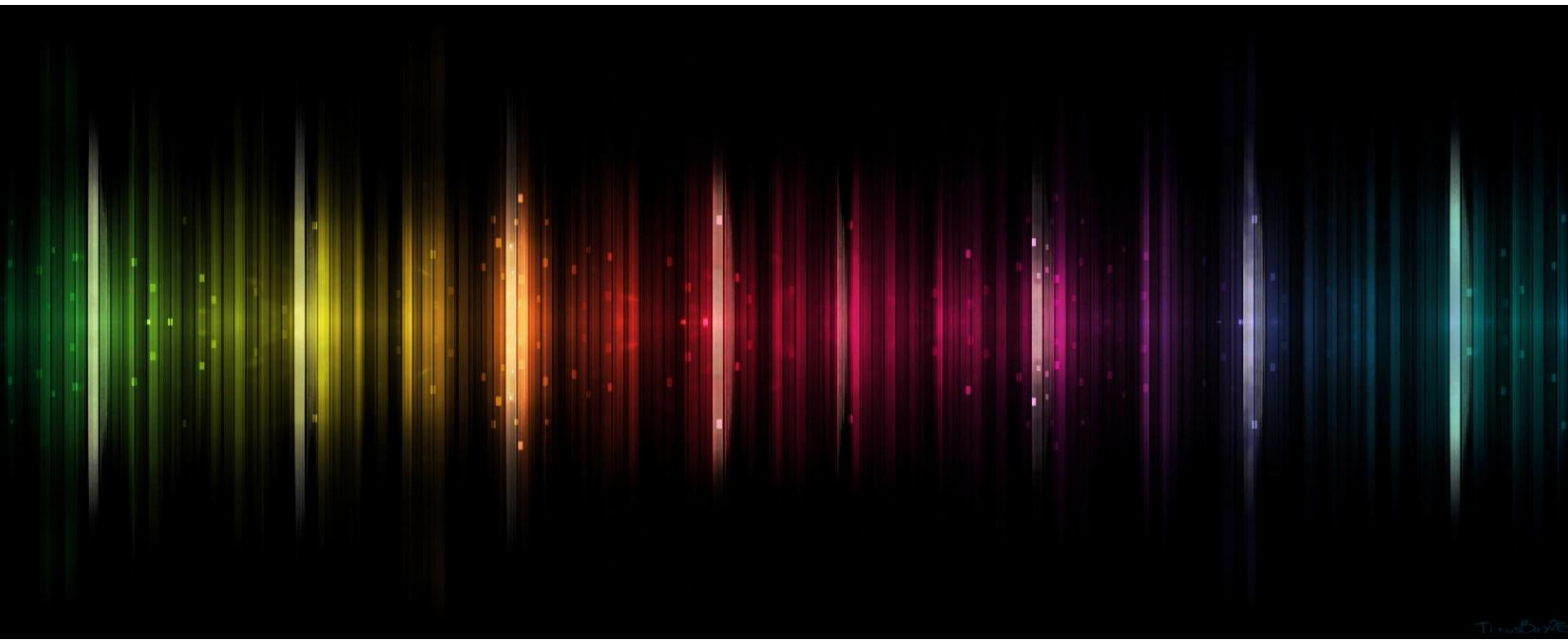
Context Project:
Human Pencils Product Planning



Student names:	Liam Clark	4303423
	Mitchell Hoppenbrouwer	4243889
	Martin Koole	4217500
	Ike Rijsdijk	4294106
	Jorai Rijsdijk	4282264

Course: Contextproject TI2806

Date: 07-05-2015

TU Delft Delft University of Technology

Introduction

This document contains our planning of the development process. We will provide a high-level product backlog to show what kind of features we want to implement. We have also created a roadmap to show when we plan to implement these features and an outline of the sprints. The product backlog will contain user stories of the product. Lastly it will contain our definition of done.

Product

Human pencils will have several features. In this section we will explain what features we want to implement. The features are in our high-level product backlog. It contains features we will implement, but also features we want to implement which might not be possible in this development process. We will also provide our roadmap in which we show our sprint plans and when we want to implement which feature.

High-level Product Backlog

In our high-level product backlog we show the features we want in our product. Some features might not be plausible to implement as of now or are not a priority. In order to clearly define the product features that are crucial to our product, we categorize each one of them according to the MoSCoW method.

Must haves: These are the requirements that must be within the final product. Without them the product will not be able to function correctly.

Should haves: These requirements are still valuable and should be within the product.
However, without them, the product would still be usable.

Could haves: These requirements will only be implemented if there is enough time left, as they will only add extra features to the product.

Won't haves: These are the requirements that probably will not be implemented, yet they can still be taken into consideration whenever the project will be continued later on.

Must Haves	
	Players are able to collaborate with each other.
	Players are able to freely roam the "canvas".
	Players are not restricted by a time limit. (Unless chosen otherwise.)
	Body movement allows the player to draw.
	Players have several colours they can use.
	Easy to join / leave of the game.
	The game is easy to play; minimal learning curve.
Should Haves	
	Players can blend colours when drawn on top of a previously coloured section.
	Eraser option to undo mistakes.

Could Haves	
	Players are able to increase / decrease the transparency of the colour they are drawing with by "pressuring the brush"; lowering their hand.
	Time restraint on certain game modes.
	Bonuses or debuffs can appear on the playing field, which will give the player(s) an advantage / disadvantage.
	Current players are given the task to invite another player to the field.
	In order to counter players sabotaging the game, we need an undo function over a predetermined time interval.
	People can submit their own drawings so other people can recreate them.
	We send / give players their creation after the game has ended.
Won't Haves	
	People are able to paint a large surface together by pointing at each other and walking around.

Roadmap

In this section we have displayed a roadmap for our game.

Sprint 1

- Deciding on a concept
- Defining game definitions
- Creating a Product Vision draft
- Creating an Emergent Architecture Design document

Sprint 2

- Creating a first demo of the game using keyboard input
- Starting with image processing
- Finishing the Product Vision
- Creating the Product Planning

Sprint 3

- Finishing Product Planning

Sprint 4

Sprint 5

- Initial input for SIG

Sprint 6

Sprint 7

Sprint 8

- Creating final report draft
- Finishing the Emergent Architecture Design document
- Final input for SIG

Sprint 9

- Finishing final report

Product Backlog

User Stories

In the following section the features are explained from a user perspective.

Must Haves

● Players are able to collaborate with each other.
As a player I want to paint with my friends/others.
● Players are able to freely roam the “canvas”.
As a player I want to be able to walk across the canvas without painting.
● Players are not restricted by a time limit.
As a player I want to be able to take my time painting.
● Body movement allows the player to draw.
As a player I want paint by using my body.
● Players have several colours they can use.
As a player I want more than one colour I can paint with.
● Easy to join/leave the game.
As a player I want to join/leave the game whenever I want.
● The game is easy to play; minimal learning curve.
As a player I don’t want to read/learn complex mechanics in order to play the game.

Should Haves

● Players can blend colours when drawn on top of a previously coloured section.
As a player I want to combine colours to create new ones.
● Eraser option to undo mistakes.
As a player I want to be able to undo my mistakes.

Could Haves

● Players are able to increase/decrease the transparency of the colour they are drawing by
“pressuring the brush” lowering their hand
As a player I want to increase the pressure on the brush by lowering my hand
● Time restraint on certain game modes
As a player I want to challenge myself by starting a timer
● Bonuses or debuffs can appear on the playing field, which will give the player(s) an advantage/disadvantage
As a player I want to gain an advantage by walking over a pickup
● Current players are given the task to invite another player to the field
As a player I want to increase the amount of players.
● In order to counter players sabotaging the game, we need an undo function over a predetermined time interval

As a player I want to undo what a saboteur has done to my painting

- People can submit their own drawings so other people can recreate them

As a player I want other players to be able to recreate my masterpiece

- We send/give players their creation after the game has ended

As a player I want to keep what I painted.

Won't Haves

- People are able to paint a large surface together by pointing at each other and walking around.

As a player I want to connect with another player by pointing. I want to paint a surface by using the space between us as a brush.

Definition of Done

We have shown our planning for the development process and the features we want in our product. But when is our product done? We will give our 'definition of done' for our features, our sprints and our product.

A feature will be done when all members of our group approve of the feature. As long as the test coverage of the product, with the new feature, is still at least 75%. All tests of the feature pass and it contains no errors. The feature also should be able to merge with the already existing system and not break other features.

A sprint will be done at the end of the week. At the end of the week we will reflect on the sprint. To say a user story/task is done depends on the task. If the task is to implement a feature, the above definition applies. When the task is a document, such as this one, every member of the group reads it, gives feedback and/or makes changes until everyone is satisfied. If the task is to research something, it is the member's own responsibility.

The product is done when it is fully tested. This means that not just the code is tested, but also the gameplay. By user testing the gameplay we know our game meets the expectations and requirements as a playable game. All the must haves from our MoSCoW table have to be implemented. These are mandatory features for our product, without these the game will not be playable. We want the should haves from our MoSCoW table to be implemented as well. These features aren't mandatory and the game will be playable without them, but we find these features will improve the gameplay. The documentation and the code should be ordered and fulfil all the requirements.