

Product Planning

VH-Danshal

Table of Contents

1. Introduction	..3
2. Product	..4
2.1 High-level product backlog	..4
2.1.1 Must haves	
2.1.2 Should haves	
2.1.3 Could haves	
2.1.4 Won't haves	
2.2 Roadmap	..5
3. Product backlog	..6
3.1 User stories of features	..6
3.2 User stories of know-how acquisition	..6
3.3 Initial release plan	..7
4. Definition of Done	..8
5. Glossary	..9

1 Introduction

Artificial Intelligence (AI) is becoming more prevalent in products around the world. Used for example to simplify tasks for us, for instance using AI in self driving cars so we don't have to drive ourselves. But not every project involving AI is about simplifying everyday tasks. An example of that is in the gaming industry, where AI is used to immerse the player with interesting behaviour and make the game more challenging. So why would the gaming industry be so interesting for AI?

Well the player of a game wants the AI to act as if the role of the AI was being played by another player. It can be frustrating for a player if the AI does something it should not do. If that would happen frequently a lot of players would critique the game very harshly, even if that was the only thing wrong with the game. That has as a consequence that the limits of AI in games need to be pushed even harder. Though AI playing alongside the player in a game is a very interesting topic, what if the player is actually being played by the AI?

This is definitely not the first time this question has been asked, since some time ago a level in the game Super Mario World, an action platformer game, has been finished by an AI. It is an amazing accomplishment, but the game is definitely not representative of the real world. There is however a field in gaming that is specialized to help solve real world problems, Serious Gaming. Tygron is a serious game about urban planning and which we are tasked with to create an AI for a specified case in the game.

In Tygron people can create their real world cases in which there might be conflict and the stakeholders can try to compromise with each other to resolve this conflict. After the case has been recreated in the game, the different stakeholders can negotiate with each other and see what kind of impact doing a certain action would have. The impact is being calculated with the help of the set up indicators to keep track of the goals for each stakeholder. If a certain action has been done this can impact other stakeholders and they will be informed about these actions in the game. So ultimately the goal will be for the stakeholders to resolve the conflict between the each other, while having visual feedback on what will happen.

For this project the specified case will take place in the TU area and involve the municipality, TU, DUWO, public housing and business facilities as stakeholders. All these stakeholders have different goals to achieve. Each group working on this project will pick one of these stakeholders and try to build the AI that achieves most or all of it's goals, while negotiating with the other stakeholders.

In the rest of this document the following will be explained: First of all in section 2 the product will be discussed with all the features we will implement for the product as well as a roadmap containing our major high release schedule. Secondly in section 3 the features of the product will be discussed in more detail in the form of user stories. Finally in section 4 we will explain our Definition of Done, in other words when we consider a task, sprint or release as finished.

2 Product

The final product that will be delivered is the GOAL agent filling the role of one of the five stakeholder as well as the adapted GOAL to the tygron connector. The GOAL agent will be able to negotiate with other agents in order to fulfill his own indicators. To be able to achieve this the given connector will need to be extended to accept actions that are not in the environment yet. In this chapter the high-level product backlog will be explained via the MoSCoW model. Also the roadmap is displayed showing the schedule for the rest of the project.

2.1 High-level product backlog

The product backlog is written with the MoSCoW model. In this model we can differentiate the features in four categories: must haves, should haves, could haves and won't haves. In the following sections we will explain these features in these categories.

2.1.1 Must haves

The following features are necessary for the product to function:

- Create the game(case) with the indicators
- Agent performs random actions in the environment
- Agent performs actions that benefit his indicators
- Agent negotiates with other stakeholders (deals)
- Connector (goal can work efficiently through connector with tygron)

2.1.2 Should haves

The product should have the following features:

- Develop different strategies to get high indicators

2.1.3 Could haves

The product could have the following features:

- Agent uses a strategy that helps give all the involved stakeholders good indicators
- An interface for the user to easily add an agent to a session
- Implement the agent so it works in more general cases

2.1.4 Won't haves

The product won't have the following features:

- Add machine learning functionality to the agent (ML that adapts goal code)
- Create more general goal agent for Tygron with high level interface that can be configured to use any action in the environment to fulfill any indicator.

2.2 Roadmap

The roadmap is planned for quarter four for from april 18th until june 17th, where the first week is used as an introduction to the project. That week will be considered as week 0 in the roadmap. The roadmap is based on SCRUM, where every sprint/week the product is improved with new backlog items.

Week	Tasks
0	Introduction to the product
1	Brainstorm about interesting cases Choose case and stakeholder
2	Create the case in the game
3	Simple interactions by agent
4	Strategy for fulfilling indicators
5	Being able to negotiate deals with other agents
6	Strategy for fulfilling indicators with other agents
7	Extra features
8	Bugfixing

3 Product backlog

The product backlog is the list of features that we want to implement in more detail with the help of user stories. In these user stories a feature is explained as how the agent needs to behave in the environment. This gives a general look at what the agent is supposed to do. In this chapter the different user stories will be explained as well as an initial release plan.

3.1 User stories of features

As the tygron client I want to be able to have stakeholder sessions with some human stakeholders being replaced with GOAL stakeholder agents.

As a GOAL stakeholder agent I want to be able to connect to the tygron environment.

As a GOAL stakeholder agent I want to be able to perform actions in the environment.

As a GOAL stakeholder agent I want to be able to retrieve information about the environment through the Tygron queries.

As a GOAL stakeholder agent when another agent requests something of me I want to be able to respond.

As a GOAL stakeholder agent I want to be able to fulfill my goals.

As a GOAL stakeholder agent I want to be able to negotiate a deal with other GOAL stakeholder agents.

As a Tygron context case configurer I want to write indicators the GOAL stakeholder agent needs to fulfill.

As a Contextgroup we want to extend functionality of the tygron connector to fit the needs of our GOAL stakeholder agent.

3.2 User stories of know-how acquisition

As a GOAL programmer I want to understand how to use the new GOAL version.

As a Java programmer I want to understand how the tygron connector works.

As a programmer I want to understand how the game tygron works.

3.3 Initial release plan

The initial release plan is heavily based on the roadmap of section 2.2.

Week	Milestone
1	Case for the game chosen
2	Created the case in the game
3	Simple interactions by agent
4	Strategy for fulfilling indicators
5	Being able to negotiate deals with other agents
6	Strategy for fulfilling indicators with other agents
7	Extra features
8	Final release

4 Definition of Done

In this section we will discuss at when we can consider something as done. This is necessary in order to establish ground rules regarding when someone can do another task when they are done with the one they were assigned to for example. We will explain when we think a backlog item, sprint and release is done.

A backlog item is considered to be done when it is implemented as code and functional. Also it needs to be tested with a test coverage of at least 75 percent and pass all these tests. Furthermore the code must be commented and styled correctly according to *Checkstyle*. The implemented code must not fail the continuous integration build. After it passed all of these things it also has to be set up as a pull request and be peer reviewed by at least two other people. When it passes all these requirements, it is considered to be done.

A sprint is considered to be done when it none of the tests fail. Also the continuous integration build should pass. Finally the entire product should be able to be shown working in a demo.

A release is considered to be done when the continuous integration build passes and it passes all the tests. Also all the code is commented and styled correctly. Furthermore it needs to have implemented most of the features. The agent needs to work in the Tygron game as the stakeholder it is implemented as. Finally the agent needs to be able to play in the Tygron game with all the other agents and perform reasonably given his goals and the means to achieve his goals.

5 Glossary

Artificial Intelligence	Intelligence exhibited by machine or software
Definition of Done	Definition of when a backlog item, sprint or release is considered to be finished
MoSCoW	Model in which features for the product are categorized into must haves, should haves, could haves and won't haves
Product backlog	List of features that can be implemented for the product
Roadmap	Plan with the goals to achieve
User stories	Small stories that describe an interaction from a specific role with the product