

# Product Plan (draft version)

28/04/2016

**CodeFox**

*Marco Houtman*

*Ronald van Driel*

*Joshua Slik*

*Matthijs Halvemaan*

*Lisette Veldkamp*

# Contents

1. Introduction
2. Product
  - 2.1. High-Level
  - 2.2. Roadmap
3. Product Backlog
4. Definition of Done

# 1. Introduction

Tygron is a software company focussed on developing the Tygron engine which is used to streamline the planning process and substantially reduces the time and costs involved in urban projects. This is done by building an urban planning project and let users take on their respective roles within the project. It is not always feasible to have all respective roles within one place which makes it harder to run the game. This can be solved by introducing virtual humans which will take on their respective roles.

The Virtual Human developed by our team will take on the role of a Housing Corporation. It will focus on finding locations to build houses which are primarily focussed on private housing.

## 2. Product

### 2.1. High-Level

#### **Must have**

Our virtual human **must** meet the following requirements:

- Able to create decisions based on environment.
- Able to create decisions based on individual gain.
- Able to interact with other players.
- Able to determine how well it did according to an individual score.

#### **Should have**

Our virtual human **should** have the following features:

- Able to judge good and bad decisions based on all available information.
- Able to use other players in beneficial ways for both parties.

#### **Could have**

Our virtual human **could** have the following features:

- Able to develop advanced strategies based on expectations.

#### **Won't have**

Our virtual human **won't** have the following features:

- Able to use and interpret natural languages to interact between players.
- Able to use online datasets to gain additional information.

### 2.2. Roadmaps

**Sprint 1:** Explore the Tygron engine. Additionally discussions are to be made to determine which role each team gets to simulate.

**Sprint 2:** Develop game environment with which the virtual humans need to interact.

**Game demo.**

**Sprint 3:** Improve game based on feedback from demo. Develop the virtual human in goal.

**Sprint 4:** Develop basic interactions of the virtual agent for all scenarios.

**Agent demo.**

**Sprint 5:** Develop reasonably decision making of the virtual agent considering feedback from demo.

**Sprint 6:** Improve strategies of the virtual agent.

**Sprint 7:** Improve virtual agent.

**Agent demo.**

**Sprint 8:** Develop agent considering feedback from demo.

**Sprint 9:** Finalize product main functionalities.

**Sprint 10:** Minor quality improvements and fixes of minor issues.

**Final demo.**

### 3. Product backlog

#### **User stories of features**

As a housing corporation agent  
I want to know if ground is going to be available soon  
So that I can consider buying the ground

As a housing corporation agent  
I want to buy ground  
So I can build residential buildings

As a housing corporation agent  
I want to build residential buildings on ground that I own  
So that I can have buildings to sell

As a housing corporation agent  
I want to sell residential buildings I built  
So that I can make profits

#### **User stories of defects**

As a housing corporation agent  
I expect no direct consequences for me when a defect occurs  
So that I can keep on running smoothly

#### **User stories of technical improvements**

-

#### **User stories of know-how acquisition**

As a developer I want to know how I can work with the Tygron Connector.

## 4. Definition of Done (DoD)

A feature is considered "done" if and only if all of the following points apply:

- All of the requirements for the feature are met.
- The feature has been thoroughly tested.
- All old features which at one point in time worked must still function correctly.
- The code has been documented:
  - The documentation has been made available to all of the members of the team (either on GitHub or Dropbox).
- The developer of the code has committed the code to the GitHub repository
- The code has been merged into the dev or master branch
- The customer is satisfied with the feature.
- When all of the above points are achieved, the ticket associated to the feature will be closed and marked as done during the scrum meeting.
- If a feature is, for any reason, aborted before completion, all documentation/code should be removed before the closing the ticket to ensure a clean working environment.
- If a feature has to be removed after it has been completed, the team will decide what to do during the first scrum meeting after the decision has been made.