



universität
wien

MASTERARBEIT

Titel der Masterarbeit

“Serious Game for Diagnosis and Therapy
in Emotion Recognition”

Verfasserin

Natascha Schweiger, BSc.

angestrebter akademischer Grad

Diplom-Ingenieurin (Dipl.-Ing.)

Wien, 2015

Studienkennzahl lt. Studienblatt:

A 066 935

Studienrichtung lt. Studienblatt:

Masterstudium Medieninformatik

Betreuer:

Univ.-Prof. Dipl.-Ing. Dr. Helmut Hlavacs

Table of Contents

List of Figures	iii
List of Tables	v
1 Introduction	1
1.1 Project EmoJump.....	2
1.2 Motivation	3
1.3 Research Question & Goal.....	3
1.4 Publication	4
1.5 Overview	4
2 Related Work	5
2.1 Serious Games Development.....	5
2.1.1 Serious Games.....	5
2.1.2 Psychological Background.....	6
2.1.3 Technical Background	7
2.2 LIFEisGAME	10
2.3 FaceSay	11
2.4 Mind Reading	14
2.5 Transporters.....	15
2.6 Summary.....	17
3 Serious Game – EmoJump	19
3.1 Game Idea and History	19
3.2 Game Concept	20
3.3 Game Structure and Screen Flow.....	21
3.4 Design Concept.....	24
3.5 Player Motivation	25
3.6 Game Mechanism and Levels.....	25
3.7 Technological Description	34
3.7.1 Technology & Tools	35
3.7.2 Game Engine Jumru 5s	38
3.7.3 Game Architecture	43
3.8 Implementation.....	44

4	Evaluation	65
4.1	Evaluation Method.....	65
4.2	Pre-Test.....	66
4.3	Observation.....	69
4.4	Post-Test.....	69
4.5	Summary & Improvements	80
5	Conclusion	81
	Bibliography	83
	Appendix	89
	Abstract	97
	Kurzfassung	99
	Acknowledgements	101

List of Figures

Figure 1 - Heart of Serious Games [5]	2
Figure 2 - LIFEisGAME: Game Mode "Build a Face"	10
Figure 3 - FaceSay GameMode "Bandaïd Clinic"	12
Figure 4 - The Transporters	16
Figure 5 - EmoJump Title Page.....	21
Figure 6 - EmoJump Options.....	21
Figure 7 - EmoJump Credits	22
Figure 8 - EmoJump Player Options	22
Figure 9 - EmoJump Level Screens.....	23
Figure 10 - EmoJump Screenshots Level 1, Level 2 & Level 3.....	23
Figure 11 - EmoJump Scoring Screens.....	24
Figure 12 - EmoJump Game Screen.....	26
Figure 13 - EmoJump Elements	26
Figure 14 - EmoJump Scene	27
Figure 15 - EmoJump Joker	27
Figure 16 - EmoJump Level 2.....	29
Figure 17 - EmoJump Level 3.....	30
Figure 18 - Clipping Log File	34
Figure 19 - Hierarchical Structure (with one active scene)	40
Figure 20 - Jumru 5s Overview	41
Figure 21 - Example Questionnaire.....	42
Figure 22 - EmoJump Composition.....	43
Figure 23 - Game Life Cycle	45
Figure 24 - Player Sprite Jumping.....	50
Figure 25 - Parallax Background.....	50
Figure 26 - Parallax Foreground	50
Figure 27 - Background.....	51
Figure 28 - Game Composition.....	51
Figure 29 - Countdown Pictures.....	52
Figure 30 - EmoJump Ground Structures.....	54
Figure 31 - Crash Scene and Game Over Scene	56

Figure 32 - Level 2 Scenario (3 Images).....	58
Figure 33 - Level 2 Scenario (2 Images).....	58
Figure 34 - Level 3 Item	59
Figure 35 - Level 3 Two Emotions.....	59
Figure 36 - Level 3 Situation Scene.....	59
Figure 37 - Scoring Board Overview.....	60
Figure 38 - Gender Ratio.....	66
Figure 39 - Age Ratio	66
Figure 40 - Enjoyment Playing Video Games.....	67
Figure 41 - Game Types.....	67
Figure 42 - Likes playing on the computer?	68
Figure 43 - Emoticons	70
Figure 44 - Part1/Question 1.....	71
Figure 45 - Part 1/Question 1.....	71
Figure 46 - Question 9 Recommendations.....	72
Figure 47 - Question 3 Difficulty	73
Figure 48 - Main Character Question	76
Figure 49 - Collecting Emotions	76
Figure 50 - Collecting Emotions Group A	77
Figure 51 - Barrier Recognition.....	78
Figure 52 - Lives Recognition	78
Figure 53 - Scene Recognition.....	78
Figure 54 - Final Score Recognition.....	78

List of Tables

Table 1 - Overview of characteristics from related games	18
Table 2 - Classification Level 1 and Level 2	28
Table 3 - Classification Level 3	30
Table 4 - Win/Loose Overview	33

1

1 Introduction

Games can be associated with notions like entertainment, happiness, enjoyment but also distraction, training, encouragement and learning. Games can be seen as playful amusement on the one hand, but can also be associated with a special purpose on the other hand. Children grow up with games irrespective whether we talk about non-video games or video games, but even elderly people use games to train their memory and concentration. An interesting fact about video gamers from the consumer study in Austria 2012 is that more than half of the respondents said that playing games can improve the skills of children, but only one per cent associated the words informative and educational with gaming [1]. In the context of the developing process, we recognized that the field of serious gaming becomes more and more popular. Games concerning education and training should be shown and tested how useful and effective they could be.

In fact there is a new interesting area of application where games are used as learning and teaching utilities and training tools. The games of this area of application are called serious games [2]. Those developments offer the possibility of a close collaboration between programmers, scientists, psychologists and other domain experts. The aim of this field of interest is not only teaching and learning through games, but serious games should rather raise quality of life, well-being and also help people with disabilities [3].

First and foremost, children with physical, sensory, or mental disabilities can gain their skills through hands-on learning, so those games can be seen as a playful support for therapies and healing processes. Studies have shown that regular players increased their cognitive capabilities, which encompass the mental perception of a person [4].

The positive results of such studies can also be seen in the huge and extensive development and design of serious games, which covers different topics that are normally not considered in common game development. Those topics are listed in the heart of serious games, which can be seen in figure 1. The topics are defined by the „Games and Meaningful Play“-program at the Michigan State University [5].

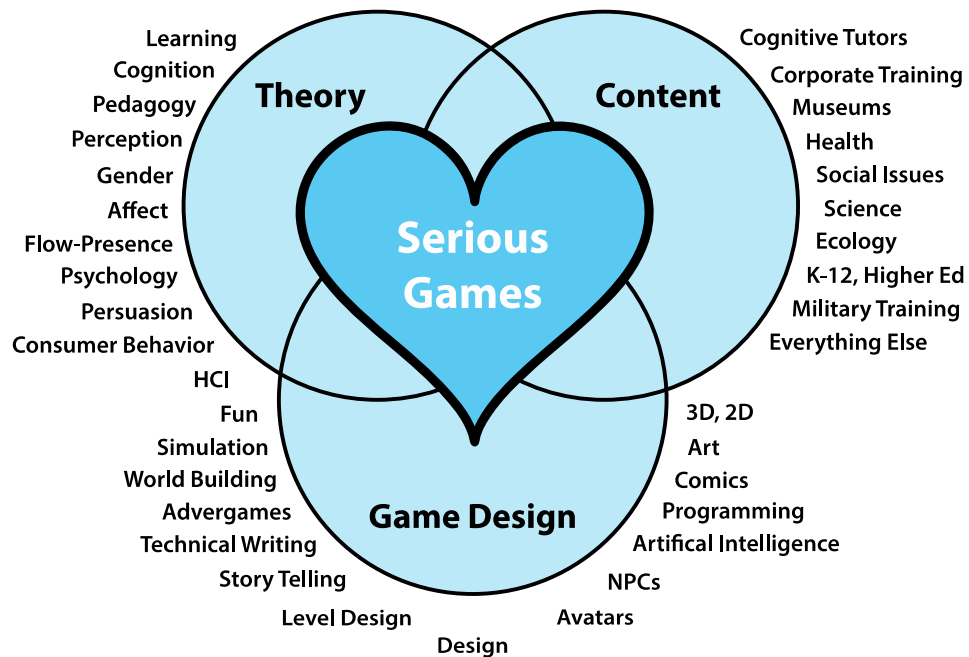


Figure 1 - Heart of Serious Games [5]

1.1 Project EmoJump

Our developed project *EmoJump* covers topics like learning, psychology, fun, world building, storytelling, avatars, programming, cognitive tutors and much more. The serious game *EmoJump* has been developed as an interdisciplinary research project in collaboration with the Department of Clinical Child and Adolescent Psychology of the University of Vienna. The project deals with the dedication as a training study for children in the field of emotion recognition. During this project process we also programmed an underlying game engine with which we parallel developed the game *EmoJump*. The game engine is called *Jumru 5s* and is based on HTML5, JavaScript and CSS3.

1.2 Motivation

Our world is full of games, whether on computers, smartphones, tablets and game consoles or still as board and parlour games, and nowadays games are also increasingly used in the educational and medical environment. The latter aims to not just being interested in the benefit of entertaining children. Furthermore, on the one hand it should help them with specific learn and training methods and on the other hand it is useful for psychologists, scientists, teachers and other experts to establish studies, statistics and new projects.

Games with such an additional benefit can be found in different application areas, but the project *EmoJump* places its focus on training studies especially for children with Autism Spectrum Disorders (ASD). The aim of this project is helping children by training their visual perception of emotions, which also delivers logging information about the children's gameplay for further psychological analysis.

1.3 Research Question & Goal

The idea behind the development of this serious game is to see how children can cope with the game *EmoJump* and how they feel, handle and look at the gameplay. We are interested in this feedback from players, because the objective is the use as helping tool for children with disabilities, so that they learn to understand their own emotions as well as the emotions of others. This thesis aims to analyse and research positive effects of using computer games in an educational context and will show how children, with no health limitations, can cope with the game *EmoJump*. The question we are interested in is if it is possible to develop a game where interface, design, usability and comprehensibleness are suitable to the needs of children and autistic children. Extensive results concerning the psychological background and aspects of the game will be presented in other master theses from students from the Department of Clinical Child and Adolescent Psychology of the University of Vienna, which had also been part of the team.

1.4 Publication

The “Programme Committee for the 5th International Conference on SGDA” accepted the project, including the game engine and the game *EmoJump*, as a full paper. Our work is published in Springer LNCS 8778:

N. Schweiger, K. Meusburger, H. Hlavacs, M. Sprung

Jumru 5s - A Game Engine for Serious Games

The Fifth International Conference on Serious Games Development & Applications (SGDA 2014)¹, 9-10 Oct 2014, Berlin, Germany.

1.5 Overview

At the beginning of this thesis, we introduce our game *EmoJump* and give a motivation of the topic serious games. Chapter 2 starts with the development of serious games, including background information about the topic itself, as well as psychological information with effects on children and also the technical background about design and the various usability aspects. Following from projects, which have already been published and tested from other development teams and are games that are finally available to train emotion recognition. Afterwards we present the characteristics and gameplay of the game *EmoJump*. This explanation and the project features present an overview about the look and feel and give an input on how the experience of playing this game will be. The frontend of the game looks like a normal entertaining screen, but invisible for children in the background, the code logs every interaction and results that are building the basis for scientific analyses. The remaining parts of the thesis are the specific technical description of the game implementation, the whole concept referring to the created objects, the game lifecycle as well as features and logging. At the outcome, the evaluation presents results from a questionnaire, which includes impressions from children while playing the game. Lastly, we present a summary and further improvements.

¹ <http://ddsgsa.net/sgda>

2 Related Work

Games with the special character of learning and training, which are called serious games, are continuously increasing since 2009 [6]. On the Internet we can find different websites that address the topic serious gaming and list different tools. Especially for children with Autism Spectrum Disorders (ASD) there are many games designed and implemented to help them improving their social skills and even to entertain them through the gaming aspects.

Zakari et al. [7] analysed a huge amount of serious games for children with ASD and categorized those games in various categories like technology platform, gaming character, user interaction and purpose. We will present serious games that are not especially designed for autistic children, but are further games that train the emotion recognition skills and the assessment of facial expressions. Before we present related serious games, we have a closer look at the development and definition of serious game development.

2.1 Serious Games Development

The development of a serious game calls for a collaboration between different teams with broad knowledge. In such interdisciplinary projects, teams for instance of the technical sector, physiological sector or psychological sector and design sector are working together, to implement and design a game, which is suitable for children in health care as well as in education.

2.1.1 Serious Games

At the beginning of our project we mistakenly classified serious games just as helping or therapy tools and did not recognize how far-reaching and complex this topic is. The term “serious game” has been used the first time in the book of

Clark C. Abt in 1970 [8]. It took a while until serious games have become more familiar, but also nowadays we can find many different explanations of this topic. A serious game has more than just a story, art and software says Zyda Michael [9], who further explains, that such games include pedagogy. So those games can be found in different application areas, like in the medicine and health section, military, education and advancements and also for different trainings. So there is a wide range of appropriateness of such special games, which should train, inform and educate the persons, which use them [10].

As an overall summary, a serious game could be seen as a game whose primary goal is not fun and amusement, but has a special purpose for education and training. That explanation is not in the interest of all people, because many of them think that education and fun should not always be seen as opposites. Thinking about a serious game should not be associated with a game that couldn't be entertaining or enjoyable, it only has a different aim than other games have. When children are playing serious games, they should not recognize or think about training or participating for a study, they should only have to learn the rules how to play the game. The effect of serious games is increasing and enclosed studies show results that serious games are getting more and more important as tools in our lives [10].

As we already saw from the results of the studies in the related work sector, most of those games achieved successful results in training and enhancement of various areas like social skills.

2.1.2 Psychological Background

The focus of our project concerning the psychological background lies on the improvement of social skills from children with Autism Spectrum Disorder (ASD), more closely to the skills of emotion recognition and comprehension. Autistic children often have limitations and impairments in handling and understanding social connections. At this juncture, autism is a disease, which has no universal and agreed cause and may can have various reasons [11]. But there are several facts about adults and children, which agree with Autism Spectrum Disorders listed from Hans Asperger in 1944: autistic children avoid direct eye contact, have uncontrolled gestures and facial expressions, they

sometimes show negative reactions or diabolicalness for their fellow human beings. A further interesting point is that many of those children have extraordinary capabilities, where Asperger is speaking of an autistic intelligence [12]. Another explanation of autism from Wing et al. describes children with autism, as young persons with difficulties in social interaction, communication and social imagination, summing up as the triad of impairments [13]. In all of these three social areas, we can think about our daily life and how we use our facial expression for social communication or gestures and emotions in social interaction. Miranda et al. presented a research of studies to show that computer-based training and therapies can improve emotion recognition skills and further also their social interaction, but they say that it is important on how facial expressions were taught to children. They have recognized limitations, when children were taught only with drawings or photographs [14].

In our project, our aim is to train the ability of emotion recognition in a more interactive and entertaining way. This ability can be separated in different levels of needs, listed by Abirached et al. [15] as a result of interviews with parents. The first level is the recognition of basic emotions, secondly, the recognition of more complex emotions and lastly, understanding the reasons behind emotional responses or also named mixed emotions. Most of the children can comprehend and recognize basic emotions; some of them can only understand that somebody is sad, when the person shows the emotion expressive, like crying. The degree of difficulty is highly increasing at level two and three, where children have problems to identify the correct emotions [15].

Our game distinguishes between those various levels of needs, to train emotion recognition constructive on their complexity. The modern medicine is not yet able to heal autism, but we could try to help children with trainings and therapies, therefore to improve their abilities and their life regarding social communication [11].

2.1.3 Technical Background

Designing serious games embraces a development with special requirements concerning for instance usability and design. Design principles were listed as a result of analysis of several studies to create a serious game that is separated in

various categories and to give an overview on what is important for children's gameplay [16]. The categories are interface, user control, identification with the game, feedback, transmission of concepts and accessibility. Further, also user testing is necessary with every game, to know the needs of the player and capabilities [16]. Also Abirached et al. listed some technological design principles for serious game development, which allow customization, enable adaptability, incorporate context and consider multiple platforms [15].

The following schema shows a summary of categories and design principles, which may be recognized by developers to optimize a serious game and were listed as results from researchers: [15] [16]

1. Interface & Platforms

The interface of a serious game should have a simply layout, which does not include too small icons to click and should have also less complexity. Handling colours in the graphical environment is an important aspect to increase the player's attention. A menu bar might not be necessary, because it may confuse the player and also the resizing of the screen should not be enabled, to avoid undesired placements. Developers may consider if their game can only be played on a PC, or also mobile by touchscreen or with a completely different technology like playing with a WII controller or handling game play by speech recognition.

2. Customization & Identification

The player should have enough time for learning the instruction, memorizing what to do and heading to new steps in the game. It could be also very helpful for the user, to go back to instructions at any time of the game. The user should also have the possibility to choose a character or to give him the chance to identify himself with the story he is playing on. Consider configuration corresponding to everyone's needs. So customization is a big part of serious games, which should not only deal with the choice of different characters, but also to give parents and therapists the possibility to adapt the game.

3. *Feedback*

Language in games should be used simple and in mother tongue. Serious games are learning and teaching games, so giving the player enough feedback that he will know what to do next is very important. Especially every action should give feedback, no matter if it is auditory or visual to show that the system recognized the action. Negative feedback may result in frustration, which should be handled in the development with care.

4. *Concepts and Context*

Solving a problem should be similar as finding a solution in real world or similar to the learning process in a natural way. This learning process can be optimized through levels or missions, where the difficulty is increasing. Games in cartoon style should be adapted with photographs from real life, to make a reference to the player's environment. Concerning emotion recognition, studies showed that children learn easier to understand emotions, if they are presented in a context or a short storyline.

5. *Accessibility*

Offering the user alternatives, so that he or she can decide between audio and video and therefore can choose the information, which he or she needs to solve a task. Developers should be aware of images or graphics that flash, because they may provoke seizures.

Those listed facts [15] [16] about serious game design and design principles are only recommendations and we will see later on, how suitable our game is concerning those points, but we think it is time to create a new and more interactive serious game that is able to train the emotion recognition skills of children. In the following section we present related projects, where all of these games try to train emotion recognition skills.

2.2 LIFEisGAME

LIFEisGAME [17] [18] is a serious game for children with Autism Spectrum Disorders (ASD) to help them improving their emotional recognition through facial expressions in different levels with various challenges. It is a joint project of a team with persons from Portugal, UT Austin and Microsoft.

The game idea presented in 2011 was to develop four pedagogical game modes [17], where every game mode has a click mechanism to solve tasks with various degrees of difficulty. In the first mode, which is called “Recognize the expression”, the player sees an avatar with a facial expression and he has to identify the correct emotion for it. The second mode is called “Build a Face” (see figure 2) and as the name implies, the player should rebuild different miens on a 3D avatar on the basis of emotions on cards. In mode 3, the player needs to use his camera to control the avatar on the screen. To be able to “become your avatar”, as this mode is named. Lastly, in the fourth mode “Live the Story” is the challenge to find the correct emotion of a short storyline.

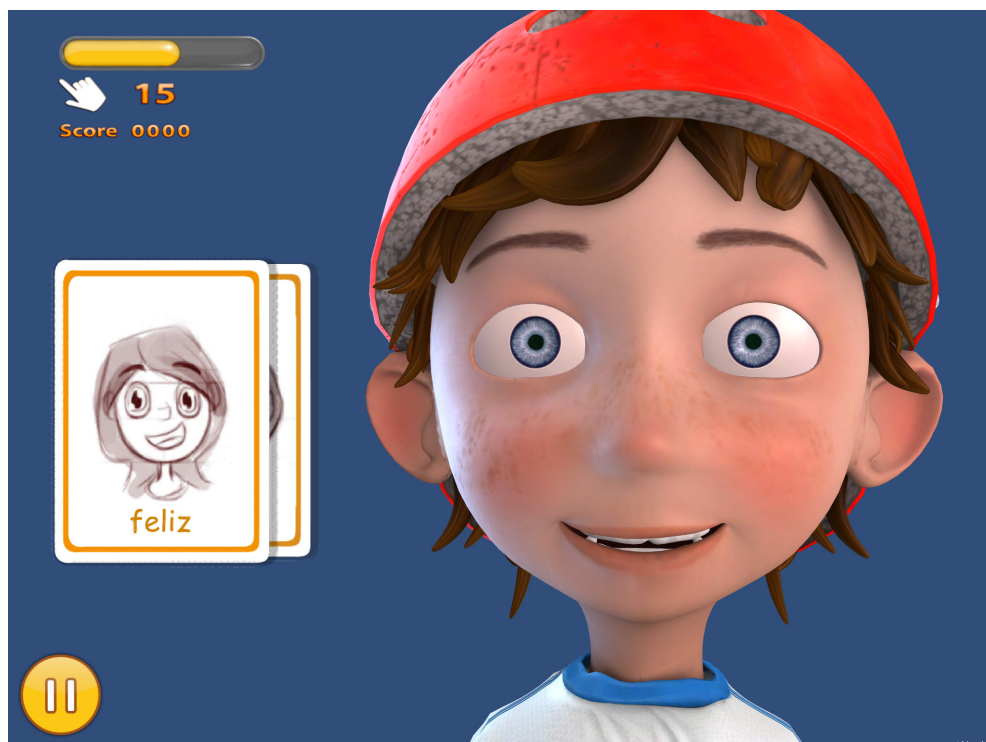


Figure 2 - LIFEisGAME: Game Mode "Build a Face"²

² <http://www.portointeractivecenter.org/lifeisgame/wp-content/uploads/2011/11/photo.png>

The authors did a user preference study with 145 children at the age from 6 to 12 to increase the attractiveness of the game. The objectives of this study were to find out the character preferences of children, in 3D (complex cartoons), 2D (simple cartoons) or photorealistic avatar images. Results show that they prefer animals in all three categories and here they like best the photorealistic avatars. Further, they also evaluated the game through a pilot experiment in 2011 with two autistic children. Both children were male and tested the game in the presence of their therapists. The authors decided to do the observation with the game mode “Build a Face”. Results show that the children enjoyed the gameplay and had fun but also had some problems with the touch-screen interface of the game. They also noticed that children favour individuality in games, why they want to do some kind of customization and choose different characters as avatars. For future work they planned to do further studies to improve the user friendliness and usability [17]. So in December 2013 the developers presented the LIFEisGAME prototype for iPad version and tested the game again with 11 children with Autism Spectrum Disorders (ASD) [19]. Now, the game consists of five different modes, where the main character of each mode is a young comic boy. Game mode 1 is now named “Recon Mee-Match” and presents a new game idea and also the modes 2-4 have been developed differently to the modes presented in 2011. Mode 5 “Build a Face” is the same as before, where the player has to draw a facial expression. The game has auditory and visual positive feedback but no negative feedback. Concerning the tests with the children, who enjoyed the gameplay, showed that they really liked the character to play with and found different advantages in every game mode. Attractive graphics and dynamic interfaces are very important and the usability of the game was very easy to understand.

The last statement is that they will improve LIFEisGAME so that it will help children with ASD and could be used as a teaching tool [19].

2.3 FaceSay

FaceSay [20] [21] [22] is not only a simple serious game, but in fact it is a training program for autistic children to improve their skills in the social sector like emotional and facial recognition. The idea behind FaceSay is the training to

recognize emotions moreover through eyes than from the mouth, because studies have shown that children with Autism Spectrum Disorder focus on the mouth. It is a game from 2008 for the PC and it is not an online game but there is a free demo version online. In the game computer animated realistic avatars are used. One of the three games of FaceSay is called “Amazing Gazing”, where the player sees a realistic face in the middle of the screen and should concentrate on the eyes in this face. The face is surrounded of other smaller faces, numbers or objects and the player has to choose the right face, number or object at which the face in the middle is looking or “gazing” at [22]. The feedback of this game is only positive and there is a nice feature programmed in addition, where the auditory commendation includes the name of the player. The next mode “Band Aid Clinic” is a game mode, where the player sees a face horizontally subdivided in three parts. One part is missing and the player has to choose the right one from a randomized selection and after choosing the correct part, the face will be animated. Figure 3 shows a demonstration image of this game mode.

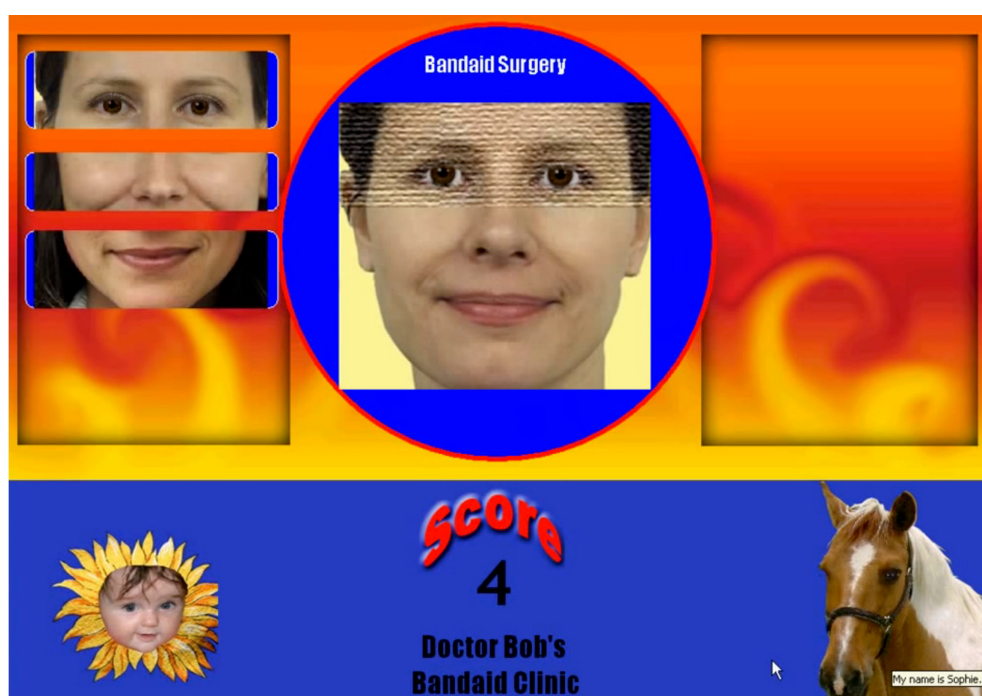


Figure 3 - FaceSay GameMode “Bandaid Clinic”³

³ Screenshot from YouTube Demo Video (<https://www.youtube.com/watch?v=wRV8LlsmIPk>)

“Follow the leader” is the third game mode and focus on training the ability to differentiate between facial expressions. The player sees two faces and has to decide if both faces show the same emotion. The simple solution is to click either on the yes or no button on the screen, but in higher levels the difficulty is increasing and the player has to change the expression of the eyes by moving them up or down to get two similar facial expressions of both avatars. Interaction in this game could be done either by touchscreen or by mouse click. The authors did different studies, but only one is free and available on the Internet. In this study, the aim was to analyse if FaceSay can help children in social life. In this controlled study they had 49 participants with Autism Spectrum Disorder (ASD) and tested three games from the FaceSay program. The analyses are based on results from pre- and post tests, in which different skills have been tested. Checking the emotion recognition skills, they used pictures and drawings of emotional faces, which have already been used in previous studies to see, how good children can assign the correct emotion to those images. Checking the facial recognition they used the Benton Facial Recognition Test [22]. In this test the children have to look at the discrimination between several pictures of faces, which are highly cropped, and to decide which facial expressions are emotionally identical. Lastly, they also checked the social skills of the children in an observation during their free time or recess and documented results in a rating system. Investigations and results are very extensive but in summary it is a success that they could suggest that the computer-based interactive simulation program FaceSay improved the participants social skills like emotional and facial recognition and social interactions. The children were entertained by those games and they even asked for playing. Improvements could be done in the design of the game. The developers see future work in gaze tracking studies because it would be interesting to find out how much attention children give to the eye section in a humans face. An interesting aspect at the end of the study is the suggestion to a multidisciplinary project, which could be more useful, because various domain experts are working hand in hand. Closing, the results demonstrated that FaceSay is an encouraging program as a social skill-teaching tool for children with Autism Spectrum Disorders (ASD) [22].

2.4 Mind Reading

Mind Reading [23] [24] is a commercial game program, which includes an emotion library, learning centre and a games zone. The aim of Mind Reading is to teach adults with Asperger syndrome or high-functioning autism (HFA) in emotion recognition. The developers describe Mind Reading as an interactive guide that uses auditory as well as visual channels for training and playing [24]. The system covers 412 emotions and mental states separated into 24 emotion groups, where a short video clip demonstrates each group. The emotion library consists of all those emotions, where the user can search for and read description, play emotional behaviour and listen to emotions. The learning centre teaches emotions through quizzes and lessons and the user gets rewards for correct answers. Lastly, the game zone constitutes of five play modes, where the user can playfully learn about emotions, constitutes of five play modes. The characters, which represent the emotional faces, are real human photographs of various ages, gender and different nationalities. It is also possible to interact with the character Harry Potter (aka Daniel Radcliffe) and control his emotions [25]. The interaction with the program takes place via navigation by the mouse. Experiments were done with three separated groups. Participants who had used Mind Reading for a specific amount of time were named Software Home Users which are also can be seen as the intervention group and others who did not use the program were called control groups. The pre- und post-tests included three different tasks, which the participants had to solve. In one task, the person had to recognize the correct emotion only of the eyes region. For the next task, the participants had to choose the correct emotion out of two possible answers by listening to an audiotape. In the last task, the persons were shown clips and they had to match the correct emotion of the person in the video. The duration between the two tests have been 10-15 weeks but this was already enough, that the test results showed improvements of the participants who used Mind Reading. Users of Mind Reading could significantly improve their abilities in emotion and mental state recognition. In relation to the test results of the typical control group, which showed that people with Autism Spectrum Disorders (ASD) have problems in choosing the correct emotion to a face, but after the training weeks, their score was significant higher than at the post test.

They also did a second experiment with the same tests, but in contrary to the intervention group before, this group also tested the software but they additionally attended to weekly meetings with a tutor. For both experiments, the results showed better emotional understanding at time 2 (after the training) than in time 1 (test before the training). An interesting aspect is the fact that the adults liked the idea of joining groups and came together for training sessions, which was maybe the main reason that they attended to that study. For future work longer studies would be necessary to improve generalisation of those results arguments, as well they planned neuroimaging studies and gaze tracking studies.

2.5 Transporters

The Transporters [19] [26] [27] is actually an animated television series and available on a DVD for training emotion recognition skills. The DVD consists of fifteen episodes, which represent short films of about five minutes. Each clip has one emotion or mental state as key issue. The emotions include the six basic emotions: happy, sad, angry, afraid, disgusted and surprised and more complex emotions: excited, tired, unfriendly, kind, sorry, proud, jealous, joking and ashamed [28] [29]. There are eight characters that illustrate the facial emotions. Those characters are various toy vehicles in a virtualized bedroom of a boy, where each vehicle has a real human face on it, which we can see in figure 4. The faces are programmed in 3D from young to old, with different nationalities and both gender. Adapted to each episode, the DVD includes quizzes about emotion recognition of every key emotion or also a combination of questions including several emotion types. There are different types of questionnaires so children can either match faces to faces, faces to emotions or situations to faces in easy or hard difficulty. The quiz returns only positive feedback by congratulations and the appearing of a reward. If a question is not answered correctly, it will be asked again and again until it will be correct. The DVD includes a detailed guide for parents and advisers how to handle and show the episodes to children and also to observe the children behaviours through watching episodes [28].

Concerning the study about The Transporters, three groups have been built for pre- and post tests, after a four-week training session. There was an ASC

intervention group with 20 children, who watched some episodes of The Transporters every day and further an ASC control group and a typical control group. They were tested in four levels of generalization including emotional vocabulary and situation-facial expression matching. Those tests before and after the four weeks have been absolved by asking the children questions and showing them pictures of faces, while the results and answers were recorded manually. The two control groups showed no significant improvements in emotion recognition but the success could be found in the results of the intervention group, where test results demonstrated the effectiveness of the program and the improvement of the emotion comprehension of the children. The fact that the children watched the episodes with their parents leaves open the question, how are the effects of the episodes on its own. Further studies may focus on an improvement on lower-functioning children with less verbal methods.



Figure 4 - The Transporters⁴

⁴ http://www.scienceinpublic.com.au/wp-content/uploads/GuideBookCover1_00001.jpg

2.6 Summary

All of those games focus on the aim to train and teach emotion recognition skills and to improve social communication skills. Those games are not limited to children, but they could also be used of adults with autism or persons who have limitations in emotional understanding. The studies succeed with adults as well as children with Autism Spectrum Disorder (ASD).

LIFEisGAME consists of various game modes to train emotional understanding. The game modes are interactive but with no adventure/jump and run character. The test results are based on observations of the gameplays, on parents' questionnaires and opinions of the children therapists. An interesting proposal of the therapists is to add instructions videos to each game mode for a better understanding [19].

FaceSay also consists of various game modes but focuses on the recognition of expressions based on the eyes of a face. The game characters are realistic human faces, but also animals took place in the game as principal supporters. Every game mode has a strategically interactive character like puzzling and you can use mouse or touch by clicking and answering. The extensive tests showed that the participants improved their social skills.

Mind Reading constitutes of a huge set of emotional material, represented visually through faces, auditory and with textual information. The program is available on DVD and consists of a database of emotions and mental states, as well as teaching tools and games to train the emotional understanding. The experiments showed significant improvements in emotion recognition but for generalization of those results, there are additionally studies necessary.

The Transporters is a series of various episodes concerning emotional behaviours, represented through toy vehicles with real human faces on it. The DVD includes also quizzes with different questionnaires about emotion comprehension. The study showed that children who watched the episodes could improve their emotional understanding significantly more than those who did not watch it [28].

	Screening	Avatars/Characters	Availability	Interaction
LIFEisGAME	3D	Photorealistic	On request	Touch or mouse
FaceSay	2D	Realistic (human and animals)	Buy and free demo online	Touch or mouse
Mind Reading	2D	Photos of real human faces	Buy and short demo online	Mouse
The Transporters	3D	Toy vehicles with real human faces	DVD	Watch and mouse

Table 1 - Overview of characteristics from related games

Most of those games have a minor gaming character and are more a representation of a learning tool than games. Table 1 shows an overview of the games, which were compared in this section. We can find puzzle games and quizzes but no jump and run or adventure aspect. None of the related games have a logging mechanism, so all of those studies have been documented and tested manually through additional equipment on how strong the affect of each of these programs on participants is. Abirached et al. [15] published in his study 2012 that several serious games lack in the gaming character and interactivity. Most of the researched games only used static photographs or static characters to interact with. For an enhancement of the emotion recognition skills of children with autism, they say that is important to check user needs and requirements to adapt games in a way that children may not loose their interest in gameplay too quick [15].

3

3 Serious Game – EmoJump

Our serious game *EmoJump* is a single-player and endless runner game that is based on our developed framework *Jumru 5s*, which we implemented parallel for serious games and includes a logging mechanism. *EmoJump* is a training game for children with Autism Spectrum Disorders (ASD) that have problems in emotion recognition, in the age between 4 and 10 years.

3.1 Game Idea and History

The game concept has been created in close project cooperation between the Department of Clinical Child and Adolescent Psychology of the University of Vienna and the Faculty of Computer Science, Entertainment Computing⁵. Originally, the assignment was an advancement of the click-through game *Emotion Detective*. In this game, the player is shown comic situations and has to choose out of a range of faces showing different emotions, by clicking on the correct emotion type.

After several team meetings we created a completely new game concept, which is no longer only a click-through game, but now a special case of a jump and run game with three main levels and a mini game. The name of this new game is *EmoJump*, as a result of the words emotional jumping. At the beginning, several meetings took place, to make decisions, which were necessary for starting the development. We determined that we would work together on storylines for each level, which include textual descriptions about every action step and also delineations of game scenarios and how different parts should be designed.

⁵ <http://cs.univie.ac.at/EC>

The three main levels of the game are mapped as the levels of needs, described in section 2.1.2 (Psychological background). Those are the basic principles for our level design with various grades of difficulties.

3.2 Game Concept

The concept behind *EmoJump* is the creation of an endless runner game, where children navigate a character through various worlds by jumping and collecting coins that are representing emotions. While running, they choose the correct emotion, which matches a scenario pictured in a bubble on the screen. Many different coins with faces on it, showing the emotions, come to the game screen and the player has to jump over or collect them. The faces of various characters show different emotion types like happy, angry, sad or scared.

After the mission statement has been formulated, the next step was to keep records of game logic, so that the development could start. The process was not to create a complete and final document with design and logic and to start programming afterwards, because this is how games in the 1980s had been built, says Ernest Adams [30]. We further programmed a suitable game engine that delivers the programming elements for the game. This is also a very different approach and resulted in many tuning, changings and modifications during the development phase. The storyline, as mentioned before, covers the most important facts about the game and includes nearly the whole concept of the game as well as pictures of the screen flow. We have several versions and the storyline was changed and expanded parallel to the development. Sometimes we also adapted the storyline afterwards, in consideration of already programmed elements and actions.

There is one main storyline that explicitly explains how to navigate through the gameplay and further storylines for level two and three. Several meetings took place to find a final version of the first storyline to start programming, but also many changes and additions can be found in these documents. The idea behind the design documents was not only to have a basis, but also to have documentations about the development for further deployment.

Adams [30] presents a process of game development, which is separated in three stages: concept stage, elaboration stage and tuning stage. The concept

stage includes decisions concerning the game concept, the audience and players, which do not change anymore. We already know that we want to design and implement an endless runner game for children, who have problems with emotion recognition, to collect emotions concerning scenarios. The next stage is called elaboration stage, where ideas have to come from general to specific arguments, so in our case, writing storylines, starting programming and creating prototypes of gameplay sessions began.

3.3 Game Structure and Screen Flow

EmoJump consists of different screenings, in our project named as stages, where the player can switch between them. The first screen, which the player sees is the menu. The menu shows the title, represented as the logo of *EmoJump* and the buttons *Start*, *Options* and *Credits* (see in figure 5). The *Options* Button leads the player to the screen showed in figure 6, where he or she can turn off or on the music and sound effects separately. The language button switches all elements immediately to the chosen language after clicking. The player can choose between English or German. The *Back* Button leads the user backward to the title page.

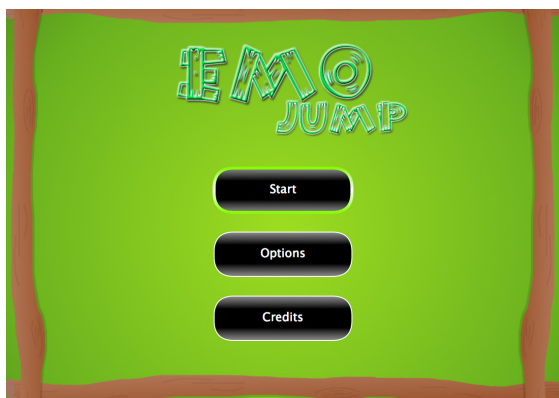


Figure 5 - EmoJump Title Page

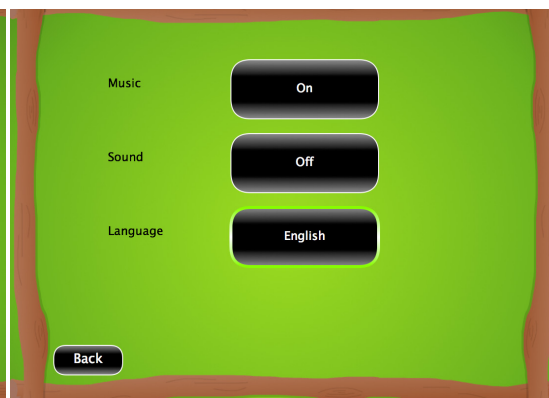


Figure 6 - EmoJump Options

Back at the title page the player can also have a look on the credits of the game, including all persons, who have ever worked on this project and also references of graphics, music and sound effects. In figure 7, we see clippings out of the scrollable credits screen.

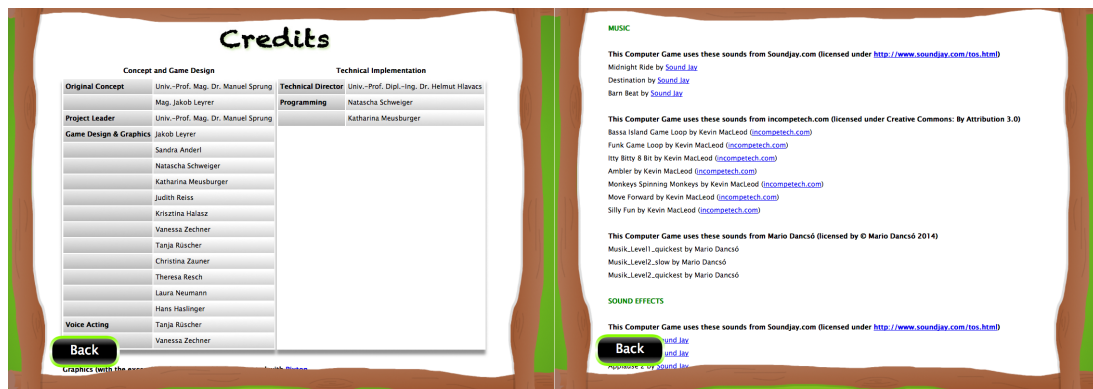


Figure 7 - EmoJump Credits

The same goes here that the *Back* Button leads you to the title page, where now the player can start the game. The *Start* Button opens the next screen, which shows two different characters. The player simply clicks the boy or the girl he or she wants to play with, to open the next screen where the user has to input the player name (see in figure 8).



Figure 8 - EmoJump Player Options

After the player has chosen character and name he reaches the level selection. The first level selection shows the three main levels: level one, two and three. After clicking on a level, an instruction video will be shown to the user. This video can only be skipped when the player watched it once. If the video has finished, the screen with the sublevels will be presented to the player automatically (see both screens in figure 9).



Figure 9 - EmoJump Level Screens

The sublevel screen is the last screen before the game starts and the player leads to the gaming stage. Figure 10 shows a screenshot of each main level setting, which are explained later on, because of their different bubble and scenario screenings. From left to right we can see level 1 with the green grass ground and the girl running. The middle picture of figure 10 shows level 2 with the boy running reversely and the right picture shows level 3, where the boy is running on an airport.



Figure 10 - EmoJump Screenshots Level 1, Level 2 & Level 3

Each level consists of multiple scenes, where the player has to collect the correct emotions that match to those pictures. After every scene, a scoring overview will be shown to the player (see in figure 11, left picture). At the end of a level, the system calculates the ratio between correct and wrong collected emotions, crashes and scenes to tell the player if he even passes the level or not. Figure 11, right picture, shows a screenshot with confetti and a golden trophy, which will be displayed, when the user plays a level successfully.



Figure 11 - EmoJump Scoring Screens

Those screenings show an overview on how the game *EmoJump* is structured and will be explained detailed in the following topics, but before we want to have a closer look on the purpose of the design decisions concerning this game.

3.4 Design Concept

As mentioned before, the whole design of *EmoJump* is a result of the collaboration between the teams, but the psychologist students created most of the graphics. We designed some graphics and decided together how graphics were placed and animated. The core graphics like characters and scenarios are creations and outputs of the software Pixton⁶, which originally is an online tool that provides elements to create comics in a very simple way by using a drag-and drop-mechanism. The light version is not available for commercial use so they had to pay an amount that the students could also download the created scenarios [31]. That was also a very important fact to store the scenes online, because so they always had the possibility to modify the created scenes, which were even more than 200 scenes for the game.

The logo of *EmoJump* (shown in figure 5) is a creation with the font Burnstown Dam⁷ and adapted with colour and shadows that fit the general view. The leading colour in the menu stages of the game is green, which has no deeper reason for this decision and maybe ought to be reassessed because of players

⁶ <http://www.pixton.com>

⁷ <http://www.1001freefonts.com/burnstowndam.font>

that have a red green colour deficiency. Wooden elements are integrated everywhere, like the frame of the screen or also later in the gameplay screen, where the lives are placed on a wooden board.

3.5 Player Motivation

EmoJump is especially designed for children aged between 4 and 10. The characteristics of the game are completely created on a comic basis and there are no realistic pictures integrated. We focused on gender equality so the player can choose between playing with the boy or the girl comic character. The user has to have the chance to identify him with the chosen character and that he or she also likes it, because the player looks at the character for a long time of the playing period [30]. In our case, intentionally the character is created in Pixton, because the player has to think about the emotions that other characters in the scenario do have, which are very similar to the protagonist. So it would not make sense, if we choose animals as protagonists to play with. The player has not only to choose the character, but also to input his or her name, which is also very important for the logging mechanism and further identification.

In section 3.3 we already stated that between the main level and sublevel selection the player is shown an instruction video. This video should increase the motivation of the player and of course it should explain the main necessary functions about playing the game in a correct way. The instructions videos are about 2-3 minutes long, because the explanation is very detailed and there is also a screencast of somebody playing the game in it, to give the player a gaming demonstration.

3.6 Game Mechanism and Levels

The game mechanism is an endless runner game, where the player has to collect things, in our case coins with faces on it, which match to a shown scenario. That is the overall definition, but each level has its own challenges. Every game immediately starts after choosing it in the sublevel menu. The gameplay setting appears on the screen and a big coin in the middle of the screen shows a countdown from 3 to 1 and “Go!” before the first scenario (Figure 12, element G)

is presented to the player. First of all the gameplay is paused and the player stands still, looking at the scenario in the bubble, so the player has enough time to try to understand at first which emotion fits to the person in the scenario. This is an important aspect, which we have already listed before and is mentioned in [16] where users should have enough time to make decisions in serious games.

The game starts by hitting the space bar and the player starts to run while the clock is ticking parallel (quickly rotating in clockwise direction in level 1/3 and anticlockwise in level 2). The clock is important in level two, which we will see in the section 4.6.1, explaining the levels. Visually the player is running, but logically the ground is moving from the right side of the screen to the left in a repetitive process. Barriers and emotions are moving in direction of the player and he has to jump over the barriers and emotions, or collect the emotions he thinks they match with the scene.



Figure 12 - EmoJump Game Screen

Game Elements

The collection of an emotion coin lets the backpack bump once. The backpack is in the left corner of the game screen, which you can see in figure 12, element C and also a zoom in at figure 13. The barrier is a crash point and when the player collides with it, he loses one life, represented as hearts on the wooden board (see in figure 12, element A). The three hearts will be

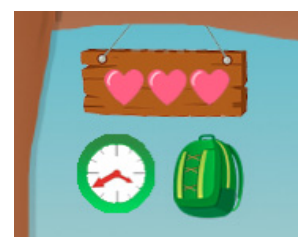


Figure 13 - EmoJump Elements

refilled with every new scene, but when the lives are equal to zero, the game over screen is shown to the player and he missed this sublevel. After the game over screen disappears, he can decide to go back to the menu or to restart the actual sublevel he had been playing.

Scenes

A scene is a psychological scenario, which always shows a person in a red t-shirt in a life situation. The child who plays the game has to interpret this situation. Every sublevel has different scenarios, randomly viewed to the player. The children have to think about the correct emotion felt by the person in the red

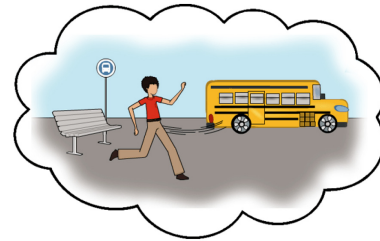


Figure 14 - EmoJump Scene

shirt. Sometimes, if it had been necessary, the face was deleted (as in figure 14), so that there is no hint for the correct emotion to recognize in the face. There are also special scenes, where the emotions are ambiguous. Ambiguous means that it is not clear what the primary emotion of a scenario is. The handling of such ambiguous emotions will be explained in the technical part, but we tried to exclude ambivalent emotions to disambiguate the decision for the player.

Joker

Another helpful supplement is the joker button. This button is always available before the player is running and while he is looking at the bubble. The joker is the auditory description of each scene, recorded by the psychology students. In each sublevel, the player has the possibility of hearing the joker for three different scenes he wants. While listening to a description



Figure 15 - EmoJump Joker

of a scene, the button is pulsing and moving and afterwards the number in the middle of the icon is decreasing from 3 (as seen in figure 15) to 2, 1 and 0.

Level 1

In Level 1, always one scene is active with one correct emotion to collect. The scenes are represented as bubbles. There are 12 sublevels, which vary in the

number of collectable emotions and speed, as well as in different backgrounds and graphic elements. The actual number of correct emotions the player has to collect is three, but also when three correct emotions passes the screen, the end screen of the scene will be displayed, because the player missed one to collect.

Table 2 shows a classification of speed and the amount of correct and wrong emotions of every sublevel from level 1 and 2:

Sublevel	Correct Emotion(s)	Wrong Emotion(s)	Speed	Theme Level 1	Theme Level 2
1	1	1	Slow	Summer	Summer
2	1	1	Quick	Summer	Summer
3	1	1	Quicker	Beach	Village
4	1	1	Quickest	Beach	Village
5	1	2	Slow	Desert	China
6	1	2	Quick	Desert	China
7	1	2	Quicker	Mountains	Building
8	1	2	Quickest	Mountains	Building
9	1	3	Slow	Winter	City
10	1	3	Quick	Winter	City
11	1	3	Quicker	Night	Night
12	1	3	Quickest	Night	Night

Table 2 - Classification Level 1 and Level 2

Level 2

This level covers the next section of emotion recognition. Children have to understand that emotions can depend on expectations and imaginations, which are not expected in reality. The player has to differentiate between the various emotions. First, the player is presented a short comic story from left to right (see in figure 16). The comic consists either of two or three pictures. The second picture represented in the middle of the comic strip should only help to clarify a situation. The first and third pictures are named as situations: belief and reality. The difficulty in this game is the fact that the child has to play the comic strip backward. To clarify this game situation, the player is visually running now from right to left and also the clock runs anti-clockwise. The first actual picture of the scenario is the reality picture, which is bigger and coloured, the other pictures are small and grey. After collecting enough emotions for the presented situation, the actual picture swaps to small and grey and the next one, the picture that shows belief, turns into a coloured and bigger one. This level has

also 12 sublevels, with different backgrounds and various speed levels, which can be figured out in table 2.

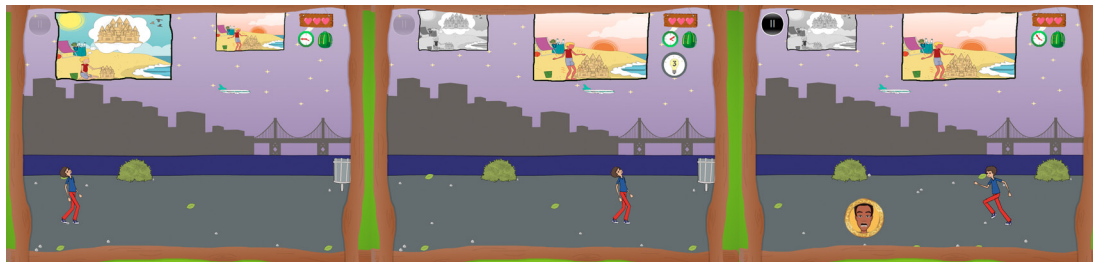


Figure 16 - EmoJump Level 2

Level 3

This level belongs to mixed emotions. The child has to understand that in some situation we can feel multiple mixed emotions. At the beginning of each sublevel we can find a mini game, implemented with jump and run character. The player has to try to collect a hovering item, which moves randomly from the right side to the left side of the screen and backwards. As we see in the example in figure 16 in the left picture, the player has to collect the item bicycle to activate the following scenario. The item always presents the main element of the upcoming scene. The amount of correct emotions in this level is always two, because the child has to understand that the feelings in a showed scene can be different. Visualized by showing the child a scene picture and after that two different mind bubbles that represent the mixed emotions, which appear on the screen (see in figure 17 on the middle picture). After tapping the space bar and the player starts running, the scene picture will be again placed on the screen and the mind bubbles before, disappear again (see in figure 17 the right picture). The idea behind this change is to animate the child to remember the scenes presented in the bubbles and to collect two emotions instead of one. As an example for a better understanding we will have a closer look on figure 17. First the player is shown the right picture in figure 17, where the boy is invited to make a biking trip. After that the player sees the two bubbles with the mixed emotions the boy with the red shirt can have. Either he will be happy to have fun with the bicycle and his friends or he is scared, because he maybe falls of

and harms himself. As explained, the two bubbles will disappear again and the first image will be shown while the player starts running.

Lastly, also this level consists of 12 sublevels, with various scenarios, items, backgrounds and speed categories (see in table 3).



Figure 17 - EmoJump Level 3

Sublevel	Correct Emotion(s)	Wrong Emotion(s)	Speed	Theme Level 3
1	2	1	Slow	River
2	2	1	Quick	River
3	2	1	Quicker	Airport
4	2	1	Quickest	Airport
5	2	1	Slow	Summer
6	2	1	Quick	Summer
7	2	2	Quicker	Buildings
8	2	2	Quickest	Buildings
9	2	2	Slow	Grassland
10	2	2	Quick	Grassland
11	2	2	Quicker	Factory
12	2	2	Quickest	Factory

Table 3 - Classification Level 3

Audio & Feedback

The audio is separated in two categories, so that the player can either switch on or off the background music or the sound effects. We use different compositions in the whole game so the menu stages have their own music, as well as each sublevel depending on the game speed. For instance, every quick sublevel has the same sound in level 1. The player runs on various grounds like grass, snow or sand. All of these grounds have their own running sound, because every run sounds a bit different. There is one crash sound if the player collides with a barrier and a landing sound after the player jumped.

Sound effects embraces the sounds of clicking a button, running, jumping, landing, countdown effect, sound of emotions, collecting an emotion, crashing and the applauses. Music files are only the background music of the title page and the music of every sublevel.

The psychologists preferred that the children only get back positive feedback. If they collect an emotion, visually the backpack is bumping and they also hear a very short sound effect. It doesn't matter if the emotion is correct or wrong, because the game gives always the same feedback. This has the reason that the children will not only collect icons because of a visual or auditory feedback that already clarifies what is the correct coin to collect. After every scene, the screen shows an overview with a picture of the scene, as well as a green check mark with the amount of collected correct emotions and a red cross with the amount of wrong collected emotions. This is only a feedback about the points and not a feedback that shows which facial expressions was the correct emotion for the current scene. We can see an example in figure 11, left picture. At the end of a sublevel, the player is given feedback how successful he was which depends on the amount of correct scenes. There is a score bar, which fills dark green for all correct scenes and subtracts after filling the wrong played scenes with light green. Below the score bar are three trophies in gold, silver and bronze. If the player succeeds, one of these trophies will be shown larger and applause rings up, where this applause is available in three different levels of power, concerning on the percentage he passed the level. If he succeeds and can head the next level, there is also confetti hovering over the whole screen (see in figure 11, right picture). In the case of losing a level, the player is shown a picture of his sad looking character and can try again.

Score Mechanism

The mechanism behind succeeding a level changed during the development process several times. Many calculations have been considered to find an optimized ratio between correct and wrong emotions and scenes.

To pass a level a certain percentage has to be achieved, but the evaluation also depends on the ratio between correct and wrong sent emotions.

If the sum of the wrong sent emotions is higher than the sum of the correct emotions the calculation factor is 1, otherwise we calculate:

$$\text{Factor} = \frac{\text{\#correct collected emotions}}{\text{\#wrong sent emotions}}$$

For this reason, we want to avoid that wrong collected emotions are stronger weighted than 1, in the case of the randomized process that there were more correct emotions sent than false ones.

$$X = \text{\#correct collected emotions} - \text{factor} * \text{\#wrong collected emotions}$$

After that, we can check the criteria if the player succeed the scenes. Therefore we need the number of correct emotions per scene, which is a value that is determined by the psychologist. The higher this value, the longer takes each scene play because more correct emotions have to be collected or maybe passed the player, when he misses to collect them.

The following calculation evaluates if a scene has been played positive, if this value is greater than 50%:

$$\text{Scene} = \frac{X}{\text{\#right emotions per scene}} * 100$$

Now, we have calculated the percentage of one scene, which will be done for every scene. If the value for one scene is greater than 50%, the sum of won scenes is increasing.

The final end result is the win percentage value of the actual played sublevel:

$$\text{Percentage} = \frac{\text{\#won scenes}}{\text{\#played scenes}} * 100$$

Depending on the percentage, the player gets different feedback:

Percentage	Trophy	Applause	Level succeed
<= 56 %	-	-	No
>= 57%	Bronze	Strength 1	Yes
>= 80%	Silver	Strength 2	Yes
>= 90%	Gold	Strength 3	Yes

Table 4 - Win/Loose Overview

Logging Mechanism

One of the most important facts about serious games is the evaluation and to see if the game has an effect on the players. For the purpose that the psychology students can later analyse or examine the gameplay of the children, we log information after each sublevel. A log will be immediately stored after playing a sublevel, as a CSV file on a webserver. This log includes some main facts like name, date, time, browser and average FPS (frames per second). Furthermore it includes, which main level and sublevel he or she played and if he or she succeeded. Lastly the log stores, how long the user played the sublevel in which speed factor and how many jokers he or she took.

The following values are also stored:

- sum of all sent emotions
- sum of all sent correct emotions
- sum of all sent wrong emotions
- sum of all collected emotions
- sum of all collected correct emotions
- sum of all collected wrong emotions

These are the main information about the gameplay and there is also detailed information logged about every scene and the players' actions in this scene.

The headers of the stored facts of every scene are:

- Scene (ID of the scene, to remember which scenario was displayed)
- Joker (Number of jokers he took for this scene – default 0)
- Right emotion (Correct emotion for this scene)
- Number right sent (Number of correct emotions which he could collect)

- Number right collected (Number of collected correct emotions)
- Number false sent sum (Number of wrong emotions sent in the gameplay)
 - False emotion (Wrong emotion of this scene)
 - Number false sent (how often this wrong emotion was sent)
 - Number false collected (how often the player collected this wrong emotion)
- Number false collected sum (Number of all collected wrong emotions)

The values “false emotion”, “number false sent” and “number false collected” are repeating in relation to the number of wrong emotions that were displayed in this scene. So if it has been a level with one correct emotion and three wrong emotions, the CSV file has six columns more than a level with only one wrong emotion. Figure 18 shows a clipping out of a log file. The example shows that a person named “Natascha” played sublevel 6 in level 2 that includes 7 scenarios.

1	Name	Datum	Uhrzeit	Browser	FPS					
2	Natascha	15.11.14	13:20	firefox	59					
3	Level	Sublevel	bestanden	nr-sent-sum-all	nr-sent-sum-right-all	nr-sent-sum-false-all	nr-collected-sum-all	nr-collected-sum-right	nr-colle	
4	2	6	ja	112	42	70	48	42		
5	Scene	Joker	right-emo	nr-right-sent	nr-right-collected	nr-false-sent-sum	false-emo	nr-false-sent	nr-false	
6	14	0	happy	3	3	2	sad		2	
7	72	0	scared	3	3	2	happy		2	
8	7	0	happy	3	3	2	scared		0	
9	32	0	scared	3	3	8	happy		5	
10	34	0	scared	3	3	7	angry		2	
11	21	0	happy	3	3	6	scared		1	
12	65	1	angry	3	3	3	happy		1	
13										
14										
15										

Figure 18 - Clipping Log File

3.7 Technological Description

We started the implementation of *EmoJump* with an existing HTML5 game framework called LimeJS⁸, but after longer coding processes, which resulted in a playable but buggy game prototype, we decided in prior consultation with our adviser, to implement our own game engine. The reason was that we found

⁸ <http://www.limejs.com>

limitations in the framework LimeJS and also focused on the part to implement our own logging mechanism, which could also be useful for other serious games. The next step in the implementation process was according to this that we started from scratch. First of all, we defined the skeletal structure and began to implement our game engine *Jumru 5s* and after that we parallel started to implement *EmoJump* based on *Jumru 5s*. In the following section, we show the programming languages and tools we used as well as the implementation itself and the programmed functionalities.

3.7.1 Technology & Tools

The technologies we used are HTML5, JavaScript, CSS3 and some PHP for storing the log files on a webserver. In this subchapter, the basics and short descriptions of those technologies are shown and why we have chosen HTML5 for the development.

Eclipse

We use the Java development environment named Eclipse⁹, which is an open source project and available on the Internet [32]. More exactly we worked with the Eclipse Java EE IDE for Web Developers in the version: Juno Service Release 2. We use the plugin Eclipse Subversive¹⁰ - Subversion (SVN) Team Provider with which we are able to directly upload changes in the project to a server, including comments and also used as a permanently backup.

HTML5

The Hyper Text Markup Language (HTML) is the core language of the World Wide Web [33]. The version 5.0 is the result of a long developing process and also a very young recommendation of W3C, passed in October 2014. HTML 5 offers some new elements concerning semantic, controls, graphics and multimedia [34]. Especially interesting in our case were the new elements audio and video as well as the element canvas. The main functionality of this

⁹ <http://www.eclipse.org>

¹⁰ <http://eclipse.org/subversive>

element is developed in the game engine, but they are further important for creating those objects in the game. Some elements from HTML 4 have been removed and instead of those, the W3C advises to use CSS (Cascading Style sheets) [34].

The decision to use HTML5 bases not only on the advantages of a new technology and on our invention. Furthermore the psychologists aimed to combine multiple serious games on a web platform with a single login for children. They want to embed games for a combined training tool and recommended us, to implement the game in HTML5.

JavaScript

JavaScript is a programming language of the web, which is directly executable in the browser and does not need any pre compilation [35] [36]. JavaScript is either directly embedded in HTML pages or embedded through external files with JavaScript code. A huge advantage of JavaScript is the live execution of the code, so that developers can see immediately changes of their work. A disadvantage of JavaScript is the different interpretation of the code of various browsers, which we also mentioned in our development process [35]. We mainly focused on the usage of JavaScript, but also PHP has been necessary to store data on the web server as a result of the logging mechanism. It is only a small part of the whole code, but the only way to ensure that files can be stored online after playing. We created many different JavaScript files and embedded them afterwards in the main web page through adding `<script>` tags with the path to the JavaScript file [35]. We paid attention to a detailed documentation and therefore we added comments to nearly every part of the code.

CSS3

JavaScript defines the behaviour of web pages, HTML the content and finally CSS defines the layout and style [36]. CSS is short for Cascading Style Sheets and is a style sheet language that is recommended from the W3C to store formatting information in a separated file [37]. CSS3 is the latest standard and all modern browsers support the modules and implementations of it. Advantages of CSS are the strict separation of layout and design, as well as a more comfortable

implantation because of the definition of central design properties [35]. We used CSS3 mainly for transitions and animations in the framework, which are afterwards necessary for the game development.

XML

XML (Extensible Markup Language) is a Meta language and used for the definition of other languages. XML files are useful because of their clearly arranged structure and they are very flexible in their creation [35]. This language was very important for us and will be also a core element for changes concerning psychological aspects. We use XML files for different gaming elements, paths to images and also level and scene definitions. Those elements often change during the development process, because the psychologists tested some scenes or made changes in the level structure. For instance, one XML file is named `Scenes.xml` and defines all used scenarios of the whole game, including structured information of every level and sublevel. Every scene has an id, the declaration of the right emotion in this scene, a URL to the picture and a path to joker, which is the audio file that describes the emotion in this scene. The following code example shows a clipping out of the `Scenes.xml` file, defining the scenes of level 1.1:

```
<sublevel id="1">
  <scene id="2" right_emotion_type="scared"
        url="images/scenes/level1/scared/L1_2_AngstuntermBett_Angst.png"
        joker="audio/joker/level1/L1_2_AngstuntermBett"/>
  <scene id="19" right_emotion_type="happy"
        ...
        ...
</sublevel>
```

3.7.2 Game Engine Jumru 5s

For the purpose of programming the endless runner game *EmoJump*, we implemented our own web game engine and named it *Jumru 5s*. The name *Jumru 5s* comes from the words “**J**ump and **R**un Framework for HTML5 Serious Games”. The engine is built upon the web technologies HTML5, JavaScript and CSS3 and we paid attention to cross browser compatibility. We focused on the browser Firefox from Mozilla, but also Google Chrome and Safari are supported. We developed our game engine for games with a special purpose and therefore we implemented logging features for further analysis.

Game Engine

A game engine should simplify game coding as good as possible and could be seen as the heart of a game. A game depends on its engine, because it manages the life of a game, encapsulates most of the game data and handles the whole lifecycle of the game [38]. Nevertheless, research shows that a game engine consists of some main components that are also valid for most of the engines and for our engine. Such components of a system interact with each other and architecture generally is divided into manager and subsystem, where every part of the system is responsible for a specific task. Alan Thorn [38] published an overview of possible components that constitute a game engine, which are the following ones:

1. *Asset Management*

A tool for loading game assets like images, audio or video files, which also supports different types of those files.

2. *Scene or Stage Management*

This management is necessary to compose loaded game data in a hierarchical structure and therefore in nodes and layers.

3. *Physics Engine*

This component applies forces like gravity to the game world. For instance if a player jumps, the physical logic should bring the object automatically back to the ground.

4. *Rendering Manager*

The rendering manager handles drawing and updating assets on the screen. The asset manager has to finish the loading process, before elements can be drawn.

5. *Media Management*

An engine should support multiple types of media files and also a management that assures the correct playing of audio and video streams.

6. *User Input*

Another important component is user input, because an engine should handle different input actions like mouse or keyboard.

In the case of developing a game engine for serious games, we added some features, to enable the possibility of further analysis and can be seen as a component, which is not listed as a main feature but also very important for the type of these games:

7. *Logging*

Collecting data about the user, his or her intentions and gameplay can be found at this component. Therefore, we can analyse the game experience and gameplay behaviour, which will be stored on the server for further psychological investigations.

Structure

Based on the knowledge of these elements, we integrated most of these important components and also added the logging feature. Our game engine is further an object-oriented engine, which results in a tree based structure of nodes. The `gamestage` is always on top of the hierarchy and can be seen as root node. The `gamestage` encapsulates all other nodes, which are necessary for the development. A stage contains several scenes, where only one scene can be active at a time. The active scene will be drawn, updated and redrawn in every game loop step. A scene again consists of various layers, which contain the leaf nodes called object entities. These object entities are not able to contain further

child nodes. The first three levels, stage, scene and layer are <div>-containers and object entities are canvas objects for drawing operations.

The following figure 19 shows an overview of the hierarchical structure with stage, scenes, layers and object entities (OE).

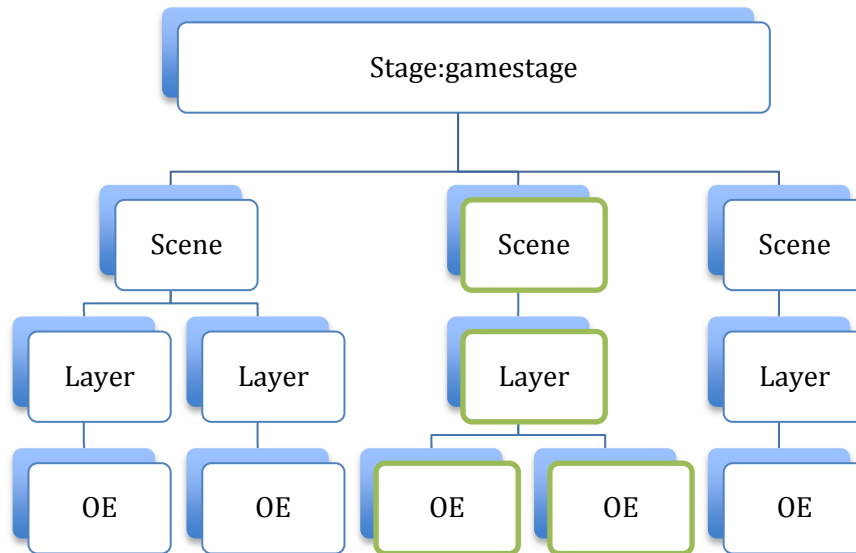


Figure 19 - Hierarchical Structure (with one active scene)

The basic node in the framework is the entity, which provides some main basic functionality. Every other object node in the engine derives from the entity node. The basic functionalities are positioning and sizing, as well as traversing mechanisms, event handling and collision detection. Object entities or also whole layers can have a motion by applying a motion object, where those elements will be updated and redrawn every game loop step while their motion is active. Object entities can be marked as dirty, if their properties like position, colour or size are changing. By marking them dirty, they will be redrawn in the next game loop cycle. This behaviour is also valid for all children of a layer, so if a layer will be marked as dirty, all child nodes will be marked, updated and redrawn. The heart of the game engine framework is called `jumruCore`. This is an object, which encapsulates important data of the game and also handles the basic game loop with the drawing, update and redrawing processes. Important

data are for instance the `gamestage`, `height` and `width`, `motions` and further properties.

Figure 20 shows an overview of the most important elements of the framework and their relations to each other. We already mentioned earlier the deriving of the element `Entity` and the possibility of adding animations. Further basic functionality, which is enabled by the engine, is asset management.

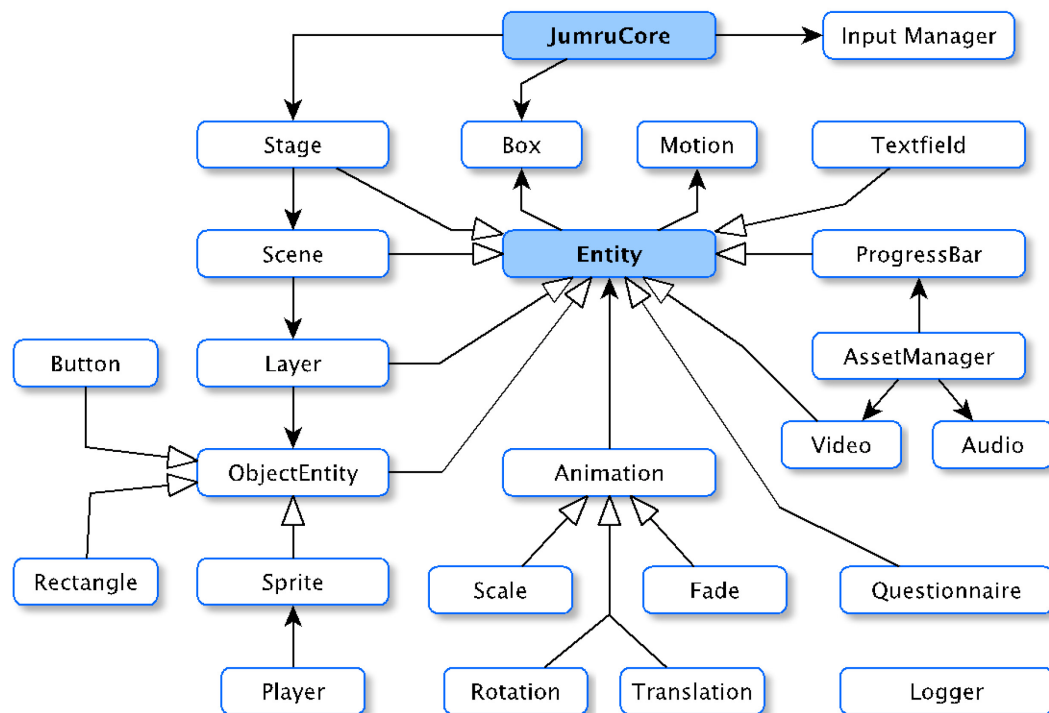


Figure 20 - Jumru 5s Overview

The engine supports various types of images, audio and video files and provides also a preloading bar for the visualisation of the loading process. Beside the `AssetManager`, *Jumru 5s* incorporates an input manager and supports keyboard or mouse input. We further implemented a rudimentary player class, which provides some basic properties and functionality. The standard player consists of different sprites that are appended to a layer. At a time only one sprite will be set to active and will be continuously updated and redrawn while the game loop is running. The standard player class provides a jump function, which includes changeable physics like gravity or the velocity of the jump. Concerning CSS3 animations, the engine offers the possibility of adding effects like scaling, fading,

rotating, translating or also colour changing to objects. All of these animations can be put together to a CSS3 keyframe animation.

The mentioned functionality can be used to build basic games, but especially for serious games we implemented the additional component logging with the invention to provide mechanisms to store data about the playing progress. For this purpose we developed two kinds of logging tools that collect information either directly or indirectly.

A questionnaire is a HTML5 form element, which collects user input and stores the form's content as a text file on the server. The text file also captures the duration, which the user needs for filling in the form elements. This can be seen as direct user feedback, where the questions and options of answers will be set in a XML file before. Figure 22 visualizes an example of how a questionnaire could look after playing a game and includes questions like "What's your name?", "How old are you?" or "Would you like to play again?".

Fragebogen	
Wie heisst du	Namenseingabe
Wie alt bist du	
Wie hats dir gefallen	0 <input type="range"/> 10
Willst du nochmal spielen	☐ ja ☐ nein
Welche Punkte haben dir nicht gefallen	☐ Handling ☐ Grafik ☐ Feedback ☐ Ablauf
Anmerkungen	Anmerkungen
Abschicken	

Figure 21 – Example Questionnaire

Beside the questionnaire, a logger is able to collect data indirectly in the background. This includes information, which is specific for the game and might be defined by domain experts. Based on the information that should be collected, the game developer can implement the logging structure.

This feature is also a big part in our developed game *EmoJump* and will be explained in detail in further sections as well as how we used the logging mechanism for the evaluation.

3.7.3 Game Architecture

The following figure (see figure 22) gives an overview on the composition of the game and their included important objects for the game life cycle, which are described in the following sections.

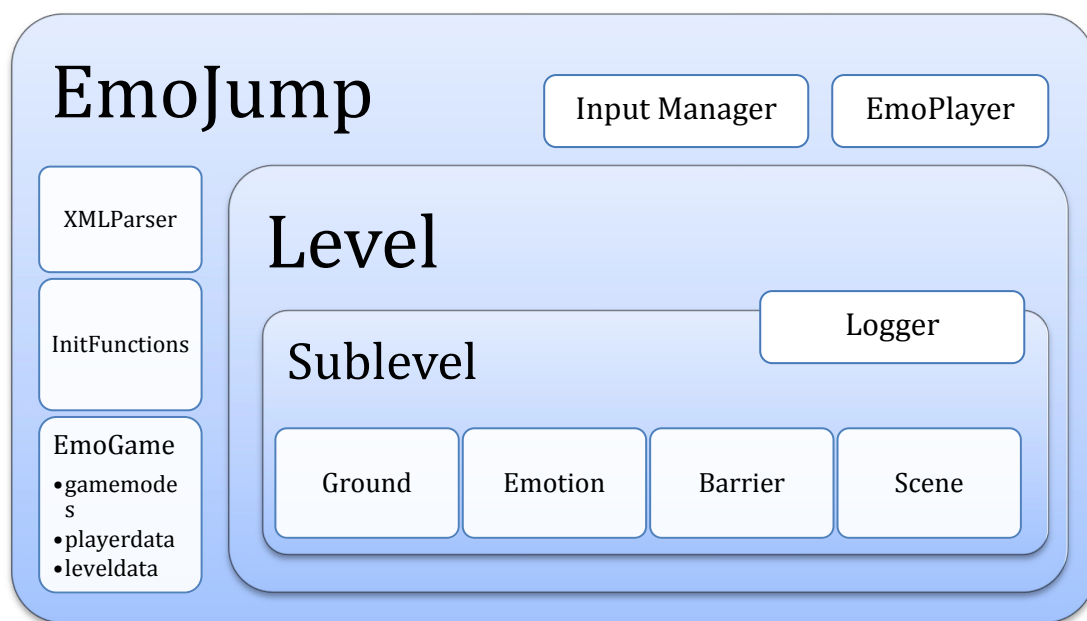


Figure 22 - *EmoJump* Composition

At the beginning of the game an XML Parser reads in data and also initializes important objects and data concerning the levels. This data is capsulated in the *EmoGame* object and these include game modes, player specific data, scenes and counter. The instance of the player is derived from the game engine player but extended with functions especially for *EmoJump*. As we have already mentioned, every level has 12 sublevels, where each sublevel has its own special elements concerning various scenes, correct and wrong emotion of this scene and grounds, normal or also with barriers.

The `InputManager` and even the menu enable the navigation by clicking or choosing elements with the arrow keys and the enter button or rather the space bar. Concerning the game mode, the arrow keys control different elements and objects. After playing a sublevel, the logger stores data on the web server.

3.8 Implementation

As before mentioned, we implemented the game parallel with the underlying game engine named *Jumru 5s*. The following declarations and specifications are closely connected to the elements, which are provided of the game engine. Figure 23 gives an overview on the game life cycle and initialisation processes.

Webpage

The HTML page `index.html` includes important elements concerning the appearance of the game like the favicon, holds all necessary JavaScript file paths and determines style tags. Further it loads the *EmoJump* menu, which will start the game life cycle. The favicon will be shown as a small logo in the browser, to the left of the URL. The page code is very short, because the rest of the game is implemented in JavaScript.

In the following code we can see the declaration of the page title and the onload event attribute, which is called after the page has been completely loaded:

```
...
<title>E M O J U M P</title>
</head>
<body onload="emojump.menu();" >
  <div style="position: absolute;">
  ...
```

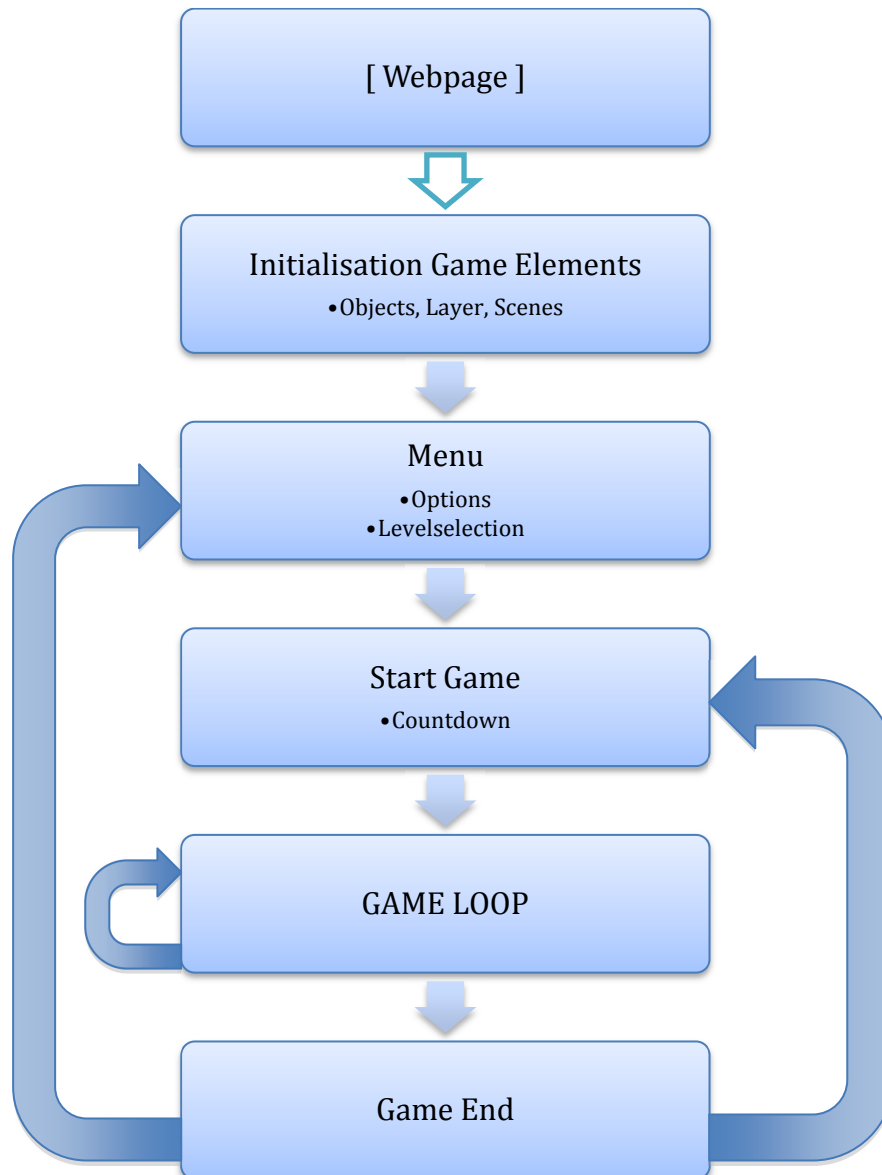


Figure 23 - Game Life Cycle

Initialization

First of all the implementation checks the browser in which the `index.html` is opened, because the browser prefix is important for determining the correct audio and video format. The prefix for Safari and Google Chrome is `-webkit-` and for Mozilla Firefox it is `-moz-` [39]. We offer two different formats of the audio and video files, to ensure a correct playback in the referred browsers: OGG/MP3 and OGV/MP4. Further a new `LanguageHandler` will be initialized to be able to set and change the language of the elements. After that, some core

elements like height, width and the `gamestage` were set, with a fixed height of 768 pixels and a fixed width of 1024 pixels.

During the initialization process the various scenes were defined. A scene at this step in the development, should not to be mistaken with the identically named scenes that show the comics. Those scenes inside the code are different from the scenarios. In this case a scene is the game screen and so indirectly the active game mode, which is displayed to the player. There can always be only one scene active at a time. In this initialization process every image, every button and text element of the menu scene will be created. As a button is created and this button has also a text in it, he will also be added to the language handler for switching between German and English.

Lastly the function `initGameElements()` creates animations, other scenes, initialisation of the logger and all elements which are necessary for the gameplay itself.

Asset Manager

Also during the initialization process the asset manager will be created and preloading of the resources starts. For preloading, the base URL to the assets is necessary and the path to a picture, which will be shown while loading assets. An `XMLParser` will be built and loads the different files including game relevant information. All relevant images, videos and audios will be added to the asset manager and loaded through the game engine. In the meanwhile the `AssetManager` also holds a progress bar, which is a new HTML5 element and visualizes the loading status, to provide some feedback for the user playing the game.

Menu

As we have already mentioned, the menu scene consists of multiple screens (see figure 5-9), where the player chooses the character, the name and the level/sublevel he or she wants to play.

The events behind buttons change the visible layers, which hold the depending elements of this layer. So if the visible screen is the start screen with the play button, the screen switches between the layers of the start screen to the gender layer. Every button has its click sound, which is an event that will be fired by clicking it. After clicking the Play button the layer swaps to the gender selection that is also defined as a different game mode:

```
/**
 * Play Button Action
 */
btnPlayCallback = function() {
    emoGame.audioButtonClick.play();
    emoGame.menuScene.removeChild(titleLayer);
    // set focus on last clicked button
    menuGender.currButton().focus();
    // set new gamemode
    jumruCore.setGamemode(emojump.GAME_MENU_GENDER);
    emoGame.menuScene.appendChild(genderLayer);
};
```

In the gender selection the buttons are defined as image buttons, where a click on the chosen image leads the player to the next game mode named `emojump.GAME_MENU_NAME`. This screen offers the demand to input the player name and shows the chosen character image. The input field is a normal text field with a defined pattern, so that the name consists of a maximum of 15 characters without special signs. Failing this, the text field will be marked red and a short error message is displayed, so the player cannot switch to the next screen until he enters a valid player name. The next layer opens up the level selection and after the player chooses a level the game mode changes to `emojump.GAME_MENU_INSTRUCTION`. In this section we show an instruction video depending on the level the player selected. This screen includes the video and a skip button, which is only clickable if the player watched the video once.

The code checks internally the chosen level and displays the correct video. After watching it, the screen switches automatically to the sublevel selection and an internal Boolean will be marked as true for watching the displayed video. So not only the layer changes, but also the next game mode will be applied again. The game mode depends also on the main level selection because every sublevel selection has its own game modes and screenings. In the following example we can see the switch code after an instruction video ends:

```
switch (emoGame.levelId) {  
    case 1:  
        menuLevel1.currButton().focus();  
        jumruCore.setGamemode(emojump.GAME_MENU_LEVEL1);  
        emoGame.menuScene.appendChild(emoGame.btnLayerLevel1);  
        emojump.playedInstruction1 = true;  
        break;  
    case 2:  
        menuLevel2.currButton().focus();  
        jumruCore.setGamemode(emojump.GAME_MENU_LEVEL2);  
        emoGame.menuScene.appendChild(emoGame.btnLayerLevel2);  
        emojump.playedInstruction2 = true;  
        break;  
    case 3:  
        menuLevel3.currButton().focus();  
        jumruCore.setGamemode(emojump.GAME_MENU_LEVEL3);  
        emoGame.menuScene.appendChild(emoGame.btnLayerLevel3);  
        emojump.playedInstruction3 = true;  
        break;  
}
```

The last step before the game play starts is the sublevel selection. As we already mentioned, before and after choosing a sublevel, the id of this level is placed internally, the function `emojump.startGame()` loads the necessary assets and continues with the game play as well as starting the game loop.

Start Game

In this case, the active scene is the preloading screen, where in the background the code loads and creates the gameplay concerning the chosen level. Some constants are set back to default values, in the case that it is not the first time the player plays a level. These constants are numbers or Booleans concerning the amount of played scenes, the taken joker and resetting the logger. An important step before the game can start is now the XML Parsing, which is already explained in the section technology & tools, XML. The correct background is drawn and also the parallax images are defined. The two ground images are set and also their direction, because in level 2 the player runs from the right to the left. Player specific data will be set and after everything is loaded and initialised and the game loop starts, but only once, for drawing the elements. Immediately after starting the game loop in the engine and drawing the objects, the game is set to pause mode, because first of all the countdown is displayed to the player. The explanation of the countdown and the process of the game loop will be explained in the following sections, after we have a detailed look on the implementation of the player, the parallax scrolling as well as joker and pause mode.

Player

Our main character will be defined in the menu selection, where the player chooses the gender. After starting the game, the gender is also set in the player constants. Before the game loop starts, the position and direction of the player are configured and also the speed of a jump. A player image is either one image or consists of multiple sprites. It is necessary to store different types of the player concerning running, jumping and looking for all two directions. Figure 24 shows an example of a sprite image, for the behaviour that the player is jumping. During the normal gameplay, the space bar, which has the key code 32, provokes the jump, while simultaneously the audio for running on the ground pauses and the audio for the jump begins. The jump of the player is handled in the game engine, but when the player hits the floor again the normal game play goes on.

The player can only activate the jump, when the character is hitting the floor, which will be verified before the action jump can be provoked:

```
// Space Bar Player JUMP
if(evt.keyCode == 32) {
    if(emoGame.player.isOnGround()) {
        emoGame.audioGround.pause();
        emoGame.audioJump.play();
        emoGame.player.startJump();
    }
}
```



Figure 24 - Player Sprite Jumping

Parallax Scrolling

This method is useful to make a very simple 3D effect in 2D games because it creates the illusion of shallow depth. The background of a game is divided into multiple images, where each of these images has a different speed level. Slower moving images are suggestive of objects, which are far away of objects that are move faster in the foreground [40]. In our code we have two layers with a different depth and speed. The ground layer which is the actual game layer, where also barriers and emotions are placed to move towards the player and the background parallax layer, where small objects are moving from the right to the left in a smaller speed as the foreground. In the following figures 25-28, we see how such a composition (see in figure 28) is configured. The examples show the fixed background (see figure 27) where the canoeist (see figure 25) is defined as a parallax in the background with a higher speed than the parallax moving in the front (see figure 26).



Figure 25 - Parallax Background



Figure 26 - Parallax Foreground

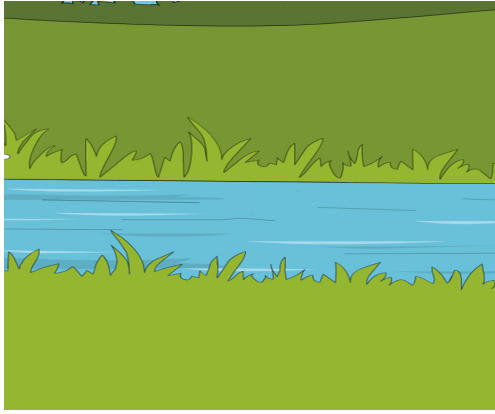


Figure 27 - Background



Figure 28 - Game Composition

Joker

The joker button is only active while the player stands on the ground and looks up to the bubble scene in level 1. Level 2 and 3 are a bit different of this scheme, because here the joker will be activated after the last scene is displayed. In level 2 it is the reality picture that releases the joker and in level 3, the last two scene pictures have to be displayed that the joker can be activated.

Pause

The pause button is deactivated at the beginning of a new level and also while the player is standing on the ground, looking up to a scene. Not before the player hits the space bar and the game loop starts running, the pause button is activated. Deactivated means that the button is not clickable and the opacity is set very low to 0.2. Pause interrupts the game and the pause scene will be displayed, where the player can interrupt the actual sublevel and can go back to sublevel selection.

Countdown

The game mode `emojump.GAME_PLAY_COUNTDOWN` is always active before a sublevel regularly starts. It is an interval method, which displays four different images until the first scene will be shown to the player:

```
// Play Countdown
emoGame.activeCountdown = setInterval(function() {
    emojump.doCountdown(emoGame.oentityCountdown);
}, 1000);
```

The `setInterval` method can be used to call a function continuously after some time has passed [40]. In our case we defined 1000 milliseconds, but it is also possible to set different values. Every second, the function `doCountdown` is called a total of four times. An internal countdown counter checks which image has to be shown and is set back to the default value after the countdown is done. Figure 29 shows the various countdown pictures, which are displayed in combination with a sound effect.



Figure 29 - Countdown Pictures

Gameloop

The gameloop is started in the pause mode after the countdown has finished. The game is now in the mode `emojump.GAME_NEW_SCENE` because after the countdown has finished, the first scene will be displayed on the screen and the level background music starts to play. Showing a new scene is an implementation that not only filters the next scene but also filters the next scene randomized out of a scene pool. This method also checks, whether it is the last scene or not, because otherwise it will not show a new scene any longer, but visualizes the final score screen.

The random scene will be handled in the method `chooseRandomScene` with the parameter of the sublevel:

```
emojump.chooseRandomScene(emoGame.levelId);
```

This method also sets the game to the pause mode `emojump.GAME_PAUSE_SCENE`. A new scene means also actions like refreshing the lives of the player and changing the sprite to the `LOOK` image, where the character looks above him to the shown scene. The randomize process is that the code defines a random number and the scene on the place of this value in the array will be taken and at once deleted, so that it is not be displayed again. Finally, the actual game mode is important for the input handler to know which reaction he has now at the

space bar, because by tapping this key, the player will start running and the game is not paused anymore. The randomize process concerning level 2 and 3 are a little bit different and will be explained later.

As already mentioned, the `emoGameLoop` only runs once at the beginning for displaying and drawing the new scene, but after hitting the space bar the game loop and also the player and the ground starts.

In the implementation we have a main game loop in the game engine and an adapted game loop for the game *EmoJump*. The `emoGameLoop` assimilates various steps in each loop pass. In each pass the code checks behaviour of the player. Checks concerning if he is jumping or if there is a collision between the player and emotions or barriers and if the mode demands for showing a new scene. A new scene is only displayed when the player is not in the jump mode and therefore on the ground:

```
// Player has to be on the ground for scene change
if(!emoGame.player.isOnGround()){
    emoGame.player.jump(emojump.stopJump);
} else {
    if(emoGame.showNewScene){
        emoGame.btnPause.setOpacity(0.2);
        jumruCore.setGamemode(emojump.GAME_NEW_SCENE);
        emojump.showScene();
    }
}
```

The logic behind the collision detections is that the primary bases are two grounds, on which optically the player is running. On these grounds, emotions and barriers are combined with the image of a ground floor into one layer. There are two ground layers defined that are directed towards the player. In figure 30 we can see the “groundStruct1” and the “groundStruct2”, where both move towards the player and after one structure has finished the moving through the screen, it will be again appended on the right side of the screen for another passageway. Every ground structure has a placeholder for one barrier and for one emotion, which either can be filled or not. The red encircled elements in figure 30 are barriers, however they are not single images but part of one ground image. A barrier is an object, which has an x and y position as well as a defined height and width. If there is a barrier in a ground image like the

wooden stand and the bush in figure 30, the code sets an invisible collidable object at this place called barrier and a Boolean concerning this object if it is active or not.

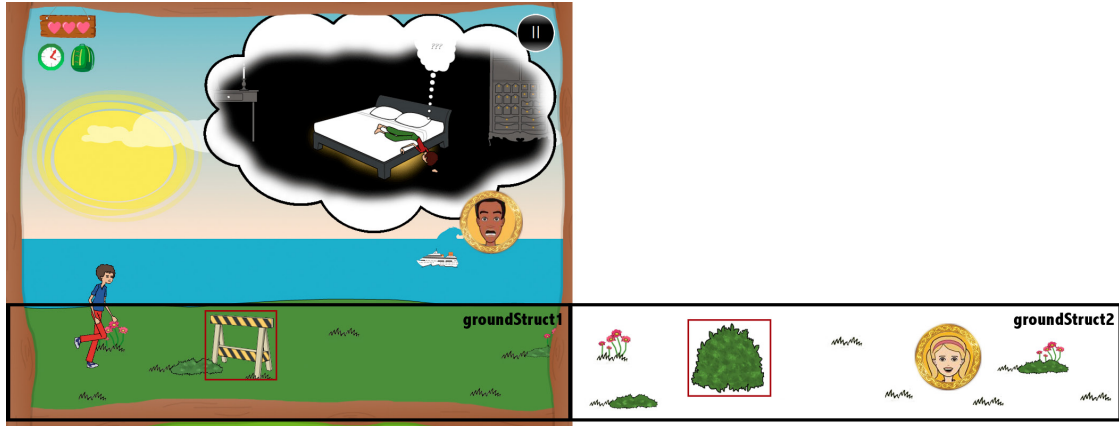


Figure 30 - EmoJump Ground Structures

The fact that a ground is a ground with a barrier placed on it, is implemented in the XML file and while the parser reads the information about the grounds, a barrier is set when the type of a ground is “crash”:

```
//ground contains barrier
if(randomGround.type == "crash") {
    groundStruct.barrier.setSize(randomGround.barrier.width,
    randomGround.barrier.height).setPosition(randomGround.barrier.x,
    randomGround.barrier.y);
    groundStruct.barrier.setActive(true);
} else {
    groundStruct.barrier.setActive(false);
}
```

In every game loop step, the code checks if the player collides with a barrier object and this depends also on the type of a ground. Grounds with the type “normal” (figure 26 shows such a ground) are not necessary to check, otherwise the object entity player and the object entity barrier are checked on a collision. Only if the barrier object is near the player and can therefore obviously collide, the check function for the collision is executed. The control if the object is near by another object should support performance. The collision detection itself handles the game engine. Such collision detection are not only used for crashing, but also if the player collides with a coin and therefore collected an emotion,

represented as coins. It is the same mechanism as the collision detection with a barrier that we recognize the ground only if the type is not normal and the coin is near by the player. An action is set when the player collects the emotion or the emotion is out of the view of the game scene (see section Emotion Handling). In summary, the most important steps of each game loop are handling a new scene and the various collision detection mechanisms.

Emotion Handling

The aim of the game is to collect the correct emotions for a presented scenario. As we have stated before, the game loop checks if a player has collected such an emotion coin. By collecting a coin, the player sees the animated backpack, hears a sound effect and in the background the counter of collected emotions increases. The opacity and activity will be set false and the canvas of the emotion coin will be cleared, so that is not visible anymore. For the game logic it is also necessary to check whether it was the last collected correct emotion. In this case a new scene will be appear and the actual scene is closed. If an emotion is not collected we implemented a control mechanism if an emotion is out of window and therefore was not collected. The counter of the sent emotions increases and also the same logic as before concerning a correct emotion will be checked. If the emotion was the last correct one, the game play of the actual scene is finished and the mode changes again to `emojump.GAME_NEW_SCENE`.

Crash & Game Over

The collision with a barrier fires up a crash. The audio of running on the ground will be paused and the crash audio is played. The lives of the player decrease and also the sprite with the hearts has to be updated. If the value of the lives is smaller than 0 the player is dead and the current scenario is over, which is shown by a game over screen. In this case the lives value will be set back to 3, which is the maximum amount on lives for further playing. There is also the Boolean `dead` that is set to true, thereby the system knows the player lost this scenario. A crash scene shows the player falling from the top of the screen to the ground and the title "BANG" and in case there are no lives left, the space bar leads the player now to the Game Over screen (see in figure 31, right picture).

The loop is paused and after the game over screen where the game mode is `emojump.GAME_PAUSE_CRASH` and the Boolean `dead = true`, there is a different code working. Game Over has its own sound effect and the game mode is set to `emojump.GAME_OVER`.



Figure 31 - Crash Scene and Game Over Scene

Continuing, the game over screen sets the `dead` value back to `false` and the mode back to the normal game play. Before heading back to the game screen, the game is in the pause mode, so the player can decide whether he wants to try again or goes back to the menu.

Sound & Music

The game separates between sound effects and music like background music in the menu pages and during gameplay. A new audio element will automatically be stored in an array in the game engine code, which holds all initialized audio files. By default, every file has its name and the sound is set to `on`. As already mentioned, in the main menu you can separately switch off music or sound effects, where we implemented the logic that there are also two constants they representing those audio characteristics: `emojump.MUSIC_ON` and `emojump.SOUND_ON`. Every time the options in the menu change concerning audio or video, the code checks the sound property of each file and adapts it if necessary. The audio files for the joker are always excluded because they are required.

Creating an audio file is done by the following code:

```
emoGame.audioCrash = new jumru.Audio('audioCrash');
emoGame.audioCrash.setAudiofile(
    emoGame.assetManager.getAudio('audio/audio_crash.' +
    emojump.AUDIO_FORMAT)
);
```

Normally it is only necessary to add the path by setting an audio file, but in our case, we handle different audio formats to play audio in various browsers.

Scene

A scene is the main element of the game play. The XML Parser, as already mentioned, reads in the individual images of the scenarios and stores information about each scene in an object. The scene object has elements like the scene id, the correct emotion type, information concerning the joker (path to joker, joker taken value, joker counter), width and height, an array named emotions counter to document the collected emotions and the playable emotions of this scene. The player always has to think about the possible emotion of the child with the red t-shirt in the scenario. More complex is the handling of the scenes concerning level 2 and level 3.

Level 2

The player has also to collect the correct emotion of a scenario but this scenario consists of two or three images and is represented as a picture story. The object, which is created, is called `Scene2` and has the same elements as the scene object of level 1 and further elements. `Scene2` inherits from `Scene3` that again inherits from `Scene`. The parsing of level 2, where either two or three images are displayed, consists of reading in one image separated in two (see in figure 33) or three parts (see in figure 32).

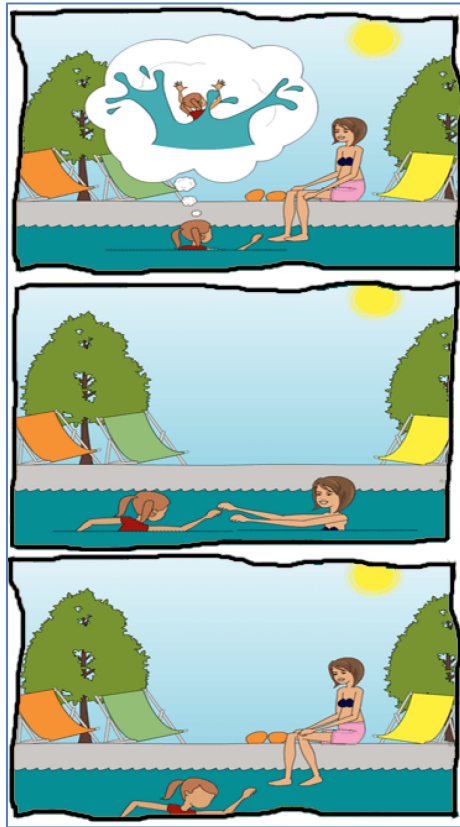


Figure 32 - Level 2 Scenario (3 Images)

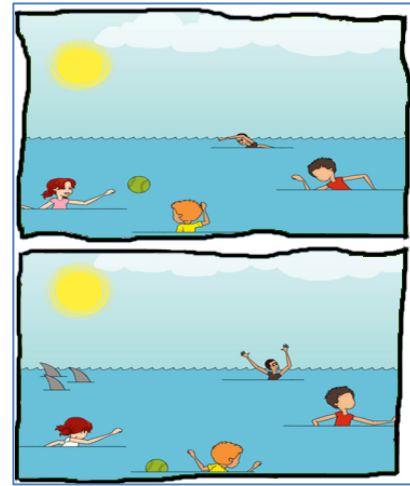


Figure 33 - Level 2 Scenario (2 Images)

The XML file includes the information how many subimages an image has. The following code snippet shows the parsing process concerning the subimages:

```
//define subscenes - situations
var subImages =
parseInt(scenes[k].attributes.getNamedItem("subimages").nodeValue);
scene.subimages = subImages;
```

Every scene object includes an array of subscenes, which represent the before explained subimages. Scenes in level 2 are always composed of one reality and one belief image. For each of those images, the code creates a subscene. The middle image, in the case of three subimages, is only a picture but not a discrete object. So the ids of the other two images are not a number but set to a defined type, called "REALITY" or "BELIEF". This is also important for the action: `scene change`. If the player has collected enough correct emotions or the last correct emotion has past the player, the code orders a scene switch from reality to belief. The player has now to finish the other subimage and after that the entire scenario changes to the next picture story.

Level 3

In level 3 a new object named `Scene3` will be initialised which inherits from the scene object. The fact that level 3 also includes a mini game expands the information per scenario at storing an item object. This is only a parameter in the scene object with the path to the image of this item. There is another image that shows two different emotions concerning one scenario that a person can have in one situation. For a better understanding see the example presented below. First of all, the player has to collect the item horse, which represents the scenario (see in figure 34). After that, he will be shown figure 36 that explain the scenario, where the girl in the red shirt is invited to go horseback riding. After that, the player sees figure 35 to explain that the girl can have two different emotions during the invitation, either she is happy or she is scared, because she can fall of the horse. The picture in figure 36 is named situation scene and the two bubbles in figure 35 are named in the code as left and right scene. This causes of early developments and a different design in a previous version, where those two bubbles were displayed parallel and not among themselves. Analogues to the parsing process of level 2, scenes have an array, with the right and left scene representing the different emotions. For now, the player starts by hitting the space bar and figure 36 is displayed again.



Figure 34 - Level 3 Item



Figure 35 - Level 3 Two Emotions



Figure 36 - Level 3 Situation Scene

Scoring

After each scene, the player is shown a summary board. The actual game mode is `emojump.GAME_NONE`, which is rather a temporary game mode, to avoid player input immediately. The board should present the collected emotions for a short time until the player can continue. To visualize the period where the player cannot skip the score screen, a chalk image, which is a smiley is fading in (see in figure 37 and 11). The game is also set to pause mode and the actual played scene will be logged. The scene board is different for each level, because various combinations of the elements have to be displayed. Figure 37 shows the hidden elements that are defined in the code and used as placeholders.

In level 1, only one bubble will be shown with the number of correct and wrong collected emotions. In level 2, also the light green bubble is filled with the values. In level 3, where two scenes are displayed, only the correct values concerning belief and reality were shown and the last two horizontally yellow boxes at the bottom are filled, to display and summarize all collected wrong emotions.

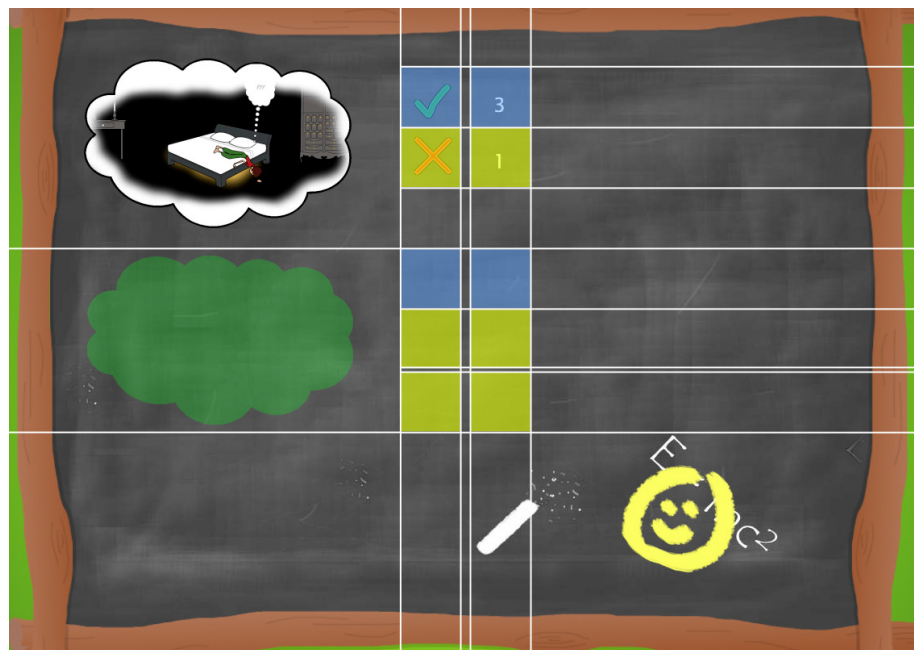


Figure 37 - Scoring Board Overview

The last check in the function `showSummary` validates if there are still scenes available to play and if not, the Boolean `isLastSummary` is set to true. After

playing the last scene, the background music is stopped and the sublevel and winning values are calculated. The calculation of the win score was already explained in the section “Score Mechanism”, where the player is displayed if he has won or lost the sublevel. If he succeeds the level, the trophies represent how good he has played and confetti is falling over the screen.

The confetti effect in HTML5 is a nice gimmick for representing success after a sublevel. The code of this effect bases on implementation recommendation of users on the homepage stackoverflow.com [41]. The confetti is an object entity and we added the Boolean `isShowing`, for controlling this effect. For playing and drawing, the confetti has its own loop, while the `isShowing` Boolean is `true`, the confetti is updated and drawn continuously.

Logging

The logging mechanism as described above can be seen as a core element for psychologist’s analyses. During the initialisation of the game elements also the logger is created:

```
emoGame.logger = new  
    emojump.Logger(emojump.baseUrl+"/Framework/logger.php");
```

This initialisation creates a game engine logger with the URL to the PHP file, which parses the data to JSON and sends it to the server. An important value is the `logIndex` that is necessary to indicate the number of the row in the final CSV file. The logger logs either a scene or a sublevel relevant information, which is the last step before the file will be stored. At two times in the game code the following action will be called:

```
emoGame.logger.log();
```

This action will be called if the player is dead or after every scene. If a player lost all his lives and therefore lost a scene, the counter of this scene will be backed up and the actual scene will be logged.

In the final logging report, this lost scene will appear twice but only if the player decides to try this failed scene again, otherwise if the player returns back to the menu, there will be no file stored on the server.

```
/**
 * log each played scene
 * @returns {emojump.Logger} this logger object
 */
emojump.Logger.prototype.log = function() {

    //according to level, log scene
    switch(emoGame.levelId) {

        case 1:
        case 3:
            this.frameworkLogger.addEntry(emoGame.scene.id);
            this.frameworkLogger.addEntry(emoGame.scene.jokerUsed);
            this.logScene(emoGame.scene);
            break;

        case 2:

            this.frameworkLogger.addEntry(emoGame.scene.parentScene.id);

            this.frameworkLogger.addEntry(emoGame.scene.parentScene.jokerUsed);
            this.logScene(emoGame.scene.parentScene.subscenes[SCENE_BELIEF]);
            this.logScene(emoGame.scene.parentScene.subscenes[SCENE_REALITY]);
            break;
    }

    //push log to framework logger
    this.flush();

    return this;
};
```

In the case of level 1 and 3 the log entry will be filled with the scene id and the number of taken jokers. After that, logging the scene will be called by a separated function: `logScene`. So now, all information concerning the game play of scenes were stored to the log array, which were already explained in the previous section. Those values were readout of the scene object, which holds values like the correct emotion type, the number of correct sent emotions or the number of correct collected emotions. In case of level 2 the `logScene` function will be called twice, once for the reality scene and again for the belief scene. Finally, the log action calls the flush action in the game engine Logger and adds the current log to the stack as well as increases the log index.

By now, this is only the logging in the game engine, but the file is not stored yet. But before the player has finished a sublevel, the code calls the function:

```
emoGame.logger.logSublevel();
```

This function handles everything of the last steps before the file will be stored on the server. The name of the file will be set and the log index will be set back to one, because now we define the headlines and the first rows of the CSV file. The first row includes the headers player name, date, time, browser and average frames per second. After defining the flush function will be called so that this array will be defined in the logger. The next array defines the appropriated values to the set under the stored headers. The following arrays include the headers of the value from the scene logs and summaries of those values.

Finishing the logging process the function `store()` in the engine logger will be called that handles the parsing of the array with the several rows and stores the data in the CSV file on the server. Starting a new game will back up the logger. Figure 18 showed us an example of a stored CSV file.

4

4 Evaluation

The aim of the evaluation is to question the design and usability as well as the comprehension of the game *EmoJump*. The whole evaluation is separated into playtesting combined with semi-structured interviews and observations of the psychology students and questionnaires. For evaluating the game we tested the game with 35 participants, who are children aged between 8 and 12 and played the game voluntary.

4.1 Evaluation Method

The participants were separated in two groups. For a better understanding we will name the groups A and B. Group A includes 10 participants who filled in the same questionnaire as group B, but each member of this group was observed from a psychology student during game play and during answering the questions. All of the participants answered some main questions before playing, which subject matters were not concerned the game *EmoJump* and we will call pre-test. After that, the children were shown the instruction video and they played the game for about 10 minutes. Members of the group A were observed and they also had the chance to ask the psychology students if anything was still unclear concerning the game. The psychology students documented their surveillances by filling in some predefined questions.

Finally, all participants answered the remaining parts of the questionnaire about design and usability of *EmoJump*, which we named post-test. Those tests include multiple choice answers as well as textual answers.

Environment & Participants

The test environment has been at the “A1 Internet für Alle Campus”¹¹ in Vienna. The participants were children aged from 8 to 12, where the average age was 11. We can see an overview of the various ages in figure 39. The gender ratio was evenly distributed, with 18 female children and 17 male children (see figure 38). All of those participants were healthy children and did not suffer on Autism Spectrum Disorders or had any other disabilities.

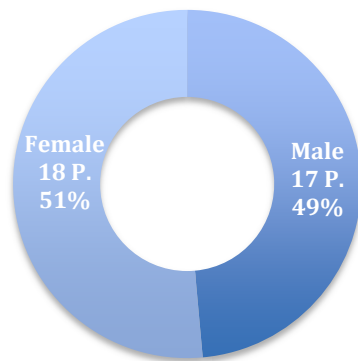


Figure 38 - Gender Ratio

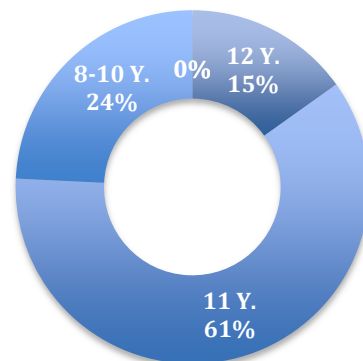


Figure 39 - Age Ratio

4.2 Pre-Test

The pre-test is a short questionnaire concerning the gaming behaviour of the participants, which they had to fill in before playing *EmoJump*. In summary, this test had five questions, which were either yes/no answers or textual answers. All answers had been transcript and translated for the evaluation.

Questions

The following questions were asked:

1. Do you enjoy playing video games?
 - a. Yes/No

¹¹ <http://a1internetfueralle.at>

2. What game are you most likely to play?
 - a. Textual answer
3. Do you have a computer at home?
 - a. Yes/No
4. Are you playing games on the computer (not game console)?
 - a. Yes/No
5. What games?
 - a. Textual answer

Results

Most of the participants enjoy playing video games, which we can see in figure 40. The diagram shows that 74% like playing games, where 46% are boys. The inner circle separates the answers yes or no in male or female answers. An interesting aspect of this question is that only one boy doesn't like playing video games, and the remaining 23% are female participants.

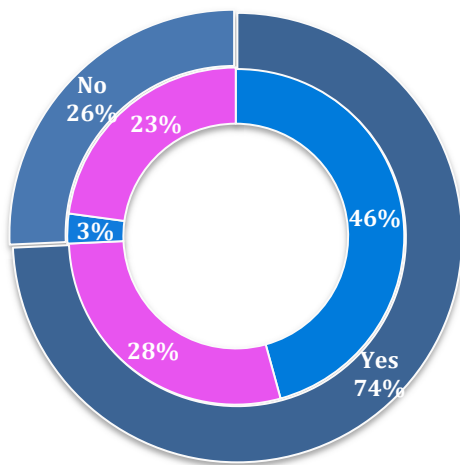


Figure 40 - Enjoyment Playing Video Games

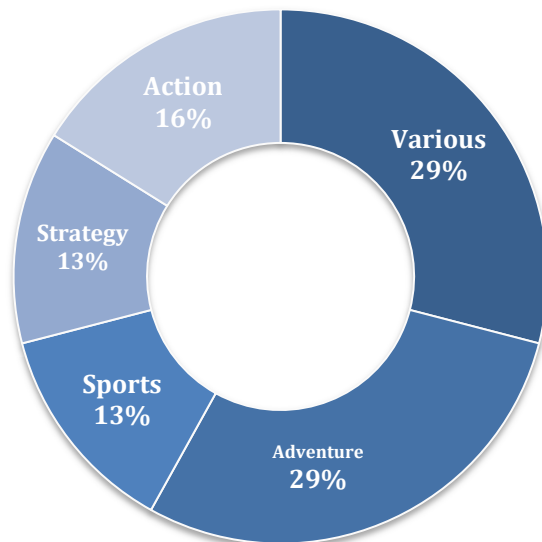


Figure 41 - Game Types

The second question concerns the different games and game types the children like to play. Nearly all of the participants answered the question except three children. The participants named their favourite game and we checked which game type (see in figure 41) those games have. Most of the children like to play

adventure games or various games on their mobile phone or on gaming platforms like Moviestarplanet¹². The game named the most was Minecraft, which is an adventure game since 2009. According to the developers, it is a game about breaking and placing blocks in a virtual world [42]. The remaining games were soccer or action games, like playing football, racing or ego-shooters. The few strategy games were puzzle games, but one interesting answer was that a child likes playing the online game “Top Eleven”¹³, where the player has to manage a soccer team by handling with money and thinking about best tactics for a good soccer season. The third question asked if the participants have a computer at home. This question is also an interesting point, if the game should be available online, so the training with serious games can be over the Internet within a login system. Only one participant has not a computer at home. The following figure 42 shows if the participants like to play games on the computer. Over 74% like playing video games, but there are also 26% left, which do not like playing on the computer, where most of them are female children. We finally asked before starting playing *EmoJump*, which game the participants like most playing on the computer and also here they named Minecraft several times. Other mentions were game on online platforms, some adventure and action games as well as an online chess and card games.

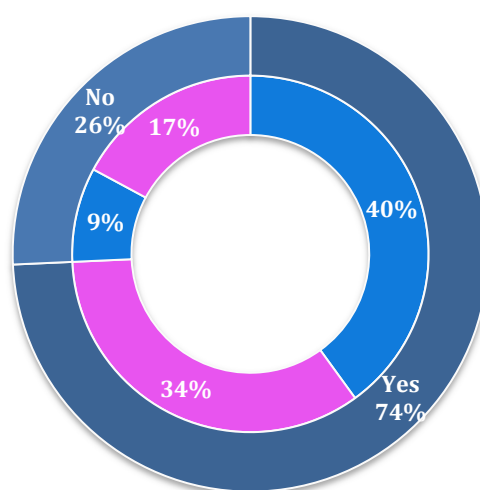


Figure 42 - Likes playing on the computer?

¹² <http://www.moviestarplanet.de/>

¹³ <http://www.topeleven.com/de/>

4.3 Observation

Group A consisted of 10 participants, which played and answered the questionnaire in presence of the psychology students. Further they documented the behaviour of the children before and during the gameplay. Unfortunately, the documentations are fragmentary and incomplete, so we can only give a short summarization of what the psychology students observed.

Most of the children were able to start the game on their own, but only one child tried to start with the arrow keys and had to be indicated to use the space bar. A few children tried the controls before the game has started, but no one had problem with the handling of the space bar during the game. Sometimes they jumped too early or too late over barriers, but on the whole they had no problems by controlling the player. Frustration could be determined after crashing several times or by reaching the game over screen. The design principles from Tomé et al. [43] indicate that concerning feedback the game should not create any kind of frustration. For further analysis, we would have to observe the children for a longer period of playtime, to find out if the game was really frustrating or annoying. The point of frustration is still an important aspect, since we primarily want to avoid that the game indicates frustration, which is especially important for children with ASD, because it should make fun and help them. This is an aspect, which has to be considered in further improvements of the game.

4.4 Post-Test

The children filled in the post-test after they played our game *EmoJump*. This test is separated in two parts, where one part asked questions about the design and logic of the game and the second part focused on the visuals, the understanding of the elements and game world.

Questionnaire Part 1

Part 1 includes 10 questions with optional answers to mark with a cross and textual answers.

The questions were translated and the answers transcribed for the evaluation. We want to know if the child has fun playing the game, how difficult it is and what the advantages and disadvantages of the game are.

- Question 1: Did you have fun playing the game?
 - Answer by marking a cross over the chosen emoticon (see in figure 43)



Figure 43 - Emoticons

- Question 2: What did you have to do in the game?
 - Textual answer
- Question 3: The game was ...
 - ...too easy for me.
 - ...just right for me.
 - ...too difficult for me.
- Question 4: What did you like most in the game and why?
 - Textual answer
- Question 5: What did you not like in the game and why?
 - Textual answer
- Question 6: Did something in or during the game make you angry?
 - Textual answer
- Question 7: Did you like to win the game?
 - Yes/No
- Question 8: Would you like to play the game at any time?
 - Textual answer
- Question 9: Is the game something for you/younger or elder children?
 - Textual answer
- Question 10: Would you show the game your friends/classmates or teachers? Why or why not?
 - Textual answer

Results Part 1

We want to evaluate the usability and design as well as the understanding of the game. Usability is not only the handling of the game, it depends also on the interface, the functionality, interaction, satisfaction, effectiveness and as well in serious games the learnability [44]. First of all, let us have a look at the topics satisfaction and effectiveness, where the questions from part 1 are suitable. Satisfaction is used to have a look if the player enjoys the game and in the questionnaire we used a Smileyometer [45] (see also in figure 43) to see which expression the children will have concerning playing *EmoJump*. If children enjoy playing a game, they are more motivated to play it again, so satisfaction can be more important than the performance of a game, to make a game recommendable [45]. All participants had answered question 1 and therefore had marked their decision in the Smileyometer in our questionnaire. The calculated average value results in 3.1 points. This result shows that the children rated the fun factor of the game with “middle”. The distribution of the values is very interesting because there are equal answers at “very much”, “middle” and “not a bit” (see in figure 44). The female participants gave in summarisation a higher value than the male participants (see figure 45).

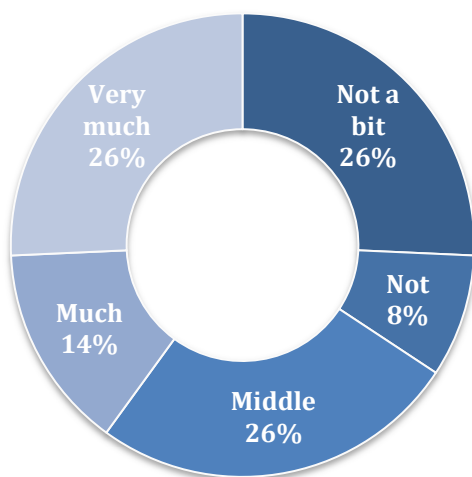


Figure 44 - Part1/Question 1

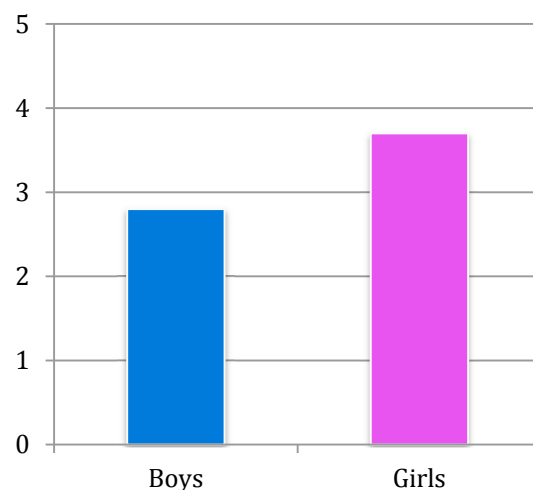


Figure 45 - Part 1/Question 1

The questions 4 to 8 can also be associated with the satisfaction the children have. Only question 7 is a yes or no answer, where the others were textual answers. Concerning question 4, the participants liked most on the game, the

music, the speed levels, jumping and even said that the game includes funny scenes. There were also negative answers, for instance that one child liked most as the game has been finished or some also wrote the word “nothing” as an answer. Question 5 and 6 are very similar, because they both deal with negative aspects of the game. Some of the children disliked that they crashed or were dead, and the situation to collect emotions unintentionally. There were also positive answers, because some of them wrote in spite of the question what they didn’t like that they enjoyed everything of the game. The last two questions concerning satisfaction should proof if the participants had the intention to win the game. More than 80% said yes and exactly half of the participants want to play the game again, though one participant was insecure in his answer.

Summarizing, the positive answers concerning in the topic satisfaction were quite good, the negative aspects are mentioned in further improvements.

The next topic is effectiveness and if the game can fulfil the personal requirements of the participants [44]. The questions 3 and 9-10 do not ask directly after requirements, but based on the answers we can reason out what the participants are thinking about the game and if he or she would recommend it. Most of the children think that the game *EmoJump* is even more a game for younger children. This could mean that the game is too easy for them, so they said that it is rather a game for younger ones. The participants who said that the game is suitable for them were also those children who were the younger section of the participants.

Recommendation

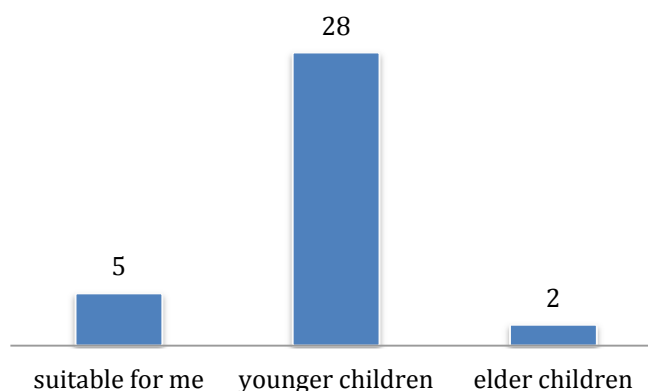


Figure 46 - Question 9 Recommendations

This assumption is confirmed with the answers from question 3. Most of the participants said that the game is too easy for them and only a few stated that it is too hard.

Difficulty

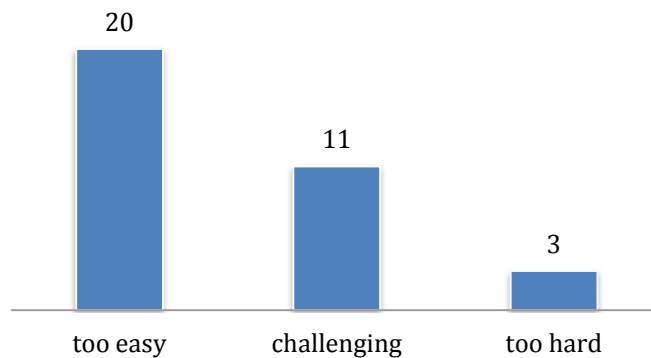


Figure 47 - Question 3 Difficulty

The question if the participants would recommend this game to their friends, classmates or teachers brings a result, we did not anticipate. Half of the participants would not recommend the game, which indicates that this game has no commercial character or is meant for a wide distribution. In fact for our purpose as a helping tool for children with ASD it is not so important for us that this result has a high value. The participants who would recommend the game, justified their decision by saying that is a funny game or makes fun. This game has the aim to help and train children's emotion recognition and focus especially on children with Autism Spectrum Disorders, so this game will may be used in training programs under the attendance of psychologists and not as a stand-alone online game.

So lastly, we want to have a look at the topic learnability and if the children understand what they had to do in this game, because this should be one of the most important things so that this serious game can be used in training studies.

Question 2 asks the participants after the assignment and purpose of this game. The most named answer was "Münzen sammeln", which translated means collection the coins. First of all, this basically shows us that the participants understood the game logic, but this answer also reveals that the children kept too much focus on collecting the golden coins. Maybe they were just concerned

on collecting the coins without paying attention to the correct emotion on it. Some of the children answered more in detail and confirmed that they had to collect the emotion, which is suitable to the shown scene. For a more reliable appreciation, the topic learnability has to be asked again in a detailed questionnaire.

Questionnaire Part 2

As already mentioned, part 2 includes questions concerning the game play and elements shown in the game. Based on these answers, we want to check if the participants have understood the logic behind *EmoJump*.

The questions and answers below were also translated.

- Question 1: Which character on the picture is the main character?
 - There are three scenes and the participants have to mark the boy or girl with the red t-shirt, which represents the main character.
- Question 2: Why did you know, which emotion is the correct one?
 - Textual answer
- Question 3: Are you seeing a barrier in this picture?
 - The picture shows a normal game play with a ground that includes a bush as barrier and the children have to mark it.
- Question 4: What keys do you need for the game play?
 - The picture shows a keyboard and the correct answer is to mark the space bar.
- Question 5: What do you do, if you like to make a pause during playing?
 - The picture is the same as in question 3 and the correct answer is to mark the pause button in the right corner of the game.
- Question 6: Where do you see how many lives are left?
 - Same picture as in question 3 and the correct answer is to mark the wooden board with the hearts in the left corner of the game
- Question 7: What do you do, if you do not understand the shown scene?
 - The participant sees a picture of game scenario where the player stands on the ground looking up to a scene. The correct answer is to mark the joker button.

- Question 8: Here you can see the scoring of another child. How good has the child played?
 - The picture shows a scoring screen from a played scene out of level 1. The child collected one correct and five wrong emotions. The correct textual answer is, that the child did not play well in this scene.
- Question 9: Where you can see, if you succeeded this level?
 - The participant sees the end-scoring screen with the green bar and the trophies, as well as the values of correct and wrong collected emotions. The correct answer is to mark the bar, which shows the child how well it has played the game.
- Question 9a: Did you succeed the level?
 - Yes/No
- Question 10: If you would choose a name for the game, which one would you take?
 - Optional textual answer. There is no correct or wrong answer that a child could give.

Results Part 2

In part two of the questionnaire we want to check the usability concerning the topics interface, interaction, functionality and features. Interface is an important aspect of each game, because an enjoyable interface motivates us to play a game. For the topic interface, the suitability of layout, icons, objects as well as navigation elements will be observed [44]. The questions 1-3, 6 and 8-9 refer to this topic. In question one we focus on the recognition of the main character presented in the scenario, for which the correct emotion had to be collected. Three images with different scenarios were shown to the participants and they had to mark the boy or girl in the red t-shirt (pictures can be seen in the Appendix). Most of the participants, exactly 20 (see in figure 48) identified the correct person, marked the comic avatar with the red shirt and the average value is 2 so they identified 2 from 3 characters in the scenarios. If we look behind the logic of the game, there is only the question if the participants recognized that they had to focus on the character in red or not. 20 from 35

participants have understood the game logic by collecting the emotion of the main character.

Main Character Question

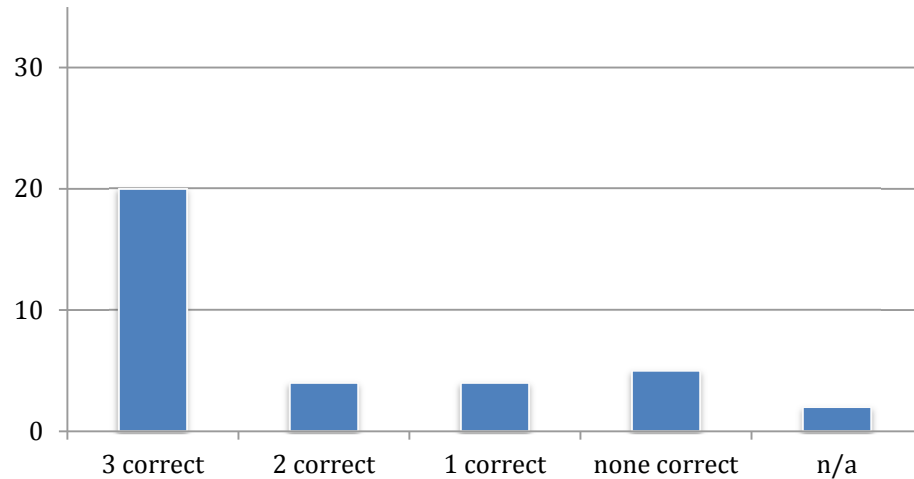


Figure 48 - Main Character Question

The next question (question 2) asked, whereof the children know which coin they have to collect. They answered different logical answers, where many of them show us that the participants not only guessed which coin they had to collect. Some of them gave an honest answer and wrote on the questionnaire that they guessed the correct emotion and about 14% (see in figure 49) really did not know what to collect.

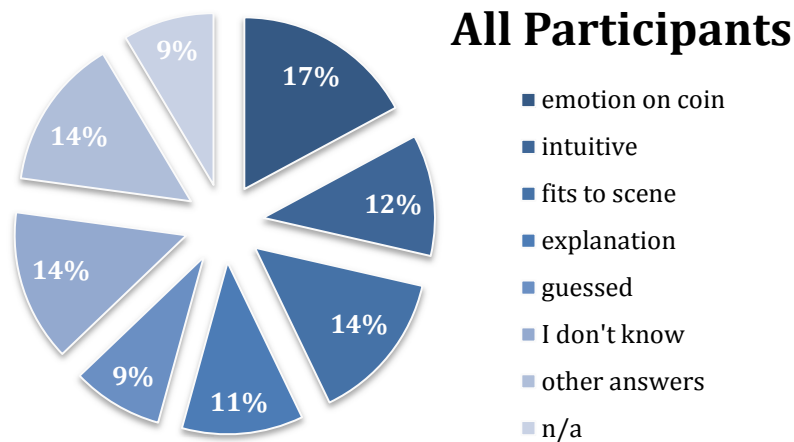


Figure 49 - Collecting Emotions

The argumentation of these results is that the interface concerning the displaying of the emotional coins is correct and is recognized, but the explanation previously and therefore the instruction video should be more informative. This conclusion can be confirmed when we compare the answers between all participants and the answers from group A, which were under observation of the psychology students. All participants of group A understood the game mechanism and the reason behind collecting emotions.

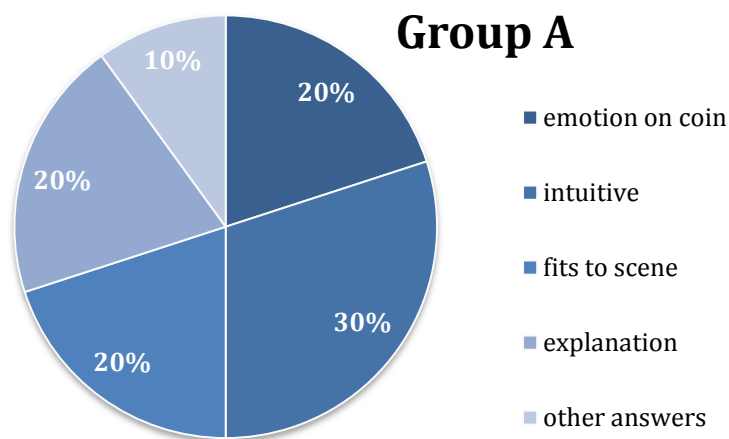


Figure 50 - Collecting Emotions Group A

Next question and therefore to find out if the participants were able to identify barriers in the game design is to mark the bush in a shown image, which represents a crash point. More than 80% marked the correct barrier, where only one participant marked a false object and 14% marked nothing. We cannot evaluate whether this 14% obviously did not know which object a barrier has been or if they ignored to mark it. Nevertheless, the barrier question brought quite good results.

The question concerning the mark where the participants can find their lives brought better results because only one person didn't mark anything. 34 participants marked the correct object, which had been the wooden board with the hearts in the upper left corner of the game screen.

Visualised results of those two questions can be found in figure 51 and 52.

Barrier

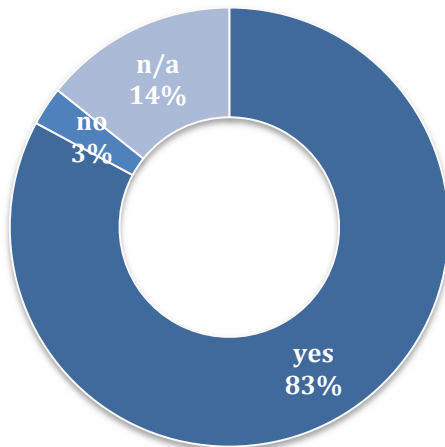


Figure 51 - Barrier Recognition

Lives

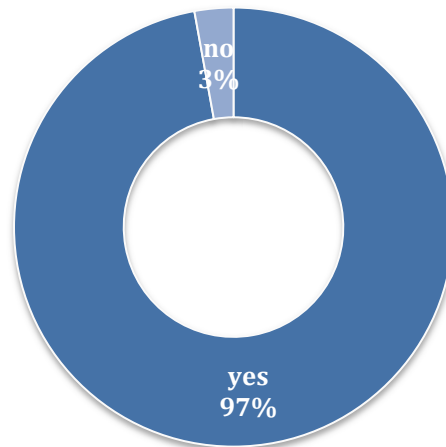


Figure 52 - Lives Recognition

Lastly, we asked the participants some questions about their understanding of the scoring interface. We want to know if they could correctly interpret the scoring of another player and why they decided so. On the one hand, they had to recognize how good a player collected emotions concerning one scene and on the other hand, if he or she succeeded a sublevel.

The resulting answers concerning a scene were good because nearly 90% (see in figure 53) recognized that a scene was played well. The other question concerned a final score with the green bar and the trophies, where the correct answer was, that the player has won the sublevel. We wanted to know where the participants could see if someone succeeds, which would be displayed through the green bar. 26 children marked the green bar, which were 74% that

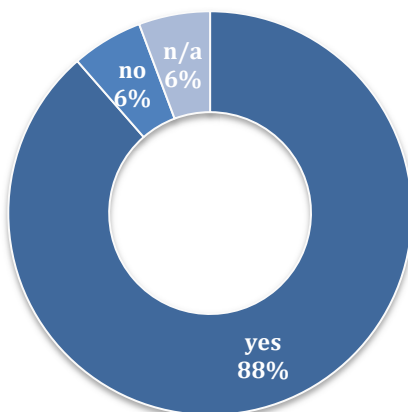


Figure 53 - Scene Recognition

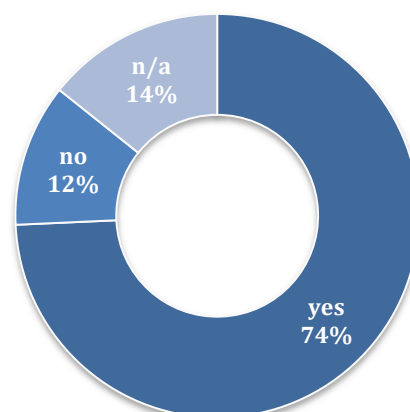


Figure 54 - Final Score Recognition

were correct in their answering, but also 4 participants' marked false objects and further 5 participants marked nothing (see in figure 54). These results showed us that the scene score screen is more logical for the participants than the final score screen.

The topic interaction was asked through question 4 from part 2, where we wanted to know which button on the keyboard is necessary for the game *EmoJump*. All of the participants and therefore 100% answered and marked the correct key, which was the space bar.

Functionality and Features is a topic at which we asked questions concerning pause and joker. We wanted to know if the participants knew where the pause and the joker button are, and how useful the joker button can be. The pause button, which is in the right upper corner of the game, was marked correctly from 34 participants. The joker button is represented as a light bulb and was marked by 32 participants, which is quite a good rate.

The next question was, what the children will do if they do not understand the scenario. Therefore one child didn't mark anything but gave us a textual answer that it will have a closer look at the scenario more exactly, which is in fact also correct but not the expected answer. Question 9 consists of two questions, where we also asked if the player succeeded the level, but the formulation is: "Do YOU have succeeded the level?". Some of the children referred the answer to the picture und some others to their own gameplay before. In this case we cannot evaluate this question, because of the different understandings and answers.

Finally, the participants could give the game their own name if they wanted to. This is only a question for our interest and how creative the children are and if they were interested in the game, so they would also name it. There were several answers and to name but a few:

- Emotion Jump
- Mega Jump
- World Run
- Coin Feeder
- Jump Smiley

4.5 Summary & Improvements

In summary, the results of the questionnaire are quite good but there is room for improvement. The children had very different opinions concerning the game, but on the whole the game can be seen as moderate till good. They liked the jumping, the music and some funny scenes and think that the game is suitable for children in their age or for younger children. This is a good aspect, the game should be simple, easy but nevertheless make fun, though most of the children the game is too easy. We wanted to create a simple game, because children with Autism Spectrum Disorder (ASD) may have limitations in playing and understanding a complex training tool. Negative arguments came up when the children crashed too often, but after the questionnaire session we improved the jump behaviour and the speed of the grounds for a better game experience. In the game version during this questionnaire, the emotional coins had not yet an appended sound effect, so future analysis may bring better results in understanding which coin the participants have to collect because of the auditory assistance. It would have been better to formulate some questions more in detail, or to include further explanations concerning the answers, because we could not proof if the children didn't know what to mark or answer because they were not prompted to do or they really didn't know the answer. The advantage of working with children is also that they are sincere and interested and answer the questions with honesty, without consideration of their arguments. Some results showed us that we have to think about various features to guarantee, that the results of the recommendation or the question to play the game again will be better. This could be achieved through a reward system, where the children have the challenge to play the game more often to win all available prizes. The questions concerning the interface deliver results, which showed us that some objects have to be designed more silhouetted against the background. Other objects could replace barrier placeholders and the light bulb is maybe not a good representation of a joker feature, because some of the children did not associated it as help for a scenario. The results offer a chance for further improvements for a game that should not only have good results in design and interface, or providing a fun factor, but also in learnability and the ambition to replay it.

5 Conclusion

The aim of the whole project and the goal of this master thesis were to develop and evaluate a game for children with Autism Spectrum Disorders to train their emotion recognition skills. In cooperation with the Institute of Clinical Children and Adolescent Psychology we designed and implemented a training tool, which can be used in psychological programs to improve emotional understanding as well as acting as an analysis tool that logs game results. The motivation was to get positive feedback of playing an evidently normal jump and run game with an educational background.

In this thesis we described the development of our serious game called *EmoJump* in detail, which is implemented with our parallel-developed game engine *Jumru 5s*. We started to have a look at related projects that are games on the computer, which also have the purpose to train the emotion recognition skills of children with ASD. These games have quite good results in their training evaluation, but most of them are programmed with a static design, which motivated us to develop a game with a jump and run character instead of a click-through mechanism. Next, we had a closer look on the roots of developing serious games and figured out what it means when a game is developed with an educational or psychological background. Concerning the technical background, we listed some various design principles, which are a recommendation to think about when developing a game for children with disabilities. The main focus of this thesis was described in the sections of the detailed description of *EmoJump* and the description of the implementation of that game. To complete the project, we presented the evaluation results of 35 participants that played the game and filled in a questionnaire, existing of questions about usability. Finally,

these results showed us that the game makes fun, is simple and easy and can be therefore a challenge for children with impairments in emotion recognition, but they also showed us that some further improvements are necessary. For a more detailed evaluation, further tests with participants with Autism Spectrum Disorders (ASD) would be interesting.

The game would be more attractive for children if it would constitute of more extensive graphics, were a game designer could be integrated to such projects. The evaluation questionnaires could also be integrated in the game, which we have already implemented in the game engine but not yet active in *EmoJump* and would simplify the reporting of the answers.

As we mentioned, the development of a serious game should not only focus on designing a game that brings fun and enjoyment but it should also act as training or teaching tool. Our logging mechanism improves the appropriation of *EmoJump* because psychologists could automatically check if children improved their skills, without using manual evaluations. This project showed us an important step of game development in the future that could really improve someone's life.

Bibliography

- [1] Dirk Bosmans and Paul Maskell. (2012, November) Videogames in Europe: Consumer Study. [Online].
http://www.isfe.eu/sites/isfe.eu/files/attachments/austria_-_isfe_consumer_study.pdf
- [2] Johannes Breuer and Gary Bente, "Why so serious? On the Relation of Serious Games and Learning," *eludamos Journal for Computer Game Culture*, vol. 4, no. 1, p. 19, Januar 2010.
- [3] Marja Kankaanranta and Pekka Neittaanmäki, *Design and Use of Serious Games*. Finland: Springer Science & Business Media, 2008.
- [4] Heike E. Krüger-Brand. (2013, Jan.) Deutsches Ärzteblatt. [Online].
<http://data.aerzteblatt.de/pdf/110/4/a126.pdf>
- [5] Erin. (2013, August) Serious Play = Serious Learning for Museums. [Online].
<http://www.edgital.org/2013/08/>
- [6] Zone 2 Connect GmbH. (2014, Dec.) SeriousGames.de. [Online].
<http://www.seriousgames.de>
- [8] Clark C. Abt, *Serious Games*.: The Viking Press, 1970.
- [7] Hanan Makki Zakari , Minhua Ma, and David Sim, "A Review of Serious Games for Children with Autism Spectrum Disorders (ASD) ," Digital Design Studio, Glasgow School of Art, Glasgow, UK, Glasgow, UK, Paper 8778, 2014.
- [9] Michael Zyda, "From visual simulation to virtual reality to games," *Computer*, vol. 38, no. 9, pp. 25 - 32, September 2005.

- [10] David R. Michael and Sande Chen, *Serious Games: Games that Educate, Train and Inform*. Canada: Thomson Course Technology, 2006.
- [11] Monks - Ärzte im Netz GmbH. (2015, January) Neurologen und Psychiater im Netz: Das Informationsportal zur psychischen Gesundheit und Nervenerkrankungen. [Online]. <http://www.neurologen-und-psychiater-im-netz.org/kinder-jugend-psychiatrie/erkrankungen/autismus-spektrum-stoerung-ass/>
- [12] Helmut Remschmidt, *Autismus: Erscheinungsformen, Ursachen, Hilfen.*: C.H.Beck; Auflage: 4, 2008.
- [13] Lorna Wing, Judith Gould, and Christopher Gillberg, "Autism spectrum disorders in the DSM-V: Better or worse than the DSM-IV?," *Research in Developmental Disabilities*, vol. 32, no. 2, pp. 768-73, 2011.
- [14] José C. Miranda, Tiago Fernandes , A. Augusto Sousa, and Verónica C. Orvalho, *Interactive Technology: Teaching People with Autism to Recognize Facial Emotions, Autism Spectrum Disorders - From Genes to Environment, Prof. Tim Williams (Ed.)*, 2011.
- [15] Bretagne Abirached, Yan Zhang, and Ji Hyun Park, "Understanding User Needs for Serious Games for Teaching Children with Autism Spectrum Disorders Emotions ," School of Information, Denver, ISBN 978-1-880094-95-2, 2012.
- [16] Rita Marques Tomé, João Madeiras Pereira, and Manuel Oliveira, *Using Serious Games for Cognitive Disabilities in Serious Games Development and Applications.*: Springer, 2014.
- [18] Porto Interactive Center. (2014, Dec.) LIFEisGAME - A game about emotions. [Online]. <http://www.portointeractivecenter.org/lifeisgame>
- [17] Tiago Fernandes, Samanta Alves, José Miranda, Cristina Queirós, and Verónica Orvalho. (2014, -) LIFEisGAME:A Facial Character Animation System to Help Recognize Facial Expressions. Paper.

- [19] Samanta Alves, António Marques, Cristina Queirós, and Verónica Orvalho, "LIFEisGAME Prototype: A Serious Game about Emotions for Children with Autism Spectrum Disorders," *PsychNology Journal*, vol. 11, no. 3, p. 22, 2013.
- [20] Evidence Based Game Design. (2014, February) FACE SAY. PDF. [Online]. <http://ebgd.be/?wpdmdl=24>
- [21] Symbionica, LLC. FaceSay™ Social Skills Software Games. [Online]. www.facesay.com
- [22] Ingrid Maria Hopkins et al., "Avatar Assistant: Improving Social Skills in Students with an ASD Through a Computer-Based Intervention," *Journal of Autism and Developmental Disorders*, pp. 41:1543–1555, February 2011.
- [23] Simon Baron-Cohen, Jacqueline Hill, Ofer Golan, and Sally Whellwright, "Mind Reading Made Easy," *Cambridge Medicine*, vol. 17, pp. 28-29, 2002.
- [24] Ofer Golan and Simon Baron-Cohen, "Systemizing empathy: Teaching adults with Asperger syndrome or high-functioning autism to recognize complex emotions using interactive multimedia," *Development and Psychopathology*, vol. 18, pp. 591–617, 2006.
- [25] Wade Juneke, "Mind Reading: The Interactive Guide to Emotions," *Journal of the Canadian Academy of Child and Adolescent Psychiatry*, vol. 16, no. 4, pp. 182–183, November 2007.
- [26] Autism Transporters CIC. The Transporters. [Online]. <http://www.thetransporters.com>
- [28] Ofer Golan et al., "Enhancing Emotion Recognition in Children with Autism Spectrum Conditions: An Intervention Using Animated Vehicles with Real Emotional Faces," *Journal of Autism and Developmental Disorders*, vol. Volume 40, no. Issue 3, pp. pp 269-279, March 2010.

- [27] Simon Baron-Cohen, Ofer Golan, and Emma Ashwin, "Can emotion recognition be taught to children with autism spectrum conditions?," *Philosophical Transactions of the Royal Society*, no. Volume 364(1535), pp. 3567–3574, December 2009.
- [29] Paul Ekman, "Basic Emotions," in *Handbook of Cognition and Emotion*. England: Wiley, 1999.
- [30] Ernest Adams, *Fundamentals of Game Design*.: New Riders, 2010.
- [31] Pixton Comics Inc. (2015) Pixton Comics Einen Comic erstellen. [Online].
<http://www.pixton.com>
- [32] Ed Burnette and Jörg Staudemeyer, *Eclipse IDE - kurz & gut*.: O'Reilly Verlag GmbH & Co. KG, 2009.
- [33] W3C Recommendation. (2014) W3C - HTML5. [Online].
<http://www.w3.org/TR/html5>
- [34] W3C Schools. (2015, January) HTML5 Introduction. [Online].
http://www.w3schools.com/html/html5_intro.asp
- [35] Philipp Rieber, *Dynamische Webseiten in der Praxis: Mit PHP 5, MySQL 5, XHTML, CSS, JavaScript und AJAX (mitp Professional)*.: mitp, 2009.
- [36] W3C Schools. (2015, January) Javascript. [Online].
<http://www.w3schools.com/js>
- [38] Alan Thorn, *Game Engine Design and Implementation: Foundations of Game Development*.: Jones & Bartlett Pub (Ma), 2010.
- [37] W3C Schools. (2015, January) CSS. [Online].
<http://www.w3schools.com/Css>
- [39] Florence Maurice, *CSS3: Leitfaden für Webdesigner*.: Addison-Wesley Verlag, 2011.
- [40] Rex van der Spuy, *Foundation Game Design with HTML5 and JavaScript*.: friendsofED, 2012.

- [41] Author Niels. (2015, January) Stackoverflow. [Online].
<http://stackoverflow.com/questions/16322869/trying-to-create-a-confetti-effect-in-html5-how-do-i-get-a-different-fill-color>
- [42] Mojang. (2009-2015) Minecraft. [Online]. <http://www.minecraft.net>
- [43] Rita Marques Tomé, João Madeiras Pereira, and Manuel Oliveira, *Using Serious Games for Cognitive Disabilities in Serious Games Development and Applications*.: Springer, 2013.
- [44] Laili Farhana Md Ibharim and Maizatul Hayati Mohamad Yatim, "iBUAT: Paper Prototyping of Interactive Game Design Authoring Tool for Children," *Serious Games Development and Applications*, 2014.
- [45] Panos Markopoulos, Janet C. Read, and Stuart MacFarlane, *Evaluating Children's Interactive Products: Principles and Practices for Interaction Designers (Interactive Technologies)*.: Morgan Kaufman Publ Inc, 2008.

Appendix

Questionnaire

Pre-Test

Playtest Dokument

Code: _____

Datum: _____

Alter: _____

Geschlecht:

Junge Mädchen

☐☐

Einführungsfragen

1. Spielst du gerne Video Spiele?

JA NEIN

☐☐

2. Welches Spiel spielst du am liebsten?

3. Hast du einen Computer zu Hause?

JA NEIN

☐☐

4. Spielst du Spiele auf dem Computer? (nicht Konsole)

JA NEIN

☐☐

5. Welche?

Fragen nach dem Spiel

1. Hat dir das Spiel Spaß gemacht?

Gar nicht

Mittel

Sehr viel



2. Was musstest du im Spiel machen?

3. War das Spiel für dich...

Zu leicht

genau richtig

zu schwer

☐☐☐

4. Was hat **dir** am **Besten** an dem Spiel gefallen? Warum?

5. Was hat dir an dem Spiel **NICHT** gefallen? Warum?

6. Hat dich irgendwas geärgert an/während dem Spiel?

7. Wolltest du das Spiel gewinnen? JA NEIN

☐☐

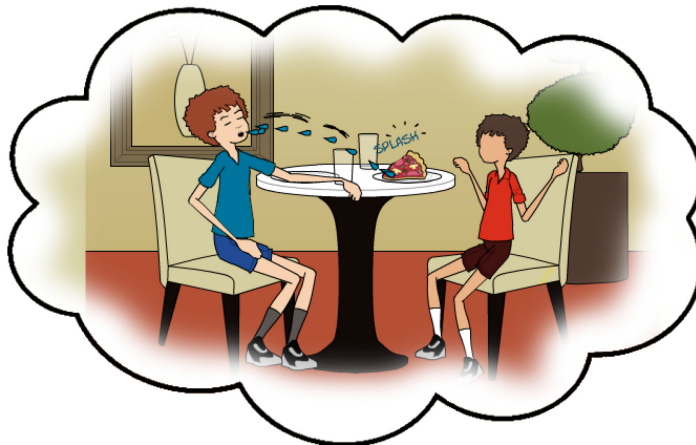
8. Würdest du das Spiel gerne noch einmal irgendwann spielen?

9. Ist das Spiel was für dich oder eher für jüngere/ältere Kinder?

10. Würdest du das Spiel deinen Freunden/ Klassenkameraden/ Lehrern zeigen?
Warum/ Warum nicht?

BITTE BEANTWORTE ALLE FRAGEN SO GUT DU KANNST!

1. Welche Figur auf diesem Bild ist die **Hauptfigur**?



2. Woher weißt du, **welche Münze** die Richtige ist?

3. Siehst du hier ein **Hindernis**?



4. Welche **Tasten** brauchst du zum Spielen?



5. Was machst du, wenn du während dem Spielen eine **Pause** machen magst?



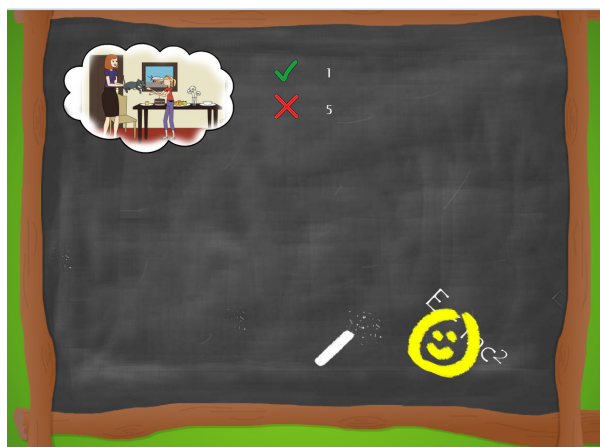
6. Woher weißt du, **wie viele Leben** du hast?



7. Was machst du, wenn du die **Geschichte** im Spiel **nicht verstehst**?

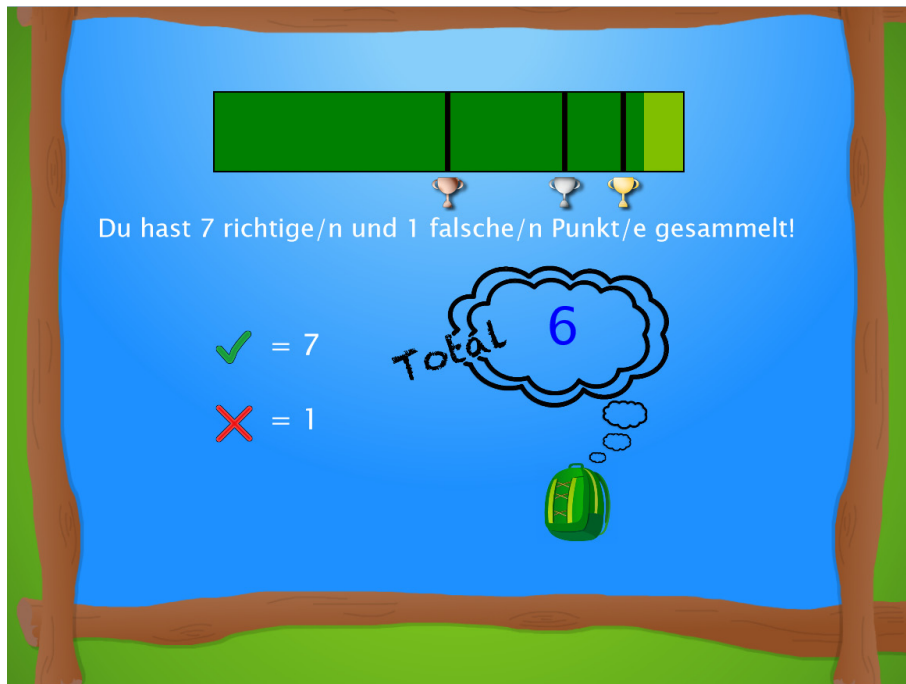


8. Das ist das Spielergebnis eines anderen Kindes. Wie gut hat dieses Kind gespielt?



Was sagt dir dieses Bild?

9. Wo siehst du an diesem Bild, ob du das Level geschafft hast?



Hast du das Level geschafft?

JA NEIN

☐ ☐

10. Wenn du dir einen Namen für das Spiel aussuchen könntest, wie würdest du es nennen?

Abstract

Games are present in our everyday lives, because nowadays they are not only used as playful amusement or occupational therapy, but furthermore they are used as learn-, training- and educational tools and are named serious games. Serious games open up a new possibility to help children with physical, sensory, or mental disabilities to train their limited capabilities through hands-on therapies.

This thesis deals with the topic serious games and which aspects and design principles are recommended concerning the development of games for autistic children. The game concept was created by close project cooperation between the Department of Clinical Child and Adolescent Psychology of the University of Vienna and results in a serious game for children with Autism Spectrum Disorders (ASD), which should train the emotion recognition skills and should help to improve their abilities and their life concerning social communication. This game is named EmoJump and is based on our developed framework as well as on HTML5, CSS3 and JavaScript. It is a jump and run game, which should train the different levels of emotion recognition by representing scenarios, where the player has to match the correct emotion.

In the case of training studies in this field of interest, also psychologists are intrigued how effective such a game could be. For further analysis of the effectiveness of our game, we implemented an underlying logging mechanism to store data about the game play and game results.

We were interested in the question if the interface, design, usability and comprehensibleness are suitable to the needs of children and which impressions children have after playing. The evaluation is based on playtestings, which include observations of gameplays and questionnaires to analyse the opinions of the children.

Kurzfassung

Spiele sind aus dem alltäglichen Leben nicht mehr wegzudenken, denn sie dienen nicht mehr nur als Unterhaltungsmittel und Beschäftigungstherapie, sondern können durch den neuen Forschungsbereich, genannt Serious Gaming, auch zu Lern-, Trainings- und Entwicklungszwecken eingesetzt werden. Serious Games eröffnen die Möglichkeit, dass speziell Kindern mit gesundheitlichen Problemen geholfen werden kann, in dem sie spielerisch therapiert und eingeschränkte Fähigkeiten trainiert werden.

Diese Arbeit befasst sich mit der Thematik von Serious Games und welche Kriterien bzw. Designentscheidungen man speziell bei der Entwicklung von Spielen für autistische Kinder beachten sollte. In Kooperation mit der Fakultät für klinische Kinder- und Jugendpsychologie der Universität Wien haben wir ein Serious Game für Kinder mit Autismus Spektrum Störung (ASS) entwickelt, welches das Emotionsverständnis trainieren und den Alltag der Kinder erleichtern soll. Dieses Spiel trägt den Namen EmoJump und basiert auf einem von uns entwickelten Framework, sowie auf den Technologien HTML5, CSS3 und JavaScript. Es handelt sich um ein Jump and Run Spiel, das die verschiedenen Ebenen der Emotionserkennung trainiert, in dem es dem Spieler/der Spielerin verschiedene Szenarien zeigt, welchen man die passende Emotion zuordnen muss. Durch den Einsatz von derartigen Spielen zu Trainingszwecken sind auch Psychologen daran interessiert, wie effektiv das Spiel sein kann. Zur Analyse der Effektivität wurde ein Loggingmechanismus einprogrammiert, welcher im Hintergrund Daten des Spielablaufs und der Spielergebnisse abspeichert.

Unser Interesse galt jedoch der Verwendbarkeit und Verständlichkeit des Spieles und welchen ersten Eindruck Kinder haben bzw. welche Vor- und Nachteile sie an dem Spiel finden können. Die Evaluierung erfolgte anhand von „Playtestings“ in Kombination mit Fragebögen.

Acknowledgements

First and foremost I would like to gratefully thank my advisor Univ.-Prof. Dipl.-Ing. Dr. Helmut Hlavacs for his guidance, who encouraged and supported me and had always motivational and constructive words during difficult times. He had not only been an advisor for me, even more a mentor, who gave my colleague and me the opportunity to realize such a project and made this thesis work possible.

A very special thank you to my before mentioned colleague Katharina Meusburger, who has not only been a team member, but more a special friend of mine. Thanks a lot for the great teamwork and everything else.

Last but not least, I want to say thank you to my whole family, especially my parents Sylvia and Paul Schweiger and my boyfriend Christoph Prettenhofer for all their warming words and patience during the whole time. Further many thanks to all my lovely friends who supported and motivated me over the last past years during studying and writing this master thesis.

Curriculum Vitae

Name: Schweiger Natascha, BSc.
Geburtsdatum: 10.04.1988
Geburtsort: Lilienfeld
Staatszugehörigkeit: Österreich
E-Mail: natascha.schweiger@gmx.net

Bildungsweg

10/2011 – heute	Universität Wien Masterstudium Medieninformatik
04/2011 – 10/2011	Universität Wien Bachelorstudium Theater- Film- u. Medienwissenschaften
10/2007 – 03/2011	Technische Universität Wien Bachelorstudium Medieninformatik Abschluss: Bachelor of Science (BSc.)
09/2002 – 05/2007	Bundeshandelsakademie (BHAK) Bruck/Mur
09/1998 – 07/2002	Hauptschule Mariazell

Publikationen

2014	N. Schweiger, K. Meusburger, H. Hlavacs and M. Sprung, Jumru 5s - A Game Engine for Serious Games, The Fifth International Conference on Serious Games Development & Applications (SGDA 2014) 9-10 Oct 2014, Berlin, Germany.
------	---

Berufstätigkeiten

05/2012 – heute	ImmobilienScout24 Österreich, 1030 Wien
05/2011 – 04/2012	DailyDeal GmbH - ein Mitglied der Google Familie, 1060 Wien
10/2008 – 01/2009	ORF Enterprise, 1136 Wien
12/2007 – 01/2008	Ski- und Snowboardschule Tulln, 3224 Mitterbach am Erlaufsee
2006 – 2010	Firma Traunfellner, 3180 Lilienfeld (jeweils in den Monaten Juli-August)