

# Product Plan

GROUP: Gamygdala-Integration

May 7, 2015

# Contents

<b>1</b>	<b>Introduction</b>	<b>3</b>
<b>2</b>	<b>Product</b>	<b>3</b>
2.1	High-level product backlog . . . . .	3
2.2	Road map . . . . .	4
<b>3</b>	<b>Product backlog</b>	<b>4</b>
3.1	User . . . . .	4
3.2	Userstories of features . . . . .	5
3.3	Userstories of know-how acquisition . . . . .	5
3.4	Release plan . . . . .	5
3.4.1	Milestone 1 - week 6 . . . . .	5
3.4.2	MileStone2 - end . . . . .	5
<b>4</b>	<b>Definition of Done</b>	<b>6</b>

# 1 Introduction

This group will create an integration of the Gamygdala engine in the programming language GOAL. In this document we will discuss all the important things about the wanted specifications of our product.

## 2 Product

In this section you can find the specific tasks our product should fulfill.

### 2.1 High-level product backlog

By using the MoSCoW method we can subdivide the high level specifications into the following.

#### Must haves

- 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.
- It must be possible to define how good or bad a belief is for goals.
- It must be possible to define relations between agents in GOAL.
- It must be possible to have these relations also present in Gamygdala.
- It must be possible to retrieve the emotional state of an agent in the same way as his beliefs.
- it must be able to setup initial emotion of an agent.

#### Should haves

- 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 for debugging.

### Could haves

- 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.

### Won't haves

- eclipse plug-in

## 2.2 Road map

- week 1: seminars
- week 2: understanding Gamygdala by testgame
- week 3: setting up goal and research on both goal and Gamygdala.
- week 4: agents in goal should get in Gamygdala their equivalent + goals. relations between agent/goals definition/
- week 5: Extra mental database for emotions of an agent. Retrieve emotions.
- week 6: debug tool + updating relations (between believes/goals agents/agents) + setting gain/decay function
- week 7: implementing ability to set custom decay function
- week 8: used for tasks that are not finished
- week 9: used for tasks that are not finished
- week 10: presentations + finishing

## 3 Product backlog

In this section you can find an overview of the backlog and release plan.

### 3.1 User

We define a user as a programmer that wants to use GOAL to develop an AI identity. While doing this, he wants to involve emotions and develop his identity in such a way that he can use emotions to trigger actions, but also actions in the environment (via percepts) that change his emotions would be very handy for the user. The user has a good understanding of the programming language GOAL. He has a very small understanding of Gamygdala.

### 3.2 User stories of features

- As a programmer I want to be able to define a relationship between beliefs and goals so that these effect the emotional state of the agent. It must be similar to define goals in GOAL.
- As a programmer I want to be able to define relations between different agents.
- As a programmer I want to be able to run my GOAL-agent and see the emotional state he is in.
- As a programmer I want to be able to query the emotional base in the same way as the believe base. This so that the agent can reason about it.

### 3.3 User stories of know-how acquisition

- I want to be able to get a better understanding of Gamygdala by reading through the documentation so that more complex things could be implemented.
- As a programmer I want to be able to get a better understanding of the integration of Gamygdala in GOAL by reading documentations or papers. This so that I can reason about it and create more complex solutions.

### 3.4 Release plan

#### 3.4.1 Milestone 1 - week 6

For week six we need to have a working version of basic integration so that the other group creating the virtual human is able to use this and experiment. If we don't give them any version to work with, it will be too late for them to use the integration of the Gamygdala engine in GOAL. Minimal features are the features we planned until week 6.

#### 3.4.2 MileStone2 - end

At the end we need to finish our product. The real minimal features are the must haves we have defined earlier. The bare minimum is thus that the features of milestone 1 work perfectly and are tested thoroughly.

## 4 Definition of Done

We consider something as done if:

- test coverage of at least 75%
- documentation of new implemented things must be present
- at least two other colleagues agreed on the implementation and also consider it as done.

## Glossary

**agent** an agent is a virtual identity that uses logical rules to derive the wanted actions and can percept events from the environment. It is AI.. 3

**Gamygdala** An emotional engine created to simulate emotions for virtual identities[2].. 3

**GOAL** A programming language developed at TU Delft for creating Virtual Humans[1].. 3

**MoSCoW** A method used to fill in requirements for a project based on difference in priority regarding the final goal of the project.[3]. 3

## 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))