

# Product Plan

Context: Health Informatics

Group: HI1 a.k.a. Geen Naam (Group Number 4)

28-4-2016

## **Group Members:**

Name:	NetID:	Studentnumber:
Jonas Dieterich	jdieterich	4317424
Maiko Goudriaan	maikogoudriaan	4302125
Ruben Meeuwissen	rmeeuwissen	4287150
Louis Sikkes	lsikkes	4298977
Thijmen Jaspers Focks	thijmendewilde	4158393

## **Table of Content**

1. Introduction
2. Product
  - 2.1. High-level product backlog
  - 2.2. Roadmap
3. Product backlog
  - 3.1. Initial release plan
4. Definition of Done
5. Appendix
  - 5.1. User Stories

# Introduction

In the health informatics context we are working on a solution to help people with various sorts of psychoses. To resolve the problem we are using virtual reality therapy. Patients get to interact with a virtual world set in a shopping street by using virtual reality glasses. The therapist gets to track the patient's movements in the virtual world using a 2D map representation that displays all characters, objects and map boundaries. The therapist is able to let the characters interact with the patient by triggering actions such as talking, moving (e.g. following the patient), interacting with other characters/objects in the world and dynamic emotions.

With this program it is possible to analyze the patient's behaviour during exposure to the virtual world. Studies have shown that this modern therapy proves to be effective and is furthermore very cost effective (reference needed from CleVR).

# Product

## High-level product backlog

### Must have:

#### Non-functional:

- Fluent motion (60fps using PC) & (90fps using Oculus)
- Processing time  $\leq 1\text{ms}$
- Difference of occurrences in VR and on 2D map ( $\leq 0.2\text{s}$ )  $\implies$  responsiveness
- Scalability (adding actions, adding large amounts of characters, adding group behaviour, monitoring features e.g. heartbeat)
- Thorough documentation
- List of prescribed tools used for code maintenance
- Patient main focus, not 2D map (helping tool)

#### Unity 3D World

- Use C# for coding
- Unity 3D (5.3.4 or 5.4b)
- Unity profiler for performance related evaluation
- .NET 3.5

#### GUI - VS

- Use C# for coding
- Use XAML
- Use WPF
- Use the MVVM design pattern
- .NET 4.5 (or higher)

#### Functional:

- Show world objects like buildings, trees, TV screen, benches
- Show subsection of the map, like good visible icons and movable part of the map.
- 2-click actions (or 1-click, hover  $\rightarrow$  click) and dragging/moving icons (to trigger walk/follow)
- Field of vision for patient

### Should have:

#### Non-functional:

- Logging of events/actions taking place

#### Functional:

- Change in world screen text (predefined options)

- Path drawing (character follows drawn path)
- Interaction with world objects (bench, police car etc.)
- All character group/list
- Group actions
- Group selection
- Network communication (minimal delay between GUI and VR computer)
- Hover on setting bar, to display essential setting options (for quick access)

### **Could have:**

#### Non-functional:

- Easy/short manual for therapist
- Touch-ready (adaptable for use with tablet)

#### Functional:

- Path drawing
- Mini-map of
- Current emotion displayed on character icon
- Keyboard shortcuts
- Other agents field of vision
- Search objects/people
- Object labelling
- Text-to-speech announcements (in entire VR World)
- Map performance monitor
- Predefined walking routes
- Start/stop and pause environment.
- Predefined actions/ list of actions to carry out for characters
- Varying map start-up settings using init window depending on kind of psychoses (with default)

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

- Voice control
- Avoid use of external packages due to compatibility & licensing issues

## Roadmap

Week	Task
1	Setup initial project, divide roles and responsibilities
2	Get all requirements from CleVr Define the project vision Make a project plan Make an architecture design Make the initial view of the map (prototype)
3	Create a static map Process input from VR world (understanding available actions, list of interactive characters and objects, map boundaries, layout of map)
4	Create static map: <ul style="list-style-type: none"><li>- Virtual characters</li><li>- Virtual objects</li><li>- Logos &amp; design of map</li><li>- Start network communication</li></ul>
5	Make the static map dynamic: <ul style="list-style-type: none"><li>- Moving objects and characters</li></ul>
6	Make the static map dynamic: <ul style="list-style-type: none"><li>- Clickable objects and characters</li><li>- UI for actions</li></ul>
7	Make the dynamic map interactive: <ul style="list-style-type: none"><li>- Execute actions: trigger speech, change emotions</li><li>- Start moving paths for characters</li><li>- Logging of events</li></ul>
8	Interactive map: <ul style="list-style-type: none"><li>- Finish moving paths for characters</li><li>- Follow characters</li><li>- Finish draft of Must and Should haves</li></ul>
9	Interactive map: <ul style="list-style-type: none"><li>- Field of vision of patient</li><li>- Group selection and actions</li></ul>
10	<ul style="list-style-type: none"><li>- Make final version of documents</li><li>- Simple manual for user</li><li>- Implement could haves</li><li>- Finetuning</li></ul>

# Product backlog

## Initial release plan

Week	Milestones
1	
2	The therapist can now see how the map will look like in global
3	The map now has boundaries and action buttons
4	The map now has virtual characters and virtual objects.
5	The map is now dynamic, meaning that it will update frequently and display the current state of the VR world
6	Objects are now clickable and when you click them a UI for actions appears
7	Actions are now executable Character are now able to move
8	Characters are now able to follow the patient
9	It's now possible to create and disband groups
10	Chosen could haves are now implemented

# Definition of Done

Working on a product can continue forever. You can always improve the product. Also you can deliver a product that isn't finished. To guarantee that the product has the right quality we create a definition of done (d.o.d.). We can divide the d.o.d. in three parts: feature, sprint and demo. For a feature to be done the following list should apply<sup>1</sup>.

1. The code should not have any compile errors.
2. The code should not violate any errors from the static analysis tools (e.g. checkstyle).
3. The feature should be implemented conform to the defined requirements.
4. The code should accept the UAT (user acceptance test).
5. All code should have understandable comments.
6. The new code should have at least 75% test coverage.
7. The old tests should still succeed, or correctly edited according to the new code.
  - a. If it is a minor (1 class is affected) change one person should have reviewed it.
  - b. If it is a larger change (2 classes or more affected) two persons should have reviewed it.
8. Relevant documents (e.g. UML or diagrams) are created, when needed.
9. Trello and the retrospective are updated according to the achieved feature.

To know when a sprint is done. This is automatically achieved when the time is over. To determine if it was a successful sprint we can look at the following list.

1. All must haves should be implemented for the sprint.
2. All should haves should be implemented for the sprint.
3. All features should be done specified in the d.o.d. for features.
4. Required documents are created and handed in.
5. No critical errors or bugs are present at the end of the sprint.

Finally we determine when we can release the product to the customers. This means we are satisfied with the product and think that the customers are also satisfied with the product.

1. All must haves should be implemented.
2. All should haves should be implemented.
3. If a must or should have has not been implemented a clear explanation is given, with an appropriate solution.
4. The product is accepted by all the programmers.
5. The product is accepted by CleVR.
6. The product is well documented.
7. The product does not have critical errors or bugs.

---

<sup>1</sup> Waters, K. (2007, July 26). Definition of DONE! 10 Point Checklist  
. Retrieved from <http://www.allaboutagile.com/definition-of-done-10-point-checklist/>



# Appendix

## User Stories

“All characters/people/persons” and “everyone” shall be interpreted as all characters except the patient present in the environment.

The users stories are given in straight letters, while the use-case is given in italic.

### Show actions:

As a user, I want to be able to let characters perform actions, so that I can change the environment.

*Interactive characters or objects can be clicked on (main icon) to list their options. These will be displayed in a circular manner around the icon.*

### Walking:

As a user, I want to be able to let characters walk to a certain destination, so that I can manipulate the environment.

*When you click on a person and drag them to a certain location, without moving through solid objects/map boundaries, the person should walk the dragged route to the given destination.*

### Walking into wall:

As a user, I want to be able to stop a character at an unmovable object, so that I prevent a collision.

*When you click on a person and drag them to a certain location, moving through a solid object like a wall that cannot be interacted with, the person should walk until encountering the object and stop.*

### Following a person:

As a user, I want to be able to let one character follow another throughout the environment.

*When clicking a person p1 and dragging the icon onto another person p2, releasing the mouse button will cause p1 to follow p2 around the map.*

### Stop following a person:

As a user, I want to be able to stop a character from following another character, so that i can influence the patient's experience.

*When a person p1 is following another person p2, clicking on the main icon of p1 will list an action “stop following p2”, which after being clicked will cause p1 to stop following p2.*

#### *Interaction with environment:*

As a user, I want to be able to let the characters interact with the environment, so that the environment becomes part of the experience.

*When you click on a person and drag the icon onto an environment icon (e.g. a bench icon), the person will interact with object (e.g. the bench by walking towards it and taking a seat).*

#### *Changing emotion of person:*

As a user, I want to be able to change the emotion of the characters, so that I can influence the patient's experience.

*By clicking on the emotion icon (top right of main icon), a list of emotions will be displayed. The emotion selected by a click will be the emotion expressed by the person.*

#### *Selecting a group:*

As a user, I want to create groups, so that I can move a group of people in a certain direction.

*Clicking on the 2D map (not on an icon) and dragging the mouse into a certain direction will create a selection window, all characters (not objects) that fall into this window will be selected after mouse release.*

#### *Triggering group actions:*

As a user, I want to let groups of characters perform certain actions, so that i can influence the patient's experience.

*The 2D map will have a sidebar (on the left of the map) containing a group icon that lists all groups. Clicking on the group icon will list all groups, by clicking on one of the groups, all possible group actions and an emotion icon will be displayed. One of the actions can be triggered by clicking on it.*

#### *Changing emotion of group*

As a user, I want to change the general mood of the selected group, so that i can influence the patient's experience.

*The 2D map will have a sidebar (on the left of the map) containing a group icon that lists all groups. Clicking on the group icon will list all groups, by clicking on one of the groups, all possible group actions as well as an emotion icon will be displayed. By clicking on the emotion icon and selecting an emotion all group members will adopt the selected emotion.*

### Changing emotion of all people in environment

As a user, I want to change the general mood of all the characters in the environment, so that i can influence the patient's experience.

*The 2D map will have a sidebar (on the left of the map) containing a group icon that lists all groups. One main group called "All" will contain all characters of the map. By clicking on the group, and thereafter on the emotion icon all possible emotions will be displayed around the emotion icon. By selecting one of the emotions all characters in the environment will adopt the emotion.*

### Triggering speech action:

When clicking on the main icon of a person, an option for speech will be displayed around the icon. Clicking on the speech option will list one/or more of the predefined words/sentences that can spoken. Clicking on one of these predefined options will trigger them.

### Changing object state

As a user, I want to able to change the state of objects in the environment, so that I can influence the patient's experience.

*When clicking on an interactable object icon in the environment (e.g. a TV), the options for this object pop up around the icon. Tapping these options will change the state of the object (e.g. the TV will be turned on or speakers will give an announcement).*

### Adding person to environment

As a user, I want to be able to add a certain character to the environment, so that I can influence the patient's experience.

*The 2D map will have a sidebar (on the left of the map) containing a "new character" icon. Tapping on this will show a list of possible characters. After selecting the character by tapping it, it can be placed in the environment by tapping the target location.*