

VIRTUAL HUMANS FOR SERIOUS GAMING

PRODUCT PLAN

Jannelie de Vries
Harmen Kroon
Jasper van Tiburg
Tom Brunner
Nando Kartoredjo



EEMC
Delft University of Technology
Delft May 4, 2016

Contents

1	Introduction	2
2	Product	3
2.1	High-level product backlog	3
2.1.1	Must Have	3
2.1.2	Should Have	3
2.1.3	Could Have	4
2.1.4	Wont Have	4
2.2	Roadmap	4
2.2.1	Sprint 1	4
2.2.2	Sprint 2	4
2.2.3	Sprint 3	4
2.2.4	Sprint 4	5
2.2.5	Sprint 5	5
2.2.6	Sprint 6	5
2.2.7	Sprint 7	5
2.2.8	Sprint 8	5
3	Product Backlog	6
3.1	User stories	6
3.1.1	User stories of features	6
3.1.2	User stories of know-how acquisition	6
3.2	Initial release plan	7
3.2.1	Week 2	7
3.2.2	Week 3	7
3.2.3	Week 4	7
3.2.4	Week 5	7
3.2.5	Week 6	7
3.2.6	Week 7	7
3.2.7	Week 8	7
3.2.8	Week 9	8
4	Definition of Done	9
4.1	Backlog items	9
4.2	Sprints	9
4.3	Final release	9
5	Glossary	9

1 Introduction

Tygron is a company that builds serious games for urban planning. A game usually consists of a scenario (such as planning a new city area) in which several players play different roles (such as major, city planner, environmental agency, etc.) So these games are meant to be played by a group of people. Tygron is interested in simulating some of these people with Virtual Humans, so that you can play the serious game also when not every player in the scenario is present. This is useful, because it is hard to make an appointment when everybody is available. By replacing the roles with Virtual Humans, it is easier to make an appointment and even be able to play the game on your own. In this way you can play the game more often and have more time to execute the plans for real.

The Virtual Human developed by us will take the role of a public service manager. Its goal is to build convenience stores, terraces and a sports center at an optimal location so that profit can be made. Our Virtual Human will play the game at the TU-Campus location with other stakeholders like DUWO and TU-Delft. Together they want to make the TU-Campus a better place to live.

During this project we will work using SCRUM. We will have weekly sprints and within each sprint we will implement a few features of our Virtual Human. By working in sprints, we have smaller goals and we will be able to check with Tygron more often. In this way we will be able to adapt to any changes Tygron wants us to make.

In this document, we will provide a high level backlog. The items mentioned in this backlog will be planned in a roadmap. Then we have a section with different user stories about how the game should work. Finally we will have a section about when the product is done.

2 Product

In this section we will give a high level backlog. The items discussed in this section will be in a roadmap, to give a high level few about the planning of this project.

2.1 High-level product backlog

Within this section the high-level features will be defined using *MoSCoW*. MoSCoW uses four categories to separate the features by level of importance. The categories are:

Must Have: Features that are of high importance. With those features the agent is considered as properly working.

Should Have: Features that are considered favorable. Without the agent should still be functional and should properly work.

Could Have: Features that are of low importance. When there will be enough time to implement, the features will be present.

Won't Have: Features that won't be implemented.

2.1.1 Must Have

- The stakeholder should be able to build a convenience store.
- The stakeholder should be able to buy land.
- The stakeholder should be able to sell land.
- The stakeholder should be able to demolish:
 - The stakeholder should be able to demolish land.
 - The stakeholder should be able to demolish buildings.
- The stakeholder should be able to have a goal, such as building at least 3 convenience stores. This is called an *indicator*.

2.1.2 Should Have

- The stakeholder should be able to build more types of buildings:
 - It should be able to build a terrace.
 - It should be able to build a sports center.
- The stakeholder should have a low Level of communicating:
 - It should have information about the changing surroundings.
 - It should be able to calculate an efficient placing of a building.
- The stakeholder should be able to achieve its goals:
 - It shouldn't go bankrupt.
 - It should reach its indicators considering convenience stores.
 - It should reach its indicators considering terraces.
 - It should reach its indicators considering the sports center.

2.1.3 Could Have

- The stakeholder could be able to understand the used language between stakeholders.
- The stakeholder could be able to negotiate with other stakeholders:
 - It could be able to barter prices.

2.1.4 Wont Have

- The stakeholder won't interact with human stakeholders:
 - It won't understand messages send by human stakeholders.
 - It won't send messages to humans.

2.2 Roadmap

In this section we describe a high level planning for the items stated in the section above. Per sprint we select a part of the product we are going to develop during this sprint.

2.2.1 Sprint 1

During this sprint, we will find out how the software works and make a start with the product vision, the product plan and the architectural design.

- Try different scenarios and stakeholders when using the software of Tygron.
- Make the product vision document.
- Make a product plan.
- Think of a scenario with different stakeholders and a conflict.

2.2.2 Sprint 2

During this sprint we will find information about our stakeholder and make a final scenario. Also we will brainstorm about how to implement the basic action for our stakeholder.

- We should choose an indicator for our stakeholder.
- We have to make a demo of the final scenario.
- We need to find out how we can make the stakeholder be able to build a convenience store.
- We need to find out how we can make the stakeholder be able to demolish land and buildings.

2.2.3 Sprint 3

During this sprint we will make sure the stakeholder is able to do all the basic actions during the game.

- The stakeholder should be able to buy and sell property.
- The stakeholder should be able to demolish buildings and land.
- The stakeholder should be able to build in context to its indicators.

2.2.4 Sprint 4

During this sprint our stakeholder should be able to reach his goals.

- The stakeholder should be able to reach its indicators target.
- The stakeholder should be able to make sure it doesn't go bankrupt.
- We need to make a demo in order to show how our stakeholder is able to reach its goal.

2.2.5 Sprint 5

During this sprint the stakeholder should be able to notice what other stakeholders do.

- The stakeholder should be able to notice the other stakeholders.
- The stakeholder should be able to keep track of the decision of other stakeholders.

2.2.6 Sprint 6

During this sprint the stakeholder should be able to calculate efficient placing of buildings.

- The stakeholder should be able to calculate if owned property is good for building.
- The stakeholder should be able to calculate if other land is good for building.

2.2.7 Sprint 7

During this sprint we could add more indicators to the stakeholder.

- The stakeholder is able build more shops
- The stakeholder is able to build parking lots.
- We need to make a demo in order to show the finished stakeholder

2.2.8 Sprint 8

During this sprint we will end everything about the stakeholder and make sure it still works with the other stakeholders made by other groups.

- run final tests with other teams.
- clean all code.
- deliver the code.
- deliver all documentation.

3 Product Backlog

In this section we will describe some user stories and an initial release plan.

3.1 User stories

In this section we will describe some user stories. Each user story describes what the stakeholder should do in a certain situation. The user stories are put in order, so the highest priority is first.

3.1.1 User stories of features

As a public service management bot
I want to buy land
So that I can build stuff

As a public service management bot
I want to build a convenience store on ground that I own
So that the indicator for the convenience store will go up.

As a public service management bot
I want to build a terrace on ground that I own
So that the indicator for the terraces will go up.

As a public service management bot
I want to demolish a building that I own
So that I can use the ground for other things

As a public service management bot
If the game crashes
I want to stop running
So that I don't mess things up.

As a public service management bot
I want to sell land
So that I have more money

As a public service management bot
I want to be notified if houses are demolished
So that I can decide if I want to build something near there or not

As a public service management bot
I want to understand the used language between stakeholders
So that I can communicate with other bot stakeholders.

3.1.2 User stories of know-how acquisition

As a programmer
I need to learn the newest GOAL code

So we can build a GOAL-bot.

As a programmer
I need to learn how to play the Tygron game
So we know how a bot should play it.

As a programmer
I need to make a Tygron scenario
So we have a scenario the bot can play in.

3.2 Initial release plan

In this section we will describe an initial release plan. Per week there will be a list of what to release.

3.2.1 Week 2

In this week we will release a draft versions of the product vision document, the product plan document and the architectural design document.

3.2.2 Week 3

In this week we will release the final versions of the product vision document and the product plan document. We will also release the final scenario where our stakeholder should play in.

3.2.3 Week 4

In this week we will release a bot who is able to buy and sell land. Also it should be able to demolish land and buildings.

3.2.4 Week 5

In this week we will release a bot who is able to build in order to increase the score of its indicators. Also is should be able to keep track of its bank account and make sure it doesn't go bankrupt.

3.2.5 Week 6

In this week we will release a bot who is able to keep track of the decisions made by other stakeholders during the game.

3.2.6 Week 7

In this week we will release a bot who is able to decide which land is the best for building a certain building.

3.2.7 Week 8

In this week we will release a bot who is able to finish its goals.

3.2.8 Week 9

In this week we will release the final bot with all documentations.

4 Definition of Done

In this section we will define when something is done. We will discuss when a backlog item is done, when sprints are done and when the final release is done.

4.1 Backlog items

A backlog item is done when it is 65% tested. We decided on 65% because 100% testing is impossible and also testing in GOAL is very hard. It is better to use user tests, by running the game with the bot then testing it via GOAL-tests. We wanted more then 50%, but we didn't want to set the minimum too high. That is why we decided on 65% test coverage by GOAL-test. Also a backlog item needs to be checked by all members, and everyone should agree that this item is implemented correctly. All test should pass and the code should be merged to the master branch.

4.2 Sprints

A Sprint is done when the features that should be tackled during this sprints are done. If a feature isn't done, the reason should be clear and it should be discussed with all team members that this feature is moved towards a next sprint. There should be a reflection about the sprint and we should contact the TA's. Problems during the sprints need to be discussed and a solution should be found. These solutions should be used for all upcoming sprints.

4.3 Final release

The final release is done, when all items stated in section must haves are implemented and at least 65% tested. We choose again 65% because, if all backlog items need to be at least tested for 65% also the final product should be tested for at least 65%. Also most items mentioned in section 2.1 should be implemented and tested.

The product owner should see multiple demos. These demos are mentioned in the roadmap in section 2.2. The product owner should also agree with the final product.

The product vision, product plan and the architectural design should be finished and delivered. Also the final report should be done and delivered.

5 Glossary

MoSCoW: a way of setting priorities in software engineering.

Indicator: a way of showing how much you have achieved your goal.