of our software product is the GOAL plug-in. We had to alter the GOAL source code to enable the usage of GAMYGDALA in GOAL. We now have a fully working plug-in that developers can use in their GOAL programs. The full GAMYGDALA functionality can be used within the GOAL environment.

# 3 Reflection from a software engineering perspective

There are unlimited ways to approach the development of a software program. To streamline the process of developing and to ensure product quality, it is necessary to adhere to software engineering guidelines and principles. This section will discuss the measures that have been taken from a software engineering perspective and in which way they have affected the final product and the development process.

## 3.1 Product

From the first day on it was clear there was a challenging task to solve: porting the emotion engine GAMYGDALA from JavaScript to Java. While ”JavaScript” and ”Java” sound very similar, these are actually two very different languages and require a very different approach from a programming perspective. Java is an object-oriented language pur sang, JavaScript, in practice, uses a more procedural programming style.

Because Joost Broekens, the original author of the JavaScript version of GAMYGDALA, has programmed GAMYGDALA using a semi-object oriented approach, we initially decided to port the classes and methods to Java one on one. The Java port was therefore immediately feature complete. However, from a software engineering perspective, the code lacked proper design patterns and thus overall quality due to the fact that JavaScript simply does not support such programming structures. Using InCode Helium (**TODO: REF**), we identified major flaws in the code and started a large-scale refactoring process to incorporate proper software design principles. During this process, we have reduced or split up very large methods, refactored methods with redundant or duplicate code and have implemented design patterns[1] where appropriate. The code review by Software Improvement Group identified additional flaws, which have been corrected in the final product.

The next and final challenge to face has been the creation of a GOAL plug-in for GAMYGDALA. The plug-in acts as an interface for the GAMYGDALA Java port, allowing other GOAL programs to utilize the emotion engine. [.. TODO ..]

## 3.2 Process

- SCRUM
- Pull-based development
- Problemen met sprints, inschatting etc. werkverdeling
- Wat hebben we geleerd?
- GitHub
- Meerdere repo’s

[1]For an overview of all design patterns that have been used, please review the Emergent Architecture Design (VH4, 2015).

## 4 Developed functionalities

# 5 Interaction Design

This section contains our interaction design report. You can read who will be the users of our final product, how we let these users test our product and what we learned from the results of these tests.

## 5.1 Users

As stated before, our final product is a fully functioning version of GAMYGDALA in the GOAL programming environment. Therefore, our final users will be GOAL programmers who want to use emotions for their GOAL agents.

To find testers we looked for students who were familiar with the GOAL programming language. Because the GOAL programming language is a language that is taught during the first year of the Computer Science bachelor (Logic Based Artificial Intelligence and the Multi Agent Systems project) there are a lot of Computer Science students that are familiar with GOAL.

We decided to ask second year Computer Science bachelor students to help us by trying to use our GAMYGDALA plug-in in GOAL. We chose some students who are also working on our Virtual Humans for Serious Gaming Contextproject, as well as some students who work on other Contextproject. This way we could collect the opinions of students who already know about our product and how it should work, as well as the opinions from students that did not use our product before.

## 5.2 How do we test

The best way to let our testers use our product is by just letting them play with it. This is why we asked them to try to give an agent some simple emotions.

For this test we used the Thinking-aloud method (Nielsen, 2012). This means that we asked our testers to say anything that came up in their mind during the test. This was the easiest way to find out if the testers enjoyed using our product and if the usage was easy enough.

## 5.3 Results

The results will be discussed here in the final report. This draft does not contain these results yet.

## 6 Evaluation

In this section you can read our evaluation about our experiences during this Contextproject. It consist of an evaluation of the product, a failure analysis, an evaluation of collaboration between our group members as well as an evaluation of the collaboration between the separate groups of our Contextprojext.

## 6.1 Product

This subsection is divided into two parts, the GAMYGDALA port and the GOAL plug-in.

### 6.1.1 GAMYGDALA port

This was the first project in which we had to work with code that we did not make ourselves. We had to work with the GAMYGDALA engine that was already made, this was an entirely new experience. The initial idea was to literally port the GAMYGDALA Javascript code to Java, this was not a lot of work. When we finished this initial port, we found out that it did not really follow the Software Engineering principles. Because our Software Engineering skills are also rated during this project, we decided to start refactoring the port. You can read more about the problems that we encountered during this refactor in the Failure analysis subsection.

Now that the port is finished, we are quite happy we the result. Although the code is not perfect yet, it is a lot better then the port that we started with. It now follows the Software Engineering principles a lot better, it uses design patterns and for example the god classes are eliminated now. You can read more about this in our Emergent Architecture Design (VH4, 2015).

It was a great experience to work on someone else's code because we encountered problems that we did not encounter in previous projects. We learned a lot of new skills that will be useful in the future.

### 6.1.2 GOAL plug-in

A lot of what we wrote about the GAMYGDALA port also holds for our GOAL plug-in. We spent a lot of time in understanding how GOAL works 'under the hood'. We needed to truly get how GOAL works before we could start making a plug-in for it. Over time we gained more knowledge about GOAL and we started making a simple calculator plug-in. This took more time then we first anticipated, but we learned a lot from making it. This knowledge was used later on to make the actual plug-in for GAMYGDALA.

Though we also worked on code that we did not write ourselves, there are also some differences between making the GAMYGDALA port and this plug-in. In the port code we worked on every class, but in the GOAL code we only added and changed code in certain classes. Making the plug-in was more about adding code and functionality and the port was more about copying the code and functionality. It was very nice to have these two experiences within one project.

## 6.2 Failure analysis

## 6.3 Collaboration between our group members

## 6.4 Collaboration between the groups

## 7 Outlook

## References

Contextgroups. (2015, May). *Virtual humans for serious gaming product vision* (Tech. Rep.). Delft University of Technology. Retrieved from https://github.com/tygron-virtual-humans/port-deliverables/blob/master/report/Deliverables/Product%20Vision%20Nieuwe%20versie.pdf

Hindriks, K. V. (2014, March). *Programming cognitive agents in goal* (Tech. Rep.). Delft University of Technology. Retrieved from http://ii.tudelft.nl/trac/goal/raw-attachment/wiki/WikiStart/Guide.pdf

Nielsen, J. (2012, January). *Thinking aloud: The #1 usability tool.* Retrieved from http://www.nngroup.com/articles/thinking-aloud-the-1-usability-tool/

Popescu, A., Broekens, J., & van Someren, M. (2013, January). *Gamygdala: an emotion engine for games* (Tech. Rep.). Delft University of Technology and The Informatics Institute of the University of Amsterdam. Retrieved from http://www.joostbroekens.com/files/Popescu_Broekens_Someren_2013.pdf

Tygron. (2015, June). *The tygron website.* Retrieved from http://www.tygron.com

VH4. (2015, June). *Virtual humans for serious gaming group 4 emergent architecture design* (Tech. Rep.). Delft University of Technology. Retrieved from https://github.com/tygron-virtual-humans/port-deliverables/blob/master/report/Deliverables/Emergent%20architecture%20design%20draft.pdf