Course data

code: IO2060-18

title: Interaction and Electronics (I&E)

date: February, 2019

institute: Industrial Design Engineering Faculty, Delft University of Technology

address: Landbergstraat 15, 2628 CE Delft, The Netherlands

# IO2060-18

# INTERACTION & ELECTRONICS

## COURSE MANUAL 2018/19


Course coordinator

Natalia Romero Herrera - Assistant Professor

Studio dream - available on Mondays: 14:30 - 17:00

n.a.romero@tudelft.nl

Course administrator

Corrie van der Lelie

C-2-070 - available on M/W/F 8:30 -16:00

c.vanderlelie@tudelft.nl


FROM PRESCAUTER.COM (AUGUST, 2013)

# CONTENTS

# COURSE DESCRIPTION

## INTRODUCTION

The course Interaction and Electronics (I&E) introduces you to Interaction Design, the discipline of defining and creating **human interactions with products, services and systems**. It aims to **optimize usability and user experience** by creating meaningful relationships between content, structure, behaviour and appearance of an interactive system (see Interaction Design Introduction on page 7). As member of a team of 4 to 5 students, you will experience a design process in which **the users, their experiences and their context are at the core**. To bring users' needs, preferences, abilities and lifestyle at the centre of the design process, you will learn to:

> **Define the user experience vision and interaction requirements of your product**[1], by applying user-centred research methods and interaction design methods, like interviews, observations, usability testing, personas, interaction scenarios, and user and system interaction flows. In this way, you will be able to create what is called **'the right design'**, a design that fits the uniqueness of a specific target group in a specific context, instead of a "one fits all" solution.

> **Make design decisions based on iterative prototyping activities,** by applying a range of simple (lo-fi) and advanced (hi-fi) prototyping techniques in design iterations to explore, test, and communicate your ideas. The more iterations the more confident you become in making and communicating decisions. In this way, you will be able to get **'the design right',** a design that translate the vision and requirements in a set of convincing design decisions that are appropriate to the target group and context of use.

## THREE DESIGN PHASES

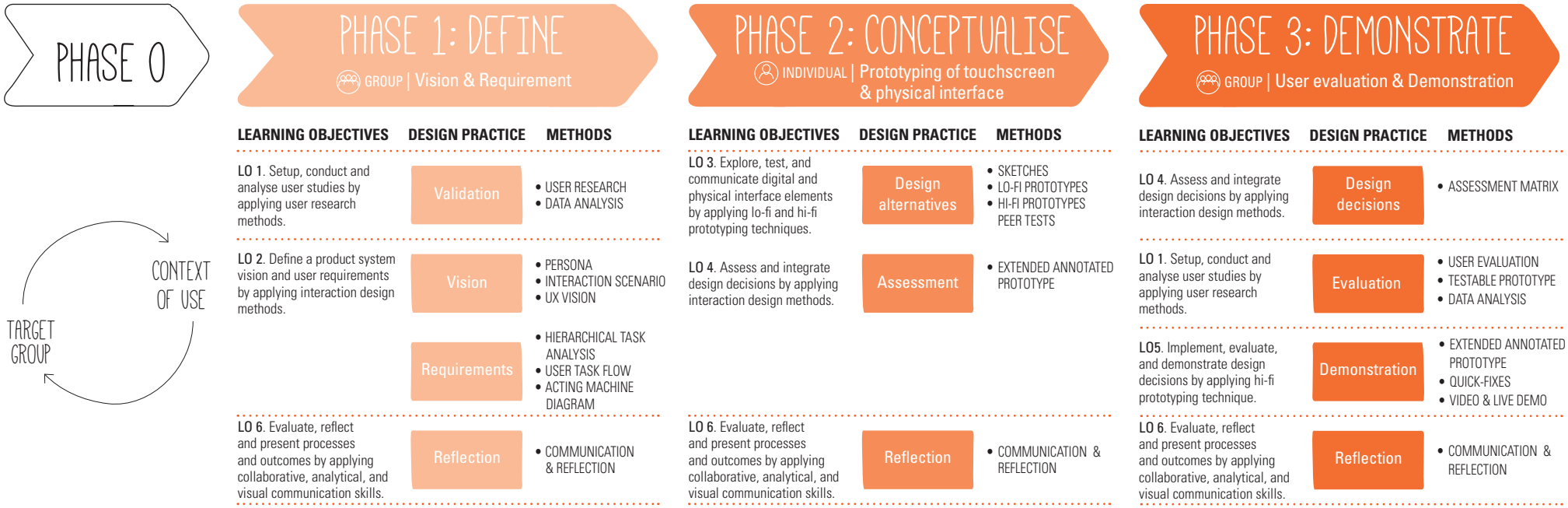
By means of team and individual activities, this course offers you a learning environment to gain hands-on experience on Interaction Design (IxD) methods and tools. You and your team will be actively involved in a complete cycle including three design phases; define, conceptualise and demonstrate (see Figure 1):

**Define** the user experience vision and interaction requirements of the product

> By means of **user research** you will define and validate users' needs and preferences

> By means of **design techniques** you will translate users' needs and preferences into a user experience vision and interaction requirements

*figure 1. Course overview*




1 For simplicity, in this course manual we use 'product' to refer to an interactive product system that encompasses more than one type of interface

**Conceptualise** the interface and interaction details of the product

> By means of applying **design principles** and **lo-fi and hi-fi prototyping techniques** you will iterate to explore, test and communicate several design directions with peers and experts

**Demonstrate** the design decisions of a final concept

> By means of **hi-fi interactive prototyping techniques** and **usability testing** you will evaluate and demonstrate the most promising concept with users, and provide quick fixes and recommendations for future cyles in the design process.

## TEACHING ACTIVITIES AND SUPPORTING MATERIAL

> **Weekly Coach sessions** are scheduled every Wednesday to actively discuss the team and individual progress, steps for the following week and the team performance with your coach. You are expected to come well prepared to each session to make it an effective collaboration.

Two tutorial and two workshops provide you with hands-on practical experience to support you with the following individual activities during phase 2: sketching, as a powerful tool for exploration and communication of ideas from a vision; and prototyping, as a tool to translate ideas into (lo-fi & hi-fi) concepts and to communicate and test design decisions:

> **Axure tutorial** A guideline has been developed for this course to give you quick access to the tutorials provided by Axure (website), which are relevant for this course

> **Arduino & Grove video tutorial** In this set of 1 hour of video tutorials you will learn how to use an electronic prototyping kit (Arduino + Grove) to iteratively prototype hi-fi interactions in a physical interface. The focus is on prototyping dynamic/adaptive feedback: interactive & multi-channel fading techniques.

We urge on you to schedule time in phase 1 of the course to go through both tutorials in preparation for the prototyping activities scheduled during the 2nd and 3rd phases of the course.

> **Sketching workshop** In this 3,5 hours working session you will learn how to use sketching to activate creativity and iteratively translate and communicate your vision into ideas. The focus is on sketching "interactions". The workshop is given by IDE drawing staff.

> **Digital prototyping workshop** In this 3,5 hours working session you will learn how to use a digital prototyping tool (e.g. Axure RP, InVision, Adobe XD) to iteratively prototype hi-fi interactions in a digital interface. The focus is on prototyping transitions within and between screens. The workshop is given by the interaction design agency Hike One.

**Notes**:

- For your sketching activities, we highly suggest you to buy a marker block so you can develop your sketches there. For the workshop, you'll be able to buy single sheets at the day of the workshop (0.20 cent a piece) so you can practice during the workshop assignments
- For your digital prototyping activities, we highly suggest you to use Axure RP as it's the tool we offer further support. If you want to use another tool, feel free to do so, yet make sure that you can prototype hi-fi interactions such as animations, user actions (swipe, tap, pinch, etc.) and system actions (pops up, carrousel, transition effects, etc.)
- For your physical prototyping activities, you will use your own Arduino + Grove kit.

The course offers content material prepared specifically for this course:

> This **course manual** provides you with description of the course, learning objectives and activities, assignments and assessments. It also contains a set of **Methods-cards** to get you deeper into the content of the methods that you will apply and reflect on during the different assignments. For detail description of some methods, the methods-cards often refers to the Delft Design Guide, therefore is suggested to have a copy: it's available in paper back and kindle edition.

> **Template and feedback forms** referred in the methods-cards are available for download in Brightspace under 'Content/Materials/Peer testing'

In addition to the course material, you are invited to deepen your knowledge by reading the books 'Designing for Interaction' and 'Microinteractions' by Dan Saffer.

## ASSESSMENT AND FINAL GRADE

Two summative assessments are used to calculate the final grade of the course. Each assessment is based on a specific rubric (assessment matrix) available in Appendix 7 and on Brightspace ('Content/Course Information/ assesment/Assessment forms'). The rubrics are developed based on a set of criteria to assess the performance of the design practices, in other words the quality of the outcomes, and the performance of applying the involved methods, in other words the quality of the process. The set of criteria used is:

> **Realistic**: The design practice represents the chosen context in an accurate, vivid, and rich manner. The methods are applied in a way that fits the context chosen.

> **Relevant**: The design practice represents a specific and distinctive, unique and original impact for the chosen target group. The methods are applied in a way that provides a descriptive and explanatory view of the chosen context.

> **Convincing**: The design practice provides extensive, fact-based, reasoning and clear presentation of the design decisions. The methods are applied in a scientific (structured, analytical and ethical) way.

> **Consistent**: The design practice provides a logic and coherent integration of previous/new design decisions. The methods are applied in connection and in continuity with previous methods.

Assessments are done by the coach and the grades are cross-checked and benchmarked by coaches in duos. For each assignment, samples of grades are discussed in plenary meetings involving all coaches and the course coordinator. Summative assessments are given in written two weeks after the deadline of the deliverable.

The summative assessment are connected to two assignments:

> Design project (team): this is a team grade on the overall performance and quality of the process and final outcomes of phases 1 & 3.
It's based on the following deliverables:
  - Final report: process and outcomes of vision & requirements, design decisions, final prototype, user evaluation, conclusions, recommendations and reflections)
  - Final interactive prototypes: touchscreen and physical interfaces
  - Final live demonstration and video: interactive demo of the concept and video to contextualise its use

> Prototyping (individual): this is an individual grade on the overall performance and quality of process and final outcome of phase 2. It's based on the following deliverables:
  - Report: that includes sketches, lo-fi & hi-fi prototypes, peer tests plan and outcomes, extended annotated prototypes
  - Source code of prototypes
  - Link to (online) interactive touchscreen prototype
  - Video of physical prototype

**Course grade IO2060-15 =**
Design Project (team grade) x 0.7 + Prototyping (individual grade) x 0.3

Notes:

> Both partial grades must be 5.0 or higher to pass the course.

> If one or more partial grades are below 5.0, that part of the course must be redone in the next year version of the course.

> If the final grade is below 6.0 and equals or is higher than 5.0, the team is entitled to do a complementary assignment. This assignment will be designed by the involved coach and the coordinator.

> If there is clear evidence of unbalanced team performance, the involved coach can decide after previous discussion with the team members and course coordinator, to give different grades to the members of the team.

# COURSE OVERVIEW

Activities in orange text: obligatory participation in instructions, coaching sessions and workshops

| PHASE | WEEK | DAY | ACTIVITY | LEARNING OBJECTIVE |
|---|---|---|---|---|
| DEFINE | 1 | Mon | **Lecture**[1]<br>*Kick-off*/registration<br>**Team work - kick-off**<br>Develop team contract and user research plan | LO.1 |
| | | Wed | **Coach session**<br>Communicate user research plan | |
| | | Fri | **Team work**<br>Conduct user research plan | |
| | 2 | Mon | **Guest lecture**<br>*IxD in practice* by Hike One<br>**Team work**<br>Work on data analysis, vision, requirements | LO.2 |
| | | Wed | **Coach session**<br>Communicate progress & results<br>**Team work**<br>Work on data analysis, vision, requirements | |
| | | Fri | **Team work**<br>Work on data analysis, vision, requirements | |
| | 3 | Mon | **Lecture**<br>*Prototyping* by Tomasz Jaskiewicz<br>**Team work**<br>Conduct vision & requirements peer feedback session | LO.2<br>LO.6 |
| | | Wed | **Coach session**<br>Communicate vision & requirements \| teamwork<br>**Team work**<br>Create user research & requirements report | |
| | | Fri | **Sketching workshop**<br>Practice sketching to explore, refine and communicate ideas<br>**Deliverable 18:00h**<br>Visual outcomes & team - project reflection | |

REMEMBER TO SCHEDULE TIME TO GO THROUGH BOTH PROTOTYPING TUTORIALS

> Axure tutorial

> Arduino & Grove video tutorial

| PHASE | WEEK | DAY | ACTIVITY | LEARNING OBJECTIVE |
|---|---|---|---|---|
| CONCEPTUALISE | 4 | Mon | **Individual work**<br>Sketching | LO.3<br>LO.4<br>LO.6 |
| | | Wed | **Coach session**<br>Communicate sketches<br>**Individual work**<br>Sketching | |
| | | Fri | **Individual work**<br>Lo-fi prototypes | |
| | 5 | Mon | **Individual work**<br>Lo-fi prototypes | LO.3<br>LO.4<br>LO.6 |
| | | Wed | **Coach session**<br>Communicate lo-fi prototypes & tests' outcomes<br>**Individual work**<br>Develop hi-fi prototypes | |
| | | Fri | **Individual work**<br>Develop hi-fi prototypes | |
| | 6 | Mon | **Individual work**<br>Peer test hi-fi prototypes | LO.3<br>LO.4<br>LO.6 |
| | | Wed | **Coach session**<br>Communicate hi-fi prototypes & tests' outcomes<br>**Individual work**<br>Create interface design report | |
| | | Fri | **Team work**<br>Update vision & requirements based on tests outcomes, assessment of individual design alternatives, design decision of final concept<br>**Deliverable 18:00h**<br>Individual documentation of sketches, lo-fi and hi-fi prototypes, peer tests and extended annotated prototypes | |

| PHASE | WEEK | DAY | ACTIVITY | LEARNING OBJECTIVE |
|---|---|---|---|---|
| DEMONSTRATE | 7 | Mon | **Lecture**<br>*User evaluation* by Natalia Romero<br>**Team work**<br>set up user evaluation plan and implementation plan | LO.4<br>LO.5 |
| | | Wed | **Coach session**<br>Communicate integration, user evaluation plan & implementation<br>**Team work**<br>Implement prototype \| pilot user evaluation test | |
| | | Fri | **Team work**<br>Implement prototype \| pilot \| user evaluation test | |
| | 8 | Mon | **Team work**<br>Implement prototype \| pilot \| user evaluation test | LO.1<br>LO.5 |
| | | Wed | **Coach session**<br>Communicate prototype & pilot test<br>**Team work**<br>Implement prototype \| pilot \| user evaluation test | |
| | | Fri | **Team work**<br>Work on data analysis \| identify small fixes | |
| | 9 | Mon | **Team work**<br>Implement small fixes \| prepare live and video demonstrators and final presentation | LO.6 |
| | | Wed | **Coach session**<br>Final presentation and demonstrators | |
| | | Fri | **Team work**<br>Create Final Design Project report<br>**Deliverable 18:00h**<br>Final design project report & presentation<br>Video demonstrator and photoshoot. | |

In weeks 5, 6, 7 and 8 assistance hours are scheduled to support the prototyping activities
both from 10:00-12:00 and 14:00 - 16:00

> Axure RP Pro in SHIFT

> Arduino + Grove in Studio 18 and 19

# INTRODUCTION TO INTERACTION DESIGN

As a result of the popularization of interactive products, services and systems, the field of industrial design has been going through a transformation. Designing interactive products, services and systems, has rapidly become an important part of the domain of industrial design. New technologies such as sensor networks, internet of things (IOT) and big data open up a richness of possibilities in the everyday context, for which dedicated products will have to be designed. To successfully enable interactions that support everyday activities in a meaningful and positive way, industrial designers had to expand their traditional set of tools, methods and techniques with new ones, coming from the emerging field of **Interaction Design (IxD)**.

*"The growing complexity and increased use of digital devices in everyday activities, such as smart alarms to wake us up, microwaves to warm up our food, smartphones to capture and share information with others, etc., has shifted the focus of design as only a form-giving process, to* ***design as a process of enabling interaction****".*

**Interaction Design (IxD)** is the discipline of **defining and creating human interactions with products, services and systems**. It aims to **optimize usability and user experience** by creating meaningful relationships between content, structure, behaviour and appearance of an interactive system. IxD focuses on designing the right technology in addition to designing the technology right. **Designing meaningful interactions requires a deep understanding of a selected target group in a specific context of use**. Interaction designers apply a **user-centred approach** and **use of prototyping techniques** in the design process to support three main design activities:

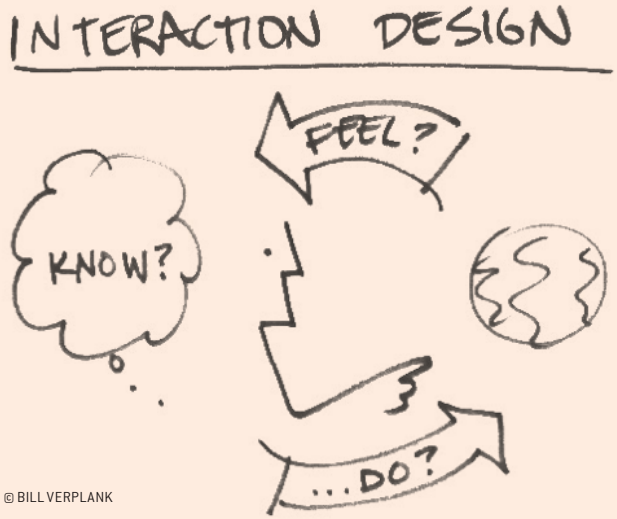



*figure 1. User centered in Interaction Design by Bill Verplank*

1) To define and validate future interaction scenarios and user experience visions
2) To explore and test interaction strategies for the design of unique user interfaces
3) To evaluate concepts with users in their context of use

For this an interaction designer has to combine creative, analytical and technical knowledge and skills coming from a variety of fields such as graphic design, product design, electronics and software design, but also from psychology, ergonomics, and ethnography. Besides, the multidisciplinary environment in which interaction designers develop projects require the ability to work in team groups and deal with complex **team dynamics**. In this manner, several skills and knowledge are potentiated to address complex relationships between users, technologies and contexts.

*"Interaction design goes beyond the engineering challenge of developing smart technologies and puts the attention to the design of meaningful interactions"*

## HISTORY

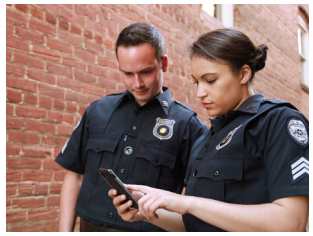
The history of IxD[2]: information and communication technology (ICT) goes back in history since the 60's and 70's with the appearance of wireframe computers and punch cards, where the focus was to operate a machine (users adapt to the machine). In the 80's with the introduction of personal computers and software of text processor for writing and spreadsheets for calculations, the focus shifted to use software. Till now the design was in hand of engineers. From the 90's the focus shifted again to perform (efficiently) a task (e.g. draw a picture, create a budget, etc.) instead of controlling a software. Cognitive psychology becomes central to align human and computer processing.

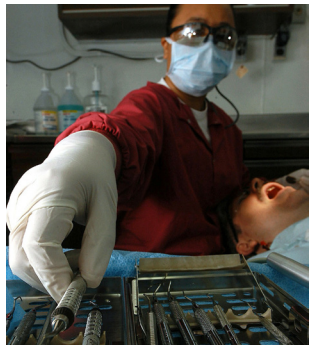
The Interaction design practice as we know it, starts from the 2000's where the focus shifts again to design for the experience to live, to learn, to work, to play (e.g. learn math, run a business, immerse into fantasy, etc.). Two recent waves characterize this present era: 'design to connect' emerged with the appearance of mobile and ubiquitous (everywhere) technologies enabling new forms of communication; and a more visionary 'transformative design' where we no longer adapt to machines but our environment and tools adapt to us as we move through life. IxD expands to design across all types of interfaces, not only screen-based interfaces with buttons and menus, but technology embedded in material objects that interact with us.

2 You can check the history overview of Marc Rettig on http://www.slideshare.net/mrettig/interaction-design-history

> HOW CAN WE MAKE PEOPLE ENGAGE WITH AN INTERACTIVE SYSTEM THAT INTENDS TO IMPROVE THEIR WORKSTYLE?

# PROJECT BRIEF 'ENGAGING IN A HEALTHY WORKSTYLE'





While at work, people can experience - consciously or not - issues related to their health and wellbeing. Competitiveness, personal ambitions, economic pressure, physical strain etc. are only some of the reasons that can create health issues in many jobs in our society. Think of a hairdresser standing, holding a scissors and doing repetitive movements for long hours; or an administrative worker who spends most of the day sitting on a desk; or a conductor of an orchestra who wants to perform at a very high level.
New opportunities arise by the design of informative and interactive technologies that can help people at work to become aware of such issues and when possible change and adjust their habits in such a way that it improves their work physically and mentally. One of the major challenges in such solutions is to design systems that people want to interact with because they are designed considering what they want to know, do and feel when interacting with it.

## PERSONAL HEALTH INFORMATICS (PHI)

Personal Health Informatics systems (PHI)[3] are ecosystems of informative and interactive technologies often including an interactive display (e.g. smartphone) and a personal physical device (e.g. a smart pillow, a smart helmet, a wearable, a smart furniture, etc.). Users can consume and produce meaningful personal information by interacting with digital and physical interfaces. PHI systems collect, analyse and communicate data about people's behaviour and experiences to activate users in managing their own health condition. Users become aware of their current behaviours, are able to reflect on their impact and act towards healthier behaviours.

As earlier mentioned one challenge designers of PHI systems face is to design interactions that a user will experience as meaningful and therefore wants to engage in interacting with the system. The design of generic solutions with the aim to 'one fits all' fail in addressing the uniqueness of users.

To address this challenge, a key aspect in designing PHI systems, is to understand the character/style the system should have when interacting with their users. For example, a system could work as a friendly reminder or as a police agent to keep a user on track with their goals. Therefore an IxD designer should be able to identify the needs, preferences, abilities of a specific target group to define a character/style of the system that has a positive effect on the user.

The character of a system is unique to a specifc user, and it is represented by unique qualities in the interactions of the system (see method-card 'Refining the design brief' p 21 for more details on interaction qualities).

## DESIGN ASSIGNMENT

As a team of interaction designers you are asked to design a PHI system that will help people at work to be more conscious about their physical and psychological wellbeing. The system will:

- provide relevant information in meaningful ways by means of a touchscreen interface
- provide realtime feedback in meaningful ways by means of a physical interface

The challenge is to design information based and feedback features that are meaningful to the user and relevant to different contexts of use.

You will decide on what kind of data the system needs to monitor to provide information and feedback, such as step counts, heart rate, back posture, hydration, sound exposure, etc. Since the focus of the project is to design a unique interaction for a specific target group, the challenge is not in designing an innovative system, but an innovative way of interacting with a system. Thus, a physical activity tracker is a perfect case to work with. (You are free to choose the system, but keep it simple).

The desired PHI system consists of a hybrid interface including a touchscreen and a physical personal device:

- The touchscreen interface offers on-screen interactions & data visualisation. You will consider different levels of engagement with data, from abstract to concrete and detailed information, depending on the needs of your user in different contexts of use
- The physical device offers quick and real-time interactions to provide adaptive feedback. You will implement an adaptive feedback behaviour to fit different contexts of use. You will decide on what sensors and actuators are needed to implement such behaviour.

Your team will be involved in the first cycle of this Interaction Design project. This means that your final design will not be the final product of the project but a solid concept with a working and tested prototype that will be used as input for future cycles. The concept will be implemented as a working prototype that demonstrates how you have implemented a unique **character/style of the system** that is relevant, convincing, creative to your target group and contexts of use, integrated in both the touchscreen and physical device. You will provide a solid argumentation of choices made, supported by several iterative prototyping activities.

3 For more information and examples, please refer to www.personalinformatics.org

| LEARNING OBJECTIVES | DESIGN PRACTICE | METHODS |
|---|---|---|
| LO 1. Setup, conduct and analyse user studies by applying user research methods. | Validation | • USER RESEARCH<br>• DATA ANALYSIS |
| LO 2. Define a product system vision and user requirements by applying interaction design methods. | Vision | • PERSONA<br>• INTERACTION SCENARIO<br>• UX VISION |
| | Requirements | • HIERARCHICAL TASK ANALYSIS<br>• USER TASK FLOW<br>• ACTING MACHINE DIAGRAM |
| LO 6. Evaluate, reflect and present processes and outcomes by applying collaborative, analytical, and visual communication skills. | Reflection | • COMMUNICATION & REFLECTION |

# PHASE 1: DEFINE

GROUP | Vision & Requirement

## GOAL

To design an **engaging interaction** for your target group, you need to define a vision and requirements of the character/style that the system will have to interact with the user. For that you want to know, what the user: > Wants to know? > Wants to feel? > Wants to do?

## WEEK 1 - 3

### DESIGN PRACTICES

You will perform four design practices in this phase:

> **VALIDATION**: First, you want to validate the pre-knowledge and assumptions you defined earlier (in the kick-off activity) by means of user research (see method-cards)

> **VISION**: Based on the data collected in user research, you want to define who's your target group, by answering the above 3 questions and identify the main characteristics of your target group that you will focus your design on, the interactions and the character of your system by means of developing a persona, interaction scenario and UX vision (see method-cards).

> **REQUIREMENTS**: Based on your vision, you want to develop requirements for both the touchscreen and physical interface of your system. The goal of the touchscreen interface is to provide rich information and desired functions around it. For that, you want to analytically describe what people wants to know (goals) and do (tasks), as well as the way people wants to do the tasks (flow) which closely connects to what they want to feel. You will do this by means of a Hierarchical Task Analysis (HTA) and a User Task Flow (UTF) respectively (see method-cards). The goal of the physical interface is to provide real time feedback and desired functions around it. For that, you want to analytically describe how the interface will behave: how many states the feedback has, how does it move from one state to the other, what does it do in a specific state. You will do that by developing an Acting Machine Diagram (AMD) (see method-cards).

> **REFLECTION**: Finally, you want to reflect on the process and outcomes of the design practices that you engaged in. First, you want to explain how efficient and effective you as a team and as an individual developed and applied the methods involved. Focus on explaining why did you do things this or that way. Next, you want to reflect on the quality of the outcomes. Focus on the criteria of assessment explained in page 4: realistic, relevant, convincing, connected. Finally, you want to identify actions that (in a next project or in the next phase) could improve your process and outcomes.

### METHOD-CARDS

User Research Plan | Persona | Interaction Scenario | UX Vision | HTA | UTF | AMD

### DELIVERABLES

Wednesday of week 3 you will present the visual outcomes of your data analysis, persona, interaction scenario, UX vision, HTA, UTF and AMD. You will develop a reflection on the team performance and project process and outcomes.

On Friday (week 3) you will upload:

1. the visual outcomes and reflection as a one page pdf to obtain a formative assessment and feedback on your work.

### SUPPORTIVE ACTIVITIES

LECTURES:
- Kick-off lecture (week 1)
- Guest lecture on IxD in practice by Hike One (week 2)
- Prototyping lecture (week 3)

### KICK-OFF ACTIVITY

After the kick-off lecture on Monday 11 February, you will meet your team and coach for the first time. You will work on a set of activities to get you started on your project and team work.

The activities aim to help you:

1. Setup an optimal environment to work as a team: you will work with the 'Team Canvas' (see method-card) to define shared and personal goals and interests, needs, skills of each member and make clear agreements to work together.

2. Setup a clear focus of your project. As a team you will agree on the PHI system you want to build (see brief with suggestion to work on a physical tracker), and to select the target group and context of use to design for. You want to identify the pre-knowledge you and your team haves of your target group and what do you think they need from a physical tracker system. Discuss the three questions of IxD (see figure 1): what do they want to know, to feel, and to do when interacting with your system.

### METHOD CARDS

Team Canvas | Refining the design brief

| LEARNING OBJECTIVES | DESIGN PRACTICE | METHODS |
|---|---|---|
| LO 3. Explore, test, and communicate digital and physical interface elements by applying lo-fi and hi-fi prototyping techniques. | Design alternatives | • SKETCHES<br>• LO-FI PROTOTYPES<br>• HI-FI PROTOTYPES<br>PEER TESTS |
| LO 4. Assess and integrate design decisions by applying interaction design methods. | Assessment | • EXTENDED ANNOTATED PROTOTYPE |
| LO 6. Evaluate, reflect and present processes and outcomes by applying collaborative, analytical, and visual communication skills. | Reflection | • COMMUNICATION & REFLECTION |

# PHASE 2: CONCEPTUALISE

INDIVIDUAL | Prototyping of touchscreen & physical interface

## GOAL

To translate your design requirements into **unique interactions**, you want to explore what design elements and aspects of your interfaces are the ones your user wants to have. Think of touchscreen design elements like: menu, navigation structure, icons, data visualisations, feedback of interactions (e.g. a loading bar), etc. Think of physical design elements like: sensors, actuators to provide feedback and interaction, labels, etc. Of the chosen design elements you will design the **layout**, ie in what order they will be placed; the **look & feel**, ie how will they be presented; and the interactivity, ie how will they work.

## WEEK 4 - 6

### DESIGN PRACTICES

You will perform three design practices in this phase:

> **DESIGN ALTERNATIVES & PEER-TESTS** (iterations): First, you want to explore different alternatives of the layout, look & feel, and interactivity of your interfaces. For that:

- you want to develop as many **sketches** (see method-cards) as needed up to the point you feel that you've covered a wide space of possibilities to make decisions on. You will assess and select the alternatives that you think better fit your vision & requirements.
- you want to validate your choices by answering the question: **how do the layout and look & feel fit your envisioned user experience and system behaviour?** You will do this by means of lo-fi prototyping & peer tests (see method-cards).
- you want to refine unique aspects of your interactions. You are not able yet to test a complete concept, but you want to experiment what are the interactive aspects that define a unique interaction by answering the question: **how do aspects such as speed, frequency, visual effects of a blinking LED or changes in data visualisation, shape the system behaviour to better fit your envision user experience?** You will do this by means of hi-fi prototyping & peer tests (see method-cards).

> **ASSESSMENT**: Second, you want to communicate and assess your design decisions during the development of lo-fi and hi-fi prototyped interactions by answering the following questions: a) what are the relevant design elements and aspects chosen and how do they fit your vision and requirements? b) how did peers experience these choices? You will do this by developing extended annotated prototypes (see method-cards).

> **REFLECTION**: Finally, you want to reflect on the process and outcomes of the design practices that you engaged in. First, you want to explain how efficient and effective you as an individual developed and applied the methods involved. Focus on explaining why did you do things this or that way. Next, you want to reflect on the quality of the outcomes. Focus on the criteria of assessment explained in page 4: realistic, relevant, convincing, connected. Finally, you want to identify actions that (in a next project or in the next phase) could improve your process and outcomes.

### METHOD-CARDS

Sketching | Lo-fi & hi-fi prototyping | | Implementation | Peer testing | Extended annotated prototypes

### DELIVERABLES

On Friday (week 6) you will upload:

1. Collection of sketches and extended annotated prototypes for the lo-fi and hi-fi prototyping activities, as as well as an individual reflection of your performance, process and outcomes all in one pdf file.
2. Source files of your prototyped interactions (e.g., the .rp file for Axure RP Pro and the .ino file of Arduino).
3. Weblink to the prototyped interactions of your touchscreen (in the pdf).
4. A short video that shows the prototyped interactions of the physical interface.

### SUPPORTIVE ACTIVITIES

**WORKSHOPS:**

- Sketching workshop on week 3.3 Friday (see p5 for details)
- Digital prototyping workshop on week 3.4 Wednesday (see p5 for details)

**VIDEO TUTORIAL:**

- Physical prototyping video tutorial available from week 3.3. To do this tutorial, you will pick up your I&E video tutorial kit at the PMG Digital lab on Friday of week 3.3.

**PROTOTYPING ASSISTANT HOURS:**

- For both touchscreen and physical interfaces, on Fridays in weeks 3.5 - 3.8 between 10-12 and 14-16. Check Brightspace for location.

| LEARNING OBJECTIVES | DESIGN PRACTICE | METHODS |
|---|---|---|
| LO 4. Assess and integrate design decisions by applying interaction design methods. | Design decisions | • ASSESSMENT MATRIX |
| LO 1. Setup, conduct and analyse user studies by applying user research methods. | Evaluation | • USER EVALUATION<br>• TESTABLE PROTOTYPE<br>• DATA ANALYSIS |
| LO5. Implement, evaluate, and demonstrate design decisions by applying hi-fi prototyping technique. | Demonstration | • EXTENDED ANNOTATED PROTOTYPE<br>• QUICK-FIXES<br>• VIDEO & LIVE DEMO |
| LO 6. Evaluate, reflect and present processes and outcomes by applying collaborative, analytical, and visual communication skills. | Reflection | • COMMUNICATION & REFLECTION |

## GOAL

To develop a **preliminary integrated concept** that showcase your **unique interactions** and **intended character/style of your system**, you want to prototype and test with real users the main effects of your design. You will focus on testing the usability and intended effect of your concept and in the end demonstrate the unique interactions that you design for your target group.

## WEEK 7 - 8

### DESIGN PRACTICES

You will perform four design practices in this phase:

> **DESIGN DECISIONS**: First, you want to develop an integrated concept by assessing the prototyped interactions developed by each member of the team. From this assessment you will select the interactions that better support your vision & requirements and decide on how to integrate them into one concept. You will do this by developing an assessment matrix (see method-cards).

# PHASE 3: DEMONSTRATE

GROUP | User evaluation & Demonstration

> **USER EVALUATION**: Second, you want to evaluate whether the unique interaction of your concept, in relation to the chosen design elements and aspects of the interfaces, have the intended effect on your target group. For that you will implement a testable prototype, develop and conduct a usability study, and data analysis (see method-cards).

> **DEMONSTRATION**: Third, you want to identify and implement the quick and easy fixes that will improve the intended effect. Then, you want to communicate and assess your design decisions during the development of the integrated concept by answering the following questions: a) what are the relevant design elements and aspects chosen and how do they fit your vision and requirements? b) how did users experience these choices? You will do this by developing an extended annotated prototype (see method-cards). Finally, you want to communicate to a broader audience the final preliminary concept of unique interactions by means of a live interactive demonstrator and a video that demonstrates the interactions in context.

> **REFLECTION**: Finally, you want to reflect on the process and outcomes of the design practices that you engaged in. First, you want to explain how efficient and effective you as a team and as an individual developed and applied the methods involved. Focus on explaining why did you do things this or that way. Next, you want to reflect on the quality of the outcomes. Focus on the criteria of assessment explained in page 4: realistic, relevant, convincing, connected. Finally, you want to identify actions that (in a next project or in the next phase) could improve your process and outcomes.

### METHOD-CARDS

Assessment matrix | User evaluation | Testable prototype

### DELIVERABLES

On Friday (week 9) you will upload:

1. Written & visual documentation of the project. In one pdf including an introduction to the context of the project, the user research plan, the updated vision and requirements, the assessment of alternatives, user evaluation plan, the extended annotated prototype and quick fixes. It will end with a conclusion and recommendation, as well as a reflection on team performance and project process and outcomes.
2. Video demonstrator
3. Source files of touchscreen and physical prototypes
4. Photoshoot of the system: one close up picture of the physical device and one picture of the integrated system in context of use.

### SUPPORTIVE ACTIVITIES

**PROTOTYPING ASSISTANT HOURS:**

- For both touchscreen and physical interfaces, on Fridays weeks 3.5-3.8 from 10:00 -12:00 and 14:00-16:00

> Touchscreen/digital prototyping in SHIFT

> Physical prototyping in Studio 18 and 19

# COURSE POLICY

## ENROLMENT & TEAM FORMATION

To confirm your participation in I&E 2018/19, in accordance with the Teaching and Examination Regulations 2018-2019 (TER) **students must be present at the kick-off lecture and kick-off workshop on Monday 11 February from 13.45 to 17.30h.**
The group composition will become final on Tuesday at and will be annouced on Brightspace. So, check to see if you are still in the same group.

## OBLIGATORY ATTENDANCE

**All activities presented in red in the course overview of page 6 are obligatory.**
Coach sessions are scheduled on Wednesday, and two prototyping workshops on Friday and Wednesday of weeks 3 & 4 respectively. .
Depending on whether you are in group A or B you will have your coach sessions on Wednesday morning or afternoon. **If you are planning any extracurricular activity that conflict with the course activities, you are not entitled to enrol in this course (without exceptions).**

## PERSONAL CIRCUMSTANCES

In case you are sick and cannot attend a scheduled activity, **you are responsible for reporting sick** by sending an email to: ziekmeldingen-io@tudelft.nl not later than 10 AM of each day of sickness.
In case of longer absence[4] that prevents the student to participate in the course as expected, **the student is responsible for contacting** the course coordinator and an academic counsellor to seek for assistance.

## FRAUD POLICY

The course deliverables will be randomly checked by the course coordinator for possible fraud. In case fraud is potentially present, the coordinator will proceed according to the rules of the IDE BSc with regards to fraud. Coaches can also detect possible fraud at the level of team work. In this case, the coach will notify the coordinator who will proceed according to the rules of the IDE BSc.

## LATE WORK POLICY

In case of a deliverable been submitted in BrightSpace after the deadline, it will be marked late. The coach assessing the deliverable will consider this in the assessment of the work. In case a deliverable could not be submitted in BrightSpace due to the assignment being already closed, the deliverable will receive a 1 and no feedback will be given. In this case, you are expected to contact your coach and explain the situation as soon as possible. Depending on the circumstances a special arrangement could be discussed and communicated to the course coordinator for approval.

**Delivered materials should be double checked to guarantee easy accessibility and readability. A late, or a non-readable deliverable will not receive feedback..**

[4] During the individual activities between weeks 4 to 6, a longer absence is one or more weeks. For the other weeks, a longer absence is two or more weeks

# APPENDICES

## 1. PLACE OF THE COURSE IN THE IDE BSC CURRICULUM

### ROLE OF USERS IN DESIGN

Before this course, you have gained knowledge and skills in involving users at different phases of your design process: 1) you know how to identify user's physical characteristics as requirements for design (e.g. *Mens en Product*); 2) you know how to evaluate a physical model of a design concept with the target group (e.g. *PO2*); you know how to evaluate the ergonomic and aesthetic aspects of a design concept with the target group (e.g. *PO1*). In this course, you will extend this knowledge and these skills by **involving users, their experiences and their contexts at different stages in the design process**.
In this regard, Interaction designers compared to traditional designers develop their concepts with users instead of for user.

### ROLE OF PROTOTYPING IN DESIGN

Before this course, you have developed physical models and prototypes as representations of your final concept for evaluation purposes (e.g. *PO2*) and analytical prototypes to predict behaviour (e.g. *Product dynamics*). In this course, you will extend your knowledge and skills by **applying prototyping techniques for the purposes of exploring, evaluating and communicating design ideas. Prototyping becomes a way to provide solid argumentation to the process of making design decisions.**

### ROLE OF ELECTRONICS IN DESIGN

In the courses *Product in action* and *Product dynamics,* you have learned how to model and assess the different functional elements as well as the dynamic behaviour of a product. In this course, you will extend your knowledge and skills by **experiencing ways to quickly implement rich behaviours in interactive prototypes** and test the effect on user experiences.

## 2. CHOICE OF PROTOTYPING TOOLS TAUGHT IN I&E

### DIGITAL PROTOTYPING TOOL

Axure RP Pro (www.axure/com) is a digital prototyping tool, well-known in the world of web interface design, and providing support with latest updates as a prototyping tool for mobile interfaces. It has been chosen for this course because it provides an excellent online support to learn and use the tool. In addition, the course offers a self-tutorial, a prototyping workshop and assistant hours to support prototyping with Axure. All TUD computer rooms provide access to Axure RP Pro 8 software. If you want to install Axure in your own computer, a one-year license can be requested at their official website (this may take a couple of weeks). However if you decide to use a different tool, the course is in transition to also support Invision. This year, we have arranged free-license to use Invision by registering with your TUD email account using this link and this code: 56-73-13-19. In addition, the company has provided us with a video demo to help you get familiar with the tool. No assistant hours would be offered during the course.
To get your more acquainted with the diversity of digital prototyping tools available we recommend you to check these two popular overviews: www.cooper.com/prototyping-tools & www.prototypr.io/prototyping-tools

### PHYSICAL PROTOTYPING TOOL

You are expected to be already familiar with Arduino (www.arduino.cc) which is one of the most known and used boards to get started with electronics and coding. In this course, you'll be introduced to Grove System (wiki.seeed.cc/Grove_System/). Grove provides you with a base shield and various modules (e.g. a button, or a heart rate sensor) with standardized connectors, to assemble interactive projects using ready-to-use electronic components. The Arduino + Grove system is a physical prototyping toolkit that is chosen to support you in rapid prototyping using Arduino, where you can abstract from the breadboard details and focus on prototyping system behavior and user experience.

It is expected that team members will contribute with their own Arduino+Grove kit to support the physical prototyping activities. In that case, as a team you should have a rich set of sensors and actuators. Since temperature and tilt switch sensors are not part of the basic kit, as a team you can request one of each at the PMB Digital Lab. If you want to have more units of one of the components in the list check if that is possible at the PMB Digital Lab. In case the PMB Digital Lab cannot provide you with your request (or your request is not in the list) then you will have to buy the component yourself.

**CONTEXTUAL SENSORS**

> Light sensor
> Temperature sensor

**INPUT SENSORS**

> Switch (P)
> Tilt switch
> Touch Sensor sensor
> Rotary angle sensor

**ACTUATORS**

> RGB color LED
> Buzzer
> Vibration Motor

# APPENDICES

## 3. TEACHING AND SUPPORTING ROLES (WHAT TO EXPECT FROM THE TEACHER(S))

The course offers you different levels of support to get the most out of the above learning activities. The following teaching and supporting roles are provided:

> **Course coordinator:** her role is to ensure an effective and stable learning environment during the course. The coordinator is available to hear out comments on the project/course of any kind. Do not hesitate to contact the course coordinator. For personal circumstances, please contact the student advisor.

> **Course administrator**: among several administration tasks, she takes care that students get access to course materials, announcements, changes in schedules, etc. In case you are having questions on these issues you can contact her for support.

> **Lecturers**: their role is to provide you with relevant theoretical information that you can apply in your course, as well as inspiring and practical information to understand the impact of the current and future role of IxD practitioners. During lecture hours, lecturers are there for you to ask questions or open relevant discussions in topics related to this course.

> **Design project coaches**: your team will be assigned to a design project coach. She/he will support you in conducting your design activities by providing you with weekly feedback and pointing you to relevant information. She/he will also support you in gaining skills in team dynamics. She/he is responsible to grade your individual and team work and provide you with feedback. In case you experience conflicts with your team, or with the activities you are expected to perform, you can discuss this with your coach.

> **Axure and Arduino+Grove advisors**: for the touchscreen and physical prototyping activities with Axure RP Pro and Arduino+Grove respectively, assistant hours are organized in which student assistants are available to answer your questions or provide you with the right pointers to solve a problem. You can find the hours and locations in the schedule on Brightspace.

## 4. PERSONAL HEALTH INFORMATICS
### COMMERCIAL EXAMPLES

Physical activity trackers (e.g. Fitbit, Nike+) mainly focus on tracking the number of daily steps as well as high intensity activities such as running, swimming, and cycling. It helps you to set daily goals and achieve them.

> Sleep patterns (e.g. NightBalance, Somnox) mainly focus on tracking your quality and rhythm of sleep. It helps you to find ways to fall asleep easily and get a high-quality sleep.

> Medical interventions (e.g. Medicine men, NaturalCycles, Fysio24) mainly focus on tracking your health condition. It helps you to follow a specific and personalised medical treatment.

> Mood trackers (e.g. Affectum) mainly focus on monitoring your psychological wellbeing in a specific context. It helps you to reflect and improve your performance based on your mood

# 5. USER ENGAGEMENT SCENARIOS AND ADAPTIVE FEEDBACK

## USER ENGAGEMENT SCENARIOS

**(GOUVEIA, KARAPANOS AND HASSENZAHL 2015[5])**

To design and measure user engagement of an application, the unit "user session" is used. A user session is defined by the moment a user opens an application, until the device (e.g. phone, tablet) is locked or the application is closed. Three types of user sessions are defined, characterised by the time duration of the session.

Glance sessions are brief interactions, in which users check for instance information on the lock screen and then turn the screen off or let the phone timeout with no additional actions or inputs. In these sessions the user wants to gain awareness by accessing quick overviews of data.

Review and engage sessions are characterized by a high number of interactions with richer feedback. They involve for instance, exploring an ongoing days' contextual feedback, while exploring textual messages.

Using these definitions, you can design for three different user engagement scenarios. Each scenario offers a different level of complexity of interaction and information. Below a summary of the three scenarios you will design for:

- Glance: brief, 5-sec sessions where users call the app to check their current activity levels with no further interaction, for example, information displayed in the lock-screen could support glance
- Review: sessions that last up to a minute where the users check at least one information piece in more detail, for example, to compare the current progress with same time progress yesterday
- Engage: sessions that last over a minute where users check for extended period of time different pieces of information in detail, for example, to visualize historical data of the last week and months and how that compares with peers.

## ADAPTIVE FEEDBACK

Interfaces with adaptive feedback provide rich experiences to user by adjusting its behaviour in relation to the realtime needs of the user and the context around. For example, a smart clock alarm monitors people's sleep cycles and adjusts the time to fire the alarm depending on the sleep cycles. Adaptive feedback could allow the alarm to choose different modes depending on the conditions of the environment: if the room is dark then the alarm fire could use colour lights, whereas in a light environment it could use sound; if it's noisy outside the room it could use a louder sound than in a quiet environment. The context can also be defined by the user's behaviour: if the user is not reacting to the alarm different modes are fired starting from subtle ones (soft vibration for example) to more obvious and explicit ones (ambient light, direct light, ambient sound, direct sound, etc.). These choices will differ and depend very much according to the personal characteristics of user (skills you gain in Mens en Product).

5 Source: Gouveia, R., Karapanos, E. and Hassenzahl, M. (2015) How do we engage with activity trackers? A longitudinal study of Habito. In proceedings of conference UbiComp '15, ACM press. Available in BrightSpace ('Content/Materials')

# APPENDICES

## 6 METHODS-CARDS

# METHOD-CARD

## TEAMWORK [1/3]

### INTRODUCTION

Teams can offer "greater adaptability, productivity, and creativity than any one individual can offer and provide more complex, innovative, and comprehensive solutions to organisational problems" (Salas et al., 2005). Also in design projects, complexity has increased and time to market has decreased. Therefore, companies consider teamwork key when dealing with the required efficiency and the increasing number of complex activities.

As the designers of tomorrow, it is therefore essential for design students to become comfortable with working in teams, and to learn how to effectively work in teams.

When learning about teamwork, an important distinction needs to be made between **teamwork** and **taskwork**. **Taskwork** involves the performance of specific tasks that team members need to complete in order to achieve team goals. […] Conversely, **teamwork** focuses more on the shared behaviours (i.e. what team members do), attitudes (i.e., what team members feel or believe), and cognitions (i.e., what team members think or know) that are necessary for teams to accomplish these tasks. (MSc IDE Thesis Dorian Kingma, 2017).

### DESCRIPTION

During the activities planned on the first Monday of the course, you are asked, as a brand new team, to perform the following activity to setup an optimal environment to work as a team.

You will use an adapted version of Team Canvas Basic, which is a strategic framework that helps team members to align on common vision. Based on experience with startups and creative groups, it is made to smoothly start collective projects, let people learn about each other and accumulate enough momentum to get going.

Team Canvas Basic consists of 3 key areas:

1. What the team is: roles and goals
2. Why the team is doing what it's doing: purpose and values
3. How the team is going to achieve what it needs to achieve: rules and activities

To start this activity, as a team appoint a facilitator to chair the session. The 'Running the session' is directed to the facilitator (see 'Guidelines to act as a facilitator').

Note: these activities are independent to the content of your project. So for now, forget about the outcomes of the project. Instead focus on the learning objectives of the course and what do you want to learn/achieve in this course.

### MATERIAL

- Team Canvas Basic recreated on a big enough piece of paper (e.g. flipchart paper or A0/A1)
- Blocks of sticky notes, one for each participant, different colors
- Sharpies or pens to write on stickies (use thick enough pens so everyone can see the writing)
- A device with a timer function

### GUIDELINES TO ACT AS A FACILITATOR

1. Introduce the team canvas as a tool to align the team members and get better at understanding goals, roles and values of your team.
2. Go through each step with the team, making sure you ask the questions for each segment. Encourage people to write their answers on stickies and talk about them with the team. Make sure to agree on all fields.
3. Use timer for each step to apply some time pressure so the team really focuses on getting to the point in each section discussion.
4. If some conversations take rather long time or seem to touch upon bigger issues, consider parking those questions them during the Team Canvas session and planning a separate meeting to address them specifically.

As a facilitator of the session, you might be asked something like this: 'How are we supposed to answer this question? What is that you expect us to say here?', etc. It is important to understand that The Team Canvas creates context for the team, rather than content, and therefore all answers are valid. Gently reply to such questions: 'How would you answer if you knew? What do you think the answer should be?'

Team Canvas Basic is working well with short-term projects and for the purpose of kicking off a new team. If you want to align on common vision and resolve conflict, or create great team bonding for a longer project, consider using Team Canvas Complete.

### RUNNING THE SESSION

Today you will conduct a 30-35 min session with your team to fill in the canvas. The session consists of the following 6 steps:

#### 1. GOALS [5 MINUTES]

Ask the team members to agree on common goals and mention their personal goals in relation to the learning objectives of the course.

Questions:

# METHOD-CARD

## TEAMWORK [2/3]

> What we as a group really want to achieve? What is our key goal that is feasible, measurable and time-bounded?
> What are our personal goals that we want to share with each other?

Examples:

> *Having very productive team meetings and proactive communication with our coach.*
> *Be open to each other and support and challeng each other when needed .*

### 2. ROLES & SKILLS [5 MINUTES]

Ask people to put their names on stickies, as well as the roles they would like to take. Since this is a learning environment, ask people to see if they can find a challenging role. A team is a safe environment to give feedback and learn from each other. If a person has multiple roles, use separate post-its. Questions:

> What are our names?
> What are the roles we have in the team?
> How are we called as a team?

Examples:

> *Max: team leader; Marie: reporting*
> *Name of the team: Super X*

### 3. PURPOSE [10 MINUTES]

Ask the team to go one step beyond their common goal, and ask them why they do what they do. Focus the discussion in your goal as IDE student, and as I&E student and member of the team.

Questions:

> Why are we doing what we are doing in the first place?
> What is something more important, which makes us pursue our common goal? Consider everyone's options. For some may be to learn how to use prototyping skills, for others to get a sufficient grade.

Examples:

> *learn how to use prototyping to design interactions.*
> *learn how to conduct user studies in real contexts.*

### 4. VALUES [5 MINUTES]

Ask the team what are the core values - the most important principles - that they want to share within the team. The team should agree on values, so everyone accepts the final set.

Questions:

> What do we stand for?
> What are guiding principles?
> What are our common values that we want to be at the core of our team?

Examples:

> *Trust*
> *Creativity*
> *Quality*
> *Transparency*
> *Mutual understanding*
> *Equality*
> *Respect*

### 5. RULES & ACTIVITIES [10 MINUTES]

Ask the team to agree on common rules and activities. Rules are group beliefs about how members should behave in a given context. Think of this as an outcome of the previous sections: a concrete set of rules and activities they want to implement.

Questions: See table below.

| MEETING RULES | WORKING RULES | COORDINATION RULES | COMMUNICATION RULES | CONSIDERATION RULES | CONTINUOUS IMPROVEMENT RULES | REWARDS | SANCTIONS |
|---|---|---|---|---|---|---|---|
| • When will the team meet?<br>• Are meetings mandatory?<br>• Is being late acceptable? | • How will decisions be made?<br>• How will team members give and receive criticism?<br>• How independently should we work? | • Who will lead the team meetings?<br>• Who will allocate assignments?<br>• Who is responsible for keeping the meetings on track? | • What means will team members use to communicate between each other?<br>• How will team members communicate outside the meetings?<br>• How quickly should a team member respond to a communication?<br>• Are all team communications public?<br>• How should a team member communicate an illness? | • Are side conversations appropriate?<br>• How will we handle disagreement? | • How will we track the team's progress?<br>• How will we evaluate the work of others?<br>• Do certain outcomes trigger a discussion? | • How will we mark achievements? | • How will we respond to a team member failing to meet obligations?<br>• How will we respond to a team member violating group rules? |

Examples:

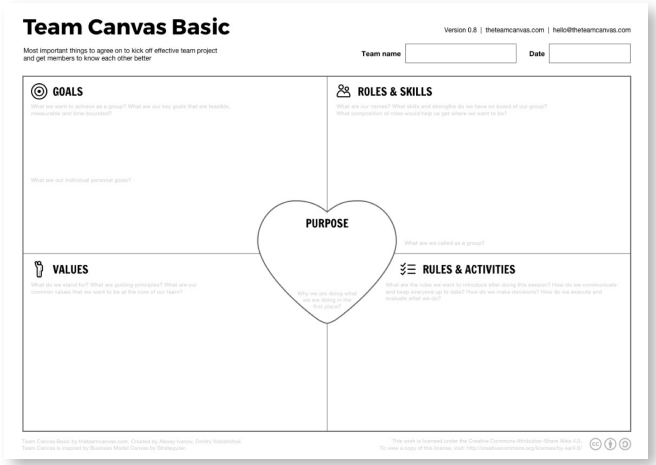
> *Keeping things within group confidential*
> *Weekly status updates*
> *Communication over Slack + Skype for calls*
> *Dinners together every second week (Max as organizer)*
> *Workday: starting from 9 to 10, meetings start at 10*
> *Keeping workday to 8 hours, except when it's needed to shorten it a bit towards more*

### 6. WRAP UP [5 MINUTES]

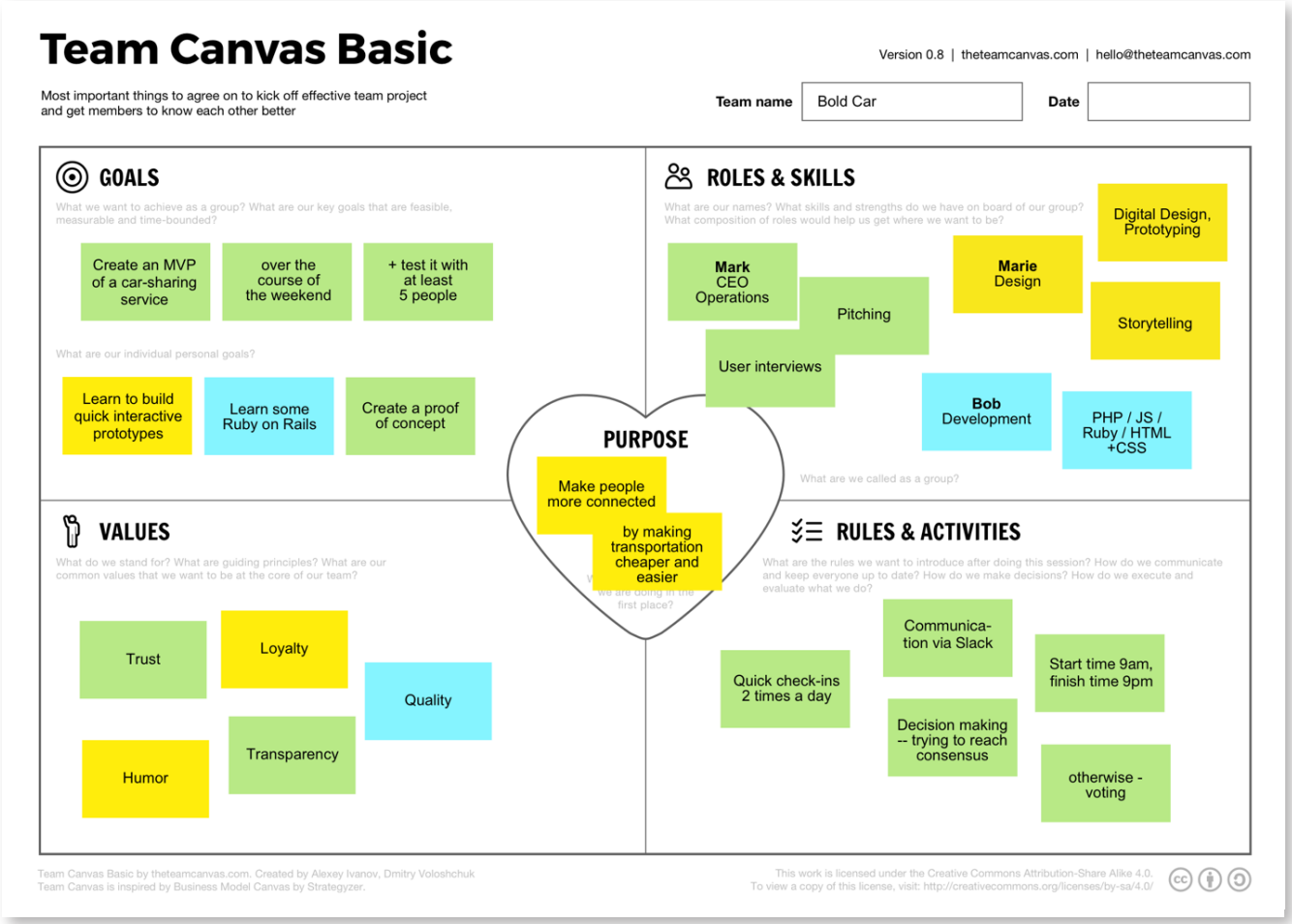
As you close The Team Canvas workshop, ask the team members to tell about one single most important insight that they gained during the workshop.

It is highly recommended to revise the canvas agreement if a new member joins the team and on several occasions during coach meetings and teamwork. In this course it is suggested to do this revision in weeks 3.3, 3.6, and 3.8.

### BASIC TEAM CANVAS TEMPLATE




### BASIC TEAM CANVAS EXAMPLE




### REFERENCE MATERIAL

> Basic canvas instruction
> More complex canvas templates

# METHOD-CARD

## REFINING THE DESIGN BRIEF [1/2]

### INTRODUCTION

The project brief gives you the starting point of a design assignment. In this case, the brief introduces you to the PHI systems and the need to design for wellbeing at work. Once you get familiar to the brief, you will make it your own by defining the specific target group and context of use you want to design for. Afterwards, you will list the pre-knowledge you already have about your target group and the interaction with a PHI system. This knowledge will lead you to envision the character/style of the product that will evoke a unique experience to your users when interacting with the system. This will give you a clear focus to define and conduct your user research.

Make sure that the user group and target group **fits your interest** and at the same time **it is accessible** since you will have to interview and observe your target group in their working environment. So, choose wisely.

### DESCRIPTION

By selecting the target group and context you will define the **who, what, where, when and why** of your future application. Start with a short **brainstorm session** in which you explore **accessible** user groups, and moments of use to explore the possible needs of that user group for the PHI system that you are asked to design. Ask yourself the following questions:

> What group of users may benefit from a PHI? Why?
> What needs/goals such user group may want to achieve with a PHI system? Why?
> In what specific situations and/or locations the user group may want to interact with a PHI system? Why?
> What information the user needs from the touchscreen and what feedback from the physical interface?

To assume what is the character/style of the system, start with a short **brainstorm** on how do you think your user wants to feel when interacting with the system, and then envision what qualities of interaction the system. For example, you assume that your users want to feel calm when interacting with the system. Then you want to connect this character with the design of unique interactions, and you do that by assuming certain qualities of the interaction. For example, to achieve a calm character the interactions that the system supports could be slow and simple. In another example, if you assume that the users want to have a feeling of control, then the qualities of the interaction could be stepwise and informative. For guidance, use the interaction qualities vocabulary developed by Lenz et al. [1] (see Figure 1 on the next page)

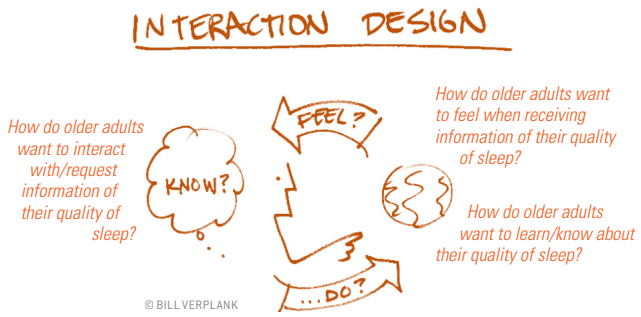
Finally, you want to summarise your assumptions regarding the character of the system and the interaction qualities by answering the IxD questions introduced in page 7 (Figure 1):

> What the users want to know when interacting with the system?
> How the users want to feel when interacting with the system?
> What users want to do when interacting with the system?

### EXAMPLE: OLDER PEOPLE WITH DIFFICULTIES TO FALL ASLEEP

Who: older people that are still active and live by their own
What: they have a poor quality of sleep
Where: living room (couch)
When: half an hour before bed time / at breakfast in the morning
Why: having time to prepare to go to bed in the evening and to reflect on the quality of their sleep may help older people to understand when they do sleep good and when not

Next, the how questions:




You could assume that older adults want to feel guided and secure. Looking at the list of interaction qualities, you could envision your system to be stepwise and constant. This will set up how to conduct your research to validate your assumption and initial vision of your concept.

| Experience | Interaction attributes | | Experience |
|---|---|---|---|
| Esteem, focus on the interaction itself, significance of the present moment, relaxing, calming, accuracy, care, appreciation of interaction/product | slow | fast | Animating, stimulating, activating, efficiency, focus on instrumental goal of interaction, expression of willpower. |
| Ritualization, every step is meaningful, rewarding, emphasis on progress and advance of the process, approaching a goal step by step, clear structure, being guided through the process | stepwise | fluent | Autonomy, continuous influence, power and right to change what's happening at anytime of the process, no barriers, fluent integration in running process, spurring instead of interrupting |
| Instant feedback makes own effect experiential, competence, feeling of own impact creates a feeling of security, you see what you do, makes immediate correction possible, nothing in between, you experience what you do, increase of competence, the instant feedback creates a feeling of recognition. | instant | delayed | Emphasizing the moment of interaction, creating awareness. Centering on the interaction itself rather than its instrumental effect. |
| Influence by intuition, control | uniform | diverging | Unusual, unnatural, amplified, grasping for attention |
| Creates feeling of security | constant | inconstant | Liveliness, suspense, you can't adapt yourself to it, unreliable, chance as an idea generator |
| Uncertainty, ambiguity, magic, handing over the responsibility (the interaction happens somewhere else), you don't put much of yourself in it | mediated | direct | Significance of your own doing, face-to-face contact, experiencing affinity, self-made, close relation to the product, feeling of constant control |
| not feeling as a part of it, feeling of distance | spatial separation | spatial proximity | Personal contact, feeling of relatedness, safety (you know exactly what you did), being a part of it, intensive examination of details |
| Deeper analysis is needed, room for variation = room for competence, room for new ideas, exploration | approximate | precise | Safety, no changes = room to concentrate on something else/competence in other fields, exact idea of result, always exact the same |
| Carefulness, awareness, appreciation, making a relationship with the thing (being gentle with it), being a part of it, revaluation of the action, raises the quality, allows to perform a loving gesture | gentle | powerful | Archaic interaction, sign of strength, power, effectiveness |
| Low challenge, no room to experience competence, no room for improvement, becomes side issue, doesn't matter | incidental | targeted | Appreciation, significance of interaction, worthy of attention, high challenge, high concentration, room for competence |
| Conscious of the significance of your own doing, assurance, security, goal-mode, seeing what is going on, expressive, very easy | apparent | covered | magic, excitement, exploration, action-mode, witchcraft, deeply impress somebody |

*Figure 1. Interaction qualities and evoked experiences, from Lenz et al. 2013*

REFERENCE MATERIAL

[1] Lenz, E., Diefenbach, S., and Hassenzahl, M. (2013) Exploring relationships between interaction attributes and experience. In *Conference proceedings DPPI'13,* ACM Press, pp. 126-135

# METHOD-CARD

## USER RESEARCH PLAN & DATA ANALYSIS [1/2]

### INTRODUCTION

To validate your pre-knowledge and assumptions, prepare a short **user research study** in context. The purpose of the study is to **observe** the natural behaviour of your users and other parties involved and to conduct short **interviews** with them to acquire more information about what people do (actions) and their reasons for it (goals and motivations).

To get the most out of your research you will need to come prepared. Therefore, you will design a **user research plan** that states what you want to know/validate and how you plan to do that by detailing the goal, activities, roles and data collection. Since you are going to collect visual/audio material, and answers from people, you also need to assess the **ethical implications** of your study and ask for participant's consent.

Once you have collected your material from the user research, you need to analyse the data from your study to extract rich and visual information about your users, their needs, preferences, and lifestyles. Finally, you will reflect on the quality of the information gathered:

> Does the data provide rich and deep knowledge?
> Did we ask the right questions? Did we ask them in the right way? Did we observe the details and the bigger picture?

Note: in case you conclude that the quality of the research is not appropriate discuss with your coach whether there is still time to do additional research or whether you will proceed with the findings and make explicit what are the limitations.

### DESCRIPTION

A **user research** plan describes the objectives of the study, the subjects and the context, the focus and the procedure. You will describe your user research plan in a brief document, which should contain the following topics:

**Objective**: What are the goals of your field study? Start by listing your assumptions and specify the information that you want to collect in order to validate your assumptions. Present this information in the form of research questions.

**Subjects & context:** Who are you going to interview and/or observe? Where are you going to observe, and interview your subjects? Identify and explain the need for different types of users in your study: first time users, regular users, etc.

**Focus:** What are you going to observe and ask in the interviews? Take into consideration that the time for you to conduct the research is very limited, so do not construct elaborate questionnaires, but try to focus on the key issues. Consider to observe people's interacting with products in the context of interest. Consider to observe the qualities of environment around. Keep your eyes and mind open to look for things outside the ordinary, since often they bring interesting design opportunities.

**Procedure:** Describe the execution of your research: How are you going to contact your subjects? and how many do you aim to study? (Note: you are expected to contact at least 3 subjects) How are you going to conduct the interviews and observations and how much time do you estimate to interview each participant? How are you going to record visual and audio material? How are you going to record their answers? Who will do what? Design your questions to be accessible to your target group, so they feel capable to answer them. An important task to coordinate is the documentation of the interviews and observations by assigning roles, for example who will make pictures (when and how) and who will write down all what you hear, especially representative quotes.

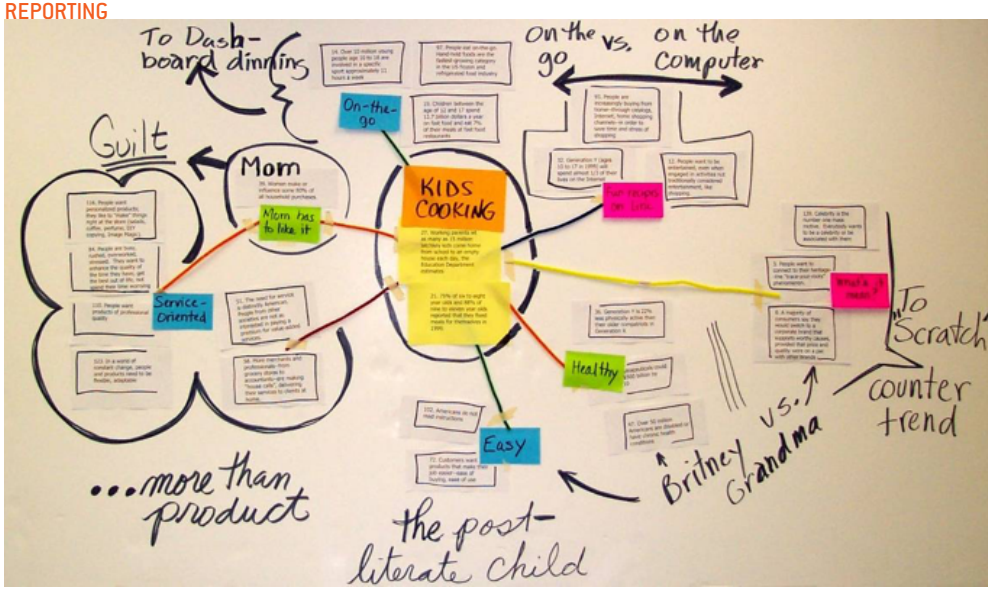
**Ethical check:** To assess the ethical implications of your study, IDE offers a platform for you to learn about **ethics in** education research. Read the information presented and fill in the form to assess the ethics implications of your plans. Prepare as well a consent form to make clear agreements with participants about their obligations and rights when performing the study (see Brigthspace for example: 'Content/Materials/Peer testing'). Include the form in your user research plan.

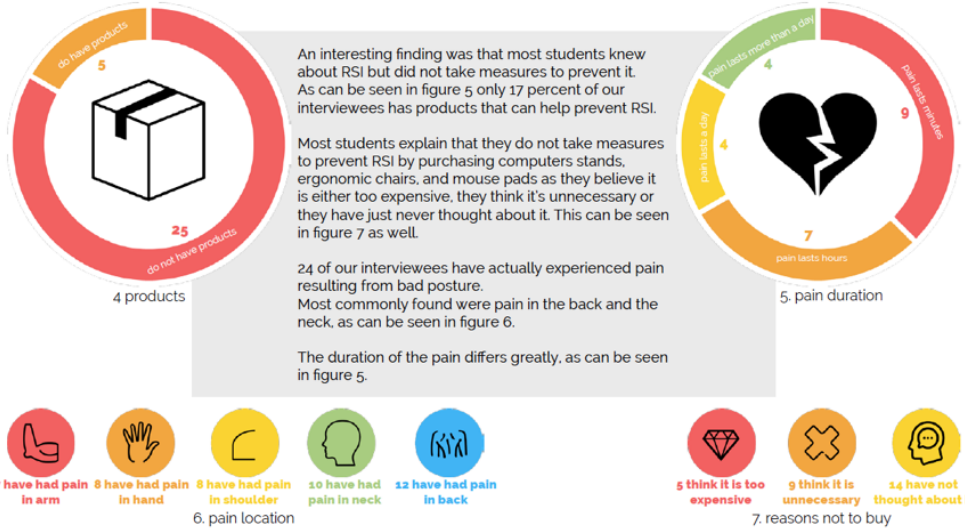
After executing the research, you've gathered lots of valuable material and data. These need to be analysed by ordering them, distilling new insights and communicating results and conclusions. To analyse your data, you start by first **collecting** your notes and bring them into a format suitable for **data analysis**. You might, for example, collect the most striking quotes, observations, impressions, etc. and put these on Post-it notes. Then try to find patterns, characteristics, exceptions etc. **Group** your notes into high-level categories, such as goals, motivations, actions, needs, influencers, barriers/inefficiencies, expectations, preferences etc. (see Figure 1). These categories should not let you lose focus on your target group and context. For visual information review the **pictures** you have taken or the **videos** you have shot and **organize** these to create a visual impression of the context you studied, giving you information that you will later use to define the aesthetics, typography, colour preferences, etc. of your interfaces. Use **your own visual material** to communicate the richness of your context; avoid the use of stock photos or pictures from the Internet, since these tend to be unrealistic or over-the-top.

# METHOD-CARD

## USER RESEARCH PLAN AND DATA ANALYSIS [2/2]

### EXAMPLES OF ANALYSIS AND REPORTING




### REFERENCE MATERIAL

#### BOOK CHAPTERS

> Ch. 4 *'Design Research and brainstorming'*, in Dan Saffer's book, pp. 69-81.

> Ch. 4 *'Design Research and brainstorming'- section 'Design implications'*, in Dan Saffer's book, pp. 84-86.

#### IXD WEBSITES

> How to conduct interviews

#### ETHICS ASSESSMENT

> Ethics in research education (IDE)

# METHOD-CARD

## PERSONAS [1/1]

### INTRODUCTION

A persona brings the data from the user research to life by visually describing the **character**, **behaviour, values** and **needs** of a target group. A persona is a factual characterisation of the target group. It is used throughout the design process by designers to involve stakeholders (project managers, software development, client, etc.) to **make design decisions** with the user in mind. It helps to avoid decision making based on personal experience or biases towards the target group.

### DESCRIPTION

Creating a convincing persona is an important step in the design process. You start by giving your persona a fitting **identity** through a representative name, a personal profile, demographics, interesting quotes (taken from your user research), life goals and motivators. **Describe and visualize** the behaviour, values and needs of your persona in general and in specific for the context and purpose chosen in your design. Consider to visualise a **'one day in the life of'** your persona in the context you want to design for. A persona should communicate **one coherent story** in a **visually attractive** and **comprehensive** way. See Figure 1 for examples.

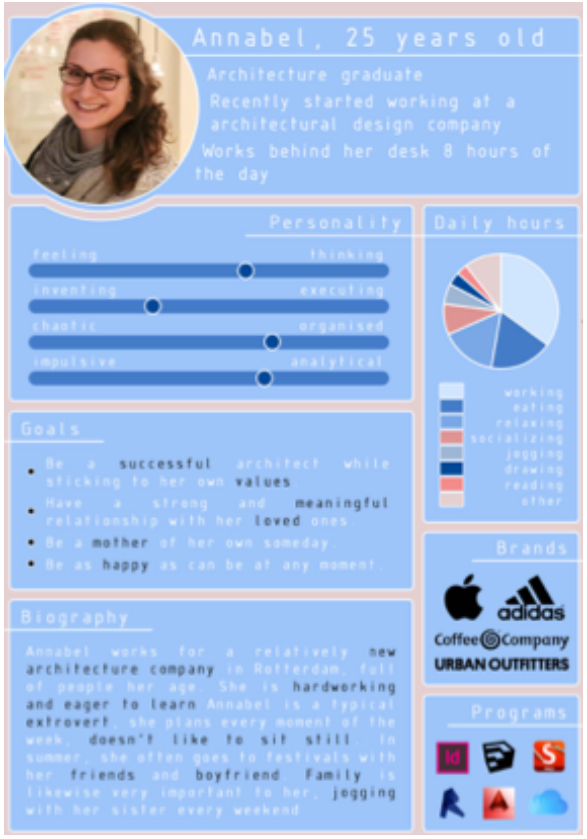
One risk when building a persona is that you build a stereotype. To avoid that, make sure that your persona is based on the results of your data analysis. Use the rich and vivid material that you collected when conducting your study: photos, video footages, user quotes, etc.

### REFERENCE MATERIAL

> Ch. 5. '*The craft of interaction design*' in Dan Saffer's book, pp. 96-101

> *'Personas'*, in the *Delft design guide*, pp. 96.

### EXAMPLE




To develop a persona that becomes a relevant tool in an IxD project, you want to visualise the three main questions you have answered from your user research about your target group:

> what do they want to know?

> what do they want to feel?

> what do they want to do?

Remember that these questions relate to your target group's needs, preferences, and abilities when interacting with a Personal Health Informatics system


*Figure 1. Two Persona examples of last year I&E studentss*

# METHOD-CARD

## INTERACTION SCENARIO AND STORYBOARD [1/1]

### INTRODUCTION

An **interaction scenario** provides **a written and narrative representation** of the relations between your **user**, your **system** and the **context of use**. A **storyboard** is a visual representation of an interaction scenario. Interaction scenarios/storyboards are often used to describe the situation before and after the system has been introduced. In this way, the scenarios help to emphasize and contextualise the benefits of the users when interacting with the system.

### DESCRIPTION

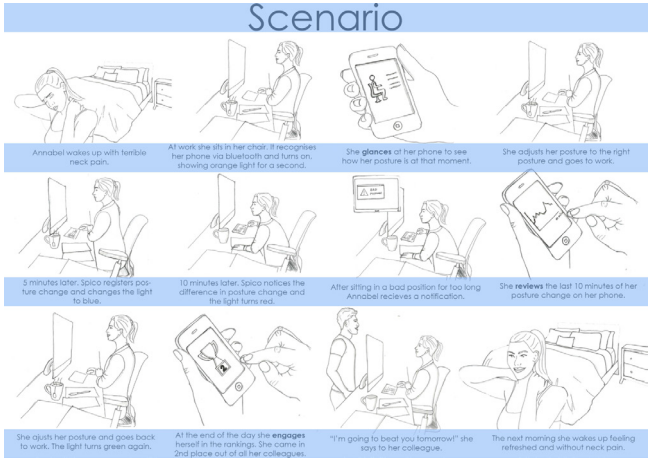
To develop a scenario, you will incorporate the specific characteristics of your persona into the scenario, to help you tailor the system to your target group. Create rich and vivid stories, which contain as many elements as you can think of that will give you directions for the rest of your design process. The story emphasises when the interaction takes place, how the interaction takes place, and what is the effect on the user. Your goal is to validate the three IxD questions you have assumed earlier:
1) how your persona wants to **perform an action** (e.g. by touching a screen or by talking to it), 2) how wants to **feel** (e.g. when being notified or when requesting information) and 3) how wants to **know** (e.g. by overviews, abstract or concrete information).
Make sure you describe the interaction of both the touchscreen and physical device interfaces of the system. For the touchscreen interface, consider different levels of engagement with the system, for example, from quickly glancing on the screen to focused interaction. For the physical interface, consider different feedback scenarios.

Illustrate your scenario with a few **storyboard-like** sketches that show how your persona feels when interacting with the system in the context of use and over time. These sketches should not yet show any user interface details like buttons, text or images, or the physical embodiment (this could very well be a black box). You will use your scenario(s) in the next phase of the project to deduct the functionalities of your system.

### REFERENCE MATERIAL

> Ch. 5 '*The craft of interaction design*', in Dan Saffer's book, pp. 101-102.
> '*Storyboard and Written scenario*' in the *Delft design guide*, pp. 97-99

### EXAMPLE




# UX VISION [1/1]

## INTRODUCTION

**User experience (UX)** refers to the experience(s) users have from interacting with systems (products, services and artefacts). The verb experience refers to people's perceptions, interpretations of those perceptions, and the emotions that are triggered before, during and after interacting with a system. These perceptions are primarily influenced by affective, meaningful and valuable aspects of an interaction, as well as by practical aspects such as utility, ease of use and efficiency of the system. A UX vision helps you to develop a visual message to communicate to the user what they will experience when interacting with the product.

## DESCRIPTION

To define your UX Vision, you need to define what users will **experience** when interacting with your product. If your main goal is to provide users an efficient process, then the interaction style of your system could 'communicate' *efficiency*. If, however, your main goal is to create an entertaining experience, then the interaction could communicate *playful* and *dynamic*. To **visualize** your UX vision, imagine your product is finished and you have to create an advertisement for a magazine or a billboard that is targeted to your user: This will contain a visual message considering the following aspects:

- Product: the name, target group and context of use of the product. You don't define details of the interfaces.
- Core experience: the feeling or atmosphere that the product evokes in the user. This could be in the form of a motto "drive safely --> get more tips" (see Fig.1), "feedback at your own pace" (see Fig. 2) and/or by means of graphical elements (colour scheme, typography, etc.)
- Core functions: unique aspects that the product will offer.

Integrate these aspects into one convincing presentation that communicates your UX vision.

## EXAMPLE


*Figure 1. UX Vision of a body posture tracker for climbers*


*Figure 2. A good/bad driving tracker for pizza deliver worker*

# HIERARCHICAL TASK ANALYSIS (HTA), USER TASK FLOW (UTF) & ACTING MACHINE DIAGRAM (AMD) [1/4]

## INTRODUCTION

The persona, interaction scenario and UX vision define the envisioned user experience of the system. The next question is: *what are the user and system requirements to support this vision?* The user requirements define the **goals and tasks** of the user when interacting with the system.

The system requirements, define the **behaviour (feedback)** of the system in reaction to the user and context. For this particular project, a simplification is made. Since the goal of the touchscreen interface is to provide rich information and desired functions around it, you want to analytically describe what people wants to know (goals) and do (tasks), as well as the way people wants to do the tasks (flow) which closely connects to what they want to feel. You will do this by means of a Hierarchical Task Analysis (HTA) and a User Task Flow (UTF) respectively.

Since the goal of the physical interface is to provide real time feedback and desired functions around it you want to analytically describe how the interface will behave: what many states the feedback has, how does it move from one state to the other, what does it do in a specific state. You will do that by developing an Acting Machine Diagram (AMD).

Note: In a non-simplified project, you want to develop user and system requirements for each interface.

## DESCRIPTION - HTA

A HTA provides a hierarchical structure of users' goals and tasks. You can create an HTA as a tree-like diagram (see Figure 1) or an indented list. You will identify all primary user goals, then break these into high-level tasks and these into subtasks using the following steps:

> Identify users' primary goals to be analysed, and give them a number. These goals are independent from each other.
  Analyse different levels of engagement with the screen to define primary goals, for example:
  - a user goal could be to get a quick overview; of their personal data.
  - a user goal could be to compare today's performance with the average.

> Break the primary goal into high-level tasks (e.g. "See overview today") and these into to subtasks detailing the activities users must perform to accomplish their goals. These subtasks should be specified in terms of objectives and, between them, should cover the whole area of interest. Number each subtask in relation to its high-level task, like 1.1, 1.2 etc.

> Optimise these procedures (check for unnecessary dependencies, repetitive tasks, etc.)

Depending on the complexity of your system you can develop a HTA for each of the primary goals or one HTA to structure all tasks.

## EXAMPLE HTA

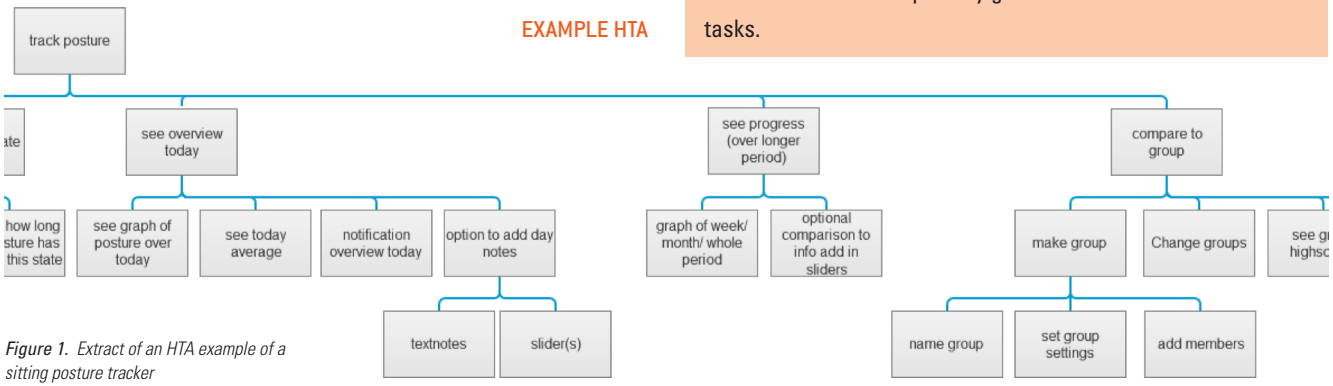



*Figure 1. Extract of an HTA example of a sitting posture tracker*

## DESCRIPTION - UTF

A UTF shows when and how the users perform the actions specified in the HTA. If well done, it can define how people wants to feel when doing certain task. For example, if a user wants to feel reassure, then confirmation after each action may be required in the flow..

It is a representation of user's sequence of interactions with the system to accomplish certain goal (see Figure 2). User flow diagrams emphasize that different user groups may perform different tasks or travel in different paths. User flows are typically attached to a persona and/or a specific entry point. For instance, two users seeking to purchase a book on Amazon may have completely different journeys - one will enter the system via the browser's address bar, while the other may search on Google.

With your persona and HTA in mind, identify the flow of different tasks using the following steps:

> Where does the user start this task?

> How they want to feel when performing that task?, think of different modes of interaction to access information and what type of feeling may evoke: For example, browsing versus searching may evoke different feeling when seeking for information: browsing is more slow and open, while searching is more restricted and fast.

> Identify in the flow; waiting times, error handling and confirmations:
  - What type of waiting times are: system requesting user input, system processing a request, connectivity, etc.
  - What happens if they get lost along the way or make a mistake?
  - How will they feel when they are done?

You can develop one UTF for the whole concept or one for each relevant user goal depending on the complexity of your system.

## EXAMPLE UTF

*Figure 2. Extract of a UTF example considering the three engagement scenarios of a sitting posture tracker*

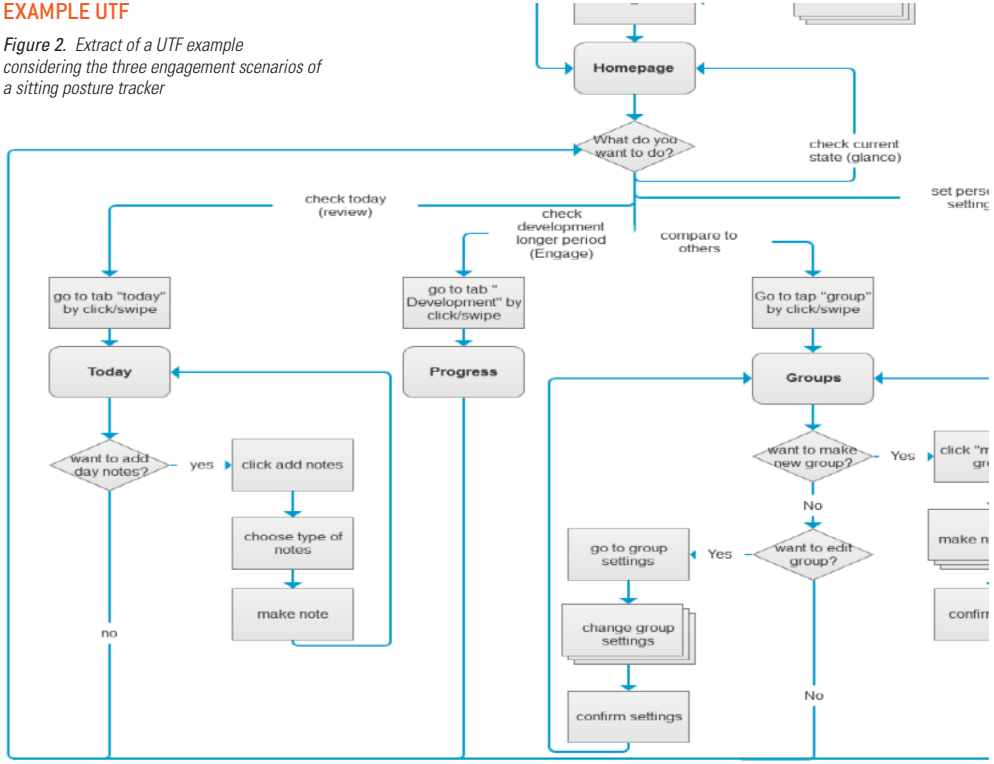
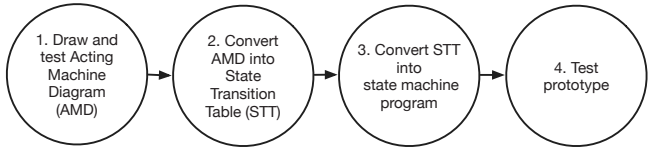



*Figure 3. The 4-step process of developing a working prototype*




## INTRODUCING THE ACTING MACHINE DIAGRAM - AMD

Acting Machine Diagram (AMD)) is a technique used in software engineering for describing and programming interactive products, e.g. standalone physical objects, such as an ATM, or elements in a game, such as an animated character in World of Warcraft. AMDs focus on modelling cause & effect, and how the product (let's call it an 'actor') reacts to them.

The advantages of the AMD technique are that:

> it makes it possible to design and test the product behaviour with the whole team. Rather than to dedicate this task to one team member the whole team can be involved with iteratively developing the product behaviour.

> with a final tested AMD, you have a solid plan for the product behaviour before you start coding, this saves so much time troubleshooting and code redesign activities.

> it supports the inexperienced programmer to create code for the interactive behaviour in microcontrollers (such as Arduino). Through the AMD technique it becomes simpler to create the control flow of code behaviours that work with various sensors and actuators. This method works for all types of interactive products you can imagine and is a great skill to acquire for future projects.

At this point we want to emphasize that for defining the product behaviour you have to change your viewpoint from the designer/user perspective (user scenario) to the system perspective. We have to imagine ourselves – as it were – inside the computer and select from the available system primitives to make designed product behaviour possible. This requires knowledge of the system you are working with e.g. on the Arduino you have to know about the capabilities and limitation of sensors and actuators and how you can work with them. Where analogRead and digitalRead are the system primitives for dealing with sensors and analogWrite and digitalWrite are the primitives for working with actuators.

The AMD technique involves 4 steps as shown in Figure 3. All four steps are described in the following text. We use the Pacman game here as a well familiar story to derive and program the behaviour of the ghost character in the game. But the technique can easily be transposed to any product behaviour that has been explored and defined with a storyboard/ interaction scenario.

## DRAW AND TEST ACTING MACHINE DIAGRAM

Remember playing the game Pacman? Through using a joystick, you can control a yellow pizza-like character, named Pacman, through a maze. Pacman's mission is to eat all the pellets (small dots) that are strewn about. When you have eaten them all, you win, and move up to the next level. One complication is that at some moments, ghost characters come out of their prison and go hunting for Pacman. If they catch him, you lose. But if Pacman eats a large pellet, he grows strong, and he can hunt and catch the ghost. The ghost knows that, and tries to flee. But after some time, Pacman's strength fades, and the ghost becomes dangerous again.


*Figure 4. Stills from the life of pacman creatures*

The previous paragraph described the behaviours of Pacman and the ghosts as a story (you can turn this into a scenario or a storyboard). In it, we recognize that Pacman has three states: 1) normal and 2) strong during play, and if he's caught by a ghost when he is not strong, he is 3) dying (Figure 5). The difference is visible in the background music and in the way Pacman appears. When he is strong, there is some extra aura visible. The difference also affects the way the player should behave: when Pacman is not strong, (s)he should avoid the ghosts, and preferably look for a large pellet to become strong.


*Figure 5. The states in the life of Pacman*

A ghost can have four states (figure 6): it starts 1) captive, waiting in its prison, unable to move. When it is released it becomes 2) angry, and goes hunting for Pacman, always moving towards it. If it collides, Pacman dies. If Pacman becomes strong, the ghost becomes 3) scared, and runs away from Pacman. If Pacman catches the ghost while he is strong, 4) the ghost dies and disappears in a puff of smoke.


*Figure 6. The states in the life of a ghost*

We see that each of the states of the ghost is characterized by a specific behaviour, e.g., whether it moves, and in which direction, and how the ghost reacts to a collision with Pacman, and (often) by a specific appearance, e.g. whether the ghost looks aggressive or scared.

Also, we see that there are certain causes which make the ghost go from one state to another: from captive to angry when it is released, from angry to scared when Pacman becomes strong, back to angry when Pacman's strength wears off, and dying when it collides with a strong Pacman.

In an AMD, we show states as circles labelled with the name of the state, and transitions between states as arrows, labelled with the cause that triggers the transition. The diagram format helps the reader to see if it is complete, and how the behaviours are related. For instance, it shows that a ghost that leaves the captive state never becomes captive again (there is no path of arrows that leads back), but that a ghost can change between angry and scared many times before colliding with Pacman. You can use the AMD to act out the ghost behaviour in a role play. At any time of its existence, the ghost is in exactly one state, which you can visualize by e.g., placing a pawn on the diagram.

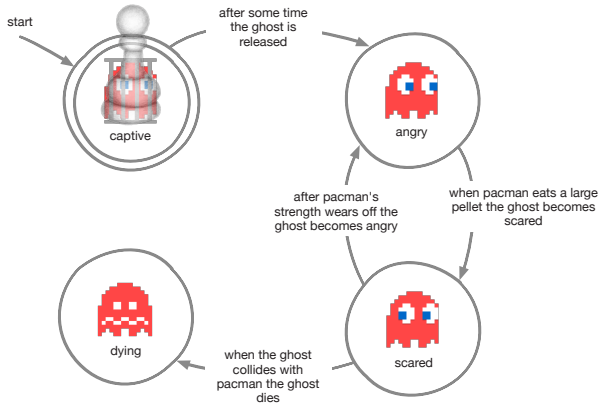



*Figure 7. Ghost states and transitions. Note: you can use a pawn to indicate the current state when you test the scenario to check that all the behaviour is modelled.*

We now complete the diagram with four aspects: the beginning, effects, in-state effects, and ongoing effects.

First, let's indicate the 'begin state'. This is traditionally done by giving it a double circle, or by drawing an incoming arrow coming from 'nowhere'. For the ghost and Pacman, the begin states are captive and normal, respectively.

Effects are things that occur at transitions. For instance, when ghost becomes scared, its colour may become paler, when Pacman becomes strong, he gets an aura. These changes which occur in transitions are also noted next to the label, typically separated from the cause by a slash '/'. If a transition has more than one effect, we separate them with ',' comma characters.

In-state effects are cause-effect relations that do not cause a transition to a different state. For example, when Pacman changes direction from left to right, its appearance changes (the mouth-side switches), but it remains in the same state. These in-state effects are shown as loop-arrows, which return to the same state as where they began.

Finally, there are ongoing effects that do not seem to have a cause: animation effects such as walking legs or bobbing eyes go on all the time.

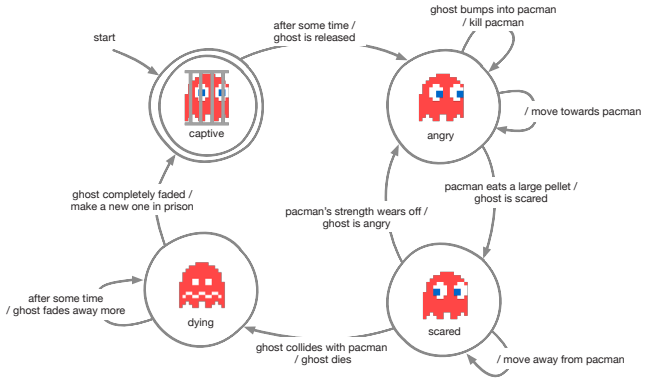



*Figure 8. More complete ghost AMD*

There are several ways to look at this. Either it is modelled as an effect that requires no cause, or it is an arrow whose condition is always true (e.g., the cause is that time has moved on). Typically, this is chosen when none of the other arrows is triggered by its cause.

Now we have a complete AMD (Figure 8). With it one of you can play the role of ghost in Pacman. Try it. You can discuss the diagram to see if something is missing, or how it can be improved. For instance, under what conditions is a ghost released from prison? Does that occur when Pacman has eaten 50% of the pellets, or if he comes too close to the prison, or what else? Is every state accompanied by a matching visual or continuous music (relaxed while there are no ghosts, tense when there are ghosts, energetic when the ghosts are scared)? When should these music moods start and stop. Should we add short sound effects to the moments of transitions (a clunk on collisions, a warning when the ghost is released)? We can add these to the arrows.

When acting out an AMD, you will have noticed that more than one cause can be active, but only one arrow can be chosen. One rule is that the ongoing effects (whose cause is always true) should be the last ones to consider, but also that some causes are more important than others: if a collision should occur at the same time as Pacman eats a large pellet, it makes a lot of difference if you first check the arrow for collisions (Pacman dies) or first check the arrow for eating the pellet (ghost dies upon the next round of checking). For most of the arrows the order does not matter, but you have to decide on a sequence anyhow. You can label the sequence where it matters in the diagram

Now, what about the AMD for Pacman you might think? Well we leave that for you to work out, it should be relatively straightforward once you can read the AMD of the ghost. In the Implementation method-card we press ahead to arrive at programming code that can model the behaviour expressed in an AMD, as in creating the program that makes everything work

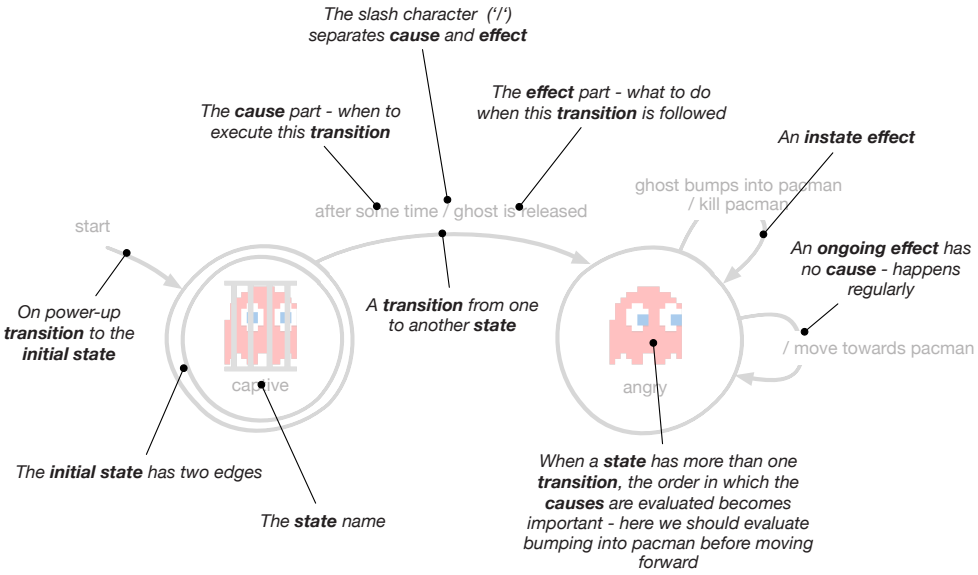



*Figure 9. Legend of State Transition Diagrams*

## REFERENCE MATERIAL

**BOOK CHAPTERS.**

> Ch. 5 *'The craft of interaction design'*, in Dan Saffer's book, pp. 104-106.

**IXD WEBSITES**

> HTA
> UTF

*Figure 10. You can draw the AMD for Pacman here yourself*

# METHOD-CARD

## SKETCHING LAYOUT, LOOK & FEEL AND INTERACTIVITY [1/3]

### INTRODUCTION

Sketching is a low-fi prototyping technique that is used to translate your exploration of ideas from your mind to a 2D world (paper). **Sketching** is a **timely and cost-efficient too**l that allows you to:

> explore several alternatives of one design aspect. In other words, explore your design space in order to develop a design that fits your target group and interaction vision: make the *right design.*

> try out different approaches and apply IxD principles that will help to develop a design that it is useful and complies to design standards: make the *design right.* To learn more about IxD principles check the reference material.

You will create different types of sketching to explore the layout, look & feel and interactivity of both the touchscreen and physical interfaces

When exploring different layout alternatives, you should be able to answer the questions: what, where and when users will have access to the information needed? For example: How does the layout or organisation of the chosen elements make pertinent information as self-evident as possible for users? What parts of the layout and navigation do you want to keep consistent and what needs to change? How does the layout convey the interaction style envisioned: think of for example, the quantity of information presented at one point in time? How will you develop a common language in layout and navigation between diverse mediums?

### LAYOUT SKETCHING - TOUCHSCREEN

**Wireframe hand drawing sketching** serve to model the touchscreen interfaces without coding and detailing the design elements used (see wireframe reference) The **layout** of a touchscreen interfaces refers to the selection of **functional/interactive** and **informative elements**, their **positioning** on the screen and the **navigation structure** between screens (seeFigure 1 - right).

The suggested steps to sketch a design alternative are:

> Based on the HTA & UTF, first explore the types of **informative elements** (such as text, photos/videos, lists, etc.) and **functional elements** (such as controls, buttons, sensors, etc.) that your interface should support. Consider how many distinctive elements are needed, and when can they be combined or showed separately.

> Next, decide on **how many screens** your interface needs to allocate these informative and functional elements.

> Next, define a **grid layout** (see grid layout reference) that allocates the informative and functional elements across screens in a consistent, clear and logic manner. Consider the dimensions of the screen and the position (portrait or landscape) to do this.

> Next, decide on the **navigation structure** across screens; which screen goes first and which second; what functional elements will be used to move from two screens.

> Finally, explore **menu elements** (such as home, settings, global and local functions, etc.) that will support users to navigate. Consider a consistent and distinctive positioning of dynamic and static menus, like the use of global menu that does not change across screens, e.g. home button and the use of local menu that changes depending on the screen, e.g. a confirmation button.

### PHYSICAL DEVICE

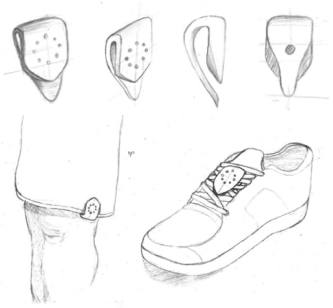
**Hand drawing sketching** serve to model the physical device interfaces without detailing of dimensions, material, etc. The **layout** of a physical interface refers to the selection of **sensors** and **actuators**, their **position** on the device and the **location** of the device in relation to the user (see Figure 1 - left). The goal is to optimise the users' understanding and experience of the feedback. The suggested steps to sketch a design alternative are:

> Based on the AMD, first explore the types of **contextual sensors** (e.g. light, noise, movement) and **interactive sensors** (e.g. buttons, touch sensor, slider) as well as **actuators** (e.g. colour LED, vibration, buzzer) that your interface should support.

> Next, explore the positioning of the device on the user's body or personal space, e.g. neck, wrist, shoes, desktop, jacket, interior object, a tool, etc.

> Finally, decide on the positioning of the elements on the physical device. Explore their absolute position with regards to the device, as well as relative position between elements. Revise your interaction scenario, to consider issues such as: where can the sensor best capture the information needed at a specific moment in time, and where actuators can best present the feedback at a specific moment (would two actuators act simultaneously, do you want the feedback to be understood independently?).

### EXAMPLE LAYOUT



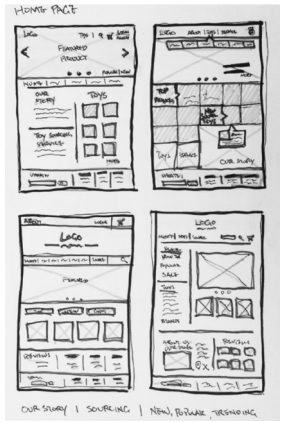


*Figure 1. Left - drawing sketches of layout physical device. Right - wireframe sketches of layout and navigation structure;*

# METHOD-CARD

## SKETCHING LAYOUT, LOOK & FEEL AND INTERACTIVITY [2/3]

### LOOK & FEEL SKETCHING - TOUCHSCREEN

By means of wireframe sketches, you will explore different visual aspects and elements to define the look & feel of the touchscreen interface (see Figure. 2).Two types of visualisations should be at least explored: general look & feel and data visualisation. The suggested steps to sketch a design alternative are:

> **General look & feel**: define a way to present the overall look & feel of the touchscreen interface so it provides an attractive, well-balanced, fresh, and harmonious appearance. Consider distinctive visual aspects to support the generic visualisation by means of a colour scheme, iconography, typography, etc.

> **Data visualisation**: define a way to visualise different levels of engagement with data (e.g. concrete, abstract, simple, complex) to users. Consider the use of metaphors, i.e. the use of a familiar concept to represent the information, e.g. a full or empty battery to represent amount of sleep; or the use of concrete visualisations, e.g. a graph. To make a graph meaningful, consider what exactly your user wants to know and how is he able to do so. If the goal is to understand progress overtime, then a reference of the average progress compare to a specific day may be better than show progress of every day in a week.

### EXAMPLE LOOK & FEEL

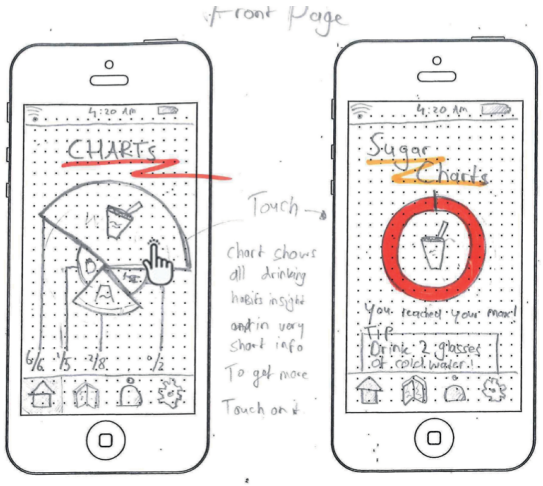




*Figure 2. Left - wireframe sketches of look & feel; Right - hand drawing sketches of physical look & feel*

### PHYSICAL DEVICE

By means of hand drawing sketches, you will explore different visual aspects of the chosen elements in the physical device interface (see Fig. 2). The context of use will play an important role to explore the **visual quality** of the general look & feel (e.g. text or graphic labels, colour background, etc.) and data feedback (e.g. light colours, vibration, etc.). The suggested steps to sketch a design alternative are:

> **General look & feel**: define a way to **communicate an attractive, well-balanced, fresh, and harmonoius appearance of the** different informative elements in the interface. Consider the use of text labels, graphics, colour scheme, etc. i.e. the physical iconography. Consider the quantity of the information needed, and work towards an intuitive interface that requires a minimum of supportive visualisation. Consider the quality of its visualisation, and work towards a harmonious set of visuals.

> **Data feedback**: define the **visual quality of the feedback**. For example, consider the brightness or intensity of an actuator with respect to the context and explore how it can communicate a pleasant and meaningful feedback. For example, a covered LED may be needed in certain context to not annoy the user..

When exploring different look & feel alternatives, you should be able to answer the question: how will you visually communicate the information needed? For example: How can changes in information be visually clear for users? How does the use of a metaphor make information self-understandable for users? How does the look & feel convey the qualities of interaction envisioned: think of for example, the variety of visual aspects at one point in time? How will you develop a common language in look & feel between diverse media?

# METHOD-CARD

## SKETCHING LAYOUT, LOOK & FEEL AND INTERACTIVITY [3/3]

### INTERACTIVITY SKETCHING - TOUCHSCREEN

By means of wireframe sketching and the use of symbols for interactive elements (see reference 'Pointing device gestures' for a collection of symbols) you will explore the interactivity aspects of the touchscreen interface (see Figure 3 for examples). Two types of interactivity should be at least explored: transitions in and between screens and screen feedback. The suggested steps to sketch a design alternative are:

> **Transitions on and between screens**: consider touch-based elements and define changes on a screen or transitions between screens, that support the quality of interaction envisioned. Consider different touch-based elements, e.g. one-double-triple tap, long press, flick, drag & drop, scroll, swipe, etc. to define the mode of navigation.

> **Screen feedback**: consider animated feedback elements and define contextual feedback of the current state of a screen, that support the intended interaction style. In your UTF you have defined waiting times, error handling, and confirmations. Consider different animations to visually characterise such states (check 'micro-animations' in the reference material).

### PHYSICAL DEVICE

By means of hand drawing sketches, you will explore the interactivity aspects of the physical interface. The interactive quality of the feedback will be explored to support the intended qualities of interaction.

The suggested steps to sketch a design alternative are:

> **Feedback**: For the interactive quality consider the time aspect and intensity of feedback. For example, think of when would an increasing or fading effect to activate/deactivate a light be needed? Should a repetitive pattern of a beep be constant and always the same or should vary depending on time or changes in context? Table 1 provides you with suggestions.

*Table 1. Suggested interactive qualities to explore feedback interactivity*

| Feedback | Time Interval | Speed | Frequency |
|---|---|---|---|
| light, vibration, beep | | | |
| blinking | | | |
| diming | | | |
| changing colour/intensity/tone | | | |
| other | | | |

When exploring different interactivity alternatives, you should be able to answer the question: how will you attract users' attention as intended? For example: what type of user and system actions are appropriate at different moments? What patterns of feedback create the intended experience on user: awareness, urgency, etc.? How does the interactivity convey the interaction style envisioned: think for example of the time and behaviour of feedback? How will you develop a common language in interactivity between diverse mediums?

### EXAMPLE INTERACTIVITY

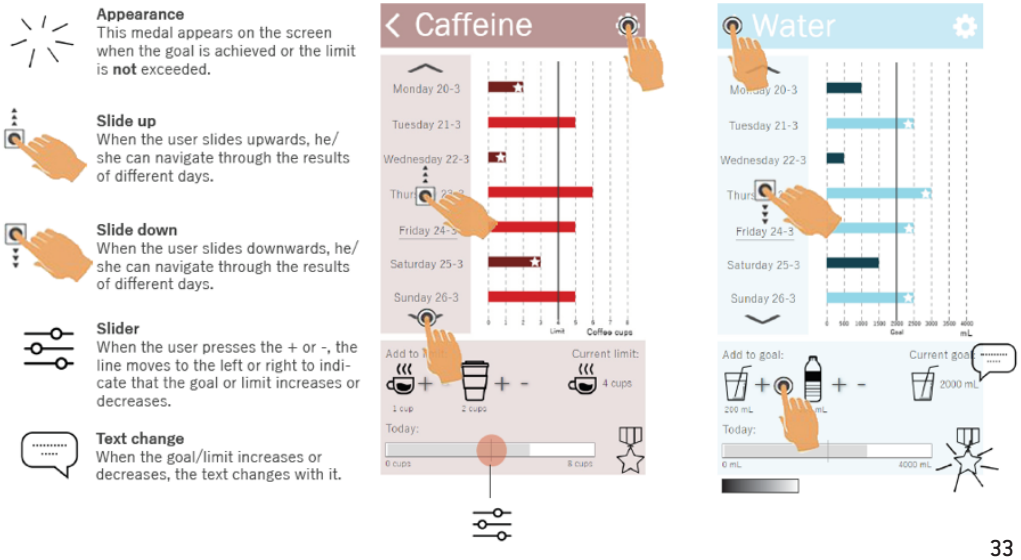



*Figure 3. wireframe sketches of interactivity*

### REFERENCE MATERIAL

#### BOOK CHAPTERS

> Ch. 5 *'The craft of interaction design'*, in Dan Saffer's book, pp. 109-113.

> Ch. 6 *'Interface design basics'*, in Dan Saffer's book, pp. 121-145.

#### BRIGHTSPACE

> The 5 infographics on IxD principles found in Learning resources: consistency, feedback, predictability, visibility, learnability

#### IXD WEBSITES

> Sketching user experience

> Typography in mobile apps

> Colour theory

> The golden ratio principle and layout

> The grid system and layout

> Aesthetics and form

> Pointing device gestures

> Micro-animations

> Wireframe

# METHOD-CARD

## PROTOTYPING LAYOUT, LOOK & FEEL AND INTERACTIVITY [1/2]

### INTRODUCTION

Acquiring sufficient skills of making prototypes is essential for interaction designers to move from sketches to more tangible and detail representations of selected ideas. Iteratively you make prototypes to understand the problem, to explore the feasibility of (technical) concepts, to test alternative solutions or to demonstrate the actual solution. The further you move in your design process the more complexity you add into your prototype. The more complexity you involve in the prototyping activity the more richness the prototype needs to communicate. In this project, you will prototype, in increase order of complexity, the layout, the look & feel and the interactivity of your interfaces using lo-fi and hi-fi prototyping techniques.

### LO-FI PROTOTYPING

#### PAPER TOUCHSCREEN PROTOTYPE

Paper prototyping is a cost-effective and efficient tool to get **maximum feedback for minimum effort** at early stages of your design. Using paper at this stage does have remarkable advantages. You can **make changes on the fly** during a test. If a design doesn't work the first time, designers can scrap it and quickly try another. For example, if a user is stuck and needs additional information, a 'Hints' button can be added on the fly. **Users feel more comfortable to be critical** of a paper prototype because it does not have a polished look. When something doesn't work well in a crisp looking prototype, users are more likely to blame themselves or their lack of experience instead of seeing critical to the prototype. Finally, with paper prototypes the whole design **team can gather around one table** with their eyes on the same design and each team member can contribute as the design unfolds. Paper prototypes for later stages in the design process pose disadvantages as they lack things such as interactivity, dynamics, sound, etc. and they usually cannot mirror the whole application to be produced. It is therefore necessary to **limit the scope of your paper prototype and use it in early stages of your design.**

To quickly validate with peers your choices for the touchscreen layout and look & feel, make a lo-fi pen & paper prototype to demonstrate your choices and test. The steps to build a paper prototype are:

> First develop a quick **test script** to describe the tasks you want your testers to try out in your interface: for the layout, the goal is to test if the informative and interactive elements are clear and distinctive in terms of their positioning/clustering on the screen, and whether the global and local navigation menus are clearly supported by interactive (menus) elements

> Looking at the wireframe sketches, start by identifying the screens your prototype must include in order to support the defined user tasks. Identify the layout and navigation elements that you want your user to interact with.

> Next create/download a **screen template using the real size** of the screen you are designing for, and print several copies. Draw the layout and navigation elements of each screen on the templates.

> Draw and cut out the **extra individual elements** that your users will place on the drawn screens when performing their tasks. The arrangement of the interface elements can thus be changed quickly and without programming, labels can be renamed, and objects added, removed, or replaced by other input widgets. The sequence of dialogs or pages should be set on the fly, which makes it easy to test multiple versions of your application's interaction design.

> Finally, do one or more walkthroughs of the test script to identify parts of the prototype that still need to be created. Prepare for different paths the user might correct or mistakenly take.

### EXAMPLE PAPER PROTOTYPE (LAYOUT AND LOOK & FEEL)

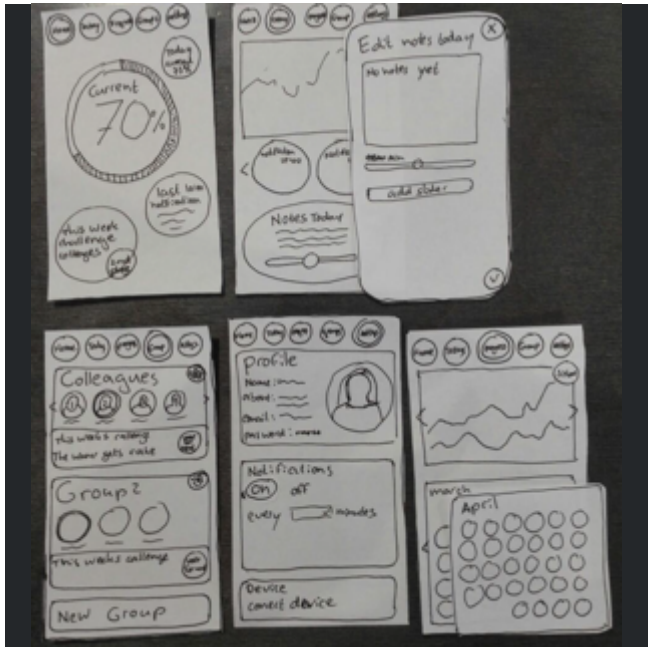



*Figure 1. LO-FI pen & paper prototype to test layout and navigation structure.*

# METHOD-CARD

## PROTOTYPING LAYOUT, LOOK & FEEL AND INTERACTIVITY [2/2]

### CARDBOARD PHYSICAL PROTOTYPE

To quickly validate with peers your choices for the physical layout and look & feel, make a cardboard prototypes to demonstrate your choices and test. The steps to build a cardboard prototype are:

> First develop a quick **test script** to describe the tasks you want your testers to try out in your interface. For the layout, the goal is to test if the sensors and actuators are position in an optimal place to capture information and provide feedback, as well as whether the positioning of the device on the users' body or personal context is right.
> Create a cardboard holder that simulates the device and its use (e.g. necklace, wristband, etc.).
> Create cardboard pieces to represent the sensors and actuators and place them on the cardboard holder.
> Then do one or more walkthroughs of the test script to identify parts of the prototype that still need to be created. Prepare for different paths the user might reasonably take (correct or otherwise).

### EXAMPLE CARDBOARD PROTOTYPE (LAYOUT AND LOOK & FEEL)

*Figure 2. LO-FI cardboard prototype to test layout of physical device*

### HI-FI PROTOTYPING

To validate with peers the unique aspects of your interactions, you want to prototype relevant interactions as experiential as possible to test the intended effect.

For the touchscreen interface you want to prototype interactions that concern with user requesting and the system displaying information. Therefore the focus is on prototyping relevant transitions within a screen or between screens. You will prototype the user interactions and the system behaviour of the chosen transitions. Consider the interaction qualities that you think are needed to achieve the intended effect: e.g., tapping on a visualisation or swiping to the right to feel in control when requesting detail information; receiving notifications of a new goal achieved with a pop up or an animation on screen to feel guided, etc. The following steps help you to organise this activity:

> Choose the digital prototyping tool that you feel confident with to build the prototype. For this course, we recommend to use either Axure RP Pro or InVision. Check Appendix 2 to learn more about the prototyping tools supported by this course and how to access them.
> Implement in the tool of choice the screens of your concept
> Select relevant transitions within and between screens where you have designed unique interactions relevant to test. Implement several alternatives to assess which one represent better your intended effect.

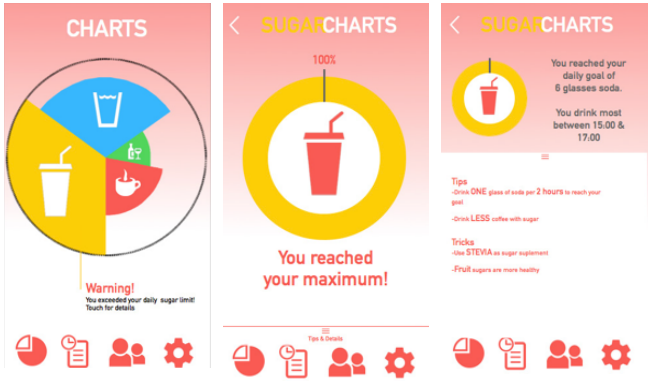



*Figure 3. HI-FI ON-SCREEN prototype to test look & feel*

For the physical interface you want to prototype interactions in relation to the real-time feedback offered by the interface. Therefore the focus is on prototyping relevant transitions of the elements used for the feedback. For example, if your feedback is a combination of LED and a buzzer, and you want to achieve a feeling of alert, you want to prototype different combinations, intensities, and changes between these elements to test their effect. The following steps help you to organise this activity:

> Prepare and choose quick prototyping material and tools to represent the device and host the sensors and actuators of choice.
> Create a small program to activate/deactivate the actuators to test how they look. Think of showing minimum, middle and maximum values to test how pleasant people perceive them.
> Simulate relevant feedback states to test how clear and meaningful they are perceived.

### REFERENCE MATERIAL

#### BOOK CHAPTERS

> 'Prototypes: paper, interactive and physical' in ch. 5 '*The craft of interaction design*' of Dan Saffer's book, pp. 114-119.

#### IXD WEBSITES

> Paper prototyping > Testing prototypes

#### VIDEO TUTORIAL (PARTS OF MOOC COURSE DEVELOPED BY HVA):

> Paper prototype video tutorial

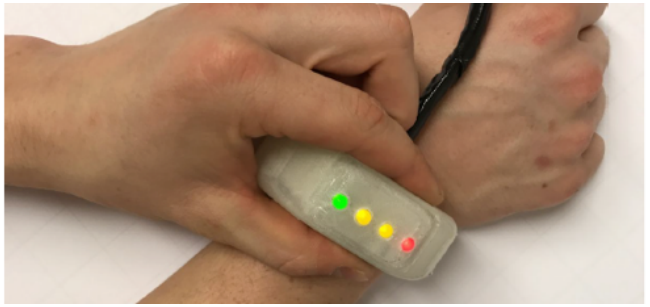


*Figure 4. HI-FI prototype to test look & feel of physical device*

# METHOD-CARD

## PEER TESTING [1/3]

### INTRODUCTION

Peer testing is a way to quickly evaluate on spot the design decisions made, for example a preliminary set of requirements or design elements and aspects of the interface. Why testing with peers? Peers may not be representative of your target group however they are quickly accessible and skilful in evaluating prototypes. They are aware of the target group but not of the design decisions that will be tested. In this project, you will peer test the vision & requirements, layout, look & feel and interactivity of the interfaces.

### DESCRIPTION

#### VISION & REQUIREMENTS

The goal is to collect feedback on the user & system requirements related to the user engagement scenarios for the touchscreen and the adaptive feedback for the physical device. A template and a feedback form are available to support you in this activity (feel free to adapt them if you have clear arguments for doing so). Copy the template into a flip-over sheet to work with. The template is intended as a communication tool: to communicate your persona, scenario, UX vision and UTF and AMD to your peers and collect their feedback, to communicate the design decisions based on that feedback to your coaches and to keep as reference during the prototyping weeks. You will use the feedback form to collect feedback from at least 6 peers. You can find a digital copy of the template and the feedback form in Brigthspace (under 'Content/Materials/Peer testing').

#### LAYOUT, LOOK & FEEL AND INTERACTIVITY

The general instructions to conduct peer-testing for both interfaces and all the design aspects intended to be tested, are the following:

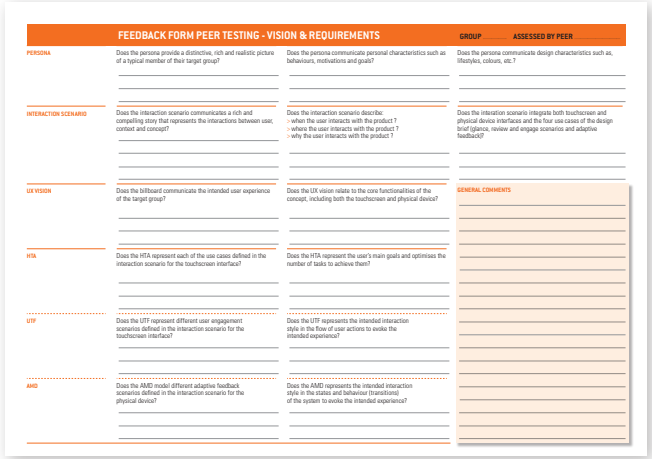
> Select participants, introduce the goal of the test and introduce the tasks. Ask participants to think-aloud when performing the tasks. This means that they verbally express what they think and experience when doing the tasks (not just what are they doing, you can observe that!). Use the peer-testing template to annotate relevant observations and prompt questions to participants after the task has been done. A digital copy of the templates is provided in Brightspace (under 'Content/Materials/Peer testing')

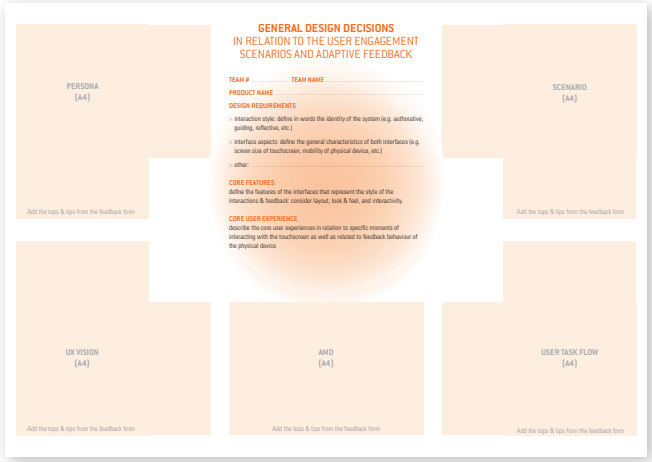
> Conduct peer-tests with at least 3 peers. Recruit different people for each iteration, so the testing will not be biased. However the same person could test the touchscreen and physical interface in one iteration.

> Analyse the data by clustering and summarising your observations. Provide a visual overview (e.g. use of a table) and list main conclusions based on the findings.

### TEMPLATES

#### VISION & REQUIREMENTS



# METHOD-CARD

## PEER TESTING [2/3]

### > TOUCHSCREEN LAYOUT

PARTICIPANT .....

| | very bad | | | | very good |
|---|---|---|---|---|---|
| **Position of Informative elements** | | | | | |
| Distinctive from other elements | -2 | -1 | 0 | 1 | 2 |
| Logical position | -2 | -1 | 0 | 1 | 2 |
| Consistent position across screens | -2 | -1 | 0 | 1 | 2 |
| Fit user & context | -2 | -1 | 0 | 1 | 2 |
| Fit interaction style | -2 | -1 | 0 | 1 | 2 |

NOTES: .....

| | | | | | |
|---|---|---|---|---|---|
| **Position of Functional elements** | | | | | |
| Distinctive from other elements | -2 | -1 | 0 | 1 | 2 |
| Logical position | -2 | -1 | 0 | 1 | 2 |
| Consistent position across screens | -2 | -1 | 0 | 1 | 2 |
| Fit user & context | -2 | -1 | 0 | 1 | 2 |
| Fit interaction style | -2 | -1 | 0 | 1 | 2 |

NOTES: .....

| | | | | | |
|---|---|---|---|---|---|
| **Navigation structure (number of screens/interaction steps)** | | | | | |
| Logical | -2 | -1 | 0 | 1 | 2 |
| Consistent position across screens | -2 | -1 | 0 | 1 | 2 |
| Fit user & context | -2 | -1 | 0 | 1 | 2 |
| Fit interaction style | -2 | -1 | 0 | 1 | 2 |

NOTES: .....

| | | | | | |
|---|---|---|---|---|---|
| **Global and local menu system** | | | | | |
| Logical | -2 | -1 | 0 | 1 | 2 |
| Consistent position across screens | -2 | -1 | 0 | 1 | 2 |
| Fit user & context | -2 | -1 | 0 | 1 | 2 |
| Fit interaction style | -2 | -1 | 0 | 1 | 2 |

NOTES: .....

### > PHYSICAL LAYOUT

PARTICIPANT .....

| | very bad | | | | very good |
|---|---|---|---|---|---|
| **Position of actuators elements** | | | | | |
| Logical position | -2 | -1 | 0 | 1 | 2 |
| Fit user & context | -2 | -1 | 0 | 1 | 2 |
| Fit interaction style | -2 | -1 | 0 | 1 | 2 |

NOTES: .....

| | | | | | |
|---|---|---|---|---|---|
| **Position of sensor elements** | | | | | |
| Logical position | -2 | -1 | 0 | 1 | 2 |
| Fit user & context | -2 | -1 | 0 | 1 | 2 |
| Fit interaction style | -2 | -1 | 0 | 1 | 2 |

NOTES: .....

| | | | | | |
|---|---|---|---|---|---|
| **Position of the device** | | | | | |
| Logical position | -2 | -1 | 0 | 1 | 2 |
| Fit user & context | -2 | -1 | 0 | 1 | 2 |
| Fit interaction style | -2 | -1 | 0 | 1 | 2 |

NOTES: .....

### > TOUCHSCREEN LOOK & FEEL

PARTICIPANT .....

| | very bad | | | | very good |
|---|---|---|---|---|---|
| **General look & feel** | | | | | |
| Fresh | -2 | -1 | 0 | 1 | 2 |
| Harmonious | -2 | -1 | 0 | 1 | 2 |
| Attractive | -2 | -1 | 0 | 1 | 2 |
| Well-balanced | -2 | -1 | 0 | 1 | 2 |
| Fit user & context | -2 | -1 | 0 | 1 | 2 |
| Fit interaction style | -2 | -1 | 0 | 1 | 2 |

NOTES: .....

| | | | | | |
|---|---|---|---|---|---|
| **Data visualisation** | | | | | |
| Engaging | -2 | -1 | 0 | 1 | 2 |
| Meaningful | -2 | -1 | 0 | 1 | 2 |
| Fit user & context | -2 | -1 | 0 | 1 | 2 |
| Fit interaction style | -2 | -1 | 0 | 1 | 2 |

NOTES: .....

# METHOD-CARD

## PEER TESTING [3/3]

### > PHYSICAL LOOK & FEEL

PARTICIPANT .....

| | very bad | | | | very good |
|---|---|---|---|---|---|
| **General look & feel** | | | | | |
| Fresh | -2 | -1 | 0 | 1 | 2 |
| Harmonious | -2 | -1 | 0 | 1 | 2 |
| Attractive | -2 | -1 | 0 | 1 | 2 |
| Well-balanced | -2 | -1 | 0 | 1 | 2 |
| Fit user & context | -2 | -1 | 0 | 1 | 2 |
| Fit interaction style | -2 | -1 | 0 | 1 | 2 |

NOTES: .....

| | | | | | |
|---|---|---|---|---|---|
| **Data feedback** | | | | | |
| Pleasant | -2 | -1 | 0 | 1 | 2 |
| Meaningful | -2 | -1 | 0 | 1 | 2 |
| Fit user & context | -2 | -1 | 0 | 1 | 2 |
| Fit interaction style | -2 | -1 | 0 | 1 | 2 |

NOTES: .....

### > TOUCHSCREEN INTERACTIVITY

PARTICIPANT .....

| | very bad | | | | very good |
|---|---|---|---|---|---|
| **Transitions on screen** | | | | | |
| Intuitive | -2 | -1 | 0 | 1 | 2 |
| Meaningful | -2 | -1 | 0 | 1 | 2 |
| Fit user & context | -2 | -1 | 0 | 1 | 2 |
| Fit interaction style | -2 | -1 | 0 | 1 | 2 |

NOTES: .....

| | | | | | |
|---|---|---|---|---|---|
| **Transitions between screen** | | | | | |
| Intuitive | -2 | -1 | 0 | 1 | 2 |
| Meaningful | -2 | -1 | 0 | 1 | 2 |
| Consistent across screens | -2 | -1 | 0 | 1 | 2 |
| Fit user & context | -2 | -1 | 0 | 1 | 2 |
| Fit interaction style | -2 | -1 | 0 | 1 | 2 |

NOTES: .....

| | | | | | |
|---|---|---|---|---|---|
| **Screen feedback** | | | | | |
| Intuitive | -2 | -1 | 0 | 1 | 2 |
| Meaningful | -2 | -1 | 0 | 1 | 2 |
| Consistent across screens | -2 | -1 | 0 | 1 | 2 |
| Fit user & context | -2 | -1 | 0 | 1 | 2 |
| Fit interaction style | -2 | -1 | 0 | 1 | 2 |

NOTES: .....

| | | | | | |
|---|---|---|---|---|---|
| **Feedback quality** | | | | | |
| Intuitive | -2 | -1 | 0 | 1 | 2 |
| Meaningful | -2 | -1 | 0 | 1 | 2 |
| Experiential | -2 | -1 | 0 | 1 | 2 |
| Fit user & context | -2 | -1 | 0 | 1 | 2 |
| Fit interaction style | -2 | -1 | 0 | 1 | 2 |

NOTES: .....

# METHOD-CARD

## EXTENDED ANNOTATED PROTOTYPES [1/2]

### INTRODUCTION

Annotations are useful to describe and explain your designs as you work on them, attaching notes to specific design elements. An extended annotated prototype serves as a communication tool to contextualise the main results from your tests.

You will use extended annotated prototypes to describe prototypes you made for the layout, look & feel and interactivity and peer tests results, as well as the final integrated prototype and the user evaluation results.

### DESCRIPTION

For all purposes, an annotation contains a descriptive note that refers to the function of an indicated element; or an explanation. It is highly suggested that an annotation contains both types of notes using the following syntax:

> [element] *in order to* [function] *with the purpose to* [intended user experience/or user goal]. *User insights* [positive/negative assessment] *because* [explanation].

For example, a blue line to indicate the average state of a battery icon (used as metaphor for step counting) IN ORDER TO provide a comparison between current daily progress and average daily progress WITH THE PURPOSE TO make user feel capable to meet the daily goal. USER INSIGHTS show that the line is unclear BECAUSE it does not indicate since when is the average calculated.

### EXAMPLE

Note: these examples are from previous year project (the proposed syntax was not yet introduced to students)

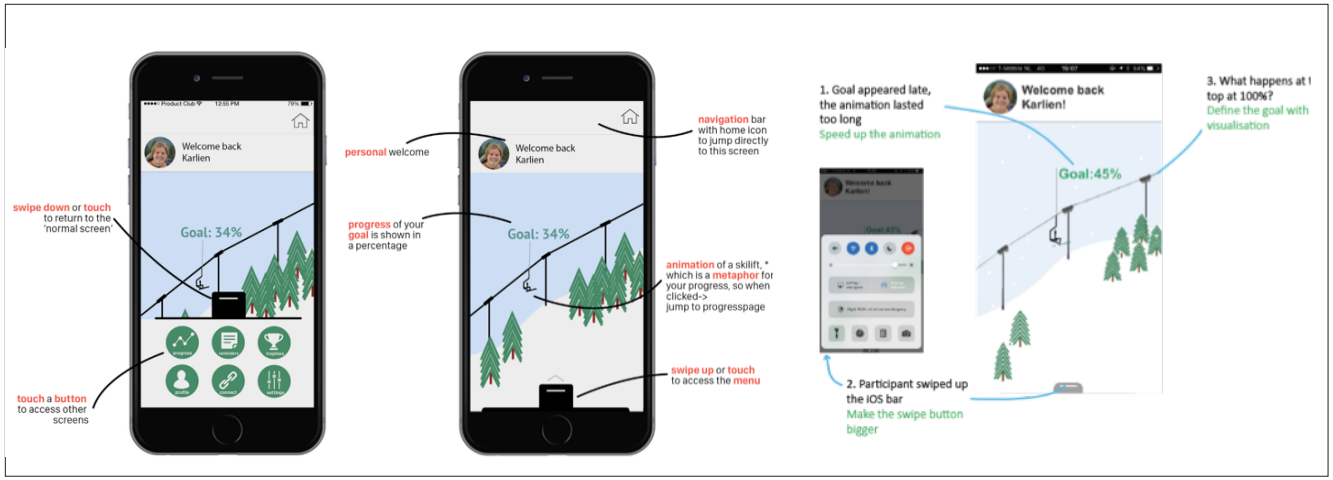



*Figure 1.* *Look & feel annotated prototype. Left: description notes; Right: insights note.*

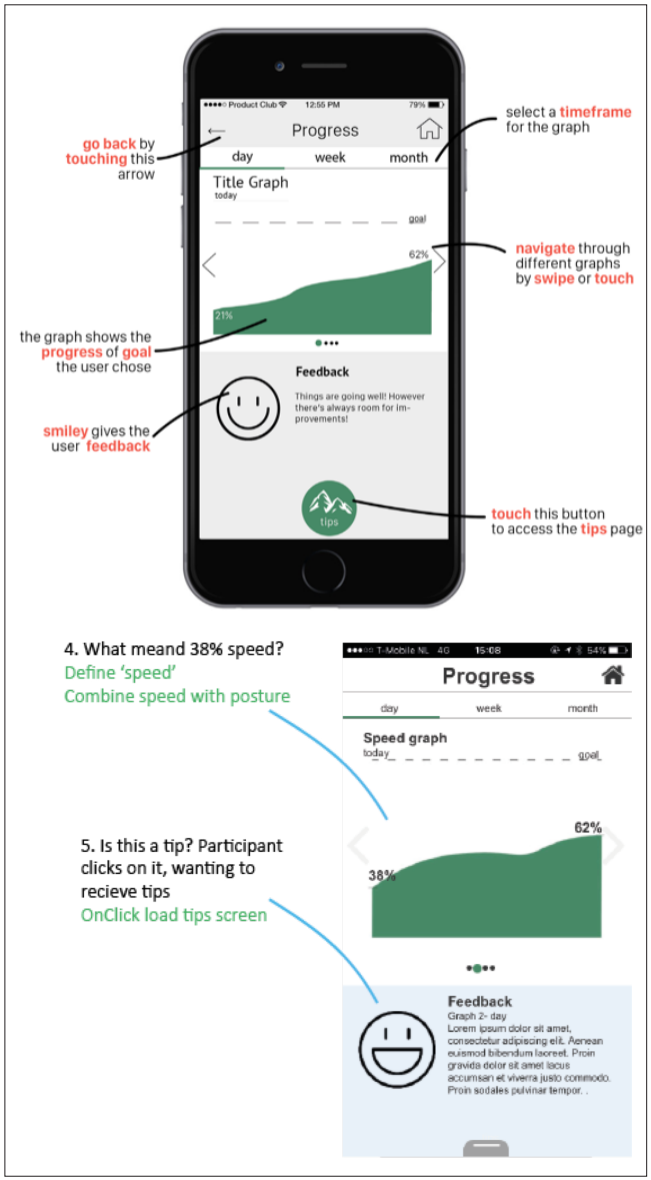


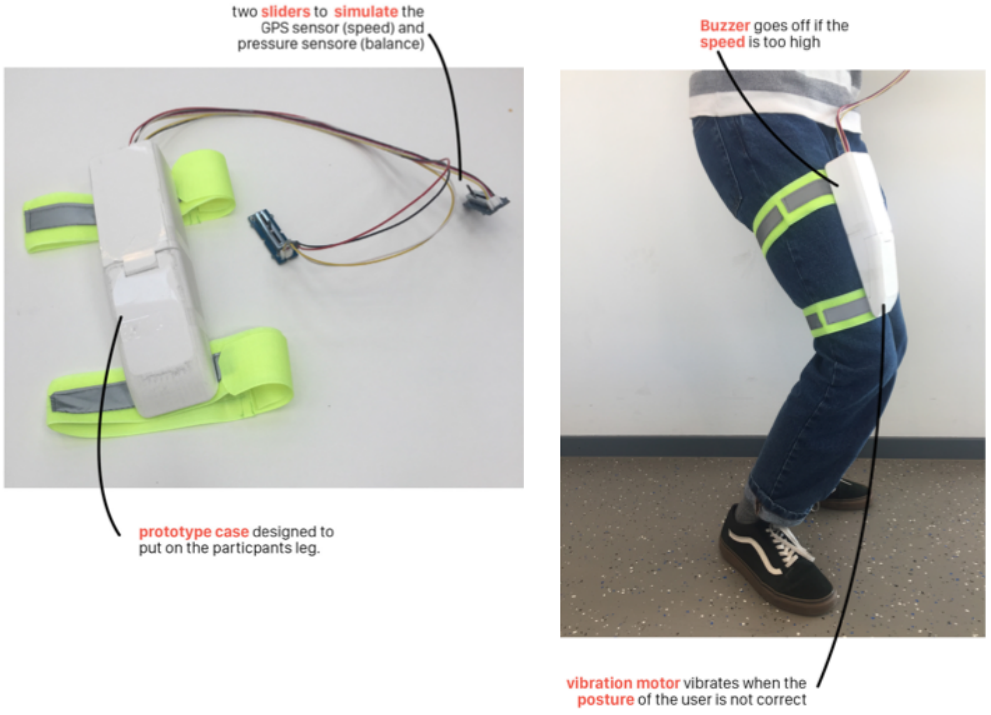

*Figure 2.* *Look & feel annotated prototype. Top: description notes; Bottom: insights note*

# METHOD-CARD

## EXTENDED ANNOTATED PROTOTYPES [2/2]

In addition to the descriptive/explanatory notes, an annotation includes a brief insight note that describes the results from the test related to the indicated element. That is what we called in this course an extended annotation. See Figure 1 and 2 for examples of touchscreen extended annotated prototypes of look & feel. See Figure 3 for an example of physical extended annotated prototype of interactivity.

*Figure 3. Interactivity. Below: description notes; Right: insights note (top– participant reacts on a vibration warning; bottom – participant gets a warning and gets confused)*




**User asked for detailed feedback with a voice** - Both the buzzer as the vibration motor are able to give feedback in just a second and therefore the user is able to react immediately. Using voice message will take at least a few more seconds and by adding the reaction time, the skier will be already past that very moment.This change won't therefore be implemented.

**Control the feedback & frequency (example special training mode)** - if the user is able to control the frequency of the feedback by themselves, the system will be able to adapt to users' progress. Therefore this change is good option to implement.

# METHOD-CARD

## ASSESSMENT AND SELECTION/INTEGRATION OF ALTERNATIVES [1/1]

### INTRODUCTION

*"Design is a process of diverging and converging. The design of a product grows from a product idea via solution principles, concepts and preliminary designs to a detailed definitive design. Design is also a process of working from a large number of ideas to a single detailed design. Designing without intuitive decisions is inconceivable. But for new, complex or unknown decision problems, intuitive decision-making is not always successful. Decision methods aim to help people in making a decision. In decision methods, you compare alternatives on predefined criteria. You look at how well an alternative performs 'on the criteria' and assign a value to this performance. By bringing together the totality of the values of each of the criteria, you calculate an overall score of the alternative. Calculating the overall scores of each of the alternatives and comparing the alternatives facilitates a decision-making process. This is what decision methods are about."* (extract from the Delft Design Guide).

### DESCRIPTION

As a team you want to assess which of the alternatives generated by each member in the conceptualisation phase. You want to evaluate whether the prototyped interactions a) connect to your vision & requirements and b) fit IxD principles to be clear and useful for your target group. There are a number of decision methods to help you compare and select between several design alternatives. Taking as reference the "Delft Design Guide" for this course, three methods are recommended: **Harris Profile, Datum and Weighted Objectives Methods**. Please refer to the "Delft Design Guide" for a description and step-by-step guidance to choose and use one of them. In Table 1 you can find the basic set of criteria that you are expected to use for this assessment. You are welcome to extend the set by justifying your changes.

### REFERENCE

> *'Harris Profile'*, in Delft Design Guide, pp 139
> *'Datum'*, in Delft Design Guide, pp. 147
> *'Weighted Objectives'*, in Delft Design Guide, pp. 151

*Table 1. Basic set of evaluation criteria for touchscreen*

| | A | B | C ... |
|---|---|---|---|
| **Layout** | | | |
| Relevant and consistent grid and navigation structure | | | |
| Specific and attractive grid and navigation structure | | | |
| Distinctive and consistent clustering of informative and functional elements | | | |
| **Look & feel** | | | |
| Logic, clean and consistent visual and graphical elements | | | |
| Neat, appealing and well-balanced visual and graphical elements | | | |
| **Interactivity** | | | |
| Creative and engaging user interaction and system actions | | | |
| Relevant and consistent interaction style | | | |
| Specific and original interaction style in relation to the target group and context of use | | | |

Since you work as a team in prototyping the layout, look & feel and interactivity of the physical device, you may not have alternatives to compare. Still, it is highly suggested to run an assessment of the outcomes, using an adapted version of the criteria in table 1. The aim is to assess which aspects are strong and which may still require some work.

# METHOD-CARD

## IMPLEMENTATION [1/4]

### INTRODUCTION

The following method will guide you to translate the Acting machine Diagram (AMD) to Arduino-grove programming code. The first step is to move the Acting Machine Diagram to a linear table – a State Transition Table (STT). The STT helps you to turn an AMD into a logic that fits programming, and to add some technical details. Then, from the STT to the programming code, you can make the connections more easily.

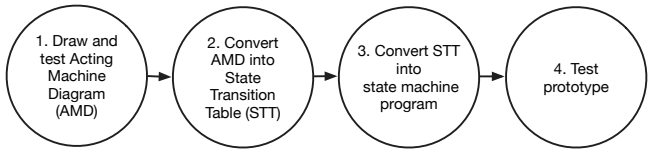



*Figure 1. The 4-step process of developing a working prototype*

### CONVERT AMD INTO STATE TRANSITION TABLE (STT)

To progress from AMD to a working prototype we require two more activities (see Figure 1). The creative heavy lifting has been done and the AMD describes a tested and tried a behaviour, next we convert the AMD into the State Transition Table (STT). This is a helpful intermediate step towards arriving at computer instructions that can be entered in a in a text editor or an Integrated Development Environment (such as the Arduino IDE).

We start with revisiting the complete Pacman AMD (figure 8 AMD method-Card) and from it we create the following table by assigning one row to each transition. As you can see it is a pretty straightforward process we start with the initial state and then follow all transitions out of that state in sequence until we are done for all states. If a state has more than one outbound transition we group them all for that state. Subsequently we order all transitions of one state such that transitions that result in a state change go first and all the in-state transitions go last.

*Table 1. State Transition Table for the AMD of Figure 8 in AMD method- card*

| WHEN GHOST IN STATE | IF-CAUSE | THEN-EFFECT | CHANGE TO THIS STATE |
|---|---|---|---|
| captive | after some time | ghost is released | angry |
| angry | pacman eats power pellet | show scared ghost | scared |
| | collision with pacman | kill pacman | angry |
| | nothing happened | move towards pacman | angry |
| scared | pacman's strength worn off | show angry ghost | angry |
| | collision with pacman | ghost dies | dying |
| | nothing happened | move away from pacman | scared |
| dying | ghost completely faded | show new captive ghost | captive |
| | nothing happened | ghost fades away more | dying |

### CONVERT STT INTO STATE MACHINE PROGRAM

The actual game Pacman runs on well know game computers such as the Atari systems, Apple II, ZX Spectrum etc. For the sake of our example here will target an Arduino style computer and use a collection of system instructions that we will not work out completely. It is possible to fit a display to an Arduino and show Pacman and the ghost, but including here all the details of make the screen work obfuscates the purpose of this text. Instead we want to show you here how to structure a program from the Pacman AMD

The collection of system instructions making the Pacman behaviour work is:

*Table 2. The collection of system instructions to make the Pacman game work on our game computer.*

| SYSTEM INSTRUCTION | EXPLANATION |
|---|---|
| movePacman() | move the pacman character according to the last joystick input |
| moveGhost() | move the ghost character according to the ghost behaviour |
| changeIcon(iconName) | change the icon appearance of a game character on the screen |
| joystickInput(): returns true / false | when user is manipulating the joystick, true is returned |
| buttonPress() : returns true / false | when the user presses the button to respawn a new Pacman character |
| startTimer(timerName) | reset the named timer |
| timerExpired(timerName) : returns true / false | check whether the name timer has expired |
| respawnPacman() | create a new Pacman character in the game |
| killPacman() | remove Pacman from the field |
| pacmanStrong() : returns true /false | check wether Pacman ate a power pellet |
| collisionPacmanGhost() : returns true/ false | check if Pacman managed to catch a ghost or vice versa |

Next, we rewrite the first STT into a version that is closer to the system instructions to ease the step of writing the actual code. Here you can see that all the cause-functions return a value of “true” or “false” (also known as boolean functions). The effect-functions do not return a value, instead they change some aspect of the game.

*Table 3. The AMD rewritten into code language, all outbound transitions have been grouped for that transition to make it clear how to structure the state machine*

| WHEN GHOST IN STATE | IF-CAUSE | THEN-EFFECT | CHANGE TO THIS STATE |
|---|---|---|---|
| captive | timeExpired(ghostTimer) == true | changeIcon(angryGhost) | angry |
| | pacmanStrong() == true | changeIcon(scaredGhost) | scared |
| angry | collisionPacmanGhost() == true | killPacman() | angry |
| | // do nothing | moveGhost() | angry |
| | pacmanStrong() == false | changeIcon(angryGhost) | angry |
| scared | collisionPacmanGhost() == true | changeIcon(dyingGhost), startTimer(dyingGhost) | dying |
| | // do nothing | moveGhost() | scared |
| dying | timerExpired(dyingGhost) == true | changeIcon(captiveGhost), startTimer(ghostTimer) | captive |
| | // do nothing | changeIcon(fadeGhost) | dying |

With STT in table 3 we can open the code editor and start to type a program. In Figure 2. you can find an overview of the structure of a typical Arduino state machine. This is the base code structure that we will make the state machine in..

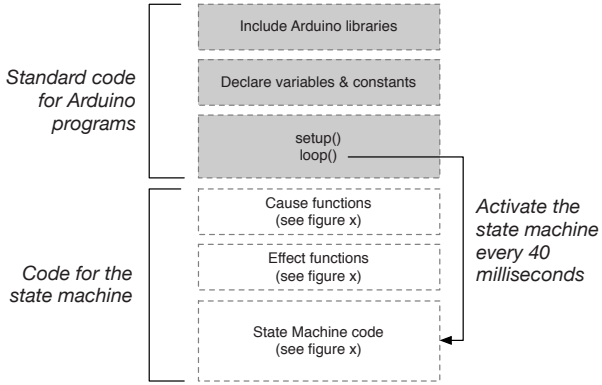



*Figure 2. On the left, the state machine program at-a-glance. On the right, the same in Arduino code. In this example we optimise activating the state machine code so we can control how often per second it is activated.*

```
// ---------------
// -- Libraries --
// ---------------

// ---------------
// -- Variables --
// ---------------

unsigned long timePoint = 0;    // store a point in time
unsigned long timeSpan = 40;    // activate machine every 40 msec

// ------------------
// -- Setup & Loop --
// ------------------

void setup() {
    timePoint = millis();
}

void loop() {
    // Every timeSpan msec do
    if (millis() - timePoint > timeSpan) {
        updateStateMachine();   // activate the state machine
        timePoint = millis();   // start a new interval
    }
}

// ------------
// -- Causes --
// ------------

// -------------
// -- Effects --
// -------------

// -------------------
// -- State Machine --
// -------------------

void updateStateMachine() {
}
```

We start writing code by declaring the variables that we need to define the states in the STT (see table 3) and to remember in which state the machine is currently.

```
// ---------------
// -- Variables --
// ---------------
unsigned long timePoint = 0;    // store a point in time
unsigned long timeSpan = 40;    // activate machine every 40 msec

const int stateCaptive = 0;
const int stateAngry = 1;
const int stateScared = 2;
const int stateDying = 3;
int currentState = stateCaptive;// store the state of the machine
```

*Figure 3. Defining constants to be able to refer to state with meaningful names and declaring the variable to remember the current state of the machine. This variable is initially set to the captive ghost state.*

Next, we move into writing the updateStateMachine() function (see Figure 4 on the next page). There are a number of ways to go about it, here we choose to use the switch-case control structure (see the Arduino language reference if you have not worked with this before). Alternatively, you can also write it completely with if-then-else control structures but it would be less legible, which makes the code more error-prone and more difficult to change.

```
// ------------------
// -- State Machine --
// ------------------

void updateStateMachine() {
    switch(currentState) {
        case stateCaptive:
            if (timerExpired(ghostTimer) == yes) {
                currentState = stateAngry;
            }
            break;

        case stateAngry:
            if (pacmanStrong() == yes) {
                changeIcon(scaredGhost);
                currentState = stateScared;
                break;
            }
            if (collisionPacmanGhost() == yes) {
                killPacman();
            }
            moveGhost();
            break;

        case stateScared:
            break;

        case stateDying:
            break;
    }
}
```

*Figure 4. A partially implemented state machine. The captured and angry states are filled out in the two respective switch-cases. The scared and dying states have not been written yet, but still the code is sound and can be tested for the captured and angry states..*

Each state listed in the STT in Table 3, has to be represented with one switch-case, let's start with the first one. The transition listed for the captive state is about the ghost being released from prison. When the ghost has been in prison long enough it will start chasing Pacman, in the code we use the timerExpired() function and pass the ghost timer identifier as a parameter (see also Figure 5. for cause function explanation). As long as the timerExpired() function returns false the state machine will remain in stateCaptive. As soon as timerExpired() returns true, the currentState is changed to stateAngry and the control flows back to the loop() (this is what break does).

Now entering the updateStateMachine() function the next time around, control will end up in the second switch-case, the one for stateAngry. Referring back to the STT there are 3 transitions listed for stateAngry. Firstly, when Pacman eats a power pellet, we have to change the behaviour of the ghost (and change the state of the machine to stateScared). The other two transitions are in-state transitions: when the ghost manages to catch Pacman, we remove Pacman from the game and there is moving the ghost. Remember that order of checking for causes is important here: if a collision should occur at the same time as Pacman eats a large pellet, it makes a lot of difference if you first check the arrow for collisions (Pacman dies) or first check the arrow for eating the pellet (ghost dies upon the next round of checking).

In general, it makes sense perform the in-state transitions last and check for any user experience issues with the order of the code for transitions that are on a similar level. In addition, after assigning a new value to the currentState value it makes sense to use the break statement to end the switch-case right then and there. To avoid performing actions that are listed in the code after that.

We leave implementing the stateScared and stateDying switch cases as an excersise for the reader.

Next up is the writing the cause function. It will always return a value depending on whether the cause under investigation is met by a user action or a game event (user presses a button or Pacman eats a power pellet). When the cause condition is met, we activate the transition and execute the corresponding effect function(s) if any.

```
// ------------
// -- Causes --
// ------------

const int ghostTimer = 0;
unsigned long ghostTimerDuration = 5000;
unsigned long ghostTimerMark;

boolean timerExpired(int timerID)
{
    switch (timerID) {
        case ghostTimer:
            if (millis() - ghostTimerMark > ghostTimerDuration) {
                return true;
            }
            break;
        default:
    }
    return false;
}
```

*Figure 5. An example of a cause function. This example shows a function that uses the timerID passed as an argument at the time of invocation, to determine if that specific timer has expired. Here it means that the ghost has been captive for more than 5 seconds (5000 milliseconds). Note that we list the variable declarations required to make the timerExpired() function work just in the lines above the function definition. Alternatively, these could also be listed in top variable section at the top of the complete program, to keep all the global variable declarations neatly together.*

An effect function has to make an action in the behaviour noticeable for the user. As such it does not have to return a value.

```
// -------------
// -- Effects --
// -------------

void startTimer(int timerID)
{
    switch (timerID) {
        case ghostTimer:
            ghostTimerMark = millis();
            break;
        default:
    }
}
```

*Figure 6. An example of the effect function to reset a timer. In our AMD of Figure 8 in the AMD method-card. this is used when the ghost is respawned into captivity when Pacman has managed to kill it. This function uses some of the variables list in Figure 4.*

```
// ---------------
// -- Libraries --
// ---------------

// ---------------
// -- Variables --
// ---------------
unsigned long timePoint = 0;     // store a point in time
unsigned long timeSpan = 40;     // activate machine every 40 msec

const int stateCaptive = 0;
const int stateAngry = 1;
const int stateScared = 2;
const int stateDying = 3;
int currentState = stateCaptive;// store the state of the machine

const int ghostTimer = 0;        // variables to time the captive ghost
unsigned long ghostTimerDuration = 5000;
unsigned long ghostTimerMark;

// ------------------
// -- Setup & Loop --
// ------------------

void setup() {
  // Add setup code here
  timePoint = millis();
}

void loop() {
    if (millis() - timePoint > timeSpan) {
      updateStateMachine();
      timePoint = millis();
    }
}

// ------------
// -- Causes --
// ------------

boolean timerExpired(int timerID)
{
    switch (timerID) {
        case ghostTimer:
            if (millis() - ghostTimerMark > ghostTimerDuration) {
                return true;
            }
            break;
    }
    return false;
}

// -------------
// -- Effects --
// -------------

void startTimer(int timerID)
{
    switch (timerID) {
        case ghostTimer:
            ghostTimerMark = millis();
            break;
    }
}
```

```

// --------------------
// -- State Machine --
// --------------------

void updateStateMachine() {
    switch(currentState) {
        case stateCaptive:
            if (timerExpired(ghostTimer) == true) {
                currentState = stateAngry;
            }
            break;

        case stateAngry:
            if (pacmanStrong() == true) {
                changeIcon(scaredGhost);
                currentState = stateScared;
                break;
            }
            if (collisionPacmanGhost() == true) {
                killPacman();
            }
            moveGhost();
            break;

        case stateScared:
            break;

        case stateDying:
            break;
    }
}
```

*Figure 7. All of the state machine put together.*

## ADVANCED TOPIC

It is perfectly possible to combine the state machine for the ghost behaviour and a state machine for the Pacman behaviour. If you create a STT for the Pacman behaviour and write a similar state machine for Pacman, both of them can run side-by-side in the loop() Figure 8..

Notice that in the code included here, nowhere is the function delay() used. Rather we use the millis() and some variables to mark the passing of some time. Using the Arduino delay() in these type of behaviours is not going work. The delay() function completely blocks the Arduino computer for some amount of time. So if we delay the ghost for 5 seconds while it is captive, Pacman will freeze and we send the game player on an unwanted break.

```
// ---------------
// -- Libraries --
// ---------------

// ---------------
// -- Variables --
// ---------------
unsigned long timePoint = 0;        // store a point in time
unsigned long timeSpan = 40;        // activate machine every 40 msec

// ------------------
// -- Setup & Loop --
// ------------------

void setup() {
    timePoint = millis();
}

void loop() {
    // Every timeSpan msec do
    if (millis() - timePoint > timeSpan) {
        updateGhostStateMachine();  // activate the ghost state machine
        updatePacmanStateMachine(); // activate the ghost state machine
        timePoint = millis();       // start a new interval
    }
}

// ------------
// -- Causes --
// ------------

// -------------
// -- Effects --
// -------------

// --------------------
// -- State Machines --
// --------------------

void updateGhostStateMachine() {
}

void updatePacmanStateMachine() {
}
```

*Figure 8. A more complete implementation of the Pacman game, with a state machine for the ghost and Pacman*

## REFERENCES

Greenberg, S., Carpendale, S., Marquardt, N. and Buxton, B., 2011. *Sketching user experiences: The workbook.* Elsevier.

Thimbleby, H., 2010. *Press on: principles of interaction programming.* The MIT Press.

# METHOD-CARD

## USABILITY & UX EVALUATION PLAN [1/3]

### INTRODUCTION

Why to evaluate a design concept with the target group and in the real context of use? For many unique reasons. You can evaluate the user experience of interacting with your design, the usability of your design, the intended impact on your target group, etc. In this project, since there are limited resources to run an extensive evaluation, your focus will be mainly on usability issues and user experiences (as envisioned in your interaction scenario and UX vision) of your design, including both the touchscreen and the physical interface.

### DESCRIPTION

There are 8 steps that you will follow to conduct a successful user evaluation, from preparing an evaluation plan to identifying and fixing relevant problems. These steps are:

1. Develop a **user evaluation plan** including the goal and research questions, the setup (materials, participants, setting), the protocol, user tasks, the test measurements and data collection.
   a. **Study goal & research questions**: In this project, there are three general goals to evaluate. It's your responsibility to make them specific to your interface tasks and define the appropriate research questions. The three goals are:
      - Determine if your subjects are able to complete the tasks using the interface.
      - Identify issues that your subjects have in attempting to perform these tasks.
      - Obtain a qualitative understanding of your subjects' experience when using your interface to perform the tasks as you envisioned earlier in your interaction scenario.

   b. **Setup**: Describe the context where you intend to perform the test (environment) and what will be the setting: which materials are you going to use to perform the test. Describe also the type of users and how many you intend to involve (Note: you should consider at least 3 participants).

   c. **Protocol**: Describe step-by-step how are you going to perform the tests. Make sure you include:
      - The introduction to your test subjects.
      - How are you going to ask the test subjects to perform the tasks.
      - Wrap-up discussion with the test subject.
      - How you plan to record information from the study.

   d. **User task & test scenarios**: The following steps will help you to define relevant user tasks and translate them into test scenarios to introduce to your participants:
      - List the task concerning the three user engagement scenarios and the adaptive feedback scenarios. E.g. glance – user checks daily progress; engage – user compares daily progress with weekly overview; adaptive feedback – user accepts a final warning from a faster and faster blinking LED.
      - Prioritise tasks by giving points: 3 points to the tasks that most users will do most of the time; 2 to the tasks that users will do occasionally; 1 to the tasks that will be performed once or twice. Select at least 2 tasks that score 3, and 1 task that score 2.
      - Translate your selected tasks into test scenarios. The scenarios should be easy to follow (use simple words). They should describe the context and the details of the tasks without giving clues on how to do it (avoid word-finding-game). See Table 1 for examples of BAD, OK, GOOD scenarios.

   e. **Test measurement and data collection**: indicate what **metrics** you will use to identify whether users succeed or not to complete a task, e.g. number of clicks done, time spend, time idle (not doing anything), etc. Consider as well, what **cues** you want to observe to assess user's experience, for example: face and verbal expressions, hands movement, body posture, tone of voice, etc. Design as well a **brief interview** to obtain user assessment on overall and specific experience with the product. Indicate the **way you will collect all that information** and who will be doing what. Make sure that each team member has one and only one specific activity to do: one member is only observing and listening, another is writing notes in a template, having a team member counting clicks, using a time watcher to count seconds, etc.

## USABILITY & UX EVALUATION PLAN [2/3]

f. **Ethical check:** To assess the ethical implications of your study, IDE offers a platform for you to learn about **ethics in** education research. Read the information presented and fill in the form to assess the ethics implications of your plans. Prepare as well a consent form to make clear agreements with participants about their obligations and rights when performing the study (see Brigthspace for example: 'Content/Materials/Peer testing'). Include the form in your user evaluation plan.

2. **Recruit participants**: Search for people that are representative to your target group to test your product in the context chosen. Start your search early. Make clear appointments with participants and be clear about what they can expect. Make them feel comfortable and express how valuable their input will be.
3. **Run a pilot**. Conduct a pilot test and consent form with one or two peers, to check if your tasks are clear, your protocol is solid, and that you will be collecting the expected data. Make improvements before starting with your participants.
4. Introduce the test to the participant. Once the instructions are clear and consent form is signed you can start with the test..
5. Force yourself to **shut up, and behave professional**. Focus on observing and avoid to all extend laughing or talking to other team members when running the test. Only remind participants to think out loud. Note: if a participant is stuck for too long, prevent him/her for getting stressed and provide some hints (make sure you take notes on that.
6. Conduct **interviews** with participants soon after they have finished all tasks.
7. When all sessions have been completed, **analyse your data** in order to answer your research questions. For the usability test, you will provide at least two outcomes:

a. a list of **usability problems.** Select the top 3-5 issues per participant. Then visualise the frequency of the top issues (see Figure 1). The issues will come from the experiences that participants have reported when 'thinking aloud' and from their answers in the to the interview questions..

b. a **task completion table** to document how well participants complete the tasks. See Table 2 for example and template..

8. Identify the three most relevant problems from the identified top usability issues and tasks with higher difficulty. Implement quick fixes. A quick fix, should not require big effort to implement and should serve as input for future iterations towards the development of a final design. See Fig. 2 for examples.

### EXAMPLE

**BAD, OK, GOOD SCENARIOS**

*Table 1. Examples of BAD, OK and GOOD scenarios*

| SCENARIO FOR THE TASK 'UPLOAD A PHOTO' | SCENARIO FOR THE TASK 'UPDATE YOUR STATUS' | SCENARIO FOR THE TASK 'SCROLL THROUGH NEW POSTS IN FACEBOOK' |
|---|---|---|
| > Find a way to upload a photo.<br>• BAD - it provides clues, it lacks of user goal, and context.<br>> Find a way to share some pictures with your friends.<br>• OK - it missess the user goal.<br>> Last night you were at a party and took some funny pictures and now you're looking for a way to share them so your friends can see them as well.<br>• GOOD - it provides a detailed context and user goal. Does not provide clues. | > Write a status update and post it to your profile page.<br>• BAD - it provides clues, it lacks of user goal, and context.<br>> Let your friends know what you're doing by updating your status.<br>• OK - it provides user goal, it lacks of context. It gives away clues.<br>> Find a way to let your friends know what you're doing and tell them you'r currently testing a website.<br>• GOOD - it provides no clues, it provides specific user goal, and detail information to perform the task. | > Scroll through the page to look at new posts.<br>• BAD - it provides clues, it lacks of user goal, and context.<br>> Look at this page and find out what's all about it.<br>• OK - it provides no clues, it hints a user goal, it lacks of detail context,.<br>> Imagine this is the first time you're checking in on Facebook today. Now go and find the first post that was published today.<br>• GOOD - it provides no clues, it provides a real user goal, specific enough to trigger natural motivation to do the task. |

# METHOD-CARD

## USABILITY & UX EVALUATION PLAN [3/3]

### FREQUENCY OF USABILITY ISSUES

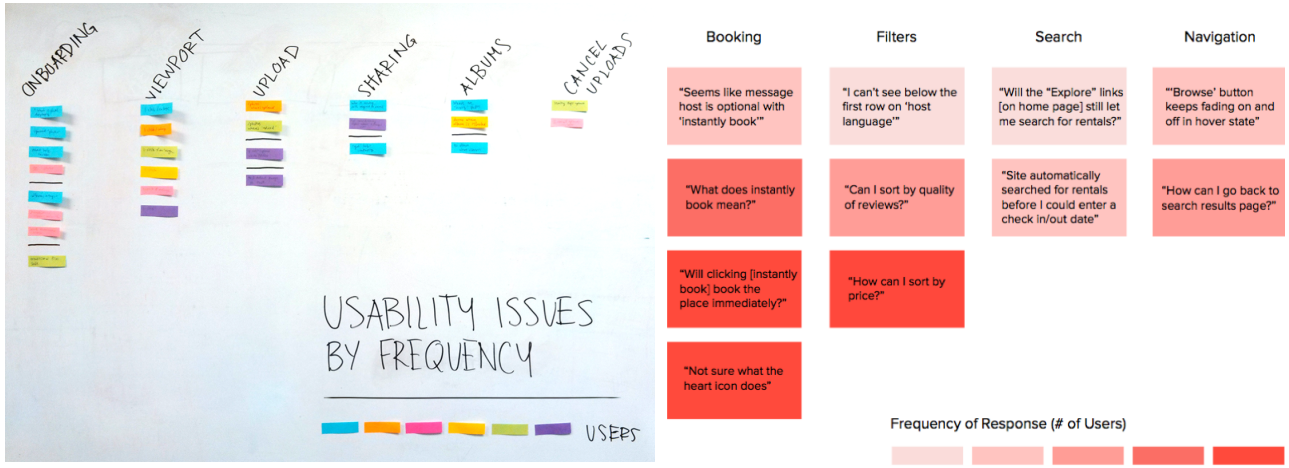



*Figure 1. Examples of visualising frequency of usability issues*

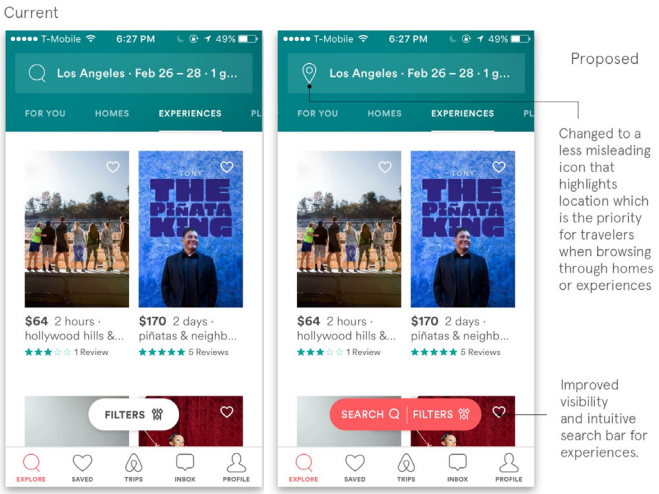



*Figure 2. Example of Quick Fixes*

### TASK COMPLETION TEMPLATE

*Table 2. Task completeion table and example*

| | Upload image TASK 1 | Post message TASK 2 | Share photo TASK 3 | Add a friend TASK 4 | Delete a post TASK 5 | Change profile TASK 6 | Logout TASK 7 | Logout TASK 8 |
|---|---|---|---|---|---|---|---|---|
| Tester 1 | 3 | 2 | 3 | 3 | 2 | 2 | 3 | 3 |
| Tester 2 | 3 | 3 | 3 | 3 | 3 | 2 | 3 | 3 |
| Tester 3 | 3 | 3 | 3 | 2 | 3 | 1 | 3 | 3 |
| Tester 4 | 3 | 2 | 3 | 3 | 3 | 1 | 2 | 2 |
| Tester 5 | 2 | 2 | 3 | 1 | 1 | 1 | 2 | 1 |
| SUM | 14 | 12 | 15 | 12 | 12 | 7 | 13 | 12 |

3: User can perform task quickly and with no trouble
2: User can perform task, but has some struggles
1: User can't perform task

NOTES

3 BIGGEST PROBLEMS

## REFERENCE MATERIAL

### BOOK CHAPTERS

> Ch. 5 *'The craft of interaction design'* in Dan Saffer's book, pp. 117-119
> *'Interaction prototyping and evaluation'*, in Delft Design Guide, pp. 131
> *'Product usability evaluation'*, in Delft Design Guide, pp. 133.

### IXD WEBSITES

> Usability evaluations
> Planning usability testing
> Guerrilla Usability testing

## 7. ASSESSMENT FORMS

### ASSESSMENT FORM - VISION & REQUIREMENTS

DELIVERABLES: Vision & requirements outputs

TEAM:

COACH:

| | |
|---|---|
| **LO.1 - Validation** | |
| **Setup, conduct and analyse user studies by applying user research methods**<br><br>**30%** | **Design practice:** validation provides realistic and relevant (accurate, vivid, rich, specific and distinctive) insights connected to the research plan. **Methods:** user research plan is designed and conducted in a convincing and consistent (structure, analytical, ethical, connected, continuity) manner, collecting rich and attractive data material. |
| | **Design practice:** validation provides to an extend realistic and relevant (accurate, vivid, rich, specific and distinctive) insights connected to the research plan. **Methods:** user research plan is designed and conducted in a convincing and consistent (structure, analytical, ethical, connected, continuity) manner, collecting a complete and inormative data |
| | **Design practice:** validation provides generic and/or incomplete insights connected to the research plan. **Methods:** user research plan is designed and conducted in an unstructured and/or vague manner, collecting generic and/or incomplete data material. |
| **LO.2 - Vision & requirements** | |
| **Define a product system vision and user requirements by applying interaction design methods**<br><br>**50%** | **Design practice:** vision & requirements define realistic and relevant (accurate, vivid, rich, specific and distinctive) user needs, experiences and system behaviour of the product. **Methods:** persona, interaction scenario and UX vision provide a convincing and consistent (structure, analytical, ethical, connected, continuity) representations of target group, context and product; HTA, UTF and AMD provide a convincing and consistent (structure, analytical, ethical, connected, continuity) translation to user and system interaction specifications. |
| | **Design practice:** vision & requirements define to an extend realistic and relevant (accurate, vivid, rich, specific and distinctive) user needs, experiences and system behaviour of the product. **Methods:** persona, interaction scenario and UX vision provide to an extend a convincing and consistent (structure, analytical, ethical, connected, continuity) |
| | **Design practice:** vision & requirements define generic and/or incomplete user needs, experiences and system behaviour of the product. **Methods:** persona, interaction scenario and UX vision provide innacurate and/or incomplete representations of target group, context and product; HTA, UTF and AMD provide basic, innacurate and/or incomplete |
| **LO.6 - Internal & external reflection and communication** | |
| **Evaluate, reflect and present processes and outcomes by applying collaborative, analytical and visual communication skills**<br><br>**20%** | **Design practice:** communication is proactive and transparent with coach and team members. Reflection on project and individual level is actionable. **Methods:** visual presentation of persona, interaction scenario, UX vision, HTA, UTF and AMD are integrated and self-explanatory. |
| | **Design practice:** communication is efficient and open with coach and team members. Reflection on project and individual level is constructive. **Methods:** visual presentation of persona, interaction scenario, UX vision, HTA, UTF and AMD are descriptive and well-balanced. |
| | Design practice: communication is inneficient and vague with coach and team members. Reflection on project and individual level is superficial. Methods: visual presentation of persona, interaction scenario, UX vision, HTA, UTF and AMD are poor and unbalanced. |

## ASSESSMENT FORM - PROTOTYPING OF TOUCHSCREEN & PHYSICAL INTERFACE

DELIVERABLES: Individual Prototyping report & Prototypes

STUDENT:

COACH:

| | | |
|---|---|---|
| **LO.3 - Design alternatives** | | |
| Explore, test and communicate digital and physical interface elements by applying lo-fi and hi-fi prototyping techniques<br><br>50% | Design practice: design alternatives explore realistic and relevant (accurate, vivid, rich, specific and distinctive) aspects of the touchscreen and physical interface. Methods: sketching, lo-fi & hi-fi prototyping and peer-testing of layout, look & feel demonstrate in a convincing and consistent (structure, analytical, ethical, connected, continuity) manner unique interactions and | 9,5-10<br>7,5-9,0 |
| | Design practice: Design alternatives explore to an extend realistic and relevant (accurate, vivid, rich, specific and distinctive) aspects of the touchscreen and physical interface. Methods: sketching, lo-fi & hi-fi prototyping and peer-testing of layout, look & feel demonstrate to an extend in a convincing and consistent (structure, analytical, ethical, connected, continuity) manner | 6,0-7,0 |
| | Design practice: Design alternatives explore in a limited or absent way realistic and relevant aspects of the touchscreen and physical interface. Methods: Sketching, lo-fi & hi-fi prototyping and peer-testing of layout, look & feel demonstrate in a scattered and/or inconsistent manner generic and/or incomplete interactions and feedback. | 5,0-5,5<br><=4,5 |
| **LO.4 - Assessment / design decisions** | | |
| Assess and integrate design decisions by applying interaction design methods<br><br>30% | Design practice: Assessment represents realistic and relevant (accurate, vivid, rich, specific and distinctive) integration of design decisions on layout, look & feel and interactivity. Methods: sketching, lo-fi & hi-fi prototyping and peer-testing demonstrate in a convincing and consistent (structure, analytical, ethical, connected, continuity) manner neat, appealing and well-balanced | 9,5-10<br>7,5-9,0 |
| | Design practice: Assessment represents to an extend realistic and relevant (accurate, vivid, rich, specific and distinctive) integration of design decisions on layout, look & feel and interactivity. Methods: sketching, lo-fi & hi-fi prototyping and peer-testing demonstrate to an extend in a convincing and consistent (structure, analytical, ethical, connected, continuity) manner logic, clean and consistent interfaces. | 6,0-7,0 |
| | Design practice: Assessment represents generic and/or incomplete integration of design decisions on layout, look & feel and interactivity. Methods: sketching, lo-fi & hi-fi prototyping and peer-testing demonstrate in an unbalanced and/or confusing manner scattered and/or inconsistent interfaces. | 5,0-5,5<br><=4,5 |
| **LO.6 - Internal & external reflection and communication** | DOCUMENTATION | |
| Evaluate, reflect on and present processes and outcomes by applying collaborative, analytical, and visual communicaion skills.<br><br>20% | Design practice: communication is proactive, transparent with coach and team members. Reflection on project and individual level is actionable. Methods: Report and extended annotated prototypes are integrated and explanatory. | 9,5-10<br>7,5-9,0 |
| | Design practice: communication is efficient and open with coach and team members. Reflection on project and individual level is constructive. Methods: report and extended annotated prototypes are descriptive and well balanced. | 6,0-7,0 |
| | Design practice: communication is inneficient and vague with coach and team members. Reflection on project and individual level is superficial. Methods: report and extended annotated prototypes are poor and unbalanced. | 5,0-5,5<br><=4,5 |

# APPENDICES

## ASSESSMENT FORM - TEAM DESIGN PROJECT

**DELIVERABLES: Final Project Design report & Presentation & Final Demonstrator**

TEAM:
COACH:

| | | |
|---|---|---|
| **LO.1 - Validation & Evaluation** | | |
| **Setup, conduct and analyse user studies by applying user research methods**<br><br>**20%** | **Design practice:** validation & evaluation provide realistic and relevant (accurate, vivid, rich, specific and distinctive) insights connected to the research plan. **Methods:** user research & user evaluation plan are designed and conducted in a convincing and consistent (structure, analytical, ethical, connected, continuity) manner, collecting rich and attractive data material. | 9,5-10<br>7,5-9,0 |
| | **Design practice:** validation & evaluation provide to an extend realistic and relevant (accurate, vivid, rich, specific and distinctive) insights connected to the research plan. **Methods:** user research & user evaluation plan are designed and conducted in a convincing and consistent (structure, analytical, ethical, connected, continuity) manner, collecting a complete and inormative | 6,0-7,0 |
| | **Design practice:** validation & evaluation provide generic and/or incomplete insights connected to the research plan. **Methods:** user research & evaluation plan are designed and conducted in an unstructured and/or vague manner, collecting generic and/or incomplete data material. | 5,0-5,5<br><=4,5 |
| **LO.2 - Vision & requirements** | | |
| **Define a product system vision and user requirements by applying interaction design methods**<br><br>**20%** | **Design practice:** vision & requirements define realistic and relevant (accurate, vivid, rich, specific and distinctive) user needs, experiences and system behaviour of the product. **Methods:** persona, interaction scenario and UX vision provide a convincing and consistent (structure, analytical, ethical, connected, continuity) representations of target group, context and product; HTA, UTF and AMD provide a convincing and consistent (structure, analytical, ethical, connected, continuity) translation to user and system interaction specifications. | 9,5-10<br>7,5-9,0 |
| | **Design practice:** vision & requirements define to an extend realistic and relevant (accurate, vivid, rich, specific and distinctive) user needs, experiences and system behaviour of the product. **Methods:** persona, interaction scenario and UX vision provide to an extend a convincing and consistent (structure, analytical, ethical, connected, continuity) representations of target group, context and product; HTA, UTF and AMD provide to an extend a convincing and consistent (structure, analytical, ethical, connected, continuity) translation to user and system interaction specifications. | 6,0-7,0 |
| | **Design practice:** vision & requirements define generic and/or incomplete user needs, experiences and system behaviour of the product. **Methods:** persona, interaction scenario and UX vision provide innacurate and/or incomplete representations of target group, context and product; HTA, UTF and AMD provide basic, innacurate and/or incomplete translation to user and system interaction specifications. | 5,0-5,5<br><=4,5 |
| **LO.4 - Design decisions** | | |
| **Assess and integrate design decisions by applying interaction design methods**<br><br>**20%** | **Design practice:** design decisions represent realistic and relevant (accurate, vivid, rich, specific and distinctive) engaging interactions and feedback of the product. **Methods:** assessment matrixes and user evaluation compare and demonstrate in a convincing and consistent (structure, analytical, ethical, connected, continuity) manner the pros and cons of the product(s) in relation to the target group and context of use. | 9,5-10<br>7,5-9,0 |
| | **Design practice:** design decisions represent to an extend realistic and relevant (accurate, vivid, rich, specific and distinctive) engaging interactions and feedback of the product. **Methods:** assessment matrixes and user evaluation compare and demonstrate to an extend in a convincing and consistent (structure, analytical, ethical, connected, continuity) manner the pros and cons of the product(s) in relation to the target group and context of use. | 6,0-7,0 |
| | **Design practice:** design decisions represent an unbalanced and/or confusing interactions and feedback of the product. **Methods:** assessment matrixes and user evaluation compare and demonstrate in a limited and/or absent manner the pros and cons of the product(s) in relation to the target group and context of use. | 5,0-5,5<br><=4,5 |
| **LO.5 - Demonstration** | | |
| **Implement, evaluate and demonstrate design decisions by applying hi-fi prototyping techniques**<br><br>**20%** | **Design practice:** final testable prototype demonstrate in a realistic and relevant manner engaging interactions and feedback of the product. **Methods:** touchscreen and physical interactive prototypes are used to evaluate and demonstrate design decisions in a convincing and consistent manner. | 9,5-10<br>7,5-9,0 |
| | **Design practice:** final testable prototype to an extend demonstrate in a realistic and relevant manner engaging interactions and feedback of the product. **Methods:** touchscreen and physical interactive prototypes to an extend are used to evaluate and demonstrate design decisions in a convincing and consistent manner. | 6,0-7,0 |
| | **Design practice:** final testable prototype demonstrate generic interactions and feedback of the product. **Methods:** touchscreen and physical interactive prototypes are unbalanced and/or confusing used to evaluate and demonstrate design decisions. | 5,0-5,5<br><=4,5 |