D2.1

Scenario, Storyboard & Multimodal Experience

Work package	**WP2 Emotional & Psychosocial Serious Game Framework**
Task	**T2.1 Scenarios, Storyboards, & Multimodal Experience**
Editor	**Lara Pittino (FAM)**
(co-)authors	**Lucas Paletta (JRD), Martijn Niessen (MCR) Gert Vander Stichele, Connor Buffel (MBY)**
Public / confidential	**Public**

Project PLAYTIME

The research leading to these results has received funding from the AAL Programme of the European Union and by the Austrian BMVIT/FFG under the Agreement no 857334, ZonMw (the Netherlands) and funding from Brussels (Belgium). It reflects only the author's view and that the Union is not liable for any use that may be made of the information contained therein.

30/08/2017

PLAYTIME partner			organisation
01	JR	JOANNEUM RESEARCH DIGITAL	JOANNEUM RESEARCH Forschungsgesellschaft mbH DIGITAL – Institut für Informations- und Kommunikationstechnologien, 8010 Graz
02	FAM	FameL	FameL GmbH Steinbruchweg 20, A-8054 Seiersberg
03	LEF	bouncingbytes	Lefkopoulos KG Jakob-Redtenbacher Gasse 9, A-8010 Graz
04	SVD	Sozialverein Deutschlandsberg	Sozialverein Deutschlandsberg Kirchengasse 7, A-8543 Deutschlandsberg
05	GGZ	GGzE	Geestelijke Gezondheidszorg Eindhoven en de Kempen Postbus 909, 5600 AX Eindhoven, The Netherlands
06	TIU	Tilburg University / Tranzo	Stichting Katholieke Universiteit Brabant, Tilburg University PO Box 90153, 5000 LE Tilburg, The Netherlands
07	MCR	mcroberts	McRoberts BV. Raamweg 43, 2596 HN The Hague, The Netherlands
08	MBY	mind bytes	MindBytes F. Roosseveltlaan 348-349, B8, 90600 Ghent, Belgium
09	GEU	200 YEARS GHENT UNIVERSITY	Ghent University Sint-Pietersnieuwstraat 25, 9000 Gent, Belgium

Acknowledgement:

The research leading to these results has received funding from the AAL Programme of the European Union and by the Austrian BMVIT/FFG under the Agreement no 857334, the Netherlands Organisation for Health Research and Development (ZonMW) and the Flanders Innovations & Entrepreneurship (VLAIO).

Contents

1 Executive Summary

Multimodal training functionalities and serious gaming should be intelligently and innovatively packaged in Playtime's software. Motoric and social activities have a positive influence on the course of the disease according to scientific studies. In future, the new training assistant will activate, motivate and stimulate cognitive processes. Personalised and standardised exercises are also intended to shed some light on the background of science and medicine as to how, in individual cases, but also in general, the course of the disease develops and which measures and forms of therapy or methods have a positive effect and which are less successful. This would be a first in the history of the measurability of progression of dementia and the disease can thus be treated be in the future. Also of interest is the planned diagnostic system for the evaluation or classification of the dementia grade.

Ultimately, the training assistant from amicasa will accompany the client within the private as well as the medical environment starting after the placement test.

This report describes the most important scenarios of the Serious Game Platform - the interplay of the individual methods and the possibilities of the application as well as the different measurements resulting from movement belt, emotional states, eye tracking etc.

Ultimately, it is about the positive clients 'experiences , the smooth application for clients and trainers, the medical success and the daily training at home or in the care unit - divided into 2 scenarios - the individual and the group training.

The consideration of the target group of those affected and the target group of relatives and caregivers were essential in the development of the storyboard.

The backend processes as well as in the app processes must be easy to understand and must not be a hurdle for the users. The continuation of this topic is to be considered later in relation to GUI.

The aim of Playtime is also the possibility to tailor the process to the personal needs and preferences of the user. Ultimately, the biography of the user should be reflected in the course of the training or serious game.

2 Scenarios

2.1 Scenario conceptual framework

2.1.1 Scenarios

Scenarios can be understood in a variety of ways, several of them have been proved to be useful in systems engineering. A scenario can be a sequence of activities, or a more or less richly branched structure of such sequences. Branches can represent alternatives or parallels, or various intermediate options. A scenario can be concrete or abstract; and it can describe either the world or the machine. Scenarios can be represented in a rich variety of ways, including text, databases, diagrams, video, animations, cartoons, storyboards, acted scenes, and collaborative workshops. All of these may have applications in systems engineering.

Scenarios can be used throughout the system life-cycle to clarify business processes; to set requirements into context for both stakeholders and systems engineers; to validate user requirements before system specification begins; to guide system specification and design; to provide the basis of acceptance test scripts; and to validate suggested tests.

Scenarios come in a bewildering array of forms, and even engineers who broadly agree on what a scenario can be used for may differ on what exactly a scenario is. In some definitions, a scenario is barely different from a description of a static situation; but most scenario engineers have gravitated towards a pragmatic usage where **a scenario means some kind of description of a set of activities**, **most commonly sequential**. This has an obvious bearing on the application of scenarios to describe how a system is to behave when interacting with its users.

A useful distinction can be drawn between concrete and abstract scenarios. A **concrete scenario** involves specific actors, places, and times:

John sets the alarm and locks the door of his chic Clerkenwell warehouse apartment at 8 a.m. on Monday the 2nd October 2000. The alarm detects a possible burglar at 11:15, and signals the HSS call centre at 11:16. Jane, the HSS operator on duty, ...

An **abstract scenario** describes what happens in generic (class) terms:

The Householder sets the alarm, and locks the door. The alarm detects a possible intruder, and signals the call centre. The operator ...

Both types of scenario have their uses in systems engineering. Some engineers (e.g. Leffingwell) strongly advocate making scenarios as concrete and humorous as possible, to bring them to life and enhance communication between stakeholders and engineers. Conversely, **abstract scenarios are closer to traditional requirements**.

Assuming, then, that scenarios are sets of activities, the simplest, and usually the first encountered, kind of scenario is a **straight sequential path from start to finish**, one step (or

task, or activity) after another. Scenarios can however branch; branches can constitute alternative paths, though there is usually one primary, normal, or 'happy day' path that stakeholders would expect. Secondary paths can simply be allowable but less favored choices, or may be followed only in response to specific undesired events, sometimes called exceptions (Alexander). In some scenarios, several paths may be allowed or be required to be followed at once; different degrees of parallelism may be distinguished, from weak – follow one or more paths, to strong - follow all paths at once.

Finally, the distinction between the world and the machine is also relevant to scenarios (Jackson). Initially, during business process modelling and requirements elicitation, scenarios should describe the **(desired) state of events in the world**. Later, during system specification, it may be useful also to construct system-related scenarios describing **intended states of events in the machine**. The popular term 'use case' implies the existence of a system and a point of view that looks outwards from a system to its 'users', suggesting that the approach is most comfortable during system specification. The older term 'operational scenario' conversely implies the existence of operations in the world, and may thus be more satisfactory for thinking about the domain and people's business requirements.

Use case

A use case is a scenario written from a functional point of view, enumerating all the possible user actions and system reactions that are required to meet a proposed system function (Jacobson et al., 1992). Use cases can then be analysed with respect to their requirements for system objects and interrelationships. A variant of it is "**user-system conversation**" using a 2-column format, a scenario is decomposed into a linear sequence of inputs from the user and the corresponding processing and/or output generated by the system. Kaindl (2000) extends this analysis by annotating how scenario steps implement required user goals or system functions.

User interaction scenario

In **scenario-based design** (SBD), the use of a future system is concretely described early in the development process. Narrative descriptions of envisioned usage episodes – user interaction scenarios – are then used in various ways to guide development of the system that will enable these use experiences. (Jacko, 2012).

A **scenario is a sketch of use**, it consists of a setting, or situation state, one or more actors with personal motivations, knowledge and capabilities, and various tools and objects that the actors encounter and manipulate.

The **narrative** describes a sequence of actions and events that lead to an outcome. These actions are related in usage context that includes the goals, plans and reactions of the people taking part in the episode.

Some summary of most important notions used:

- **Scenarios**: "sequences of actions aimed at accomplishing some task goal"
- **Goals**: "partially specified states that the user considers as desirable"
- **Use cases**: "particular cases of how the system is to be used", "classes of scenarios"

- **Functions**: “effects achieved by some entity”
- **Tasks:** “pieces of work that a person or other agent has to (or wishes to) perform”

IEEE Standard for requirements

Requirements express the user wises and needs. The IEEE Standard is as follows, “A condition or capacity **needed by a user to solve a problem** or **achieve an objective**.” “The <system> shall be able to ...” express system to be built, a composite system.

- Example: “The ATM shall accept a cash card.”

Goals

Relationship between representations and scenarios that express activities in terms of goals.

- Member Has Video for Rent **By-Scenario** Rent Available Video

Member Has Video for Rent: A member of the video store has a copy of a video title from the store for rent.

2.1.2 Representations of scenarios

As in other areas of systems and software engineering, there are perhaps more schools of thought concerning notations and representations of scenarios than on any other topic. Among other representations, engineers have proposed Text, Databases, Diagrams, Video, Animations, Cartoons, Storyboards, Acted Scenes, and Collaborative Workshops as ways of describing scenarios. All of these different representations may well be useful in specific situations and with appropriate groups of people.

One key function of scenarios is to help people communicate about the systems they are involved in - whether they have a stake in using systems when they become available, or in developing the systems. Stakeholders are often unfamiliar with the language used by engineers. Scenarios offer a way of bridging the gap and helping to ensure that different groups of people understand each other. The issue of choice of representation would ideally be a purely pragmatic one - people would select the best for their situation. Some 'methods' are quite ideological about notations, others not at all. Here we look as pragmatically as possible at some of the best-known representations.

UML Use Cases

The idea of the Use Case (Jacobson) supposes that the world of business is divided into neatly addressable cases where a user interacts with a system. The use case graphical notation is appealingly simple - a manikin connected to an elliptical bubble. Each bubble means a use case, and a list of several of them can be arranged to form a diagram. But even if they are arranged in a vertical list, as they usually are, this does not imply sequence, according to UML. Visually the arrangement is certainly misleading, and in practice it is natural for UML users to put the first step at the top and the next one after it, if the steps are in fact sequential.

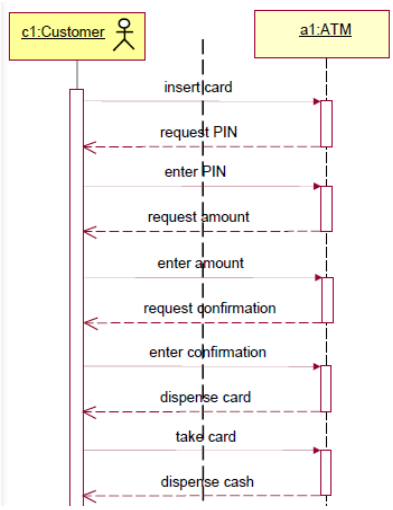


Figure 1. UML sequence diagram.

Flowcharts

Flowcharts are a traditional means of representing scenarios at different scales. They became unfashionable in software engineering through being used early on at very low level to represent program structure. However they are still useful, and remain popular in business process modelling. Modern notations such as UML flowcharts have different symbols, but deliver broadly the same results. Flowcharts tend to emphasise sequentiality and decision-making at the expense of describing roles and system involvement.

Sample Scenario Representation

Scenario “Rent Available Video”:

- A member of the video store identifies himself/herself to VSS (Video Store Software).
- VSS shall check the identification.
    - *By-Function: Member Identification Check*
- If the identification is successful, VSS shall start a transaction and present a selection of video titles.
- The member selects a video title that is available and indicates the intent to rent (a copy of) the video.
- VSS shall book this rental on the account of the member and ask the clerk to hand out a video copy to the member.
- The clerk hands out a copy of the video title and acknowledges this to VSS.
- VSS shall again present a selection of video titles.
- The member does not select further titles, but initiates the termination of the transaction.
- VSS shall issue a good-bye message and terminate thetransaction.

Sequence diagrams

Sequence diagrams have a clear modern notation (one form of which is part of the UML). They are clear and easy to read, and they give actors full weight as part of the process being described, unlike, for instance, dataflow diagrams and flowcharts. They are equally capable of describing single or multiple threads within scenarios providing the number of agents involved does not become unreasonably large.

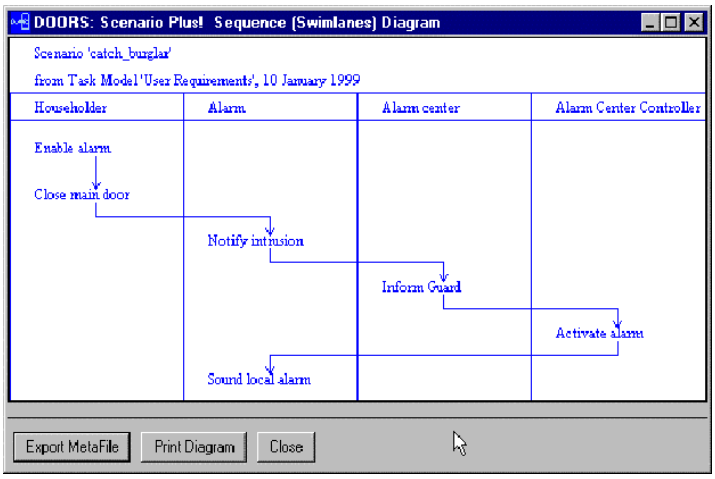


Figure 2. Sequence diagram.

2.2 Tablet based Serious Game Scenario

This point describes how to use the system from a user's perspective. It concerns the procedure and application of the system and methods. The presentation refers to some examples which can be supplemented and extended in this form. In other words, any number of methods and options can be assigned to the heart (software) of PLAYTIME.

Entry and recording of personal data – incl. diagnosis and diagnosis repetition

Man	**Machine**
Data entry (personal data)	Saves on server
Query diagnostics	If missing – suggest Minimental test, otherwise query for repetition
Entry of biographical data	Saves on server
	System proposes difficulty level and training process
Extension of personal data and the personal profile	Saves on system
	System proposes renewed validation
Validation is performed again	Results are included in compilation of further training

Adaptation of user hierarchy

Man	Machine
Granting access rights for the different data types (who can access what)	
Every authorized supervisor (on different levels) creates a new user (user, trainer) with own access rights	Saves new user incl. use and application rights
User manages his/her own data	Saves on system
User creates new clients	Saves on system
User manages his/her own clients	Saves on system
Supervisor has access to user and client data	Saves on system and information goes to user
Supervisor has access to user and client data	Saves on system and goes back to user for confirmation

Input and creating training content

Man	Machine
Input of training data according to categories (red, yellow, blue, orange – movement, music/noise/dancing, perception, memory training/knowledge) by different persons – admins, med. staff, trainers, users, prof. and semi-prof. nursing staff	Saves on server based on respective person, topic, or task
Input of image and video material by uploading previously saved files – Wi-Fi, Bluetooth, cable connection	System creates appropriate file in the appropriate size – request to assign tags and select saving names. Saves on system
Direct capture using terminal (smartphone, tablet, Wi-Fi, camera, etc.) and direct upload	System creates appropriate file in the appropriate size – request to assign tags and select saving names. Saves on system
Manual and automatic assignment of tags to respective content, people, and topics as well as selection of a saving name and location	Saves tags, file names, and links to the location
Extension or modification of existing content	Extension database (procedure as above)
Entry of new content and topics	Extension database (procedure as above)

Compilation of own trainings (with and without consideration of previous diagnosis and while considering experiences of the system's artificial intelligence)

Compilation without diagnosis and recording of previous trainings

Man	**Machine**
Query of single or multiple keywords	System returns tagged results
Selection and individual compilation of one or more topics	System saves the topic in the desired length and order
User can change the order at any time	System saves the new order
User can reduce and/or increase the number of tasks at any time	System saves the new compilation

Compilation with diagnosis and recording of previous trainings

Man	**Machine**
Entry of the client for whom the training should be created	System displays all entries for the selected person
Entry of framework conditions that apply to the training unit (e.g., time, type of tasks and exercises, etc.)	System creates a new training unit with optimized degree of difficulty based on diagnostic results and experiences of previous training units
Review and confirmation of the proposed tasks	Saves the training unit
User can change the order at any time	System saves the new order
User can reduce and/or increase the number of tasks at any time	System saves the new compilation

Adaptation of the training process, length, and degree of difficulty for a training session (automatic and user-controlled)

Changes – automatic through artificial intelligence

Man	**Machine**
Automatic adaptation of the degree of difficulty to the user's daily routine (easier or more difficult tasks)	System searches for lighter or more difficult categorized tasks, i.e. adapts the degree of difficulty accordingly, and logs the training process

Permanent adaptation of the training desired	System saves the training under the new difficulty mode or degree
Training should be maintained as previously compiled	System logs the changes and sequences, but does not resave the training
A new training should be saved with the adapted degrees of difficulty – the old one is retained	System saves the new training

Changes – user controlled

Man	**Machine**
Manual adaptation of the degree of difficulty to the user's daily routine (easier or more difficult tasks)	System searches for lighter or more difficult categorized tasks, i.e. adapts the degree of difficulty accordingly, and logs the training process
Permanent adaptation of the training desired	System saves the training under the new difficulty mode or degree
Training should be maintained as previously compiled	System logs the changes and sequences, but does not resave the training
A new training should be saved with the adapted degrees of difficulty – the old one is retained	System saves the new training

MoveTest (What is a person capable of doing?)

Man	**Machine**
Installing the belt and linking to the software	System recognizes the belt
Creating the client in the system	System saves the information
Selection of existing clients	System displays the person's data
Specification and monitoring of the exercises and tasks to be completed by a trainer	System records and saves movement sequences, patterns, and the different activities
	System creates a motion profile

MoveMonitor (What exactly does one do in a period of 7 days?)

Man	Machine
Installation of the belt and activation of the sensors	System starts recording
Client completes normal daily routine (7-day observation period)	System records and saves data
Trainer receives the belt and links it to the software via USB	Data are transmitted to the evaluation software
	System creates a movement profile

Expectations of the artificial intelligence

Man	Machine
Prerequisite: Creation of the client incl. basic classification	Saves personal data
Process starts – recording of the training processes	System learning (basic classification, daily recording, emotional attitude towards different tasks, are motivational factors effective or not?)
Emotion checks (in between the exercises), both with the client and the trainer	Saves results and reacts to them
Motivation (select particularly popular exercises, countdown, applause, voice input)	Simulates, learns from, and saves different scenarios
Response to the client's daily schedule by adapting the training accordingly (order of the exercises)	e.g., 4. exchanging a math problem with a puzzle if the first 3 could not be solved
Adjusting and modifying the level of difficulty during training	e.g., reducing number of puzzles, general knowledge exercise from category B instead of A

Timeline

Man	Machine
Manual creation of a training program with definite time specification	Automatic creation of a training program with definite time specification
Manual compilation of the training with different exercises on one or more selected topics	Automatic compilation of the training with one or more selected topics

Individual arrangement, reordering, and modification of the individual exercises during the training unit (e.g., number of exercises – add or delete, extend shorten the training, change number of exercise blocks – e.g., add or remove math exercises)	System performs and saves the changes on command
	System maintains original training and stores the new version on command (person, date, etc.)

2 modes – adaptable and non-adaptable (disabled for automatic changes)

System receives rules (adjustable for every user) – the 5 pillars of the training must be retained

Statistical evaluations (prerequisite: database) – which data should be captured

Man	**Machine**
Evaluations of individual clients (training course)	Tosses the data
Evaluations of client groups of a specific user or trainer	Tosses the data
Superordinate evaluation of the entire saved data	Tosses the data
Selected analysis based on country, person groups, different levels, men/women, etc.	Tosses the data
Display of analyses and evaluations in diagrammatic form	Displays the data graphically

New methods and functions (what can be implemented in PL)

Man	**Machine**
Drag and drop for puzzle, memory etc.	
Creative method – free drawing	
Creative method – assembling geometric shapes (Tetris) – rotating and moving	
Zoom function	
Larger fonts and keys	
Adapt design to visual weaknesses	

Interactive mat (group training)

Man	Machine
Charging of electronic components	
Plugging and connection of the power supply units	Connecting and keeping the connection
Turning on tablet and connecting to the system	Connect and maintain the connection
Selecting participants and training	Identify already personalized participants, save new participants
Dice casting (live using Bluetooth dice or virtual on the tablet?)	System detects the number of eyes and points to the correct playing field (LED on/off?)
Placing cone in the right position	System proposes personalized or general tasks
Task completion	Reward (sound, voice input, etc.)

2.3 Caregiver Serious Game Scenario

Man	Machine
Creation of the client (input of data)	Saves data
Selection of existing client	Loads data
	Initializes personalized game environment (avatar, characters, etc.)
	Selects & displays personalized starting serious game scenario
Makes a decision in the serious game scenario	Saves decision data. Displays the results of user decision
	Displays cognitive feedback
	Selects & displays subsequent personalized serious game scenario

2.4 Sensorimotor Serious Game Scenario

MCR will determine how to consider features from measuring of sensorimotor processes and include physical activity.

2.5 Eye-Tracking Serious Game Scenario

Table 3.Scenario "Eye-Tracking Anti-Saccadic Task"

User action	Machine action
User selecting to play "Ant-Saccadic Task" (AST)	
	AST procedure determines Areas-of-interest (AOI) for user interaction, initializes eye tracking and starts to count AOI entry and dwell times in AOIs.
	Starting the task by displaying the "cross" symbol. Next action displays the stimulus and counting the time the user gazes at various AOIs
Upon display of stimulus, the user intends to look away to opposite side of screen.	
	The AST procedure measures the delay and the percentage of dwell time in the correct and in the wrong AOIs.
	The AST procedure performs a summary on the statistics of AOI dwell times.
	The AST procedure inputs the statistics into a a classifier to determine whether the eye movement behavior was determined to be "dementia like" or "not dementia like".

3 Storyboards

The technology of using storyboard is successfully used for the visualisation of user interfaces, for example, as a consequence of screen displays. The application of storyboards is analogous with the MockUp technique in software development (paper prototype, user story, application case). A mock-up in software development means a rudimentary throw-away prototype of the user interface of to-be-developed software.

Mock-ups are especially implemented in early development stages so that requirements can be determined better in cooperation between management and applicants. Usually it refers to a back-bone of user interaction elements without functionality. Often programming is not featured but its development is done in an image editor.

So-called mock-objects are used for testing, especially for the module test (unit test), i.e., the testing of individual components. They are placeholders for objects that are necessary for the testing of other components and which are, for example, not yet available at the beginning of developments (for example, Test Driven Development, TDD). Later, mock-objects are used once the initialization of (available, functioning) object would become to extensive or not possible due to a lack of interface to a productive backend-system or due to legal causes (Bundesdatenschutzgesetz).

3.1 Interactive Game Storyboard

Playtime overview – synopsis of all components and functions

As for individual training, the initial essential factor is providing and managing content. The following description illustrates how content can be integrated into the system and who are possible content providers.

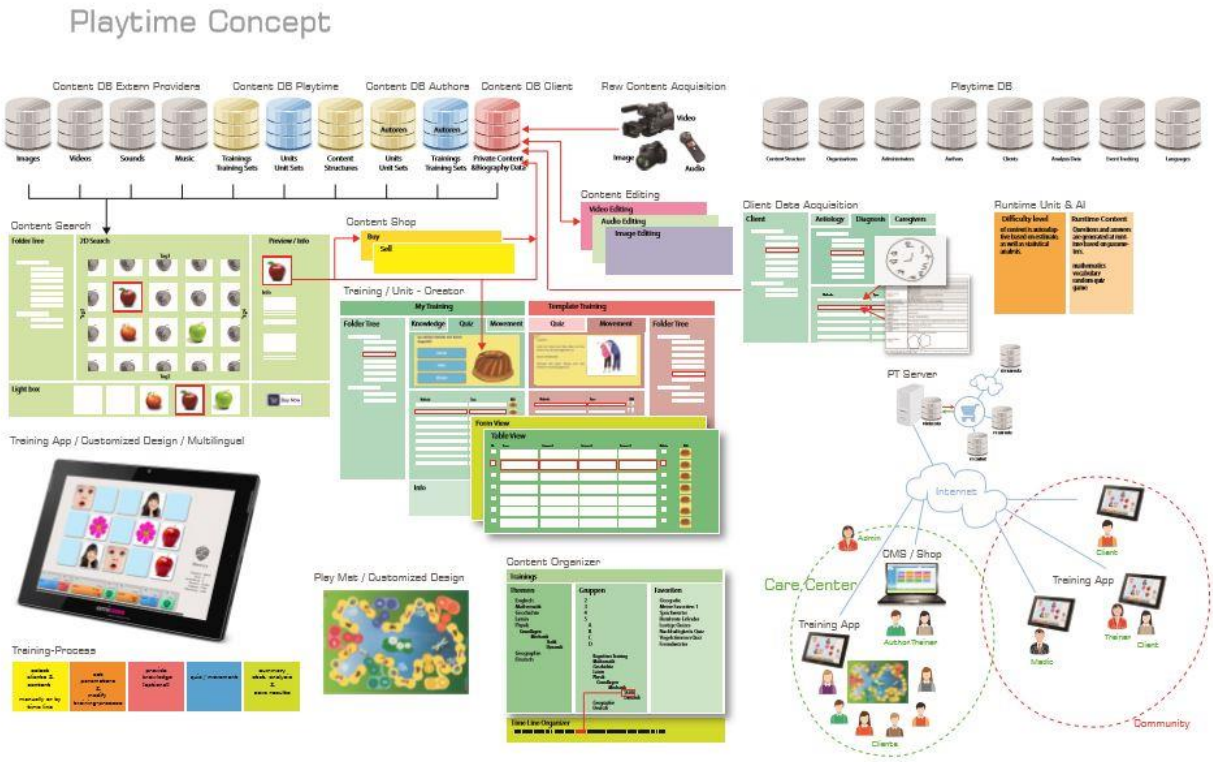
Two scenarios to consider:

The gameplay and the content are specified for an unknown group.

The gameplay and the content are intended for a registered user.

In the first case there are plenty of tasks and content which can and should be selected from. A search section assists this: making it possible to select the content differentiated according to preconfigured colour, keywords, shapes, visual language, video criteria, sound criteria etc. It must also be possible to directly save data in the system via camera, microphone, video camera, tablet or smart phone and from there assign it to a defined person as well as a topic.

Playtime Concept - Final Vision

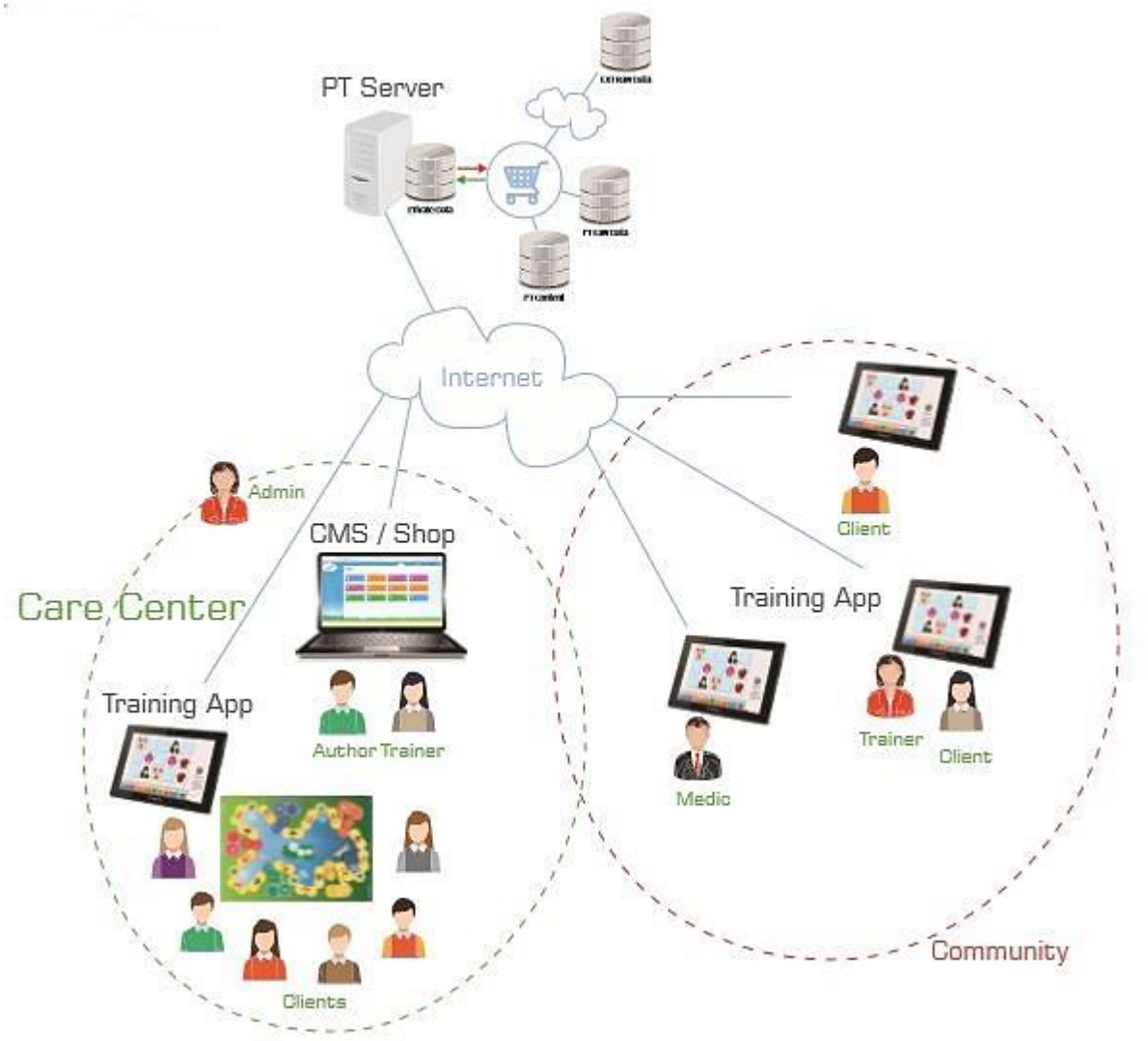


In principle a data bank making data available for individual training (Serious Game) as well as group training (Mat Game) is the basis of all the content. This means that the basis for both variants is valid for both providing data and collating content. The system only differentiates between group and individual training sessions after creating a training session or composing a game with movement exercises, quiz questions, puzzles, mathematical tasks, memory, etc.

Individual, personal content as well as general content is available in both cases. The system is currently designed so that the tasks are installed, the degree of difficulty is defined and the training has to be tailored and individually assembled in advance of the training session.

Target setting in the Playtime Project: The content and tasks can be manually or automatically moved during a training session, the degree of difficulty can be adjusted. At best the system reacts to the registered and recognized people and assembles suitable, tailored tasks and exercises or searches and distributes suitable questions and exercises during a group training

session using the interactive mat (it recognizes the participant using the cone).



Game description – multiplayer game with mat

Preparing at home:

Ensure that the tablet is fully charged. Keep the user name and password within reach, and (optional) bring an extension lead and plug for the tablet.

Preparing on site:

Register on the tablet with a user name and password using the pink icon Playtime and corresponding appliance. Fold out the game mat, unpack and position the cones.

Game instructions must be defined in advance and managed via the tablet during the game.

Each player has a cone assigned to him/her. All cones are placed on the start space. The players take turns to roll the dice in a clockwise order. The player with the green cone starts rolling the dice.

→ Question correctly answered: The player remains on the action space until it is his/her turn again. The player rolls again in the next round and moves into the next circle in the direction given.

→ Question incorrectly answered: The player remains on the action space until it is his/her turn again. The player continues the game in the normal direction in the next round.

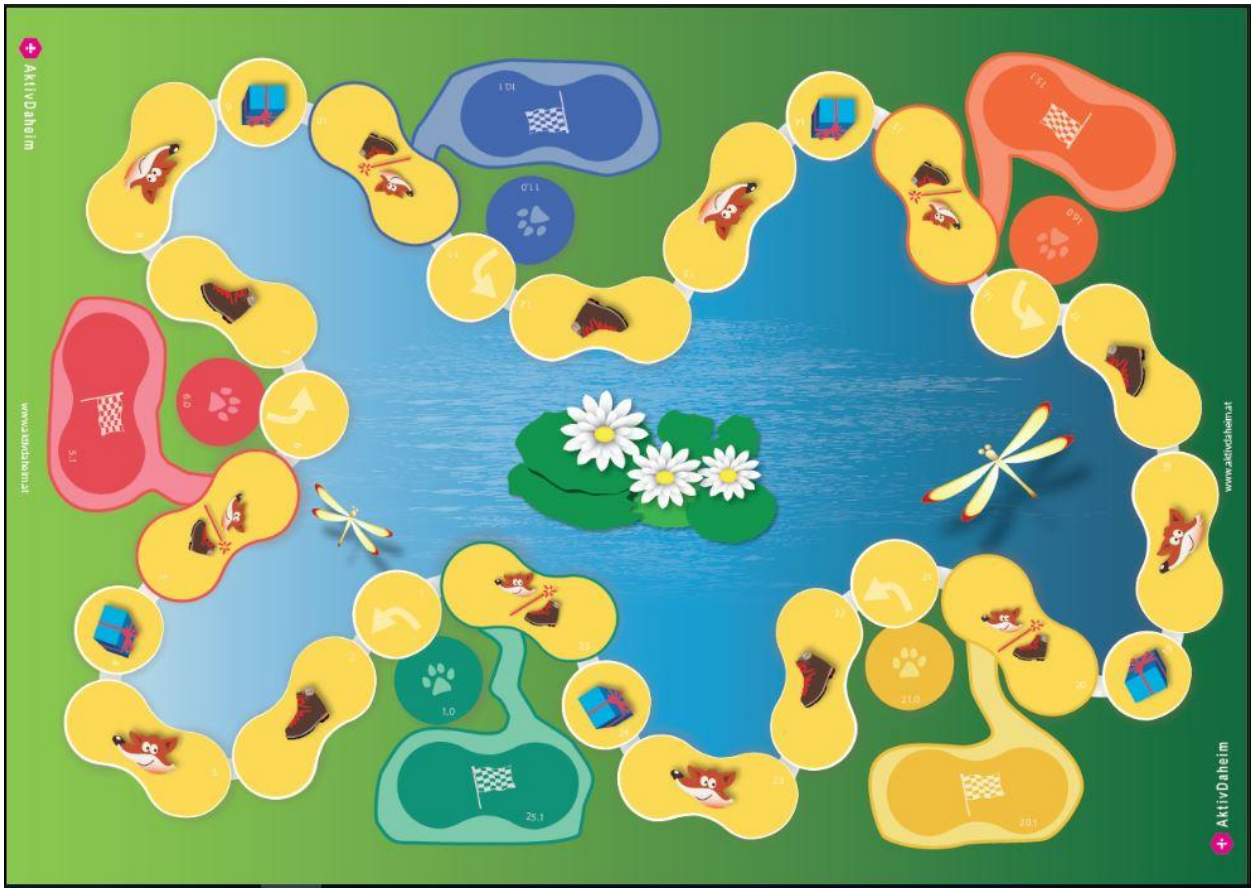
Description of the playing fields:

Fox = Cognitive exercise

Shoe = Physical exercise

Present = Surprise (e.g. move two spaces forward)

Shoe + Fox = Player can choose between cognitive und physical exercise



The basic game space arrangement from the AktivDaheim Project is to be kept. The design and GUI is to be redesigned for a unified concept. The results from the AktivDaheim Project field tests have shown that the clients benefit from sitting around the game mat and starting on different places. Thus every player has a space with a defined colour assigned to him/her – this has become evident as being less confusing than the original mat design in which all players started on one position.

Summary – What is new for Playtime (applies mainly to multiplayer training – but also to individual training):

- New design for game mat and application
- Optimised GUI (more functional, simpler, larger elements and buttons, improved contrasts etc.)
- New methods (Drag & Drop – creative mode, camera pen, possibly scent organ*)
- Motivation and emotion between individual exercises
- Participating players assigned to client data already saved in system
- Resulting in diagnosis retrieval and division into levels of difficulty for the client
- Resulting in assigning levels of difficulty during the game due to recognition of the game cone and assigning the tasks according to requirements
- The timeline makes it possible to intervene in the level of difficulty during the game to take into consideration the clients' mood and condition on that day (particularly important for the movement exercises)

* **Development project:**

SCENT Organ with up to ten different scents

- Scent bottles marked by means of a RFID TAG
- Motherboard with CPU to read the scent bottles and
- communication via Bluetooth with Android tablet PC
- Development – specific software to map functionality.

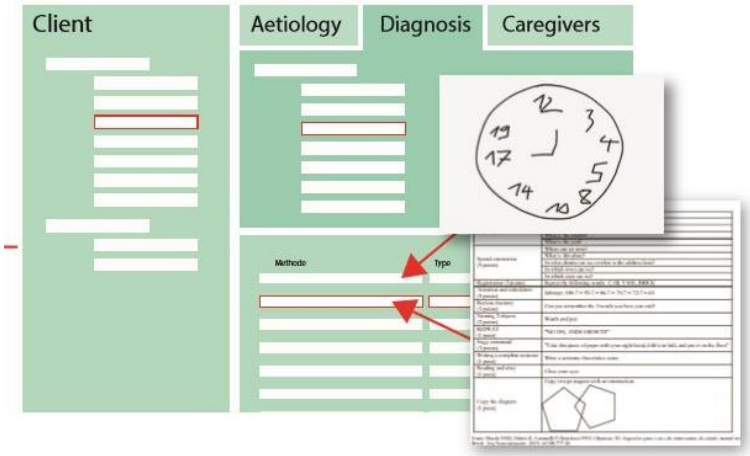
Development costs estimate for precisely defined technical specifications ~ 6,000.00 EUR

3.2 Tablet based Serious Game Storyboard

Data is the basis of the whole content structure. This can in turn be captured individually but also obtained from available data banks (refer to Overview Playtime Concept). The following illustrations describe the way to create a training session – from providing data to the training unit with a modifiable and controllable timeline.

Shooting photos and videos, recording audio files which can be played directly into the system and saved in the data bank.

Client Data Acquisition



Raw Content Acquisition

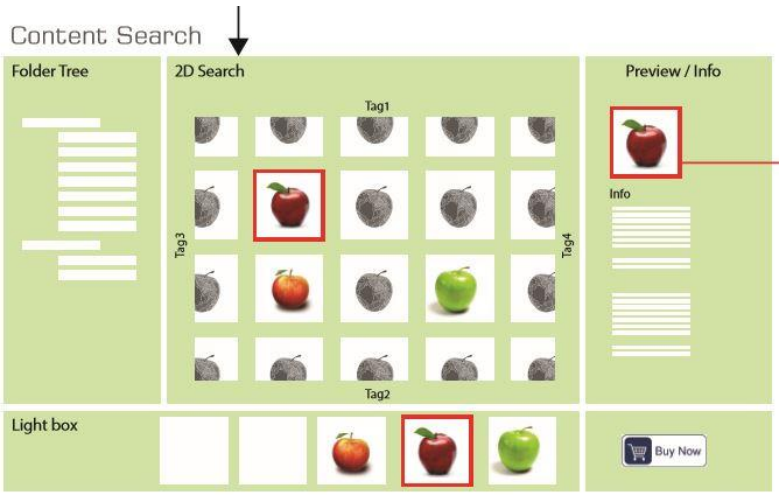
The raw content acquisition refers to the upload of the picture and video material of the client, instruction videos, etc.

Data can be recorded via a camera, tablet, smart phone etc. which can be used for every training session as well as for biographical work.

Content Search

Creating training units begins after successfully entering data, content, tasks, images, songs, videos etc. These can of course also be accessed via all available activated content.

It is possible to hold a detailed and targeted content search via a simple presentation. Different tags substantiate the result. For example (all apples are shown) searching for an apple with a white background and red. The system approximates the search criteria and makes systematic suggestions in a raster. The Light Box contains the chosen selection from which the best image can be chosen.

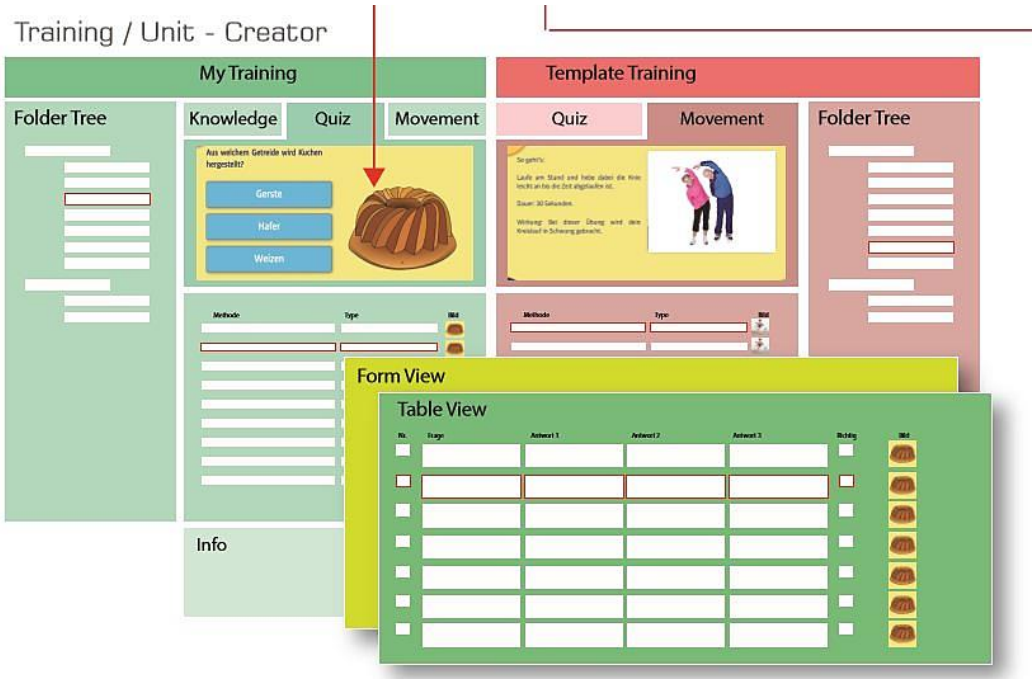


Trainings-Creator

Tasks and exercises according to the client's ability can easily be summarised in the Training Creator. The trainer can collate the training units to be tailored and individual but also very general. The aim in doing so is for the trainer to supplement the training units with those from colleagues or redesign them – at a glance. The prerequisite is always activating the content for the active person.

A colour system (a further search criterion which supports the five pillars of Alzheimer disease therapy) acts as colour coding for task definition and assignment

- Cognitive – orange
- Movement – red
- Perception – blue
- Singing, music – yellow



Training App

The result – a training process which intelligently and intuitively reacts to the trainee's needs or can be manually customized and modified. The system should contain exercises and tasks at levels 1, 2, and 3, whereupon it is only in a position to learn with repeated use.

System learning factors:

- How the task is solved (e.g. error search image – how many clicks did the client need – logged)
- Solution speed or time to solve or after time has run out – how many tasks the client has completed
- Define communication: timer, temporary screen: done, time is running out, repeat exercise etc.
- Specifications must be defined – that is the basis for Artificial Intelligence
- Number of errors – e.g. which level of difficulty the exercise has
- Complexity of image shown (puzzle)

ATTENTION:

There must be comparable values in the test phase (this applies to all field tests – data of all tests is recorded in the background), possibly in two test groups: on the one hand a fixed timeline, times exactly defined, time is running out etc. on the other hand a system which follows the client's ability.

- Test group 1 tests the system's intelligence (adjusting the degree of difficulty)
- Test group 2 always tests in the same mode to show comparable data or gains in comparison to past training sessions

The training process is illustrated in five actions:

1. Choice of client and corresponding content
2. Parameter selection and training process modification
3. Optional: possibility to transfer knowledge before it is retrieved (can also be applied for biographical work)
4. Action – completing the training – quiz and movement exercises
5. Training summary and analysis – securing the data

3.3 Sensorimotor Serious Game Storyboard

3.3.1 Introduction

Within the PLAYTIME Suite, a Motion Analytics component will be incorporated. This component will be embedded in an evaluation module which can only be accessed by a trainer (healthcare professional). The evaluation measurements could take place before or after the group sessions (see section 3.2) or on individual appointments. The trainer can select a specific user and perform evaluation measurements using McRoberts sensors (DynaPort product range) to gain insight in the physical functioning (physical activity and physical capacity) of the subject.

3.3.2 Evaluation measurements

To perform an evaluation, a trainer opens the PLAYTIME app, logs in and selects a subject who will be evaluated. If the subject does not have an account yet, it needs to be created first. Next, the trainer can either select different evaluation measurements:

Start a week measurement that gives insight in the movement patterns of a subject (what does a person actually do?).

The trainer couples a DynaPort MM(2) sensor and assigns it to a subject and measurement instance (visit) via the PLAYTIME app.

The trainer programs the DynaPort device for a measurement of 1 week and hands the device over to the subject with instructions on how to wear the device.

After a week (e.g. during the next group session) the subject returns the device to the trainer. Via the evaluation module of the PLAYTIME app, the trainer uploads the data to the (cloud-based) analysis server. The data will automatically be linked to the correct subject and visit. The outcomes of the analysis will be added to the central database of PLAYTIME

The algorithms developed in WP3 (Task 3.2) will be incorporated in the analysis

The integration of McRoberts' sensors (DynaPort MX and DynaPort MM2) is part of WP4.

Start a short measurement that gives insight in the physical capacity of a subject (what is a person capable of doing). The following 'modules' will be used in the first field trial within PLAYTIME (together, the 3 modules form the Short Physical Performance Battery (SPPB)):

Balance: 3 modalities

Gait: short gait (4 meters)

Sit-to-Stand (STS): Repeated STS (5 times)

Modules that could be added to the capacity measurements (e.g. for the second field trial within PLAYTIME) could be the following:

Timed-up-and-go

Stair test (in development, currently being validated by an academic partner of McRoberts)

6-minute walk test

Glittre test (in very early development)

To start a measurement, the following actions have to be performed:

The trainer couples a DynaPort MX sensor and assigns it to a subject and measurement instance (visit) via the PLAYTIME app.

The trainer programs the DynaPort device for a short measurement by selecting a pre-defined protocol and hands the device over to the subject with instructions on how to wear the device.

Guided by visual and audio feedback by the PLAYTIME app, the trainer performs the measurements. When the measurements are finished the trainer uploads the data to the (cloud-based) analyis server via the evaluation module of the PLAYTIME app. The data will

automatically be linked to the correct subject and visit. The outcomes of the analysis will be added to the central database of PLAYTIME

The algorithms developed in WP3 (Task 3.2) will be incorporated in the analysis

The integration of McRoberts' sensors (DynaPort MX and DynaPort MM2) is part of WP4.

During these measurements, data is collected which will be uploaded to the central database (the API for this will be developed in WP4). The data in the database can be linked to the emotional or cognitive status of the subject (output of the serious game developed in WP2).

3.3.3 Exercises during the game

During the Amicasa game, exercises will be incorporated that should train the subject on the aspects mentioned above (e.g. balance exercises, strength exercises etc). Questions about exercising could also be incorporated to increase the awareness of the subjects of the importance of being physically active.

The exercises and questions can be incorporated in the board game (see section 3.2) or in the personal app where the exercises can be tailored to the subject's capabilities and progress (see section 3.3). At regular intervals (e.g. before people start playing the game and after playing the game for a while (10 weeks or so)) the measurements mentioned above can be performed and evaluated whether a subject scores better, the same, or worse. The outcomes can be used to tailor the exercises to a subject and to give feedback on progress.

3.4 Eye-Tracking Serious Game Storyboard

3.4.1 Application of Eye Tracking in PLAYTIME

Eye tracking will be applied in the PLAYTIME suite as a means for assessment but also as a means of gaze based interface for serious gaming. In this sense an eye tracking component will not propose an individual software package but merely provide functions that are part of the playful multimodal training component in PLAYTIME.

The contribution of the Austrian research institute JRD to PLAYTIME is the implementation of measurement technologies that identify behavioral and psychophysiological markers for cognitive mechanisms in people with dementia during gameplay, such as, at home. Based on estimated mental state of the user, further game content will be personalized by adjusting the level of gameplay to the person with dementia.

3.4.2 Calibration Session

JRD will provide a software for web camera based eye movement analysis at a Tablet PC. This software will analyze the video stream by detecting the face, localizing the eyes, and apply eye tracking for the estimation of orientation of eye balls towards the screen of the Tablet PC. The accuracy of the eye tracking is sufficient to conclude with simple statistics of eye movement features during gameplay at home and will estimate features of users' cognitive control.

The eye tracking calibration procedure will start with a simple game to have a playful procedure. By focusing on simple dots on the screen a timeout procedure starts and will make the points disperse in a kind of firework after a predefined time interval. The user has to examine five calibration points and is able to continue after a predefined fixation time interval.

After calibration, the eye tracking can be used and it will be able to save the calibration configuration in association with the user name.

3.4.3 Games based Assessment of Cognitive Control

Assessment of cognitive control capacities will be performed in the multimodal playful intervention component by means of eye tracking analysis of selected serious game components.

At the time of the implementation of this deliverable, the following components are planned to be implemented as a playful cognitive control tests:

- **Antisaccade test**. There are several implementation considered. One imagines that the user has to hunt for bad rabbits which are looking out of holes in random time intervals. Hunting is achieved by looking with considerable dwell time at a rabbit which will make the rabbit disappear. However, once a good rabbit accidentally looks out of a hole, the user has to move its gaze away from it since otherwise she would be penalized for hunting good rabbits. In this sense, this game will implement a gaze interface for the interaction with the game content.
- **Visuospatial comparison test**. This test is implemented by an image pair where the right one will incorporate 5 small differences in the image that do no pop pout immediately and need scanning the images and look back to the original image to compare the image content. Eye tracking will be applied to accompany the visual comparison and the eye movements and analyse the features of the viewing behavior for implicit assessment.
- Trail making test B. The gamified implementation will use a gaze interface to activate the sequence of letters and numbers according to the correct order.
- Smooth pursuit test. The gamified version will display birds flying across the screen and users will try to follow as well as possible the curve of the animals.
- Delay-of-gratification task, Go/no go test. A concrete implementation plan will be outlined in more detail for the evaluation in the second field test.

The analysis of the data from eye tracking will be performed as follows. Data from eye tracking will first be filtered with respect to the frequency of available data. Data captures with frequencies below 5 Hz will be discarded. Filtered data from eye tracking will then be classified into fixations, saccades and blinks according to the methodology of (Salvucci & Goldberg, 2000). The time stamps associated with the fixations of the stored eye tracking data will be matched with the training sessions and the corresponding areas of interest (AOI) on the screen. With respect to these AOIs, further eye movement features will be elaborated, such as, dwell time within a specific AOI. From the dwell time measures and related AOIs we will then estimate means/medians, standard deviations SD of dwell time. The quality of eye tracking data will be

related to the mean dwell time that is being measured on a specific AOI, and comparing these values to those that have been received in lab like environments. If the difference is below a certain threshold the data quality evaluation will be positive.

The eye tracking data of JR will be stored in association with time stamps in order to match eye movement statistics to training sessions played with the app on the Tablet PC. Neither video data nor any visual content about the study participant will be stored. IP addresses will also not be stored.

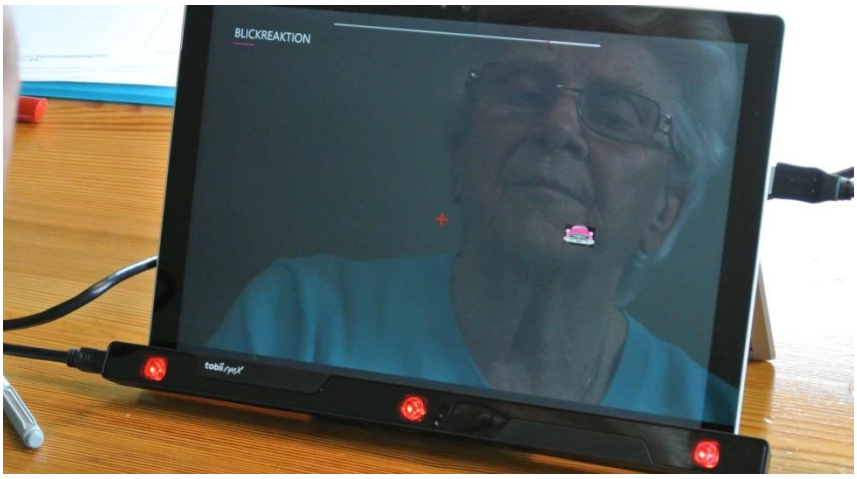


Figure 1. Eye tracking (mobile interface attached at the bottom of the tablet PC) applied to an Anti-Saccadic task for the assessment of cognitive control.

4 Dementia Caregiver Game Scenario and Storyboards

4.1 Dementia Caregiver Game Scenario: Description of Actors (Caregiver & Software)

4.1.1 Purpose

The purpose of this document is to give an insight into the architectural setup of the Dementia Caregiver game in Articulate Storyline. The key-functionalities of the game will be illustrated from a high-level approach. More substantial information about the business meaning (scientific foundations) of parameters, variables, … and the technical implementation of this Articulate file in an online platform (with the connection with a database) is not covered in this document.

After reading this manual, the reader should be able to find his way through the set-up of an Articulate game and more specific the Dementia Caregiver Game

4.1.2 High-level application structure

The functionality of the Dementia Caregiver game is essentially based on an interaction of the game screens with JavaScript coding.

The screens can be divided in 4 different types:

- Start screen
- Scenario explanation screen
- Question screen
- Result screens (= Dashboard screens)

More information on the screens can be found in the section ‘User interface’

The JavaScript code drives:

- Start screen: parameters are loaded from the database into the Articulate file (more explanation on the parameters in section 3)
- Question screen: loading of the scores of the determinants for the possible answers
- Result screen: loading of the results based on calculations of the scores of the determinants and the final scores of the previous scenario.

An overview can be found in Figure 2.

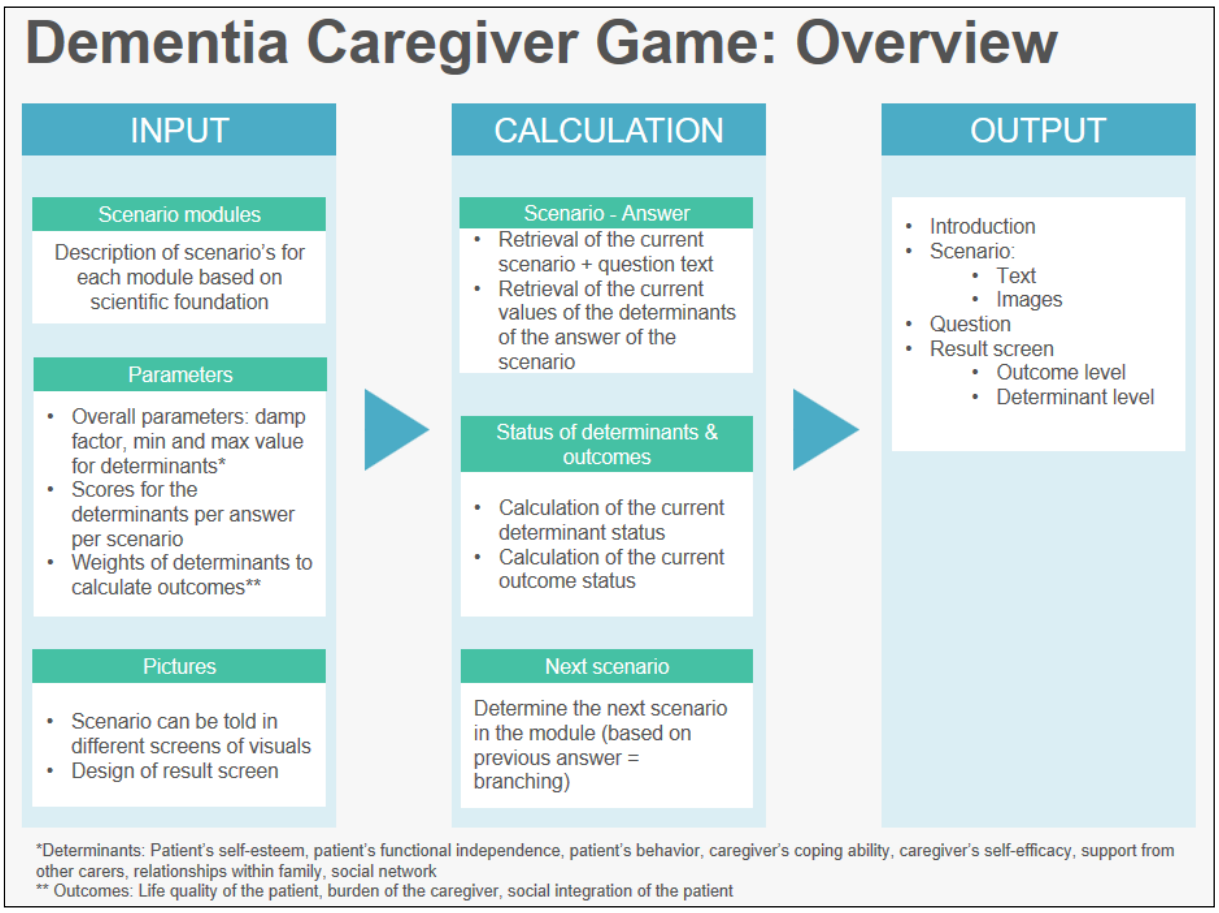


Figure 2 Overview of the Dementia Caregiver Game

4.1.3 Elements of Dementia Caregiver Game

4.1.3.1 Modules

KEY CONCEPT

- The caregiver game consists of different modules (each module consists of a different family setup). A module is a combination of scenarios on a specific topic. A scenario in the dementia game has 3 possible answers.
- In each module we ensure that there's a good balance of the scenario's based on following 3 parameters:
    - Nature of the problem (cognitive, functional, behavior,…)
    - Which of the 3 outcomes are influences the most (see paragraph outcomes)
    - In which setting the scenario takes place (Home, social environment,…)
- Scenario's can be selected based on the home situation.
- Within a specific module, every player follows its own "parcours": The sequence between scenarios within a module has a "branched tree structure" and is based on the decision of the player. Depending on his/her answer on a scenario, (s)he will get

different scenarios to respond to. At this stage, the modules of the Dementia Caregiver game are not yet branched and is a topic to develop in the next version 2.0.

ALGORITHM

Per scenario in a gaming module, the game engine records following variables:

- The current module number e.g. M1
- The current scenario number e.g. M1Q1
- The current answer number e.g. M1Q1A1

4.1.3.2 Determinants

KEY CONCEPT

- The determinants in the game are parameters related to the patient and his environment system that will be affected (positively or negatively) by the player's decisions. The determinants are determined by scientific foundation work.
- In the Dementia Caregiver game, there are 8 different determinants:
    - Patient's self-esteem
    - Patient's functional independence
    - Patient's behavior
    - Caregiver's coping ability
    - Caregiver's self-efficacy
    - Support from other carers
    - Relationships within family
    - Social network
- By observing the impact on the determinants, the player can get an insight into the consequences of his/her decisions.
- The status of the determinant is quantified by a value between a pre-defined minimum and maximum range.
- The determinant variables that can be parameterized are:
    - The determinant names (in Articulate)
    - The status of each determinant at the onset of the game >> is 0 score (so no impact) (JavaScript)
    - The minimum and maximum value for the determinants (JavaScript)
    - The number of determinants present in the game

ALGORITHM

- After each player decision on a scenario, the impact on the different determinants is calculated.

- The current status of a determinant is calculated from the previous status and the impact score of the current player's decision.
- The magnitude of a decision's impact can be modified by applying a "damp factor" on the impact scores (to be parameterized) and is based on following principles:
    - Making the impact more or less stronger
    - The closer you get to the maximum or minimum value of the determinant, the less the impact is
    - There's always a trend towards "the middle"/neutrality.

Discore(t) = Current status of the determinant at scenario t (= after t scenario's)

Siscore(t)= The score of the determinant for the current scenario t (not considering the previous scenarios)

Min = minimum value of the determinant

Max = maximum value of the determinant

Cf1=damping factor

$$D^i_{score}(t) = D^i_{score}(t-1) + S^i_{corr}(t) \quad \text{(E1)}$$

$$S^i_{corr}(t) = S^i_{raw}(t) * cf_1 * nf^i(t-1) \quad \text{(E2)}$$

$$nf^i(t-1) = \frac{|max - D^i_{score}(t-1)|}{max} \quad if \;\; S^i_{raw}(t) \geq 0 \quad \text{(E3)}$$

$$nf^i(t-1) = \frac{|min - D^i_{score}(t-1))|}{min} \quad if S^i_{raw}(t) < 0 \quad \text{(E4)}$$

4.1.3.3 Outcomes

KEY CONCEPT

- Decisions also have impact on the outcomes (parameters that are derived from a combination of different determinants.
- In the Dementia Caregiver game you have 3 different outcomes:
    - Life quality of the patient
    - Burden of the caregiver
    - Social integration of the patient

DESCRIPTION

- The impact value of the determinants are calculated as a weighted average from the determinants

- The weights are determined based on scientific sources and the sum of the weights of the different determinants is 100%

ALGORITHM

$$W = \begin{pmatrix} w_{11} & w_{12} & w_{13} \\ w_{21} & w_{22} & w_{23} \\ w_{31} & w_{32} & w_{33} \\ w_{41} & w_{42} & w_{43} \end{pmatrix} \quad \text{(E5)}$$

$$O_j(t) = \sum_{i=detIndex} D^i_{score}(t) * w_{ij} \quad \text{(E6)}$$

With wij=weight of determinant i in outcome j

Oj(t)=Result of outcome j after t scenarios

Discore(t) = Current status of the determinant at scenario t (= after t scenario's)

4.1.3.4 User Interface

KEY CONCEPT

- Although the game focuses on the content in the first place, much attention is paid to the user interface considering game mechanics/learning mechanics and emotions. A welcome screen with user instructions and the use of many different pictures, alerts, colors, etc. allow the game to be played in a very dynamic environment.
- Dashboards have attractive designs with creative visual elements.

DESCRIPTION

There are a number of screens that make up the user interface (see Storyboard)

- Start screens:
    - Welcome screens (slide 2)
    - Description of determinants (slide 3)
    - Description of the outcomes and the connection with the determinants (slide 4)
    - Description of the family setting of current module (slide 6)
    - Navigation button to the scenario screens ("start button")
- Scenario explanation screens:
    - Description of the scenario via multiple scenario screens (slide 7)
- Question screens:
    - Description of the question ("What would you do in this scenario?") + 3 possible answers (for the Dementia Caregiver Game) (slide 8+9)
- Dashboard screens:

- Result screen of the outcomes of the current answer (not taking into account previous answers) by using sun & clouds. The result screen will probably be adapted in the next stage of the development. The arrow in de picture of the outcomes indicating a trend (positive or negative based on the previous results. (slide 10)
- When clicking on one of the outcomes: screen with result of current situation on the determinants (slide 11)

4.2 Caregiver Serious Game Storyboard

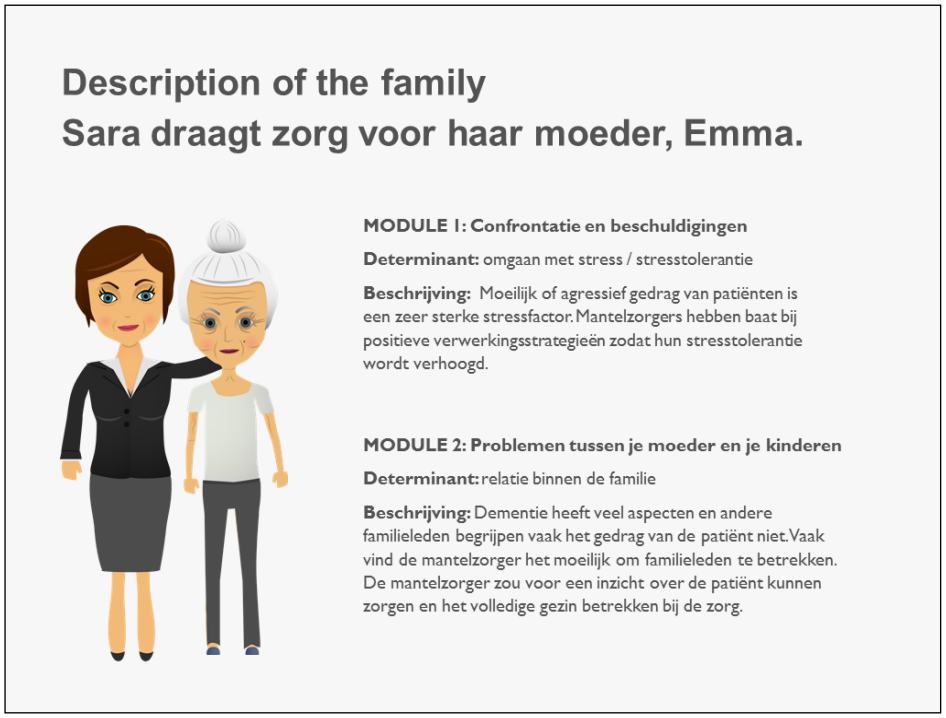


Figure 3. SERES Dementia Game – Description of the family.

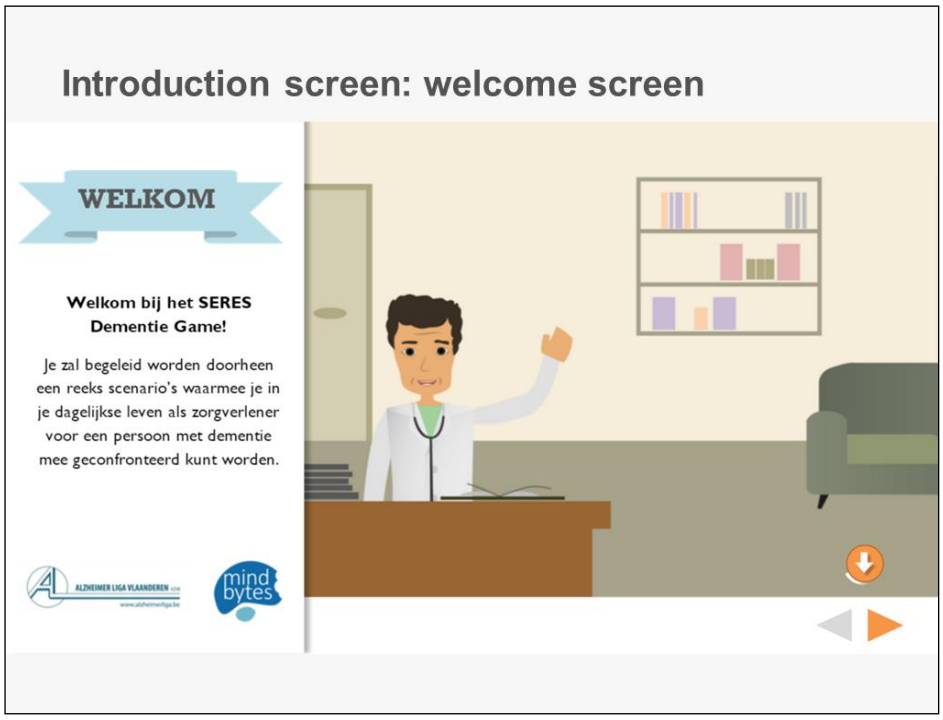


Figure 4. SERES Dementia Game – Introduction screen: welcomes screen.

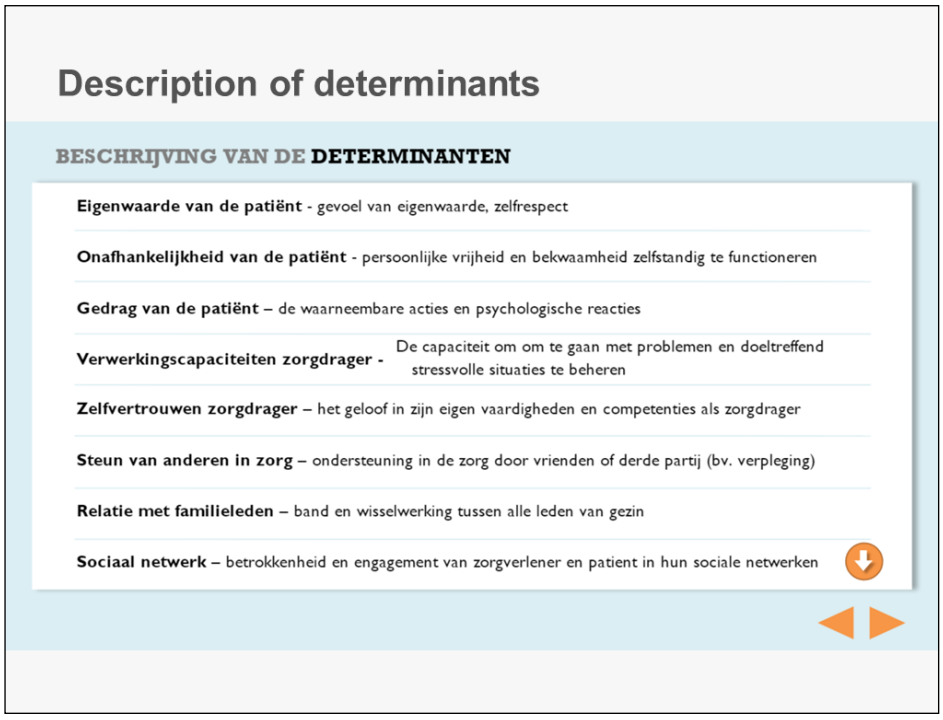


Figure 5. SERES Dementia Game – Description of determinants.

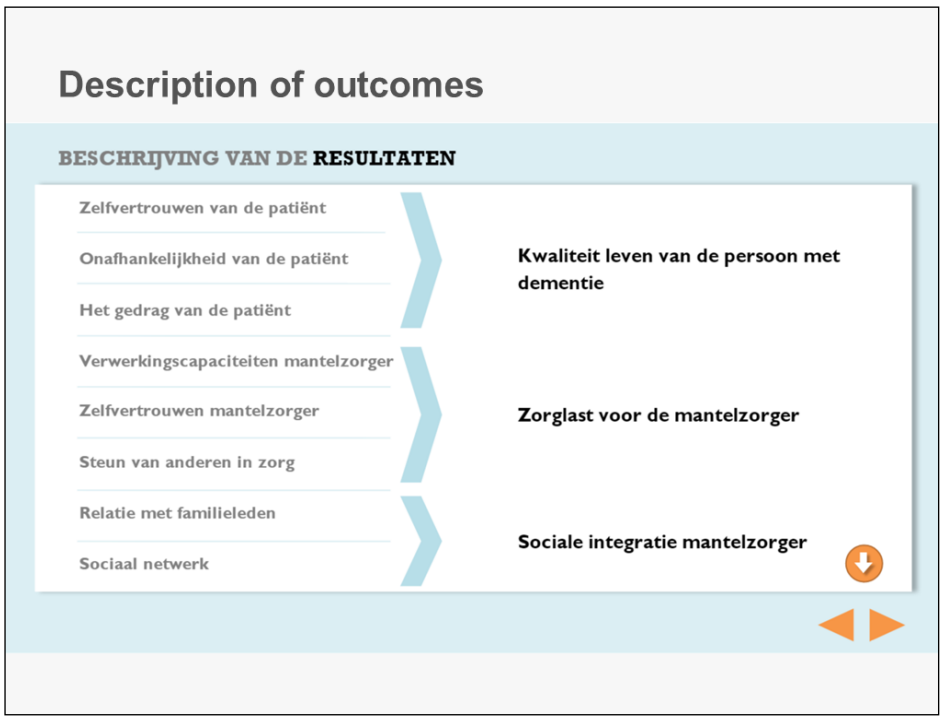


Figure 6. SERES Dementia Game – Description of outcomes.

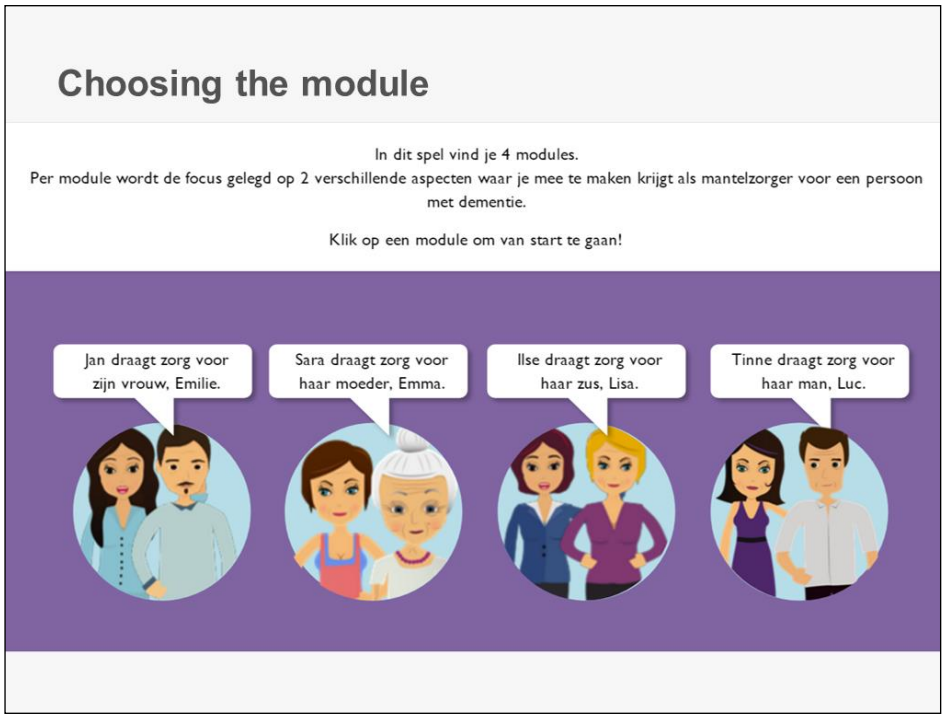


Figure 7. SERES Dementia Game – Chosing the module.

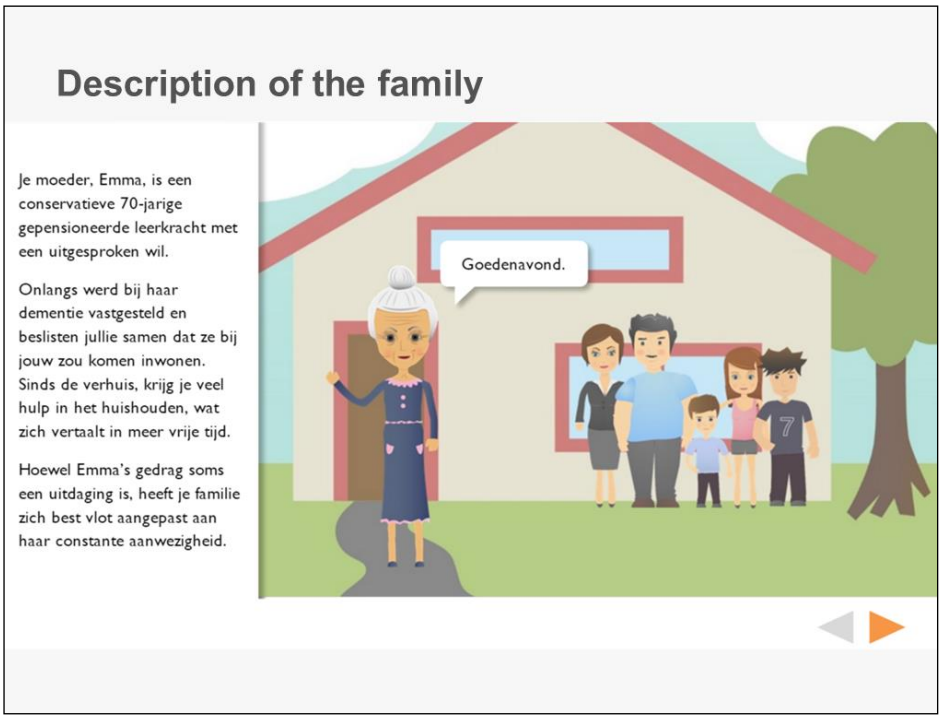


Figure 8. SERES Dementia Game – Description of the family.

Figure 9. SERES Dementia Game – Description of a scenario.

Figure 10. SERES Dementia Game – Question screen.

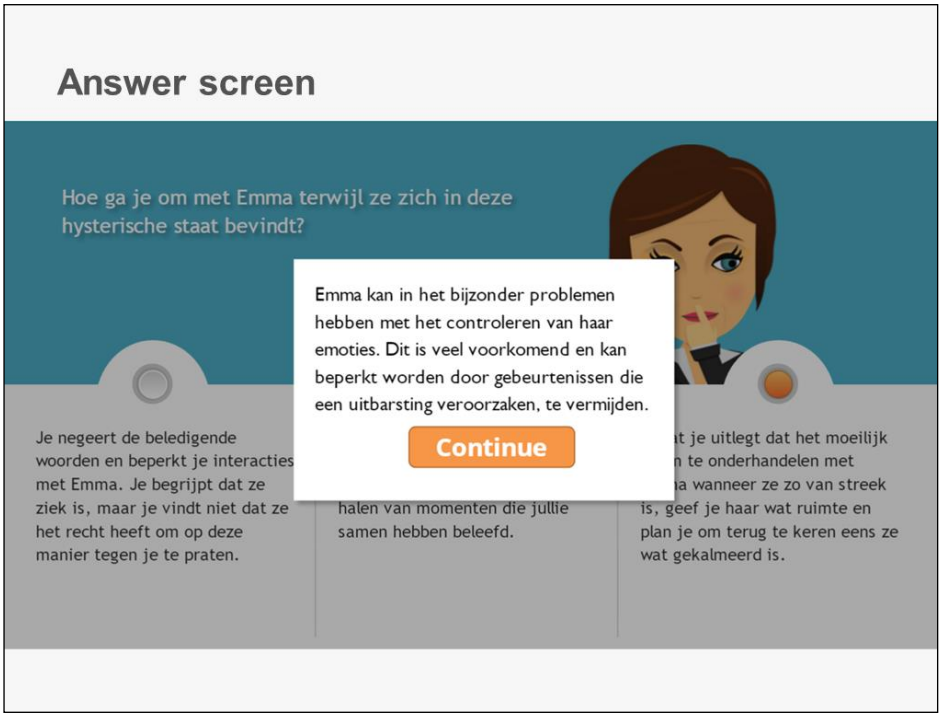


Figure 11. SERES Dementia Game – Answer screen.

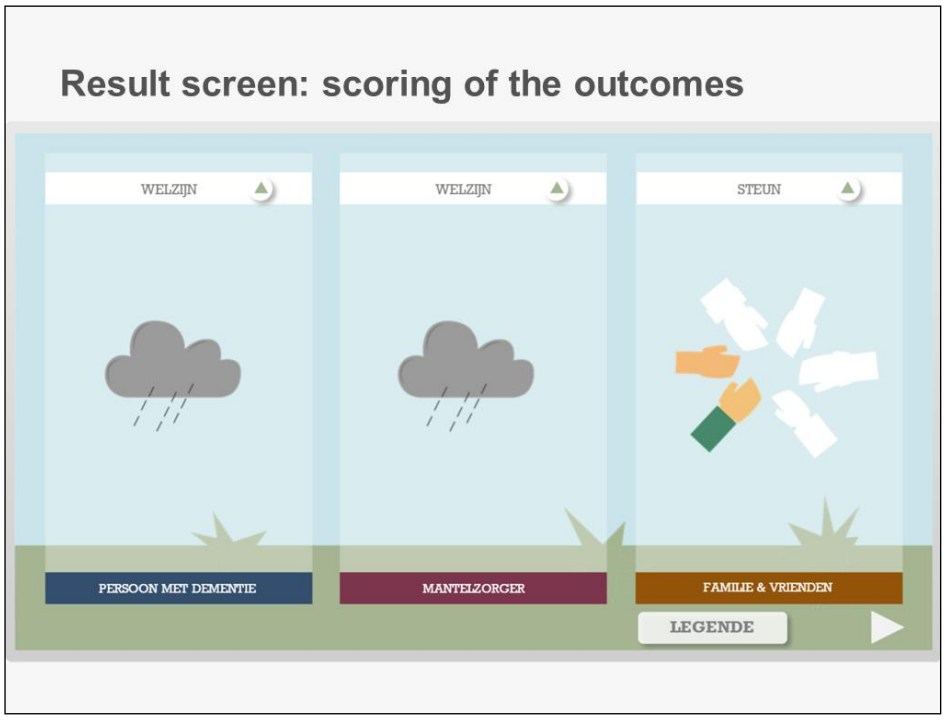


Figure 12. SERES Dementia Game – Result screen: scoring of the outcomes.

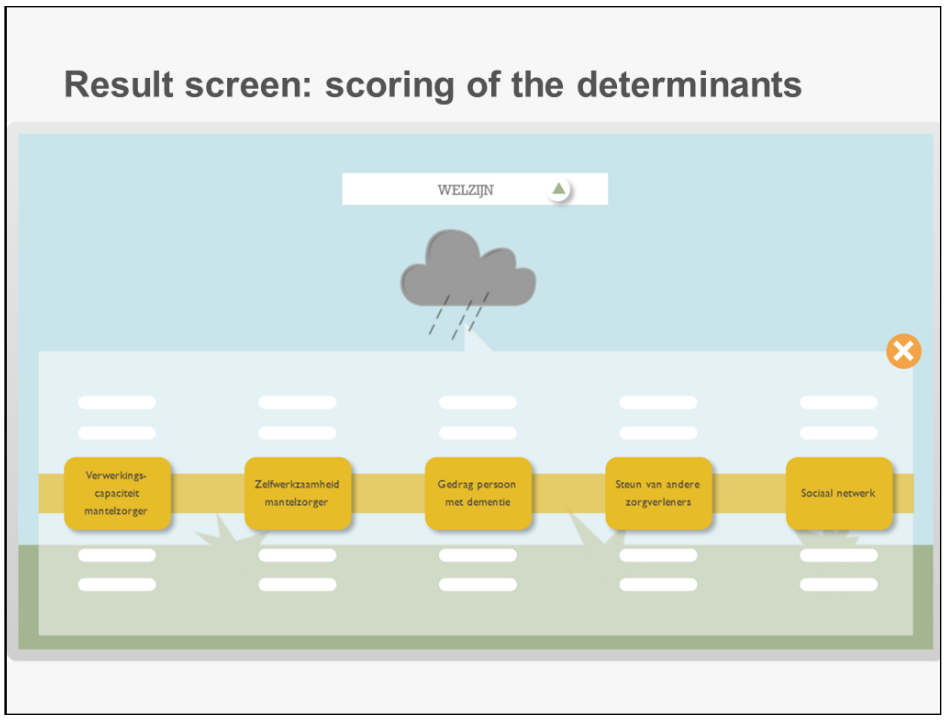


Figure 13. SERES Dementia Game – Result screen: scoring of the determinants.

5 Conclusions and Outlook

The PLAYTIME project, with all its methods and application options, is on the one hand intended to be a daily companion in dementia care and professional training and on the other to give sufferers stress-free, varied and fun access to modern media and training forms. The combination of cognitive and physical exercises should be embedded in a serious game, which is suitable for one or more players. Special care must be taken in certain scenarios to ensure that the different grades of dementia are considered in the application. At best, the options to easily adjust the difficulty levels should be created here. The trainer must always have the opportunity to take into account the daily needs and sensitivities of dementia clients. The scenarios must provide for an adjustment of the training duration, number of exercises, sequence of exercises and selection of exercises. Not only in the preparation of the training, but also directly with the client on-site.

The goal for PLAYTIME to be used both in the group with care, as well as individual training with trainer or relatives and by the client alone, will be a great challenge especially with respect to the last mentioned point. The menu navigation must be clear, the GUI has to be optimally prepared and tailored to the target group. A guide through the system - from powering on the tablet to connecting to the power cord – faulty operation by the client must be intercepted by the system. This means for example: no premature unwanted termination of training, no sudden failure due to low battery level, etc.

Also the feeding of a training session into the system must be simple, precise and fast for trainers or users in the background. The structure of the underlying database is crucial. Important are comprehensive search functions, image and video databases, the rapid upload of personal data such as private photos and videos for biography work, and the ability to intelligently manage your own and third-party data well.

6 Glossary

Table 1. *Glossary.*

Notion	Description
Use case	A use case is a **scenario** written from a functional point of view, enumerating all the possible user actions and system reactions that are required to meet a proposed system function (Jacobson et al., 1992). Use cases can then be analysed with respect to their requirements for system objects and interrelationships. A variant of it is “user-system conversation” using a 2-column format, a scenario is decomposed into a linear sequence of inputs from the user and the corresponding processing and/or output generated by the system. Kaindl (2000) extends this analysis by **annotating how scenario steps implement required user goals or system functions**.
User interaction scenario	In scenario-based design (SBD) The use of a future system is concretely described early in the development process. Narrative descriptions of envisioned usage episodes – user interaction scenarios – are then used in various ways to guide development of the system that will enable these use experiences. (Jacko, 2012). A scenario is a sketch of use, it consists of a setting, or situation state, one or more actors with personal motivations, knowledge and capabilities, and various tools and objects that the actors encounter and manipulate. The narrative describes a sequence of actions and events that lead to an outcome. These actions are related in usage context that includes the goals, plans and reactions of the people taking part in the episode.
Storyboard	The storyboard technique is successfully used to visualise user interfaces, for example, as a result of screen contents. The application of storyboards essentially corresponds to the mock-up technique in software development (paper prototypes see User Story, Use Case). A software development mock-up refers to a rudimentary disposable prototype of the user interface of software to be created. Mock-ups are used in particular in early development phases in

order to be able to better determine user interface requirements in cooperation with clients and users. It is usually a mere skeletal structure of the controls without additional functionality. As, often there is no programming, the creation takes place in an image editing program instead.

So-called mock objects are used for testing in particular for the unit test, the testing of individual parts. They serve as placeholders for objects which are for instance not available at the beginning of the development but are essential in order to test another part (see also Test Driven Development, abbr. TDD).

Later, mock objects are used if for example the initialisation of the (existing, functional) object is too elaborate or not possible in a test environment due to a lack of connection to productive back-end systems or is not permitted for legal reasons (see also Federal Data Protection Act).

7 Abbreviations

Table 2. *Abbreviations.*

Abbreviation	Description
FAST	Functional Assessment Staging
GDS	Geriatric Determination Scale
MMST	Mini Mental Status Test
POR	Point-of-regard
QOL-AD	Quality of life by Alzheimer disease
QUALIDEM	Quality of life in dementia
UCD	User centered design

8 Bibliography

(**Salvucci & Goldberg, 2000**) Salvucci, D. D., & Goldberg, J. H. (2000). Identifying fixations and saccades in eye-tracking protocols. In *Proceedings of the 2000 symposium on Eye tracking research & applications,* pp. 71-78. ACM.