

# Final Report

GROUP: Gamygdala-Integration:

B.L.L. Kreynen

M. Spanoghe

R.A.N. Starre

Yannick Verhoog

Joost Wooning

June 15, 2015

# Contents

<b>1</b>	<b>Abstract</b>	<b>3</b>
<b>2</b>	<b>Introduction</b>	<b>3</b>
2.1	Problem description . . . . .	3
2.2	User requirements . . . . .	3

# 1 Abstract

## 2 Introduction

In this section you can find a brief introduction to the context problem and to the solutions applied by group 3.

### 2.1 Problem description

The research problem is stated by a company called Tygron. They provide local authorities with a game in which their city is fully simulated. In this game the council members can discuss difficult matters on where to build certain structures. By doing so they gain an understanding of everybody's needs and responsibilities. The main question that rises is the possibility of replacing one of those players with an Artificial Intelligence solution. Since the players of this game mostly interact in an emotional way, as real humans do, these bots should feel emotions too.

This topic is very interesting because it has many applications. Both further research and different industries can profit from this. For this context project the students work in a large group that is subdivided into different smaller groups each with specific tasks. In total there are 4 groups of 5 students that work together. Together the whole group will make a proof of concept by creating a game in the Tygron engine[4], creating an interface between the Tygron engine and GOAL and creating both a plug-in and an integration of the Gamygdala emotional engine in GOAL. The specific task of this subgroup is to provide the group that will create an agent with an integration of Gamygdala. It is not our task to implement the engine itself, since it is already present, but it is our job to integrate it in GOAL in a way that programmers, like the other groups, can make use of this when creating virtual humans.

In this report there is given a overview of the user requirements first. Then a list of the implemented software product is explained in detail. After that, you can find a reflection of the product and process from a software engineering perspective. This is followed by a detailed list of implemented features. A section on the Human Computer Interaction will then explain how the product and users interact. Finally a conclusion and outlook will list all the important findings and future improvements.

### 2.2 User requirements

In this section you can find the specific tasks the product should fulfill. By using the MoSCoW method we can subdivide the high level specifications into the following. The subsections decrease in importance or priority. This means that for example the MUST-haves are way more important to implement than the COULD-haves. This method is very handy and more information can be found in the glossary or references.

### **Must have**

- An agent in GOAL must have an equivalent in Gamygdala.
- The agent's goals in GOAL should also be present in Gamygdala for that agent. This means that if you define a goal for an agent in GOAL that the Gamygdala engine must also know of this. New goals must be communicated to Gamygdala.
- It must be possible to define how good or bad a belief is for certain goals.
- It must be possible to define relations between agents in GOAL.
- It must be possible to have these relations also present in Gamygdala. Again, it must be the case that when a relation is created in GOAL, the Gamygdala engine knows about this relation. Otherwise no emotions between two agents can be calculated.
- It must be possible to retrieve the emotional state of an agent in the same way as his beliefs.
- It must be possible to setup an initial emotion of an agent.

### **Should have**

- It should be possible to set the gain to a specific value.
- It should be possible to set the decay to a specific function.
- The Gamygdala goals and its properties should be possible to change. This also includes the relations regarding beliefs and goals.
- It should be possible to display the emotional states in the Simple IDE[5] provided with GOAL for debugging.

### **Could have**

- It could be possible to set a custom decay function.
- It could be possible to change relations during runtime.
- It could be possible for an agent to reason about other agents emotional bases.

### **Will not have**

- Eclipse plug-in

## References

- [1] GOAL programming language <http://ii.tudelft.nl/trac/goal>
- [2] Gamygdala emotion engine <http://ii.tudelft.nl/~joostb/gamygdala/index.html>
- [3] MoSCoW method [http://en.wikipedia.org/wiki/MoSCoW\\_method](http://en.wikipedia.org/wiki/MoSCoW_method)
- [4] Tygron engine <http://www.tygron.com>
- [5] Simple IDE <https://github.com/goalhub/simpleIDE>