Vrije Universiteit Brussel

FACULTY OF SCIENCE AND BIO-ENGINEERING SCIENCES
DEPARTMENT OF COMPUTER SCIENCE

Towards Automatic Debriefing of Serious Games

Graduation thesis submitted in partial fulfillment of the requirements for the degree of
Master of Science in Applied Sciences and Engineering: Computer Science

Anas Helalouch

Promoter: Prof. Dr. Olga De Troyer
Advisor: Prof. Dr. Christophe Debruyne

Academic year 2015-2016

Vrije Universiteit Brussel

FACULTEIT WETENSCHAPPEN EN BIO-INGENIEURSWETENSCHAPPEN
VAKGROEP COMPUTERWETENSCHAPPEN

Towards Automatic Debriefing of Serious Games

Proefschrift ingediend met het oog op het behalen van de graad van Master of Science in de Ingenieurswetenschappen: Computerwetenschappen

Anas Helalouch

Promotor: Prof. Dr. Olga De Troyer
Begeleider: Prof. Dr. Christophe Debruyne

Academiejaar 2015-2016

Abstract

Serious Games are used to educate while playing. Therefore they are a powerful tool to support a learning process, but most of them don't include a debriefing session, which is the best way to reflect on what was learned during the game. This work explores the topic of debriefing in the context of a specific game based on artificial intelligence and developed at the WISE lab, i.e. BullyBook. The ultimate goal of this work is to set a first step in the direction of an automatic debriefing system that wouldn't require the presence of a facilitator.

In this thesis the different steps taken to concretise this project are documented. It started with an investigation of the related work to extract meaningful elements that can be used in our debriefing strategy. The next step was to gather information in the game to use it in the debriefing. This is done by means of game states that represent snapshots of the states of the game during gameplay. The next step is to represent the game states in a way that would be understandable, and to accomplish this visualizations and textual feedback were used. Three visualizations in total are explored, each focusing on a different aspect of the game states: the time, the characters and the interactions.

An Ontology was created to represent the game states and a prototype of the Debriefing System was implemented. The System was then evaluated by means of a Pilot Study involving 5 participants, with promising results on the understanding of the outcome of the game.

Samenvatting

Serious Games worden aangewend voor het leren door het spelen van games. Het zijn daarom krachtige hulpmiddelen om zo het leerproces te ondersteunen. Echter, de meerderheid bevatten geen debriefing. Dit is opmerkelijk nu een debriefing de speler op een effectieve manier toelaat te achterhalen wat hij tijdens de sessie bijgeleerd heeft, en wat er nog bijgeleerd kan worden. In dit werk wordt dieper ingegaan op het onderwerp van de debriefing in de context van een spel gebaseerd op artificiële intelligentie en ontworpen in het WISE-lab, m.n. BullyBook.

Met dit onderzoek wordt beoogd een eerste stap te zetten in de richting van een automatisch debriefing systeem dat de aanwezigheid van een facilitator niet vereist. Het onderzoek overloopt en documenteert de verschillende stappen die werden ondernomen om aan dit project een concrete vorm te geven. In de eerste plaats wordt na vergaring van relevante literatuur de meest geschikte elementen aangekaart die in het debriefing strategy gebruikt kunnen worden. Dit wordt gerealiseerd door middel van game states, i.e. momentopnames van de status van het spel. Vervolgens zullen de game states overzichtelijk moeten worden gepresenteerd, op zodanige wijze dat ze begrijpelijk overkomen. Hiervoor werd gebruik gemaakt van visualisaties en tekstuele feedback. In totaal werden drie verschillende soorten visualisaties onderzocht en uitgeprobeerd die elk focussen op verschillende aspecten van de game states: de tijd, de personages en de interacties. Een Ontologie werd ook samengesteld rond de game states.

Ten slotte werd een eerste prototype van het Debriefing System geïmplementeerd. Het Debriefing System werd op zijn beurt geëvalueerd met behulp van een Pilot Study waarbij 5 vrijwilligers aan deel hebben genomen. Dit leverde optimistisch stemmende resultaten op zowel m.b.t. het verstaan als de uitkomst van het spel.

Declaration of Originality

I hereby declare that this thesis was entirely my own work and that any additional sources of information have been duly cited. I certify that, to the best of my knowledge, my thesis does not infringe upon anyone's copyright nor violate any proprietary rights and that any ideas, techniques, quotations, or any other material from the work of other people included in my thesis, published or otherwise, are fully acknowledged in accordance with the standard referencing practices. Furthermore, to the extent that I have included copyrighted material, I certify that I have obtained a written permission from the copyright owner(s) to include such material(s) in my thesis and have included copies of such copyright clearances to my appendix.

I declare that this thesis has not been submitted for a higher degree to any other University or Institution.

Acknowledgements

First of all I would like to thank my promoter Prof. Dr. Olga De Troyer and advisor Prof. Dr. Christophe Debruyne for their guidance and support throughout this work.

I would like to thank my parents for their patience and unconditional support.

I also would like to thank the persons who participated in the evaluation for their time and effort.

And finally I would like to thank my friends and family for giving me the much-needed laughs on the heavy work days.

Thank you all.

Abbreviations

ATTAC	Adaptive Technological Tools Against Cyberbullying
COD	Character-Oriented Debriefing
COV	Character-Oriented Visualization
IOD	Interaction-Oriented Debriefing
IOV	Interaction-Oriented Visualization
NPC	Non-Player Character
RDF	Resource Description Framework
TOD	Time-Oriented Debriefing
TOV	Time-Oriented Visualization

Contents

List of Figures

List of Tables

1 Introduction

Serious Games are created with the specifically designed goal of teaching people while they play. When using serious games to teach or to raise awareness about real situations, a so-called debriefing plays an important part. Debriefing is described as “ the occasion and activity for the reflection on and the sharing of the game experience to turn it into learning ” (Crookall, 2010). In order to enable the learner to acquire new knowledge, it is needed to experience this knowledge in other settings than the one in which the game was played (Nicholson, 2012). Most digital serious games however don’t include a debriefing phase as part of the game. Usually the debriefing is done by having a facilitator sit together with the learner and discuss his results and past actions in the game. This approach can be costly and time consuming for large applications. Nowadays serious games are mostly digitalized, and including an automatic debriefing as part of the gaming experience would improve the fluidity of the experience for the learner. However, achieving an automatic debriefing for games that are based on an AI (Artificial Intelligence) approach is not obvious, because the player may not be able to understand the underlying mechanism that lead to the final result. The goal of this thesis is to investigate how we can provide automatic debriefing for non-linear serious games. As this is a rather new research domain, we decide to start with a concrete case and investigate how to incorporate debriefing into this serious game.

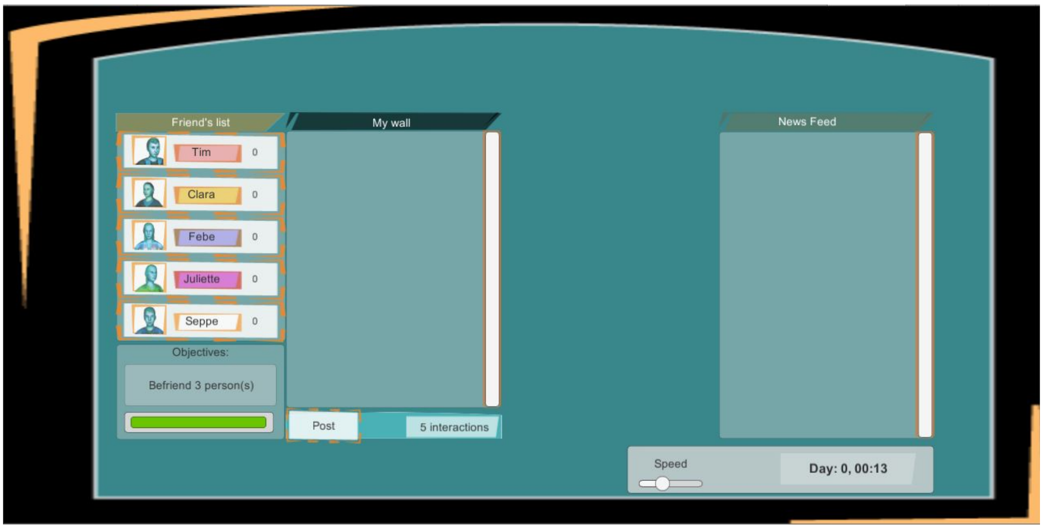


Figure 1.1: Screenshot of the first level of BullyBook.

As part of the Friendly ATTAC project (*Friendly ATTAC*, 2012) , a research project with the aim of investigating how game technology could help with the problem of cyberbullying, the BullyBook is being developed at the WISE department of the VUB. BullyBook simulates a social network like Facebook with the interactions and capabilities that go with it and with the specific didactic aim to raise awareness about Cyberbullying (see figure 1.1, a screen of the game). In this thesis a first step is made towards an automatic Debriefing System for this serious game.

After examining the structure and the underlying model used by the game for its interactions, a debriefing strategy was defined. The Debriefing System lies at the center of this strategy. The system was designed and a first prototype was implemented. For the sake of reusability of this system, we also created an ontology and rules that would allow to apply the debriefing in different contexts. Our goal is not to have a fully reusable system that could be used on all serious games, but to strive for a generic approach where many elements could be reused. The first step in designing this Debriefing System was to determine which elements would need to be captured during the gameplay. These elements can be stored in game states, which can be compared to snapshots of the states of the game during gameplay and which are also used for the save functionality in entertainment games. In our system, these game states are automatically stored each time an interaction occurs.

The next step in our debriefing strategy is to give information about the interactions that took place in the game and provide feedback about these interactions, in a way that is understandable for the target audience of the serious game. We decided to opt for two methods to represent this. The first method uses a direct and textual feedback to the player about his actions and how they resulted in the game outcome. For this the underlying reasoning model used by the game was integrated in the Debriefing System but with little details to avoid making it too complex. The second method consists of providing visualizations. Three different kinds of visualisations were designed for this, each focusing on a different aspect of the game states. The first visualisation puts the focus on the time of interaction. This allows the player to inspect what happened in the game over time in a linear way. In the second visualisation the characters of the game are the most important elements. The idea of this visualisation is to show the relationships between characters in the game. Finally the interaction-oriented visualisation is the most complete one, and focuses on the interactions. With this visualisation a general overview of the interactions is given to the player. The three visualisation approaches were implemented as intractable visualisations.

The thesis also discussed some extensions. One of these possible additions is the replay functionality that would allow the player to replay certain sequences of the game as part of the debriefing. Putting the player in the context of a bad decision during the game and showing why this particular decision was not the best option could help the learning process. Having an extensive feedback about the actions in the form of a dynamic questionnaire is also one way to automatize the debriefing process. This dynamic questionnaire could be used to ask questions to the system about the outcome in the form of a dialogue.

1.0.1 Research goals

The goal of the thesis is to explore the topic of debriefing in the context of a specific AI-based serious game developed at the WISE lab, i.e., BullyBook. The research questions can be formulated as follows:

- How can we explain the game outcome of BullyBook to a casual player?
- Which game data are relevant to track for this purpose?
- Which visualization techniques are suitable for representing this data and allow the player to explore the data and their relationships?
- What is the right amount of detail to expose? How much information is too much? What types of data do we present and how do we provide the player insight into the game outcome?

1.0.2 Methodology

The research methodology applied for this thesis work consists of a number of steps. Firstly the related work was investigated to see how debriefing was done for other serious games. However, debriefing is also used in other domains and focusing only on existing debriefing strategies for serious games proved to be insufficient. Therefore, we broadened the scope. The next step consisted of examining the BullyBook game and the theoretical principles (e.g. the personality model) it uses. The implementation of the game was also investigated. The next step was to come to a solution. Based on findings from related work, we started by determining which elements would need to be logged during the gameplay to allow for a debriefing afterwards. Then a debriefing strategy was designed by trying out different visualization techniques to show the data. This design was used to implement a first prototype of a Debriefing System for BullyBook. And finally an evaluation was conducted to verify the validity of the Debriefing System.

1.0.3 Outline of dissertation

The first chapter of this dissertation gives a general introduction to the work together with the motivation, the research goals and the methodology that was used. The second chapter contains a description of the background and literature study that was performed. Three different topics were examined: serious games, BullyBook and debriefing. Chapter 3 describes the method towards the solution. First, an analysis of the problem and the approach to solve it are discussed. After this the design of the solution is described. In chapter 4 the implementation and the prototype of the Debriefing System are given. Chapter 5 focuses on the evaluation of the prototype, and outlines the pilot study that was conducted to assess the Debriefing System. The results of the evaluation and a discussion are also included in this chapter. Finally a general summary is given, together with the limitations and future work in chapter 6.

2 Background

2.1 Serious games

The first step towards finding a fitting and complete definition for Serious Games is to start by examining what a game is. Although it is easy to identify a game, capturing the exact elements that make a game can be tricky. Philosophers have been discussing this matter for decades and many definitions emerged from this. A research on this domain yielded following results:

A game is an activity that must be fun, separate (fixed location and time), uncertain, non-productive, governed by rules and fictitious (Caillois, 2015) .

This definition by a set of characteristics seems outdated. A game does not necessarily have to be circumscribed in time and space and examples of games based on augmented reality can be found to contradict this characteristic. Games can also serve a purpose and therefore be productive, as we'll see later with Serious Games. A more recent definition was given by Chris Crawford. According to Chris Crawford's a set of characteristics can be found to describe all games: Representation, Interactivity, Conflict and Safety. With according descriptions for the characteristics:

> "Representation: ... a game is a closed formal system that subjectively represents a subset of reality.

> Interactivity : The most fascinating thing about reality is not that it is, or even that it changes, but how it changes, the intricate webwork of cause and effect by which all things are tied together. The only way to properly represent this webwork is to allow the audience to explore its nooks and crannies to let them generate causes and observe effects. Thus, the highest and most complete form of representation is interactive representation. Games provide this interactive element, and it is a crucial factor in their appeal.
>
> Conflict: The player is actively pursuing some goal. Obstacles prevent him from easily achieving this goal.
>
> Safety: A game is an artifice for providing the psychological experiences of conflict and danger while excluding their physical realizations." (Crawford, 2003)

This definition is complete and contains more elements that can survive the evolution of games. Finally, a first transition to Serious Games can be made using Zyda (2005) definitions for Games, Computer Games and Serious Games.

He defines a game as a physical or mental contest that is played with certain rules and the goal is to amuse or reward the player. For the video games the nature of the contest is only mental. As for Serious Games, they are played on a computer with specific rules and with the goal to train while entertaining.

This coherent definition places Serious Games as a continuation of Games and Video Games, but it also limits Serious Games to be computer-based. However, Serious Games can for example also make exclusive use of playing cards, an application of this can be found in Lasse Hakulinen's experiment where cards are used to teach sorting algorithms (Hakulinen, 2011) .

Numerous sources reduce Serious Games to its most simple definition :

- "Serious Games are games that have other objectives than pure entertainment and they include all aspects of education: teaching, training and informing." (Hakulinen, 2011)
- "Games that have a certain positive goal beside entertainment (education, health promotion, etc.) are referred to as Serious Games." (Shoukry, Göbel, & Steinmetz, 2014)

A more precise definition is given by Clark C. Abt :

> "Reduced to its formal essence, a game is an activity among two or more independent decision-makers seeking to achieve their objectives in some limiting context. A more conventional definition would say that a game is a context with rules among adversaries trying to win objectives. We are concerned with serious games in the sense that these games have an explicit and carefully thought-out educational purpose and are not intended to be played primarily for amusement. " (Abt, 1987)

To summarize, the following can be said about Serious Games, given the definition for a game:

- They can be played in virtual or real environment
- They have a primary purpose which can be education, training, marketing, health, advertisement, public policy or strategic communication objectives.
- They can be played alone or with multiple players

2.2 BullyBook, a Serious game based on the iATTAC system

BullyBook is a serious game currently under development in the Wise department at the Vrije Universiteit Brussel. The main purpose of the game is to raise awareness about cyberbullying. To accomplish this a simulation of a social network has been created, and to ensure that the social interactions with the Non-Player Characters (NPC) remain realistic, the iATTAC system was used.

2.2.1 iATTAC

iATTAC is part of the Friendly ATTAC project, and presents a system for realistic interactions in a social network environment between autonomous game agents (Cebolledo & De Troyer, 2015; Gutierrez & De Troyer, 2015). The underlying model used for iATTAC relies on different principles such as personality models, moods, rituals etc.

Cyberbullying

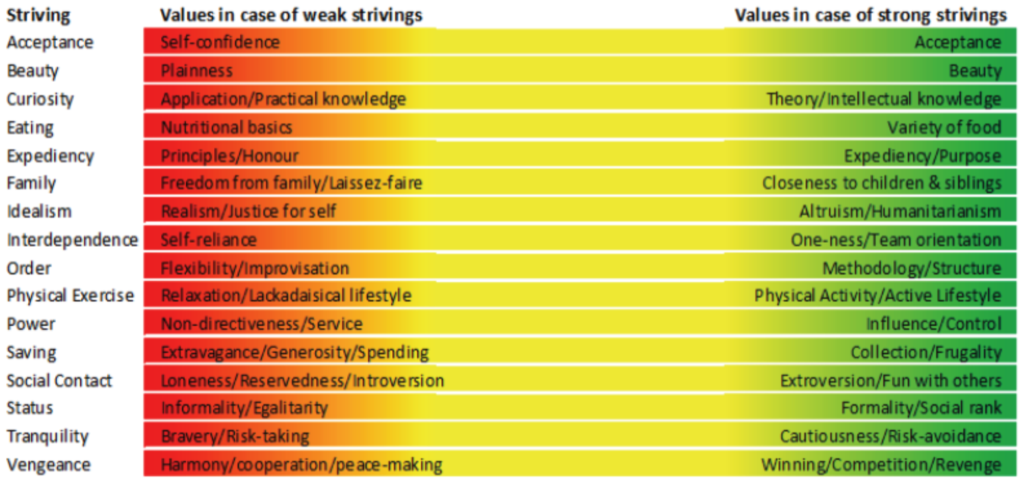
Cyberbullying, or bullying on the internet, is a phenomenon that came with the inevitable rise of internet technologies in the last years. Social Networks more specifically have highly contributed in the existence of this phenomenon. iATTAC distinguishes 3 types of characters in the bullying process : The bully, the victim and the bystander.

Personality models

Each character in the iATTAC system has an autonomous behavior, which is achieved by using personality models. More specifically Reiss personality model (Reiss, 2008) serves as foundation for this purpose. This model consists of 16 basic desires as depicted in Figure 2.1. According to the associated theory, every human being tries to fulfil these needs, giving priority to the desire with the lowest value, while the value for the needs decreases over time. The pace at which the value decreases is determined by the individual personality of every human being. In fact, a personality can be described by how fast the distinct values associated with the 16 desires decrease. The personality model forms the base of the personality in the iATTAC system, but other components are added to form a complete personality. Firstly, the Additional Values can be adopted in certain cases. These take the form of rules that can enforce the behavior of agents. An example of this would be to have an Additional Value to characterize the “ not standing up for victims ” in a bullying scenario. Lastly the moods are also an important piece of a personality. The moods in the iATTAC framework focus on the emotions of characters, and for this end the 6 emotions model (anger, disgust, fear, joy, sadness and surprise) has been used. The aim of adding moods to the personality is to be able to add facial expression to represent the emotions of the NPC. However, in the current implementation of BullyBook, moods are not used.

Social interactions

To handle social interactions between characters in iATTAC, Eric Berne’s transactional analysis is used (Berne, 1961) . At the core of this framework lie the rituals. In a classical bullying ritual we would identify 3 types of characters (bully, victim and bystander). If this ritual is executed every character’s personality is affected (either positively or negatively). For example after being the victim of a bully, one might have its basic desire for safety increase in value. Ritual define how social interactions occur, what type of character is involved and how it affects the personality of said characters. Another element taken in consideration by the ritual is the precedence of actions, so that counter-rituals can be applied. This is particularly useful in the bullying scenario, where a victim or bystanders can react to the bullying.



Striving	Values in case of weak strivings	Values in case of strong strivings
Acceptance	Self-confidence	Acceptance
Beauty	Plainness	Beauty
Curiosity	Application/Practical knowledge	Theory/Intellectual knowledge
Eating	Nutritional basics	Variety of food
Expediency	Principles/Honour	Expediency/Purpose
Family	Freedom from family/Laissez-faire	Closeness to children & siblings
Idealism	Realism/Justice for self	Altruism/Humanitarianism
Interdependence	Self-reliance	One-ness/Team orientation
Order	Flexibility/Improvisation	Methodology/Structure
Physical Exercise	Relaxation/Lackadaisical lifestyle	Physical Activity/Active Lifestyle
Power	Non-directiveness/Service	Influence/Control
Saving	Extravagance/Generosity/Spending	Collection/Frugality
Social Contact	Loneness/Reservedness/Introversion	Extroversion/Fun with others
Status	Informality/Egalitarity	Formality/Social rank
Tranquility	Bravery/Risk-taking	Cautiousness/Risk-avoidance
Vengeance	Harmony/cooperation/peace-making	Winning/Competition/Revenge

Figure 2.1: The 16 basic desires in the Reiss personality model (Reiss, 2008)

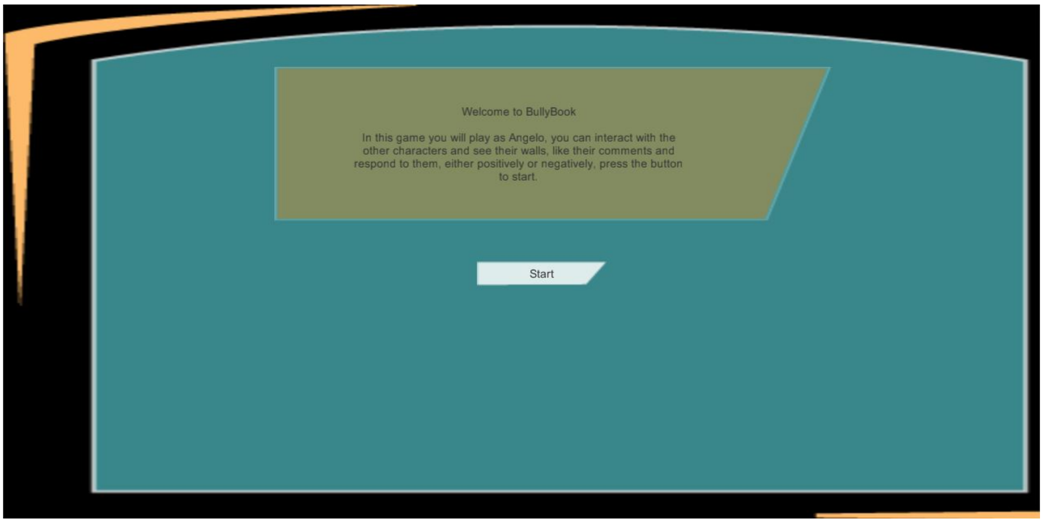
Secondary components

Next to the personality and social interactions, other components are part of the iATTAC system. Although these components are currently not of high importance in the BullyBook game, they still constitute essential parts of the whole system.

- Location : A location can have a decisive effect on the social interactions involved. A hallway is usually more likely to be the scene of bullying than a classroom.
- Personal agenda : A planned schedule might determine what action a character will take in a given moment.
- Memory : Each character has a memory of executed actions, which can also influence the next actions.

2.2.2 BullyBook

To have people understand the reality of cyberbullying, the BullyBook game was developed using Unity3D. The main idea is to have players participate in a social game, largely based on the social network Facebook. The player incarnates a character named Angelo who has a number of friends. These friends interact with each other, as well as with the main player. For every



Welcome to BullyBook

In this game you will play as Angelo, you can interact with the other characters and see their walls, like their comments and respond to them, either positively or negatively, press the button to start.

Figure 2.2: The start screen of the BullyBook game.

action that is occurring, the player can intervene, or he can initiate interactions. These interactions can be in the form of a dialogue or by liking/unliking posts.

The first screen introduces the player to the game by having a short explanation of what is expected. Once the player presses the start button the game can start.

Currently there is only one level playable, and figure 2.2 shows the start screen for this level and figure 2.3 shows the main screen.

The objective of the level is shown in the lower left corner of the screen. In this case the level is to befriend 3 persons. No more indication is given to the player on how to achieve this goal. On the lower right corner the in-game time is given. The speed at which the time progresses can be adapted using the bar left to the time. The main screen is composed of 4 columns. In the first column the list of friends (figure 2.4), the objective and the progress towards this objective are displayed. A color code is used in the background color of the picture that represents the friends of Angelo : green means the friendship is ensured, red means the relationship between Angelo and the character is in a bad state, and if no color is applied then the interaction can still go both ways.

In the second column, the player's wall is represented. This wall can contain posts made by other characters or by the player himself. In this case (figure 2.5) Tim posted a message on Angelo's wall about a book. The player (or Angelo) can then either Like, reply or do nothing about this post.

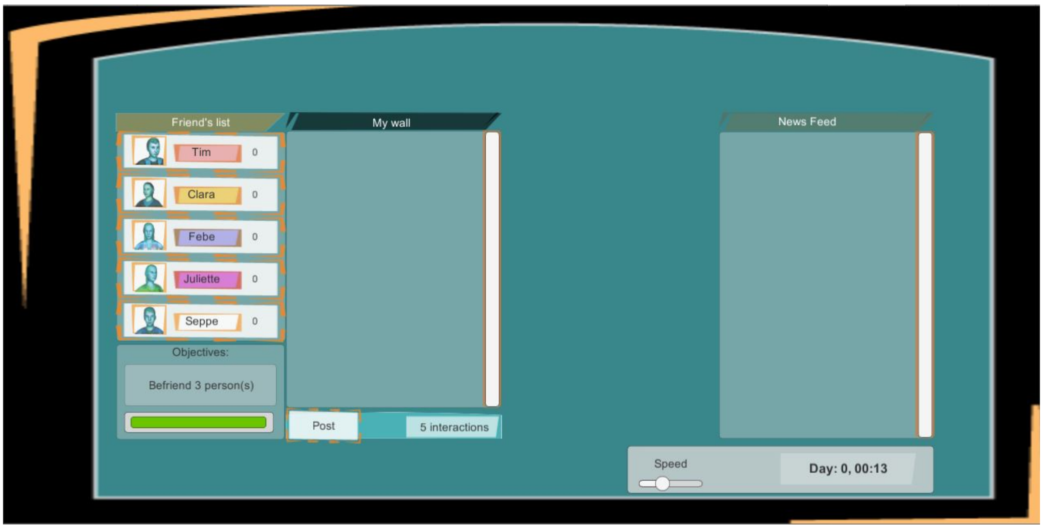


Figure 2.3: The main screen of level 1 in the BullyBook game.

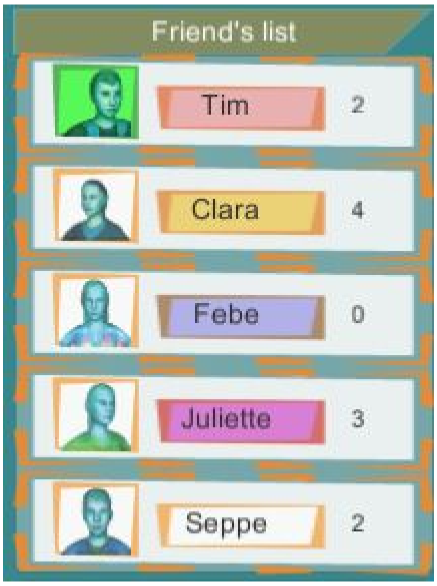
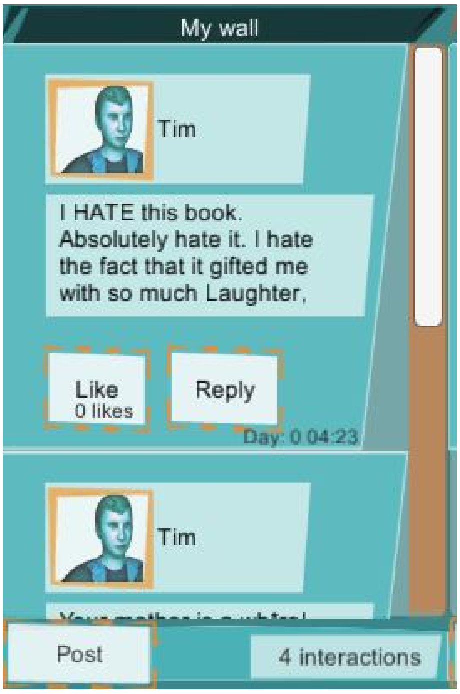


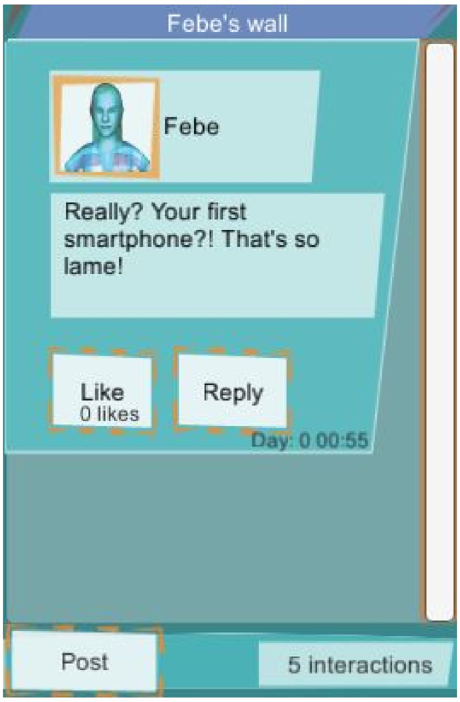
Figure 2.4: The list of friends of the main player.



I HATE this book. Absolutely hate it. I hate the fact that it gifted me with so much Laughter,

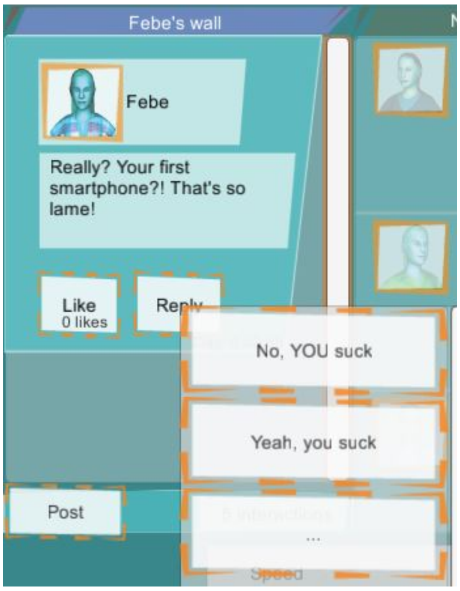
Figure 2.5: The display of the main player's wall.

In the third column of the screen we can see the wall of a selected friend, in figure 2.6 this friend is Febe. Similarly to the player's wall, only posts made on his wall by himself or other characters, are shown.



Really? Your first smartphone?! That's so lame!

Figure 2.6: The wall of a friend.



Really? Your first smartphone?! That's so lame!

No, YOU suck

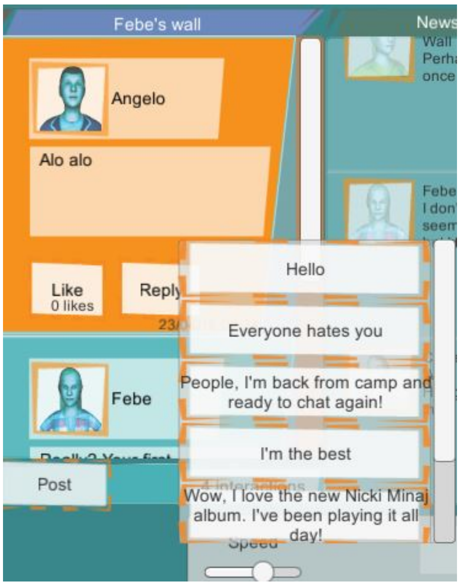
Yeah, you suck

Figure 2.7: The player has the possibility to reply to a post made by a friend

On a friend's wall, the player can either reply to existing posts (figure 2.7). Incidentally this figure (2.7) shows a simple example of how one can assume the position of the bystander in a bullying scenario.

The player can also decide to posts new messages on a friend's wall as shown in figure 2.8.

Finally on the rightmost column of the screen the newsfeed is shown (figure 2.9), this newsfeed is populated by the interactions that occur in the game. By clicking on one of the interactions they appear in the third column to allow further interaction.



Alo alo

Hello

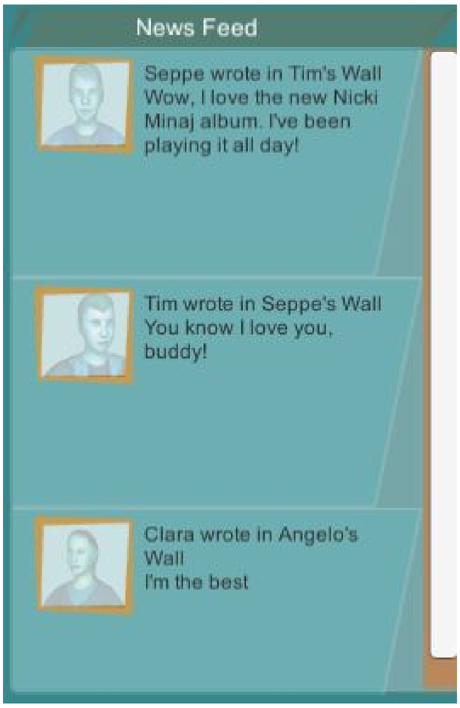
Everyone hates you

People, I'm back from camp and ready to chat again!

I'm the best

Wow, I love the new Nicki Minaj album. I've been playing it all day!

Figure 2.8: The player has the possibility to initiate a social interaction with other characters.



Seppe wrote in Tim's Wall
Wow, I love the new Nicki Minaj album. I've been playing it all day!

Tim wrote in Seppe's Wall
You know I love you, buddy!

Clara wrote in Angelo's Wall
I'm the best

Figure 2.9: The newsfeed containing all social interactions that occurred

2.3 Debriefing and related work

2.3.1 Debriefing

Debriefing in Serious Games can be described as the activity of (formally) reflecting on the gaming experience to turn it into learning (Crookall, 2010). This process of retrospection usually happens when the game is finished, but in-game or pre-game debriefing is also a possibility. Three phases can be observed in the general debriefing process (Fanning & Gaba, 2007): description, analysis and application. The first phase is the description of the events by users or players, usually in their own words. The second phase, the analysis, involves a reflection on how emotions were involved during the simulation. Finally the application phase tends to generalize the individual perceptions. During this phase, playersâĂŹ actions and resulting events are compared to real life events and conclusions are drawn.

Debriefing is one of the most important steps in the Serious Games and without it or other analysis mechanism, Serious Games are nothing more than entertainment in a serious context (Pavlov et al., 2015). However, not much literature can be found about this topic. In what follows a short overview of some existing serious games will be presented together with the used debriefing strategy or proposed debriefing frameworks. The ultimate goal of this process is find elements that could be reused in the Debriefing System for BullyBook.

2.3.2 Related work

The first realization is that very few Serious Games include explicit post-game debriefing, but rather use in-game feedback as a learning mechanism (Barbosa & Silva, 2011) or assume that the game does not require specific analysis of user behavior to facilitate learning (Coenen, Mostmans, & Naessens, 2013)
.

OxyBlood

OxyBlood is a 3D web-based Serious Game developed to teach young students the basic functioning of our circulatory system (Barbosa & Silva, 2011). The goal for the players is to draw the path of red blood cells and to propagate oxygen in the body.

The learning elements included in the game take the form of explanatory videos (Figure 2.10) for each new anatomy part the user reaches. As for the debriefing, no clear mention of it is made, but after each level an inquiry

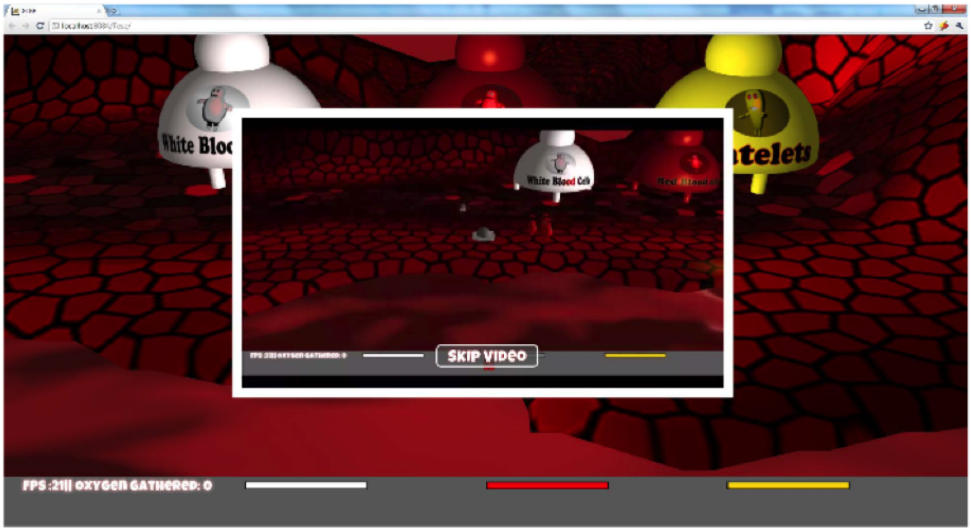


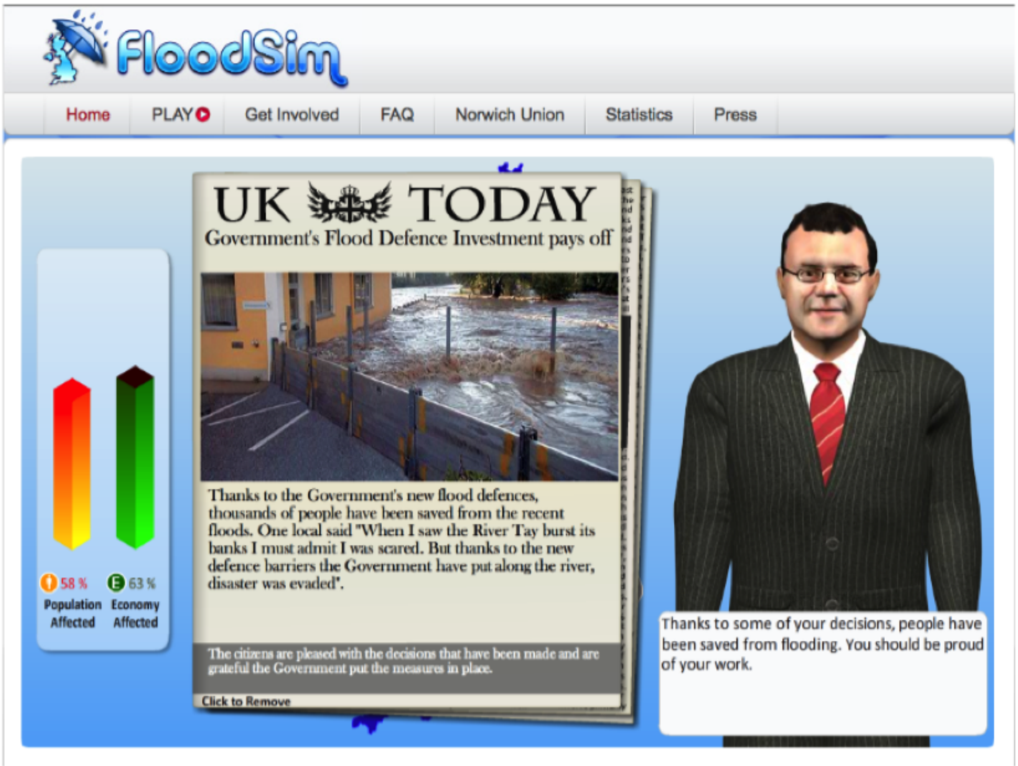
Figure 2.10: OxyBlood video tutorial. (Barbosa & Silva, 2011)

based on the Likert-scale has to be completed by the user to assess if the learning goals are reached. But it is not clear if they intend to use the result of these inquiries for individual feedback or simply to improve the game.

FloodSim

FloodSim is a serious game that tries to raise awareness about the issues surrounding flooding policy and citizen engagement in the UK (Rebolledo-Mendez et al., 2009) . The user actions determine how flooding can be avoided and an in-game dialog tells the player whether his action contributed in a positive way (Figure 2.11).

Although no debriefing at all is done in the game, the authors discuss the possibility of a comparative decision making strategy by using data of other players. This can lead to the concept of comparative debriefing strategy, where the idea is to compare the player's behavior with other players' to better learn.



Government's Flood Defence Investment pays off

Thanks to the Government's new flood defences, thousands of people have been saved from the recent floods. One local said "When I saw the River Tay burst its banks I must admit I was scared. But thanks to the new defence barriers the Government have put along the river, disaster was evaded".

The citizens are pleased with the decisions that have been made and are grateful the Government put the measures in place.

Thanks to some of your decisions, people have been saved from flooding. You should be proud of your work.

Figure 2.11: Feedback in FloodSim (Rebolledo-Mendez et al., 2009) .

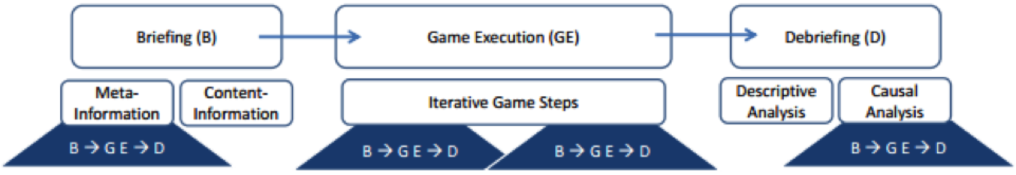


Figure 2.12: Different stages in the revenue management framework (Cleophas, 2012) .

Framework to debrief serious games about revenue management

(Cleophas, 2012) provides a framework to design and develop serious games for revenue management, where the goal is to maximize overall revenue by adapting combinations of products and prices. This framework assumes three stages: the Briefing of the game and the conditions, the Game Execution and finally the Debriefing (Figure 2.12). Note that the game execution consists of different steps that are each to be concluded with a debriefing session.

- Descriptive Analysis: The result indicators are analyzed and compared for all players, in case of multiplayer games. This analysis facilitates autonomous reflection as users can interpret the results themselves and do not necessarily require a trainer. However, this only makes sense if other players are involved in the game as reflecting on one's own scores can be done during the execution in a single player game.

- Causal Analysis: The relation between user actions and resulting events are analyzed individually.

- Concurrence. If the game assumes concurrence play between different users, the game should include this in the debriefing. This can be done by having an individual reflection about the player's action. In the case of a participatory serious game, the effect that one player's action can have on other players should be included in the causal analysis.

- Conditions. If during the briefing phase the explanation of the game conditions were not sufficiently clear for the players and led to an incorrect understanding of the game, it should be corrected in the debriefing. The player should also be able to trace back to each of the user actions to subtract information. If targets or goals exist in the game, the debriefing should reflect on the player's results with respect to these goals.

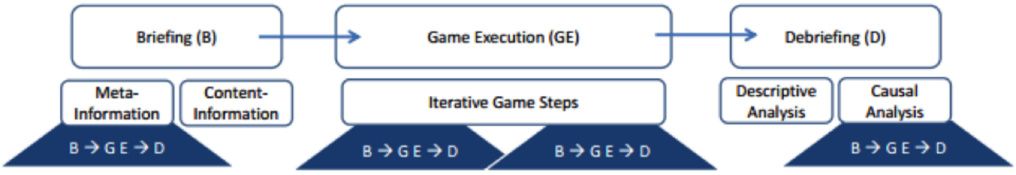


Figure 2.13: Design aspects to be considered for Serious Games about revenue management (Cleophas, 2012) .

- Competition. The aspect of competitions or scores can be modulated by either taking either comparative performance indicators or absolute indicators.

Structural Debriefing is a debriefing activity aimed at helping students learn about causal relationships, feedbacks, accumulations and delay in black-box simulations (Pavlov et al., 2015). This (extensive) strategy is based on Structural Debriefing Protocol, which is a step-by-step description of how to debrief an activity using concepts of System Dynamics . This protocol is tested on LITTLEFIELD, a serious game that simulates a small factory that produces make-to-order electronic equipment. Firstly, different combinations of simulations and debriefing sessions are presented as possible scenario (Table 2.1). Here again we note the possibility of iterative cycles (Design E) of simulation and debriefing. Interestingly the cycle starts with a debriefing before the actual game, but according to the associated explanation this corresponds more with a briefing of the game conditions.

Table 2.1: Different possible scenarios for debriefing, from (Pavlov et al., 2015).

Design A	Design B	Design C	Design D	Design E
Simulation	Short debriefing Simulation	Simulation Short debriefing	Simulation Full debriefing	Short debriefing Short simulation 1 Short debriefing Simulation 2 Short debriefing

The Structural Debriefing Protocol consists of 8 steps:

1. Identify variables in the game: An entity that has a values that can change in the course of the game or stay constant. In the case of LITTLEFIELD this could be Cash balance, Revenue, Raw Materials, etc.

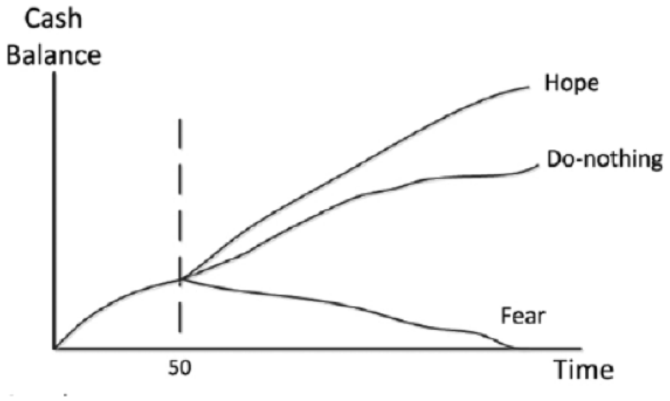


Figure 2.14: Example of reference modes for the cash balance variable (Pavlov et al., 2015)

2. Make reference modes: Graphs that show behavior of the variables over time (figure 2.14), including 3 trajectories for ?hope?, ?fear? and ?expected? (also used: ?best case?, ?worst case? and ?expected?).

3. Define momentum strategies: in this phase the users identify the strategies they employed to reach a certain goal.

4. Dynamic hypotheses: An explanation of the underlying system behavior is attempted using a causal diagram (Figure 2.15).

5. Model construction: A system dynamics model of the black-box simulation is created. This step requires an extensive knowledge on System Dynamics and takes a lot of time (Figure 2.16).

6. Model validation: The designed model has to be validated by testing for unit-consistency.

7. Strategy testing: Once the model has been validated it can be used to explain the outcomes of the adopted strategies. During this phase new strategies can also be implemented and tested to see which the most profitable strategy is.

8. Write a final report about findings.

As could be expected this very extensive Debriefing strategy showed good results in the case of LITTLEFIELD. This can mainly be explained by the carefully organized attempt to understand the underlying system supporting the black-box simulation. As mentioned, this debriefing process takes a lot of time and including it dynamically in Serious Games would seriously damage the ludic aspect of said games. Therefore only using parts of these steps for an (automatic) debriefing seems a safer solution.

2.4 Conclusions and application to BullyBook

While most of the time it is not included in the development cycle, some Serious Games could probably benefit from having a reflection process in the form of a formal debriefing (Pavlov et al., 2015). However, certain games do include a debriefing stage, or leave room for it, and different types were distinguished. An example can be seen in OxyBlood (Barbosa & Silva, 2011) where after completion of each level the user is asked a few questions to assess if he has learned anything. This method seems appropriate for this game as the underlying model is an unambiguous source (Biology) and incorrect answers can easily be corrected and explained. In the case of BullyBook, this could also make sense, but certainly not without guidance. User behavior is more complex to analyze than it is for OxyBlood and tracking user actions with more details seems unavoidable. Using a comparative decision making strategy, as suggested in (Rebolledo-Mendez et al., 2009) , only makes sense as an in-game guiding method. Seeing what others did in a particular situation before taking a decision could increase the learning value during these decisions if the player is lost. However, this approach offers little advantage in a post-game debriefing strategy. The debriefing framework described in (Cleophas, 2012) seems on the other hand to contain more useful elements, even though it is designed for Serious Games about revenue management. This framework also advocates for repeated debriefing sessions after each level in the game execution. Finally the Structural Debriefing Strategy (Pavlov et al., 2015) was explained using LITTLEFIELD. This approach describes an interesting approach by making use of system dynamics to ask the player to reproduce the underlying model of the game. The causal loop diagram especially could be incorporated in an automatic debriefing process.

3 Method

3.1 Analysis

3.1.1 Introduction

In light of the findings from the related work, it seems evident that a strategy to debrief a Serious Game and to facilitate learning is needed. While it is clear that very little proof of formal debriefing in other Serious Games could be found, some elements, however, could be reused in a debriefing scenario. The main question when it comes to debriefing is: What elements are essential in the underlying model of the game? In other words, which elements and players' actions will impact the player's success or failure and how do we measure them? Another integral point of the debriefing is how we present this information to the player in a way that allows reflection on past actions. In this section we will analyze this problem for the specific case of BullyBook and make a first step in the direction of a generic debriefing process.

3.1.2 Main Requirements

Giving insight to the player on his actions is not an easy task when it comes to games with a nonlinear gameplay, such as BullyBook. These kinds of games don't have a fixed scenario where the user is asked to perform a set of actions to accomplish a predefined purpose. In BullyBook the objectives can be reached in a large number of ways. Each action can cause an interaction involving other NPCs and eventually lead to a step forward or backwards towards the objectives. Non-linear games are often confused with what Squire Squire (2008) calls Open-Ended Games. The latter is a type of game where the player is free to choose his actions, but not necessarily in order to accomplish an objective. In Open-Ended games the player can play for hours without the notions of success or failure ever being of importance.

Designing a system that gives a debriefing to the player, must involve the key elements in the game. Only describing the underlying model(s), used by the game to drive the actions, success and failure, is not a sufficient approach as it does not give any indication on how the actions of the player have affected the outcome of the game. If the outcome is not related to the actions of the player, it may be hard for the player to understand the debriefing. It is comparable to telling a student that the solution he provided for an exercise is wrong because he did not follow the theory. Most students will still not understand what exactly they did wrong. The teacher should indicate precisely which steps in the solution are wrong and why. Therefore the player's actions need to be captured and used in the debriefing. Depending on the debriefing scenario, the information given to the player should provide sufficient information to allow the player to answer the following questions:

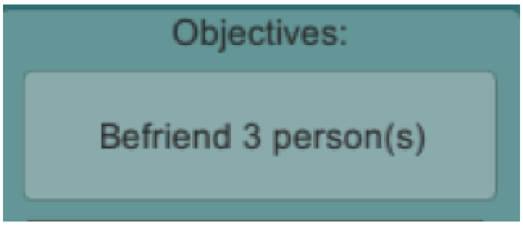
- Why did I win/lose?
- How did my actions affect the outcome of the game?
- What interactions could I have performed to improve my results?

These requirements are all related to the nature of the game, and place the interactions among characters and player(s) at the center. In case of BullyBook, the main objective is to raise awareness about cyberbullying. Therefore it should not only be evident to the player why he lost, but also what does losing mean when he/she would apply the same strategy in real life. In other words the goal of the debriefing for a serious game is not to make sure that the player will play the game better next time (by providing better answers or performing better actions), but that the findings made in the course of the game will be reflected in the player's real life behavior. In case of BullyBook, this would mean: affecting the perception the player has about cyberbullying and provide him insight into effects of cyberbullying related behavior.

3.1.3 Capturing relevant data

During the course of the BullyBook game, the player is confronted with many different elements that contribute in the success or failure of the predetermined goals. These elements take varied forms such as the time, moods, personality, interactions between the player and a NPC (Non-Player Character), or interactions between NPCs. Not only the freedom given to the player in choosing his actions but also the autonomous behavior of NPCs result in an open scenario where various outcomes can occur. As these different elements may influence success or failure in a complex way, finding the appropriate debriefing strategy (not too difficult to understand but powerful enough to realize the objectives of debriefing) may be tricky. In the following, we describe a first attempt of capturing the essence of the learning process in distinct variables. This approach will be implemented (Chapter 4) and evaluated (Chapter 5). The results of the evaluation will allow us to revise our debriefing strategy (future work).

First of all let us concentrate on the static elements of the game that may impact the game. Other than the player's attributes, one important aspect of BullyBook is the list of friends or NPCs (figure 3.2), and characteristics associated with these friends. This will certainly need to be logged in order to proceed with a debriefing.



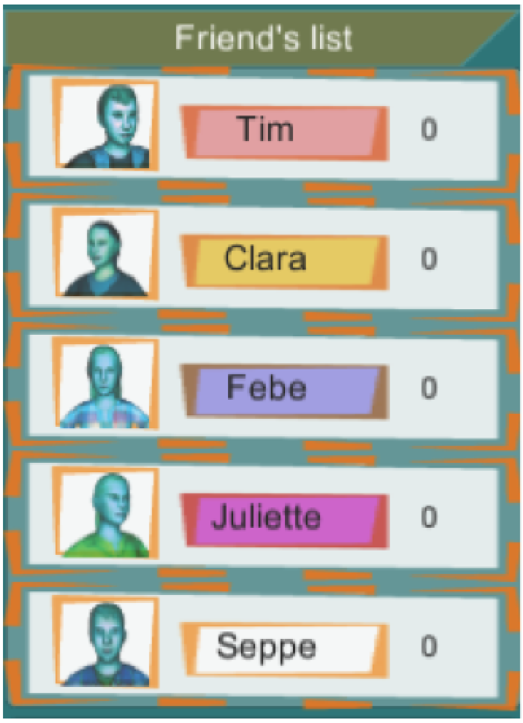
Objectives:

Befriend 3 person(s)

Figure 3.1: The excplicit objectives in BullyBook

Also the explicit objective of the level (figure 3.2) together with the player's progress towards this objective will be captured.

Some elements are however variable, such as the time (figure 3.3). As there is a limited time to accomplish the level, capturing this information can also be beneficial to the debriefing.



Friend's list

Tim	0
Clara	0
Febe	0
Juliette	0
Seppe	0

Figure 3.2: List of characters in BullyBook

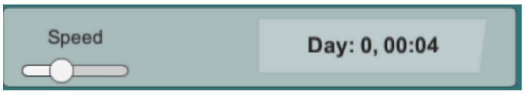


Figure 3.3: The time of play in BullyBook

The key elements in any social game are the interactions between members. First of all we have the interactions between NPCs as illustrated in figure 3.4. Although the player does not have any impact in this occurring, these interactions might however have an influence on the NPCs moods as well as give an opportunity for the player to intervene, and will therefore be captured as well. Interactions between the player and the NPCs will also be saved to be used in the debriefing. As the interactions are a key structure in BullyBook, the logging of data will be centered on them.

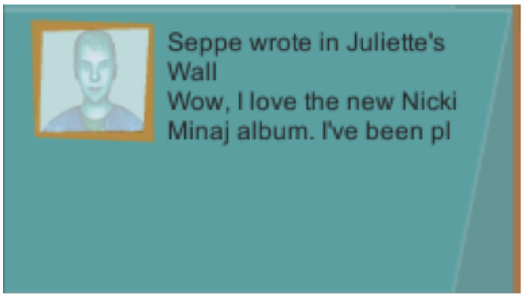

Seppe wrote in Juliette's Wall
Wow, I love the new Nicki Minaj album. I've been pl

Figure 3.4: Interaction between NPCs in BullyBook

The use of game states as described in (Harpstead, Myers, & Aleven, 2013) can be used for this purpose. Every time an interaction takes place a game state will be logged containing the interaction type and content, the time of action, the values for the personality characteristics of the friends and the player.

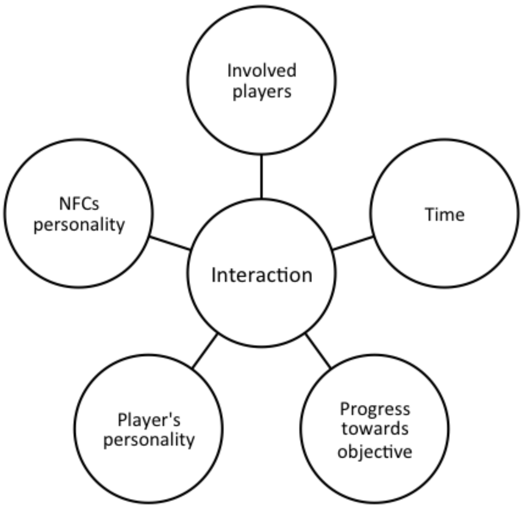


Figure 3.5: Graphical representation of a game state

3.1.4 Representing the data to induce learning

Having relevant data about a player's interactions is one thing, but finding a way to represent this data for a reflection process is a different challenge. The ultimate goal is to have players gain awareness about cyberbullying by being confronted to realistic settings where cyberbullying might take place and observing the effects of certain behavior. So the explicit game objectives given to the player (e.g. to befriend 3 people in level 1) are not the ones that really matter in BullyBook. The primary objective is to understand how cyberbullying might take form in daily conversations on social networks. To accomplish this objective, visualizations are appropriate as they allow to show a large number of information in a compact and readable way. The other advantage of visualizations is that the player can interact with them, and that would make the debriefing an active process. Three different conceptual visualizations that could be used in this first step towards debriefing will be presented. These visualizations each focus on different aspects and center the data around one variable of the game. The purpose of using these three visualizations is to see what elements in the game best capture the attention of the player.

Time-Oriented debriefing

The time-oriented reflection focuses on the interactions from the players given their time of occurrence. The standard way to do this is to have a time line of events, which the user can click to gain information about an interaction at a specific moment. This would be complemented by a content rich description of the interaction and its effect on the values of the different personality characteristics.

In addition, a replay button could allow the player to return to that specific moment in time and to replay that situation after having received the insights from the debriefing. The main advantage of this representation would be its linearity and ability to perceive effects over time. The major drawback of this approach would be that it would not permit to see the evolution of the characters' values and how they reacted with each other or with the player.

Character-Oriented debriefing

The second option puts the focus on the characters, not only between the main player and NPCs but also among NPCs. A way to represent this would be to have a geometric figure with at each extremity a character. The relationships between characters could be selected, or more information about a specific character could be viewed. A way to could be achieved is to use a circle that would be used to represent the different interactions that took place, the system would also allow the possibility to replay the game before this particular interaction in order to improve the outcome. This visualization resembles the underlying structure of the game states, which also revolve around the interactions. Having the time of action integrated in the visualizations also helps structuring the events. However this visualization has the same disadvantage as the previous visualization when it comes to having a clear overview of how the interactions affected the characters.

Interaction-Oriented debriefing

Finally the last debriefing approach is interaction-oriented and puts the interactions between players at the core of the visualization. The players could be linked together every time an interaction occurs between them, using the same representation as in previous visualization. This visualization would look similar to the previous one but the effect of interactions on characters would be more visible. This visualization is the most complete one but might cause an overload of information.

Textual indications and replay possibility

As mentioned in (Fanning & Gaba, 2007) the use of mixed-media modalities presents advantages when we come to a post-experience analysis. Therefore it would be interesting to add some textual feedback to the debriefing to give the player a direct explanation of the outcome. This could be done as in (Johnson, 1994) where a questionnaire is provided to ask simple questions to the system regarding the outcome and the decision-making of agents in the game. Another interesting addition to the debriefing would be to allow users to replay certain moments in the game. As presented in ((Harpstead et al., 2013) , this functionality offers two advantages: firstly it allows the user to understand the outcome more easily by returning to the exact same situation and judge the effects by himself, and secondly it allows us to evaluate the user to see if the prior debriefing proved to be helpful in light of his actions.

3.1.5 Conclusion

In this chapter we looked at how the findings made during the investigation of related work could be used in the conception of a debriefing strategy. The purpose of the debriefing strategy is to make sure the player can view not only what happened during the game, but also to have a better understanding of why these actions resulted in a positive or negative outcome towards the objectives of the game. To create a system that would allow such a reflection on the game, elements of BullyBook have to be captured in what we called game states. These game states save the information that played an important role in the result of the game. Next we discussed the methods to represent this information to the player. A textual feedback is one part of this representation, the other is a visualization that would focus on a specific element of the game model. In total three visualizations were discussed, focusing on the time, characters and interactions. Finally the replay functionality that allows the player to replay certain actions in the game could also be integrated in the debriefing strategy.

3.2 Design

3.2.1 Introduction

In the previous chapters a first idea was given of what information would be shown to allow reflection on past actions in the game. The way this information would be presented is also discussed and three approaches have emerged from this. In this chapter we provide the design of the Debriefing System. This will be discussed in different steps. First a system context diagram presenting an overview of the system will be given. Next, followed by lower level flow representations. In order to apply reasoning and allow reusability of this debriefing strategy, the ontology and reasoning applied in this context will be described. Finally the prototypes of the debriefing modules will end this phase and allow us to move to the implementation process.

3.2.2 Context Diagram

A first step towards the development of a new system is to create a System Context Diagram. Such a diagram provides an overview of the system, the boundary between the system and the environment and the interactions between the system and the environment. It also has the advantage of simplicity and the possibility to decompose parts of the diagram to show more details. This approach is also consistent with the Systems Development Life Cycle's incremental commitment philosophy (Kay, 2002) . This strategy consists of waiting to make technological decisions until the system's architecture is fitted for the functional requirements prescribed in prior stages.

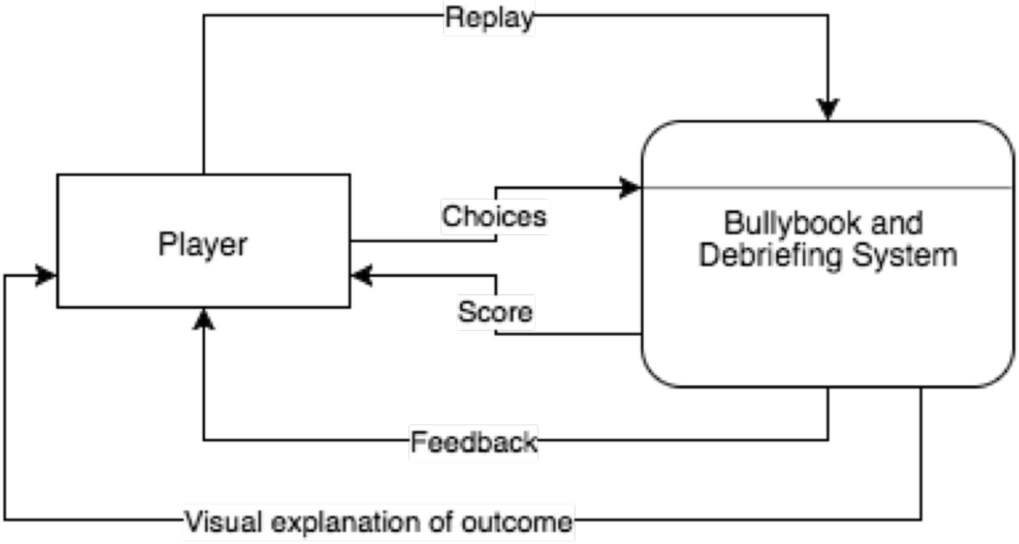

Replay
Player
Choices
Score
Bullybook and Debriefing System
Feedback
Visual explanation of outcome

Figure 3.6: Context Diagram of the Debriefing System

Figure 3.6 shows the System Context Diagram for our complete system consisting of BullyBook and the Debriefing System, using Gane and Garson symbols (Gane & Sarson, 1979) for Data Flow Diagrams. In this picture, the user of the system is considered an external entity (or source/sink) and interacts with the system. There are 5 main interactions possible between the player (or user) and the Debriefing System:

- The player makes choices in the course of the game
- The system returns the score or outcome of the game
- The system gives a textual feedback to the player about his actions
- The system presents a visual explanation of user actions
- The player can replay certain parts of the game

3.2.3 Data Flow diagrams

This big picture of the system can be decomposed in Data Flow Diagrams of different levels to iteratively include more details about the structure of the system. The different levels of the Data Flow Diagrams respect the interactions with the system described in the Context Diagram.

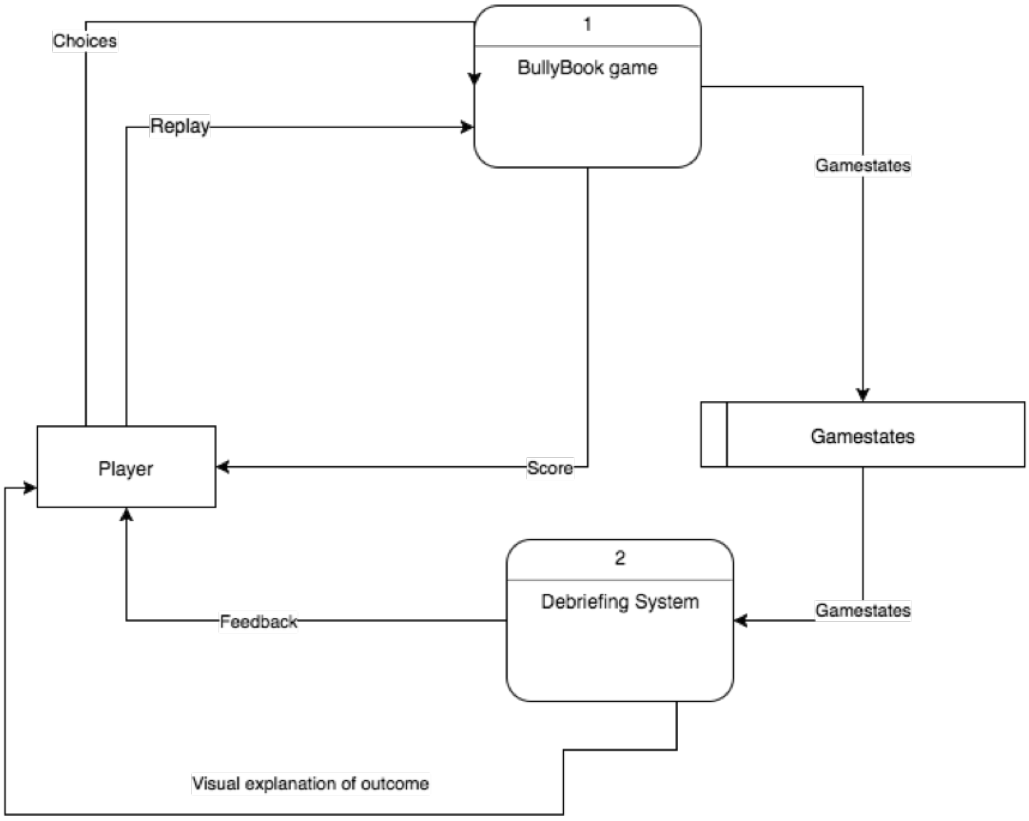


Figure 3.7: Level 0 of the Data Flow Diagram for the Debriefing System

Figure 3.7 shows the level 0 Data Flow Diagram of the system. While the interactions are kept, the system is now composed out of two processes: the BullyBook game and the Debriefing System. The first notable fact is that the game is included in the whole System. This has been decided to include the replay strategy as part of the complete Debriefing process. To allow this the game has to be modified to include this functionality. Another new element in this Data Flow Diagram is the Game states data store. The idea of keeping track of the game states is described in the previous section and captures the important elements of the game. These states are stored and exchanged between the game and the debriefing module. They are used to perform a data-based debriefing and allow to return a textual feedback and visual explanations to the user.

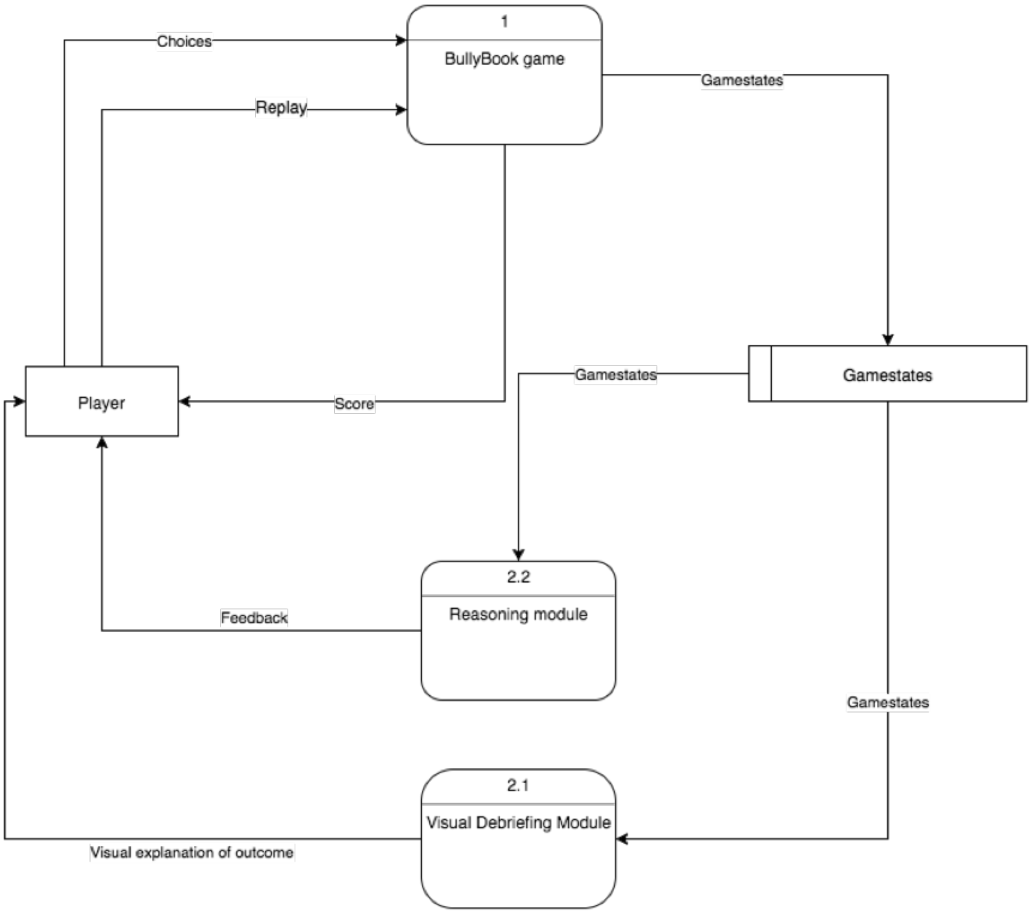


Figure 3.8: Level 1 of the Data Flow Diagram for the Debriefing System

The Debriefing System has further been decomposed in figure 3.8. It shows that the Debriefing System has been decomposed into 2 modules: the Reasoning module and the Visual Debriefing module. The Reasoning module will mainly be responsible for gathering and inferring information about the game states that could later be used to give a textual feedback to the player. More details about this module will be given later in this chapter. The Visual Debriefing module will focus on the visual aspects of the debriefing, its task involve: selecting game data and filtering the irrelevant information, providing the information in a visual way and allow the user to interact with the visualizations so that more details could be obtained, if required.

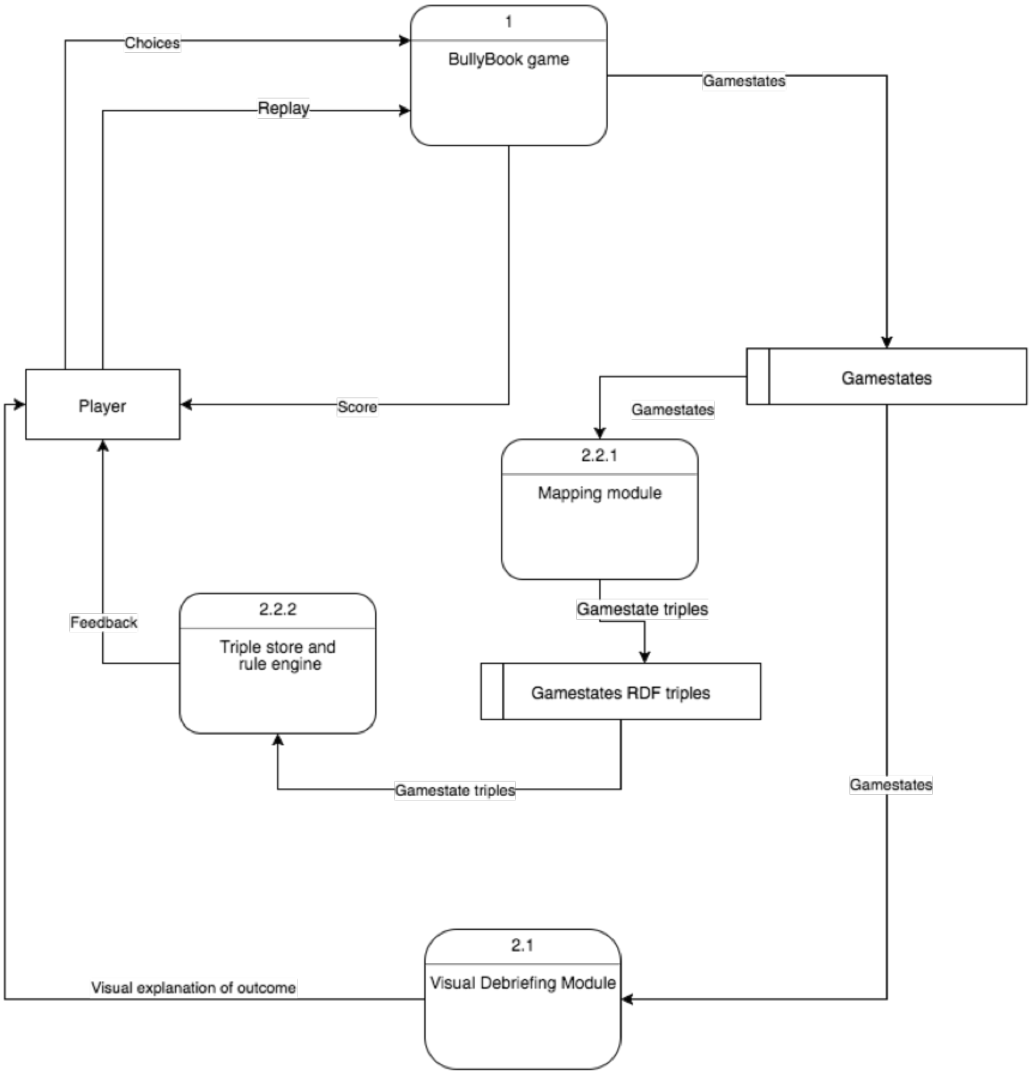


Figure 3.9: Level 2 of the Data Flow Diagram for the Debriefing System

In the Data Flow Diagram of level 2, the Reasoning module has been decomposed into a Mapping module and a triplestore with rule engine. The reason for this is to enforce the presence of RDF triples. These triples are used to be applied with an existing reasoner and to generate an appropriate feedback concerning the game states.

3.2.4 Ontology and Reasoning

Although we are developing a debriefing for the BullyBook serious game, we are aiming for a generic debriefing strategy towards automatic debriefing of Serious Games. Implementing a solution that is exclusively designed for BullyBook would make little sense in this context. Therefore an ontology has been created to allow reusability of some key concepts used for the Debriefing System of BullyBook. One of these key concepts is the game state, which is common to most games. Using our ontology approach permits other similar serious games to apply the same approach as we did in this work.

The graphical illustration of the Debriefing Ontology in figure X was created using W3C recommendations for the diagrams (Reynolds, 2014) . The complete description of the Ontology has also been added to this thesis. The Ontology focuses on the game states to allow easy manipulation of key elements for the Debriefing. The purpose of the ontology is to make a structured representation of important elements in the game.

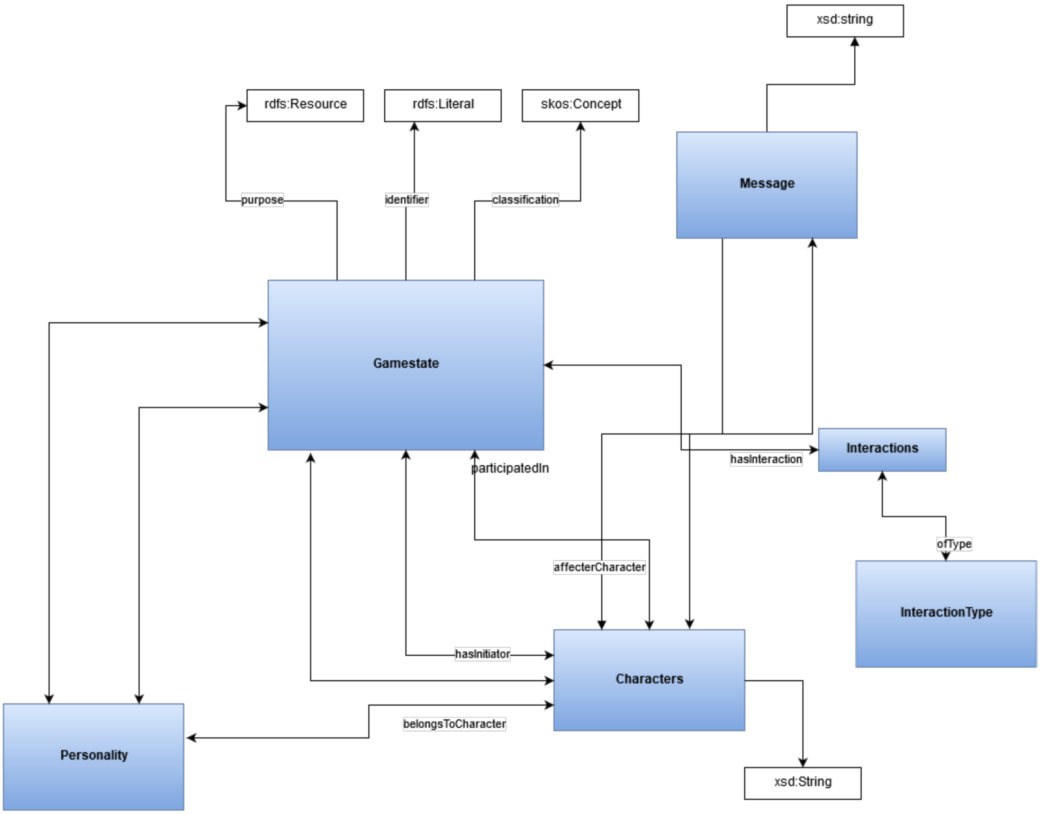


Figure 3.10: Debriefing Ontology

Next to offering a generic solution, the use of RDF triples also allows us to use rules in order to reason over the data we have at our disposal in a scalable way. This could be used to infer information about past actions and generate an automatic feedback that would go beyond simple fact stating. With the current implementation of BullyBook, using rules to infer information does not present an evident advantage compared to the more classical imperative approach. But using an ontology allows us to validate our approach by applying it to other games. One way rules could be used in a nonlinear gameplay, is to produce another sequence of actions that could have resulted in a better outcome that the ones chosen. This is particularly helpful when suggesting a better action to a player.

3.2.5 Wireframes

In the analysis three approaches were discussed to apply a debriefing after the game. Each of these strategies focuses on a specific element that is relevant to the gameś outcome. In this section wireframes were designed for each of the 3 debriefing approaches. These prototype wireframes could be inspected for their usability and easily be adapted. In what follows we present the prototype wireframes for the different debriefing approaches proposed.

Time-Oriented Visualization (TOV)

The time-oriented reflection puts the focus on the time of interaction. Therefore a timeline is displayed on top of the screen with the times of occurrence of the interactions (figure 3.11 and 3.12). Upon clicking one of these time slots, a more detailed description of the interaction is given, with firstly the content of the interaction. For BullyBook, this interaction can be a dialog or a ?like? action. If the action is a like or a reaction, the original post is also shown. A textual feedback is also given to the player about his action, together with a representation of the change in the values of the characteristics of the targeted Non-Player Characters (which can also be done using graphs). Finally a suggestion will be made to the player when he clicks the “improve action” button, giving the best action he could have performed in this situation with the given selection of interactions (see figure 3.12).

The TOV presents the main advantage of taking the time of occurrence of an action into account, which allows one to see the impact of an action on the evolution of the personality models over time. It is also the model that mostly resembles the flow of the game, but this linearity comes with the disadvantage of providing little information about the overall relationships between the interactions.

Character-Oriented Visualization COV

The second visualization (figure 3.13) places the characters at the centre of attention. The idea is to present the relationships between the characters in a clear way. This allows us to have an overall picture of the interactions that occurred between characters, however without providing details about the relationships and time aspects between interactions.

To accomplish this a geometric figure with a character in each corner is represented and lines are drawn between them if they had an interaction in the course of the game. An icon with a colour code is used for this purpose. Good interactions will have a green background, neutral interactions will be blue and finally bad interactions will be red. These interactions are clickable

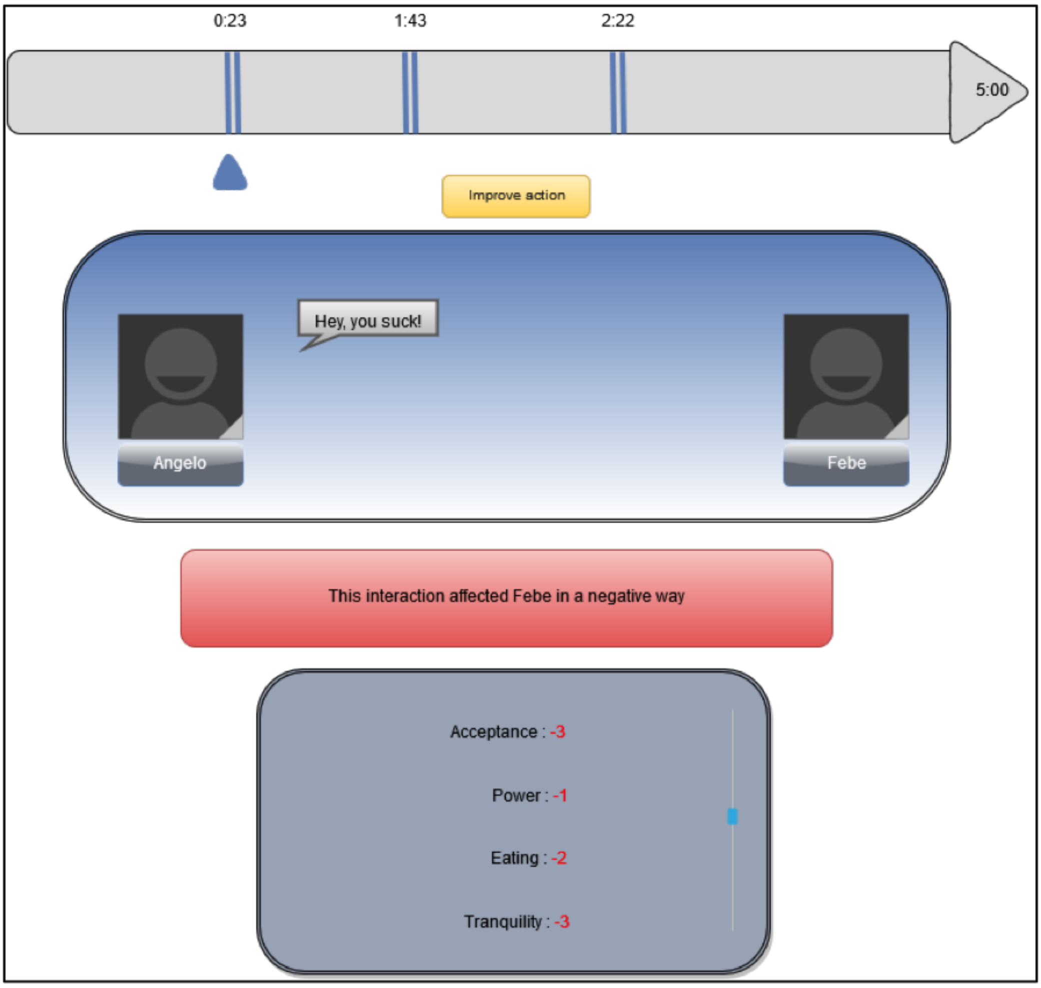


Figure 3.11: Wireframe for Time-Oriented Visualization prototype showing a good interaction

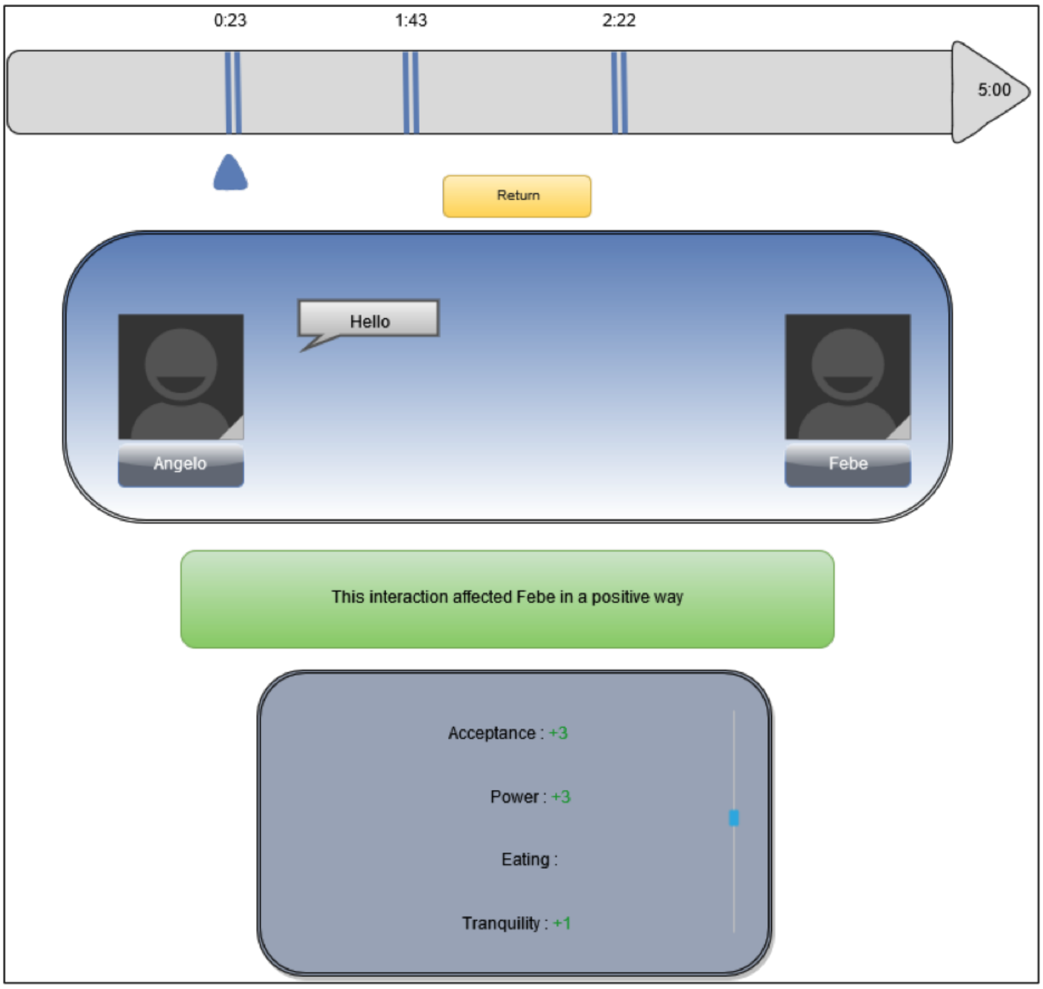


Figure 3.12: Wireframe for Time-Oriented Visualization prototype showing a bad interaction

and similarly to the time-oriented visualization a more detailed description of the interaction is shown. The main drawback of this visualization, like for the TOV, is the lack of information on the relationships between the interactions.

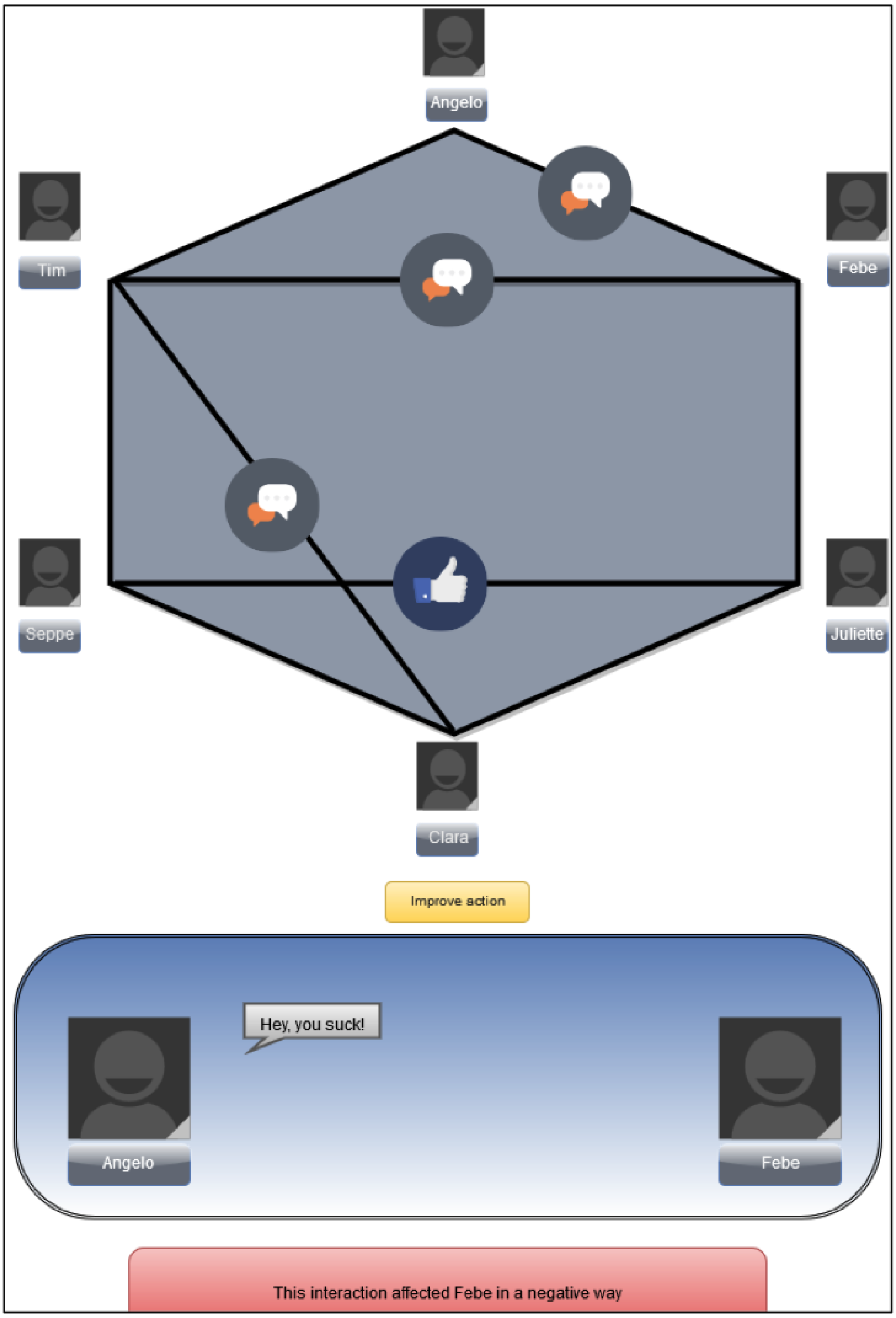


Figure 3.13: Wireframe for Character-Oriented Visualization prototype

Interaction-Oriented Visualization (IOV)

In the last visualization the interactions are the most important element. For every interaction that took place in the game a line is drawn between the characters involved containing an icon using the same colour code as in the COV. If the post is a reaction then the icon is put under the original icon, the same way as done in the game. This results in a graph where the characters are the nodes and the interactions are the edges.

The main disadvantage of the previous visualization is here corrected by having a nested structure for the interactions, which allows one to see all the characters involved as well as their contribution to the personality characteristics. However this correction comes at the price of complexity of the visualization and might be problematic when the number of interaction is high.

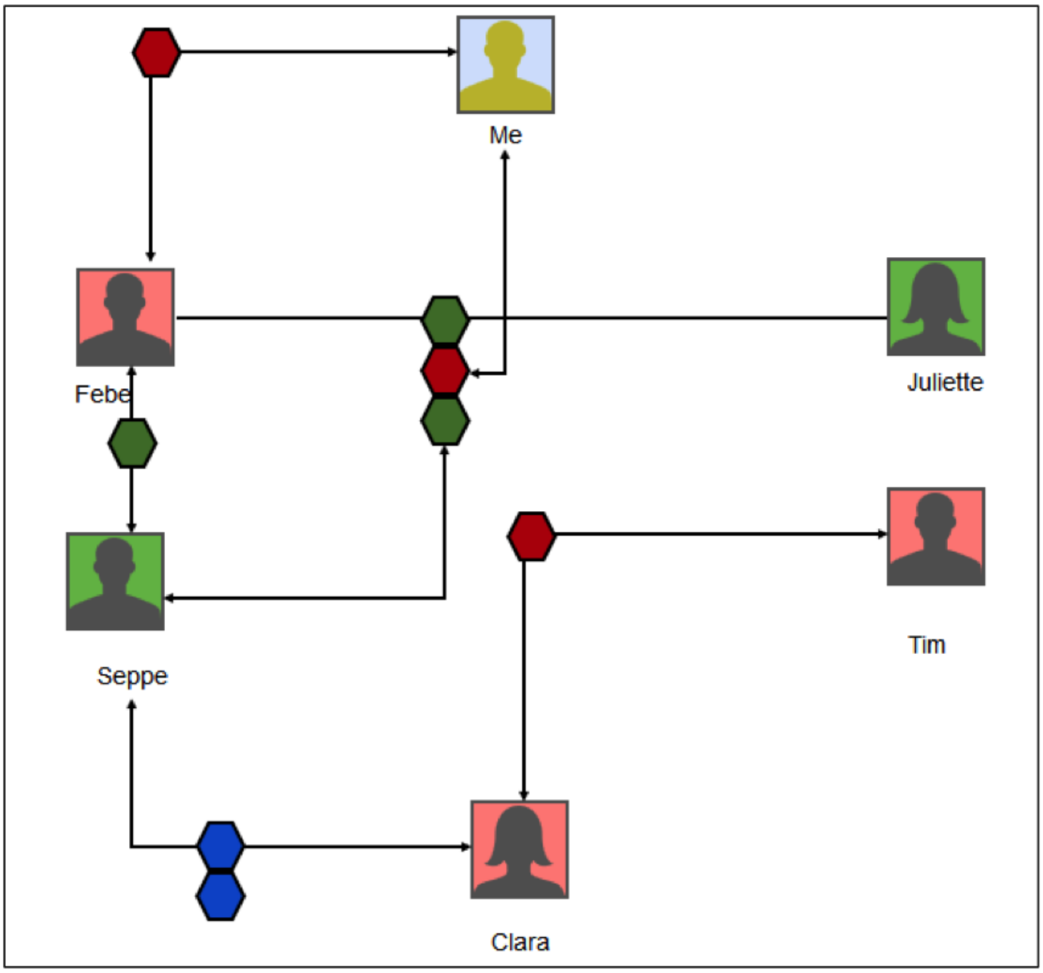


Figure 3.14: Wireframe for Interaction-Oriented Visualization prototype

3.2.6 Conclusion

In this chapter we looked at the design of our Debriefing System. We started by using a context diagram to show an overview of the system, the environment and the interactions between them. Then we deconstructed our system in layers using Data Flow Diagrams. This allowed us to represent all the components present in our system : the BullyBook game, the mapping module, the triplestore with rule engine and the visual debriefing module. This was followed by the Debriefing Ontology that we defined for the purpose of this game, as well as for reusability in other games. Finally three wireframes were shown for each of the 3 debriefing modules discussed in the analysis (chapter 3.1).

4
Implementation

The previous chapters outlined our debriefing strategy and allowed us to design a first solution. In what follow we explain how this design is implemented using different artefacts. Firstly the architecture of our solution will be presented using a technical schema containing the flow of data between each module. Then the database structure will be given, followed by the debriefing module and the way it works. Finally, some screenshots of our implementation will be given.

4.1 Technical architecture

While figure 3.9 depicts the overall flow of data in the debriefing process at a high level, figure 4.1 presents how this data flow will be realized in the implementation.

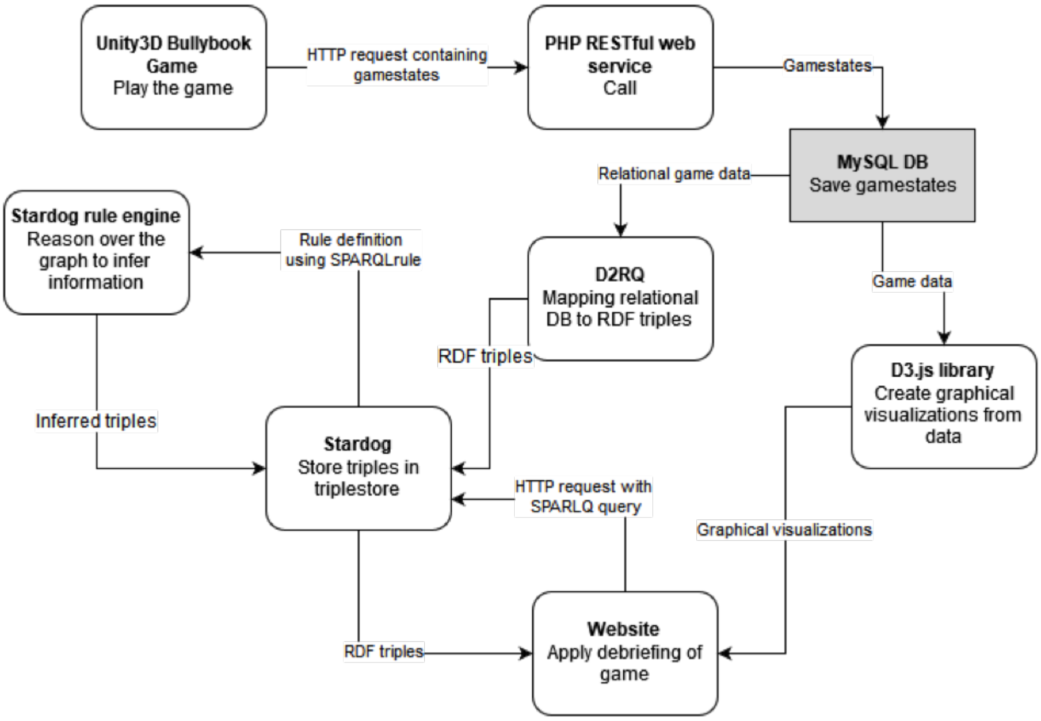


Figure 4.1: Technical schema of our implementation, containing the data flow between modules

The starting point of the Data Flow Diagram is the BullyBook game that is developed using Unity3D and scripted with C# . In the course of the game different elements need to be stored in order to allow for the debriefing in a later stage. The data is stored in a relational database. The choice was made to use a relational database because of its simplicity of use with the visualizations. The goal is to eventually replace the relational database with the triplestore. The structure of this database is provided in the next section. To store the data a PHP web service is called using a RESTful API. The HTTP request sent to the web service is then executed to store the data in the MySQL database. In order to represent the data in a graph-like structure and to reason over it in an efficient way, the relational instances are converted into RDF triples by populating an ontology with a mapping from relational databases to RDF. In the debriefing of the game we would for example be able to determine which dialogue would have had the best outcome in the decision tree using simple rule definitions. The generated triples are stored in a Stardog triplestore (*Stardog*, 2016) , which allows one to convert relational database to RDF triples. However this feature is not supported in the community edition of Stardog, therefore another mapping tool, D2RQ mapping tool, is used. One of the main advantages of Stardog is that it has its own rule-engine, and the possibility to have user-defined rules. To do so, SPARQL rules can be created and imported in Stardog. The next step in the debriefing process is supported in the form of a web application, where the user can interact with the system to view his results and progress. The debriefing model will be described in chapter 4.4, but evidently this is where all the previously mentioned mechanisms come together. The original triples as well as the inferred information can be queried using the HTTP API provided by Stardog. This data complemented with the D3.js visualizations will form the eventual debriefing.

4.2 Database structure

The choice was made to work with a relational database as well as with a triplestore for this prototype for the sake of simplicity and time-consumption. In the future work of this project this implementation using two data stores should be removed and a fully RDF implementation should be accomplished.

The database structure captures the essential elements of BullyBook and leaves the door open for future additions to the game. For instance, n the current implementation of BullyBook moods of players are not yet taken into consideration, but when this will be the case, this would not require a major change in the database.

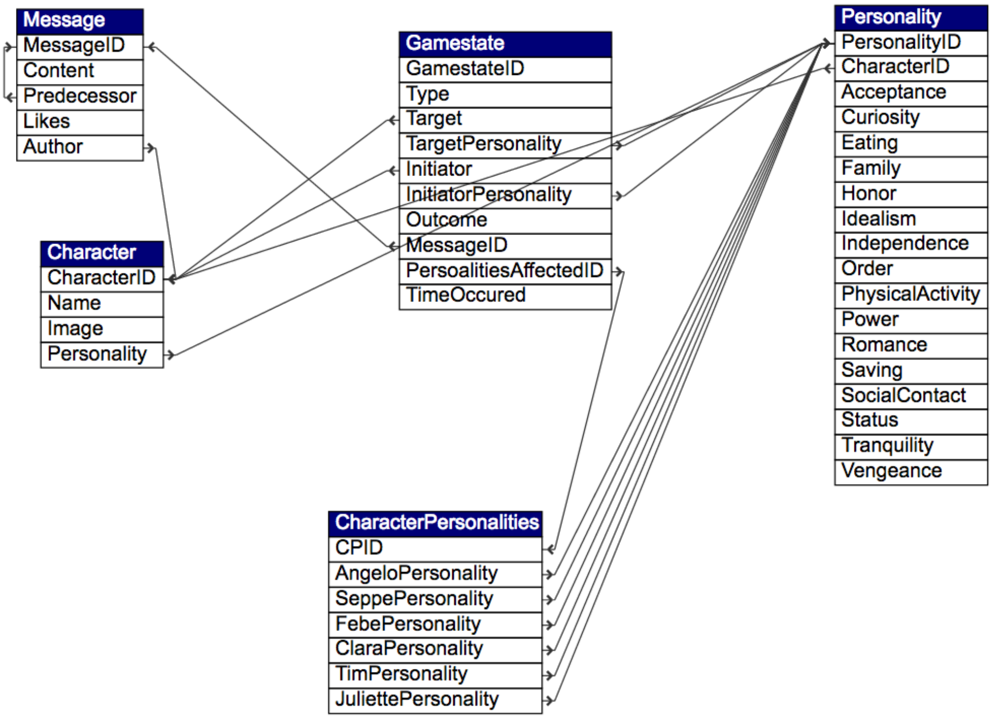


Figure 4.2: Database structure of our Debriefing System

The main table of the database is the one containing the game states, i.e. Gamestate table. As described in the analysis this concept of game state captures the important elements of the game. The Personality table contains a value for each of the 16 attributes described in Reiss' personality model. The Message table contains the content of the posts made, as well as a reference to a possible preceding message. The number of likes, as well as the author, are kept in this table. The Character table contains the main information available about the characters in BullyBook, which also includes a reference to their initial personality. The CharacterPersonalities table stores the personalities of each character at a given moment in time; because at each significant event in the game the personality of the characters can be affected.

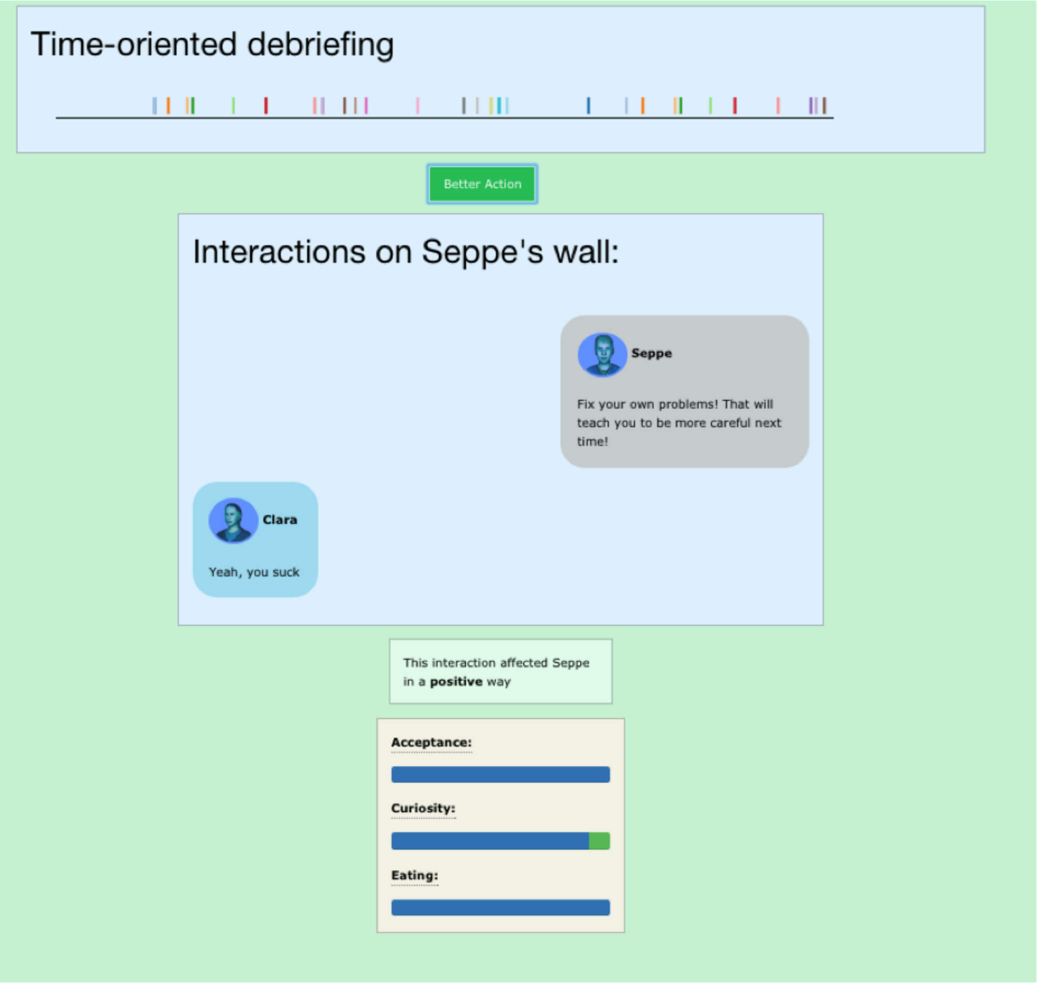
4.3 Reasoning

The reasoning module is used in the Debriefing System to infer new information about the game states. As the current version of the game only has one level playable and lacks the integration of key concepts in the iATTAC framework (such as the location, moods,...) , the applied reasoning in the Debriefing System has a limited range. The main aspect explored with the reasoning is the suggestion of a better message given a specific message and context. Stardog allows us to use it's own reasoning engine, that permits the creating of rules using Stardog rules, that use SPARQL extended with new bits such as (If, then,..) . Applications of reasoning have been used in the code, such as simply retrieving messages that belong to a same tree of messages. In what follows an example of rule used to retrieve a better interaction, is shown.

```
 [] a rule:SPARQLRule ;
 rule:content """
   PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>
   PREFIX gr: <http://purl.org/goodrelations/v1#>
   PREFIX :<urn:test:>
PREFIX bb: <file:///Users/Metzo/Documents/Thesis/UnityProject/d2rq-0.8.1/vo
  IF {
     ?original a bb:Message.
     ?improved a bb:Ritual.
      ?interactionOriginal bb:hasMessage ?original.
     ?improved bb:hasInteraction ? interactionOriginal.
     ?improved bb:hasOutcome ?outcomeImproved.
     ?interactionOriginal bb:hasOutcome ?outcomeOriginal.
     Filter (?outcomeOriginal > ?outcomeImproved).
  }
  THEN {
     ?original bb:betterInteraction ?improved.
  } """.
```

4.4 Debriefing System

The debriefing module is divided in 3 modules: the time-oriented debriefing, the character-oriented debriefing and the interaction-oriented debriefing. The differences between these debriefing strategies have been explained in chapter 3. However, they have in common that they describe what occurred in the game. This common part is described in the next section. Figure 4.3 shows an example of a debriefing (in this case a Time-Oriented Debriefing). It is composed of different parts, which will be described further down.



Time-oriented debriefing

Interactions on Seppe's wall:

Seppe

Fix your own problems! That will teach you to be more careful next time!

Clara

Yeah, you suck

This interaction affected Seppe in a **positive** way

Acceptance:

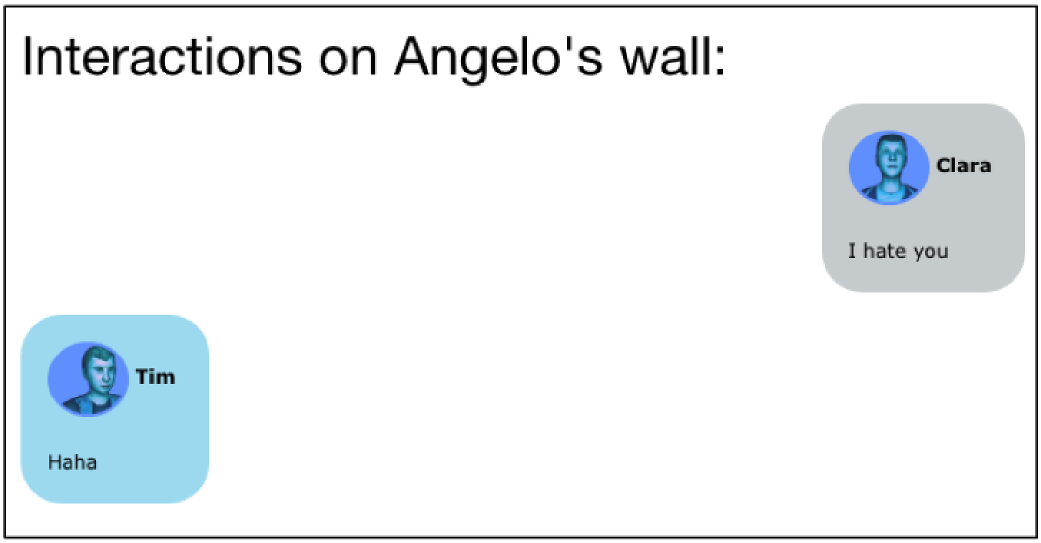
Curiosity:

Eating:

Figure 4.3: The Time-Oriented module of the Debriefing System

4.4.1 Common Elements in the Debriefing Modules

The debriefing is designed to allow reflection on the game actions and their effect on the outcome of the game. In the case of BullyBook these game actions take the form of posted and liked messages, called interactions. Therefore the debriefing focuses on these interactions and tries to present as much information as possible about them to the user in an understandable way.



Interactions on Angelo's wall:

Clara

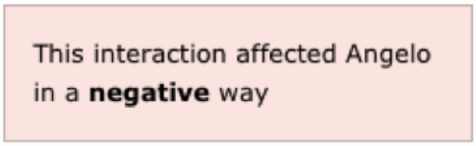
I hate you

Tim

Haha

Figure 4.4: Message wall of Angelo in the Debriefing System

Figure 4.4 shows an example of interaction that occurred between Tim and Clara. Different graphical elements are used in this panel. First the picture and name of both characters are displayed using the same images as the one used in the BullyBook game. Next a reproduction of the activity wall of the target character is shown; in this case it is Angelo's, who happens to be the main player of the game. The interaction depicted is a bullying scenario between Clara and Time. If the message is a reaction on some other message, then this initial message is shown on the right of the screen. The message, i.e. a reaction, on which this interaction screen focuses, is shown on the left in a dialog box with a blue background. In this particular example, Tim reacted to Clara who attempted to bully Angelo.



This interaction affected Angelo in a **negative** way

Figure 4.5: Textual feedback in the Debriefing System

Figure 4.5 shows the textual feedback that is given to the player. In this case we can see that the reply Tim posted after Clara's initial bullying message had a negative effect on Angelo's personality values.

In figure 4.6 we have a more explicit view on how this effect affected Angelo. The 16 values in Reiss' personality model are displayed together with how the selected interaction affected them. To highlight the evolution a color code is used: blue indicates the previous value, green indicates what was added by this interaction and red shows how much the value has decreased due to the interaction.

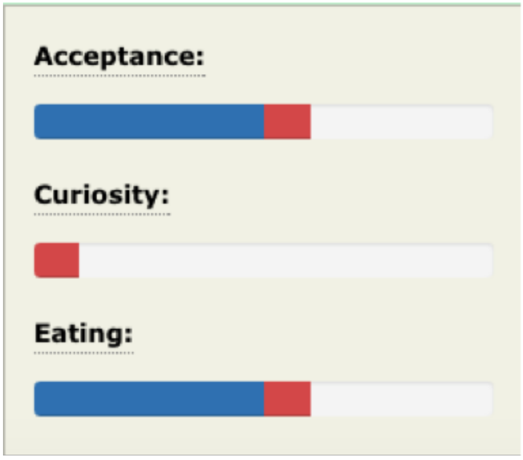
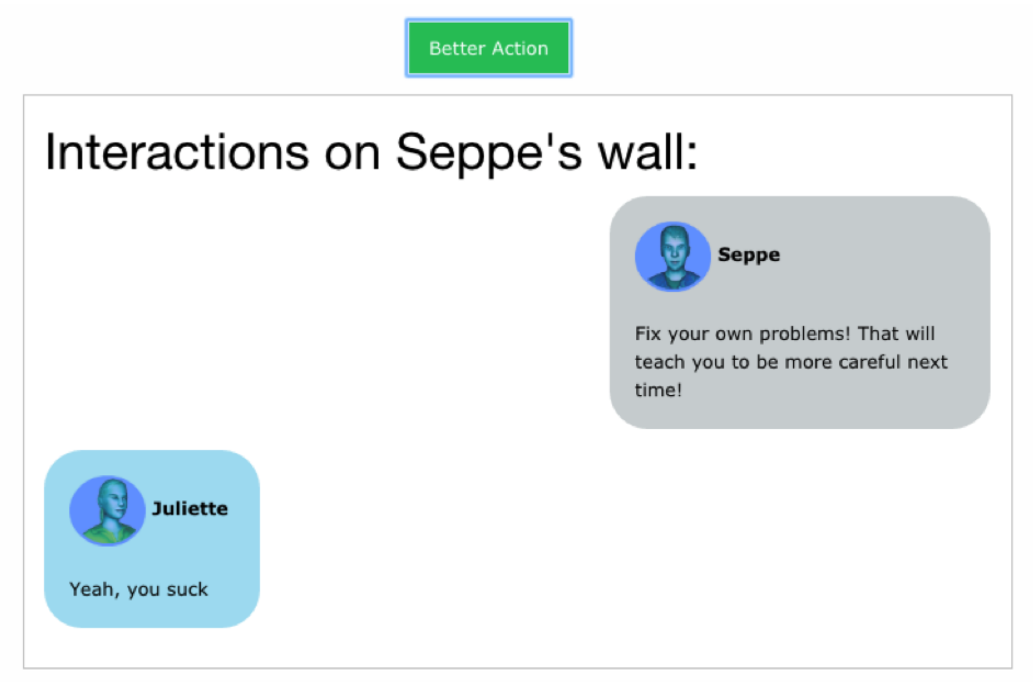


Figure 4.6: Evolution of personality values due to the selected interaction

When a reply that had a bad outcome is posted, an extra button is made visible in the interaction screen. This button shows a reply that would have had a better outcome if it was chosen instead. In this case we can see that that in figure 4.7 Juliette participates in the bullying, which has a bad impact on Seppe's personality values. If however she would have chosen the reply depicted in figure 4.8 Seppe's personality would have been affected positively.



Interactions on Seppe's wall:

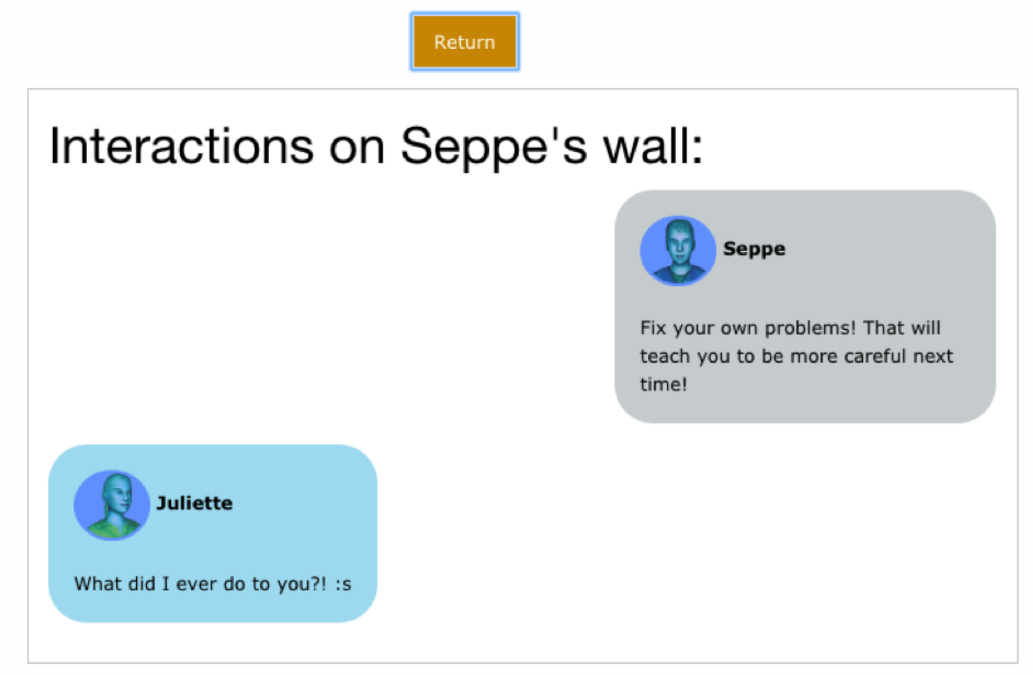
Seppe

Fix your own problems! That will teach you to be more careful next time!

Juliette

Yeah, you suck

Figure 4.7: Example of bad reply in an interaction in the Debriefing System.



Interactions on Seppe's wall:

Seppe

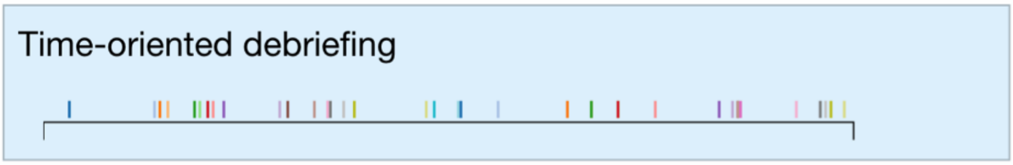
Fix your own problems! That will teach you to be more careful next time!

Juliette

What did I ever do to you?! :s

Figure 4.8: Example of a better interaction in the Debriefing System.

4.4.2 The Time-Oriented Debriefing



Time-oriented debriefing

Figure 4.9: The timeline in the Time-Oriented Debriefing

In the time-oriented visualization the most import element is the time of occurrence (figure 4.9). This was implemented using the timeline provided by D3.js (Bostock, 2015). This visualization is very simple to understand. It allows the user to select a time-slot (colored bar) to obtain more information about the interaction that occurred at that time. Upon selection of a time-slot, more information about the associated interaction will be displayed in the common panel.

4.4.3 The Character-Oriented Debriefing

The Character-Oriented Debriefing puts the focus on the character (figure 4.10) and tries to give a good overview of the relationships between characters during the game. This is done using a D3 chord diagram in the main screen. This visualization represents the relationships between characters dynamically, when hovering over a character name, only the relationships associated with that character are shown (FIGURE 4.11). Two interactions are possible with this main visualization. Either a relationship is selected between two characters, which will lead to a second screen where a list of all the interactions between these two characters is shown (4.12). The elements in this list are clickable and show more information about that particular interaction in the same fashion as described earlier. Secondly the name of the character can also be clicked, which leads to a screen where the interactions involving that characters are shown in a timeline (figure 4.13). Bellow the timeline the personality values of the character are shown for the selected time. This allows to see how the personality values evolved over time for a character. Each time an interaction is selected the interaction details are shown in the bellow panel.

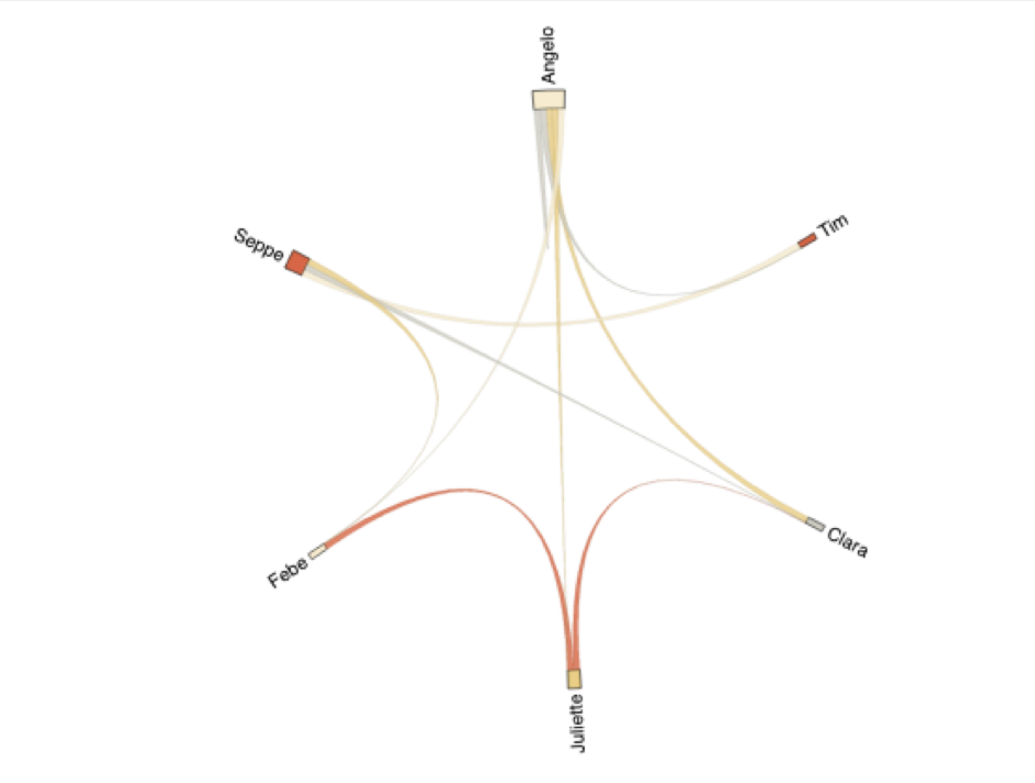


Figure 4.10: The main screen of the Character-Oriented Debriefing

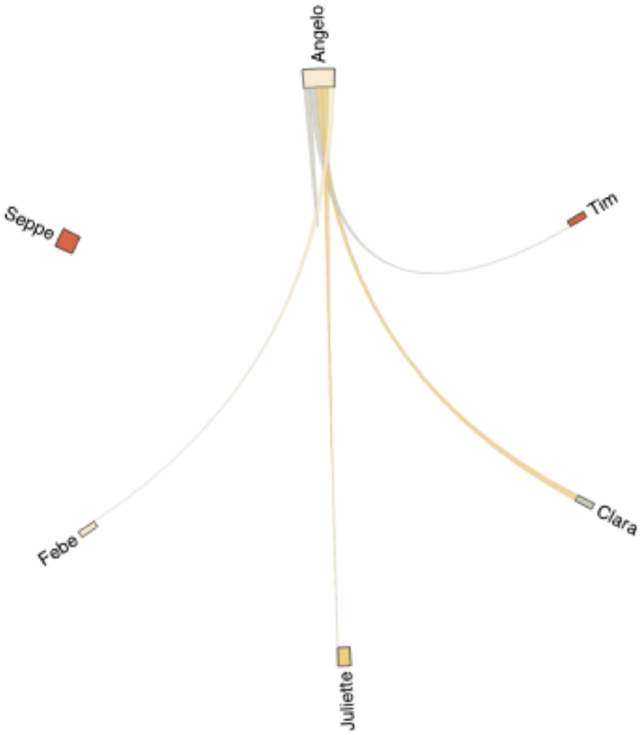
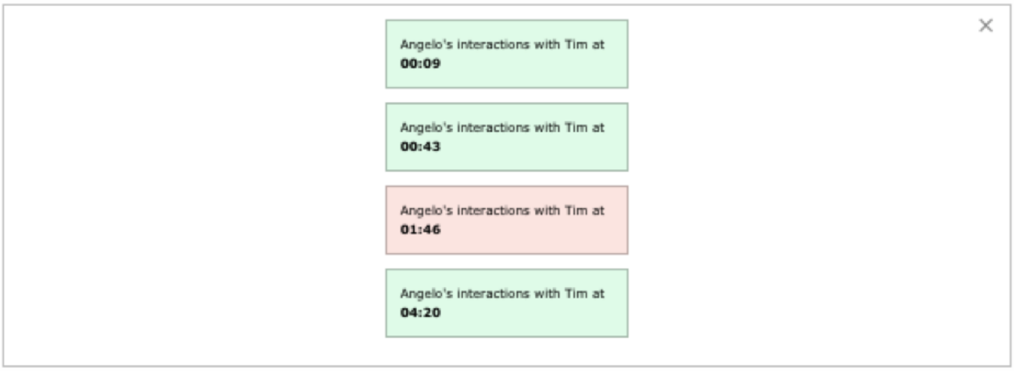


Figure 4.11: Hovering over a character shows the relationships of that character with other characters



Angelo's interactions with Tim at
00:09

Angelo's interactions with Tim at
00:43

Angelo's interactions with Tim at
01:46

Angelo's interactions with Tim at
04:20

Figure 4.12: List of interactions in the relationship between the 2 selected characters

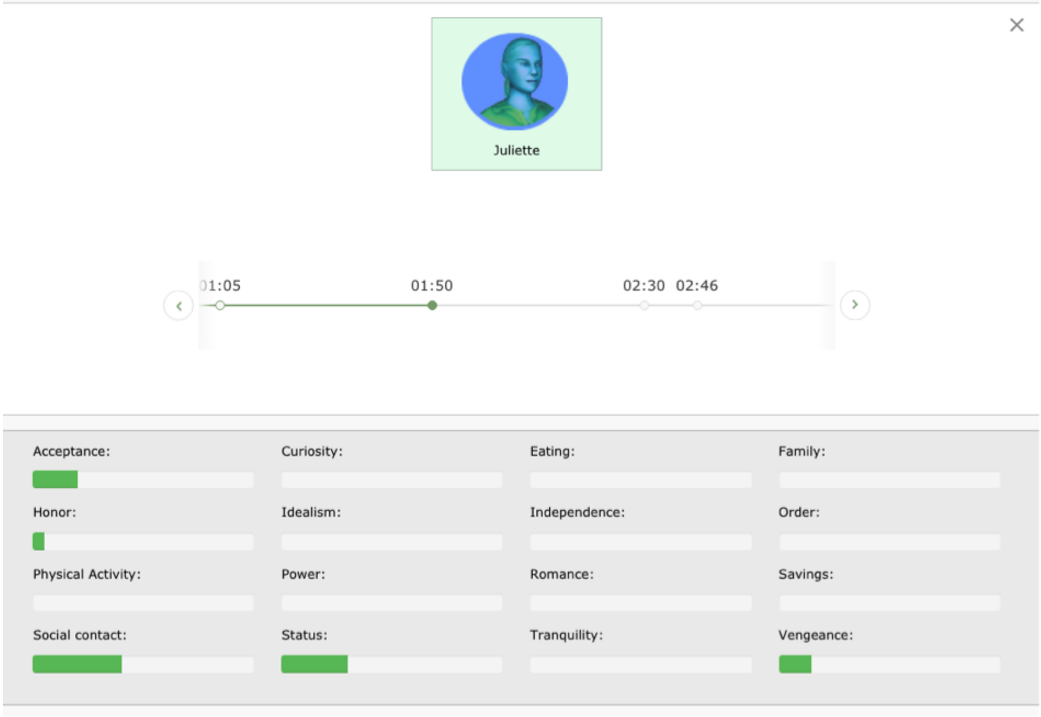
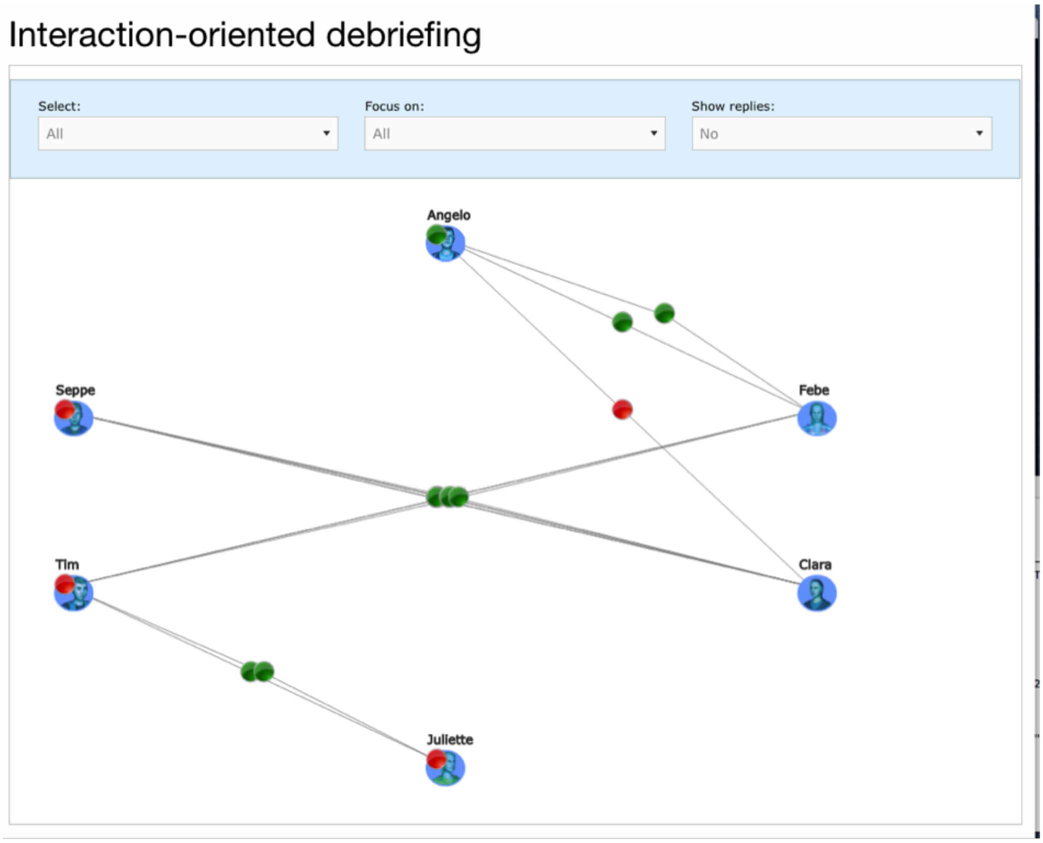


Figure 4.13: Selecting one character in the COD shows the interactions for that character over time.

4.4.4 The Interaction-Oriented Debriefing

The last module focuses on the interactions that took place in the game. The 6 characters of the game have a fixed location in the screen, and for each interaction between characters, a circle is added to the visualization and linked with the involved characters. The circle is colored red if it had a bad impact on the outcome of the game, and green if it had a good impact as shown in figure 4.14. The circles are clickable to get more information about the associated interaction. This information is shown bellow in the common panel. Because this visualization might be overcrowded with circles, filters were added. These filters allow to select only the good/bad interactions, to focus on one specific character or to hide the replies and only show the original posts. Figure 4.15 shows an example of how these filters can be used.



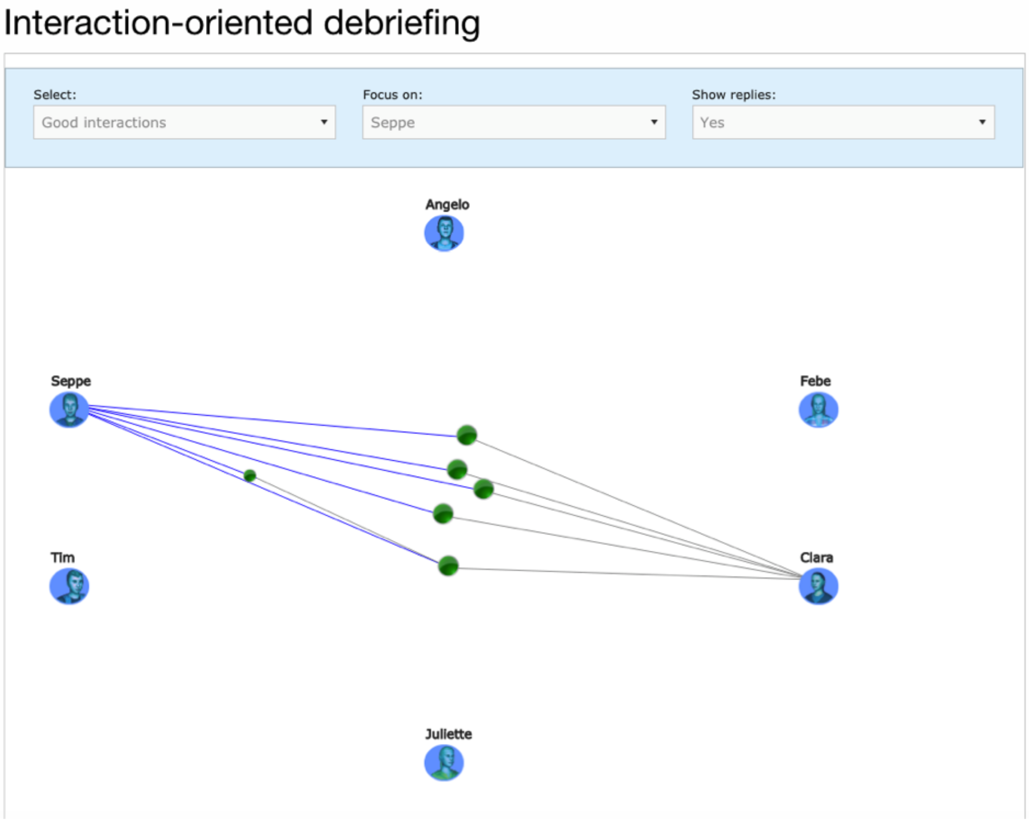
Interaction-oriented debriefing

Select: All

Focus on: All

Show replies: No

Figure 4.14: Interaction-Oriented Debriefing module



Interaction-oriented debriefing

Select: Good interactions

Focus on: Seppe

Show replies: Yes

Figure 4.15: Filtered Interaction-Oriented Debriefing module

4.5 Conclusion

In this chapter a first implementation of our Debriefing System was introduced. The implementation integrates two main components: the visualization and the feedback. For the visualization the D3.js library was used, first with a timeline for the Time-Oriented debriefing, then with a chord diagram to represent the interactions between the characters and finally to draw connections between the interactions and the involved characters in the interaction-oriented debriefing. All these 3 modules share a common part that contains the feedback about the interaction. In this part a message is shown to inform the user if the selected interaction had a good or bad outcome in the game, then an explanation of this is given using the personality values and how the selected interaction changed these values. In the case a reply that had a negative outcome is selected, then a button appears to suggest another reply that would have resulted in a better outcome in the game.

5 Evaluation and results

In this section the evaluation of our solution will be discussed. The specific aspects we want to assess as well as the method employed to accomplish this will be described. It was decided to opt for a pilot study in order to gain some first insights about how useful the debriefing system is. Pilot studies (Lazar, Feng, & Hochheiser, 2010) are often performed to evaluate the feasibility of a project, by having a small scale experiment that allow to get a first evaluation of the system.

The purpose of the evaluation is to verify if the proposed debriefing system allows a better reflection on past actions and understanding of the outcome. The purpose is also to evaluate how much the two main components used for the debriefing, i.e. the visualization and the feedback, contribute to the effect of the debriefing. To accomplish this, a user experiment has been set up and multiple questionnaires have been composed. We also evaluated the usability of the system, using questions from the System Usability Scale (Brooke, 1996) to collect information on the factors that might have contributed to the usability.

5.1 Setup

To perform this study, a user experiment with 5 participants was conducted. The participants were between 18 and 27 years old and were all frequent users of social networks (mostly Facebook).. The most frequent age demographic for users of social media is from 18 to 29 (Duggan & Brenner, 2013) and according to MacDonald and Roberts-Pittman (2010) cyberbullying not only affects teenagers but also college students. Therefore, and because BullyBook is targeting a more mature audience, we selected people between 18 and 34 .

The experiment was performed in a closed setting and with the presence of an evaluator. Participants received a clear description of what to expect of the game in the form of a verbal presentation, but without too much information about the automatic debriefing system.

5.2 Methodology

The participant will first play the game, after which he will be asked to fill out the post-game questionnaire and answer some open questions. Then the participant will use the debriefing modules, he will be asked to find certain information about his past actions using the debriefing system and finally he will be invited to fill out the post-debriefing questionnaire. In this way, the user evaluation is divided in two parts:

5.2.1 Post-Game evaluation part

The participant first plays the game and finishes (or not) level 1 on his own without guidance. Then, the participant is asked the following questions to evaluate if he understood the outcome of his interactions and how they affected the Non-Player Characters:

- I understand the outcome of the game
- My actions had the effect I expected they would have on other characters
- Deciding on my reaction to a bullying post was easy
- Predicting how my interactions would influence other characters was not easy
- It is clear to me which interactions in the game lead to a good/bad outcome
- I know feel more secure on how to deal with bullying situations on social networks

Two open questions are also asked regarding the strategy applied during the game.

- Which strategy did you apply to complete the first level?
- Did you change your strategy in the course of the game?

5.2.2 Post-Debriefing evaluation part

The participant experiments with the 3 different debriefing modules sequentially in a random order, and answers the questions given to him. After each session he is asked to answer a number of closed questions to evaluate his (changed) understanding of the outcome of the game, using a written questionnaire. The focus of this evaluation is to assess whether there is a better understanding of the outcome of the game and what elements of the debriefing contributed to this. Afterwards the freedom is given to the participant to critique the system in an open interview session.

The first set of questionnaires concern the experience with each of the modules separately:

- It was not easy to obtain a clear overview of all interactions that took place in the game
- It was simple to understand how I could interact with the visualizations
- I got a clear overview of which interactions I performed in the game
- I had trouble understanding how to interact with the visualization
- I could see how my interactions affected other characters in the game
- I found it difficult to understand what the visualizations represented

For each of the 3 debriefing modules additional questions are asked that focus more on the visual aspects of the specific visualization.

Time-Oriented Debriefing:

- Showing the interactions based on time was a good way to recall my interactions
- More details about an interaction on the timeline could have helped me to inspect my interactions
- Using a timeline is not a good way to represent the course of the game.

Character-Oriented Debriefing:

- There was enough information in this visualization to understand the interactions between characters
- It was easy to navigate through the different screens of this visualization
- I got enough information to understand the impact of the interactions on the characters

Interaction-Oriented Debriefing:

- The visualization contained too many elements
- This visualization was a good summary of my game session

Final questionnaires:

After experimenting with each of the 3 debriefing modules the participant is asked to answer some questions about the usability of the debriefing system :

- I think that I would like to use this system frequently.
- I found the system unnecessarily complex.
- I thought the system was easy to use.
- I think that I would need the support of a technical person to be able to use this system.
- I found the various functions in this system were well integrated.
- I thought there was too much inconsistency in this system.
- I would imagine that most people would learn to use this system very quickly.
- I found the system very cumbersome to use.
- I felt very confident using the system.
- I needed to learn a lot of things before I could get going with this system.

A set of questions is also asked to evaluate what they thought about the feedback they received in the debriefing. These questions were the same for all 3 modules:

- I found it interesting to see the values for the personality attributes of all the characters
- Receiving an example of a better reaction was helpful to understand the issues related to cyberbullying
- I did not get enough information in the suggestion for a better interaction to understand why it was better
- I understand why a suggested reaction would have been better in the game
- I donâĂŹt understand why values for personality attributes are given for each interaction
- It is clear to me which interactions in the game lead to a bad/good outcome

This is followed by final a questionnaire, similar to the one given after playing the game, to see if the participant understood the outcome of the game better:

- I now better understand the outcome of the game
- I now understand better why my actions had a certain effect
- It would now be easier to decide on a reaction to a bullying post
- It would now be easier to predict how my interactions would influence other characters
- It is now more clear to me which interactions in the game lead to a good/bad outcome
- I now feel more secure on how to deal with bullying situations on social networks

5.3 Results

In what follows the results of the evaluation will be given. Some of the results are influenced by some limitations in the game that will be discussed later in this chapter. To check the consistency of the participants' answers, certain questions were rephrased negatively and included in the questionnaires. In this chapter the values answered for these questions are transformed to conform to the other question (a 2 becomes a 4, a 1 becomes a 5). No inconsistency were found in the results for the questions that were rephrased negatively.

On the first questionnaire that was taken after playing the game, the average result was 3,1 on the Likert Scale. These questions focus on the understanding of the game and the interactions that took place, and relate to the original understanding of the game's logic without using any debriefing system. See figure 5.1 for the results. For the two open questions regarding the strategy they used to accomplish the objective of the game, most participants replied that they used common sense in their replies to bullying posts.

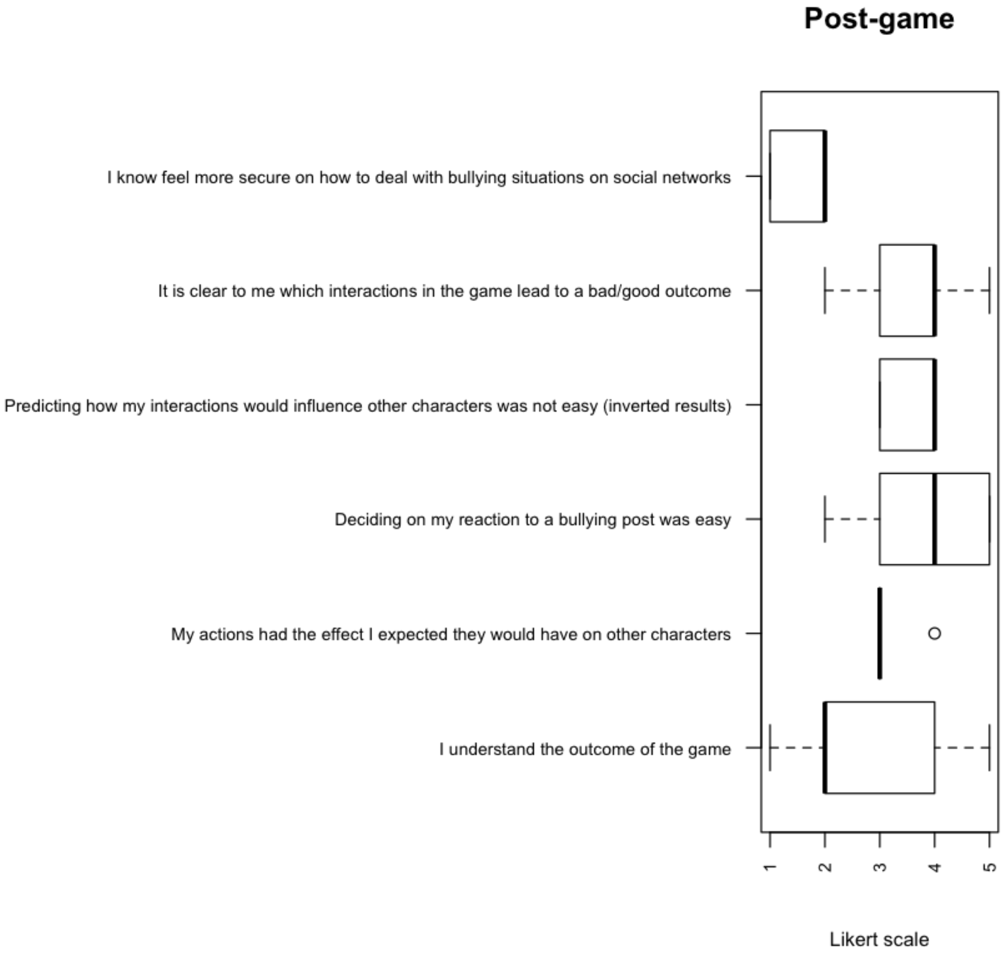


Figure 5.1: Box plot representation of the answers provided for the post-game questionnaire

For the second set of questionnaires the focus was put on the 3 visualizations in the debriefing system. Figure 5.2 shows the results of the questionnaire concerning the time-oriented debriefing. The average score for this debriefing module is 3,8.

The second module received a score of 4,1 and a box plot representation of the answers provided for this questionnaire is shown in figure 5.3.

The interaction-oriented debriefing had the best results of all 3 debriefing modules, with an average score of 4,4. Figure 5.4 shows the boxplot representation of the questionnaire associated with this module.

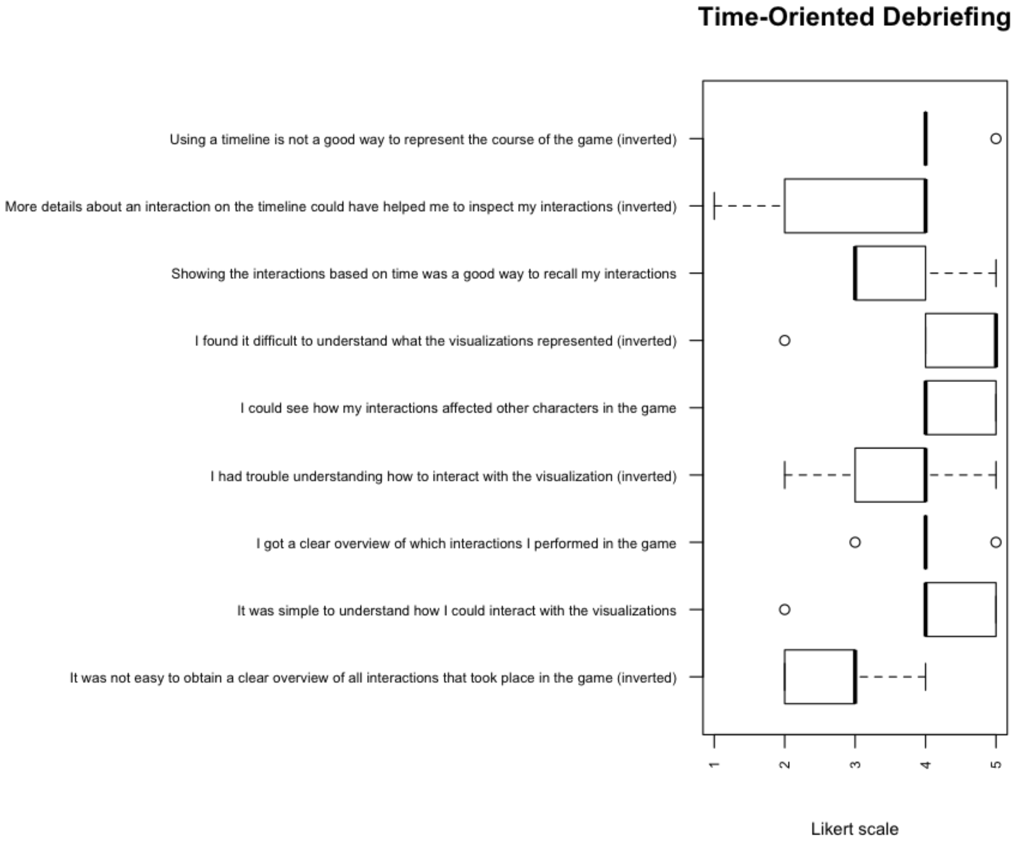


Figure 5.2: Box plot representation of the answers provided for the questionnaire about the Time-Oriented debriefing

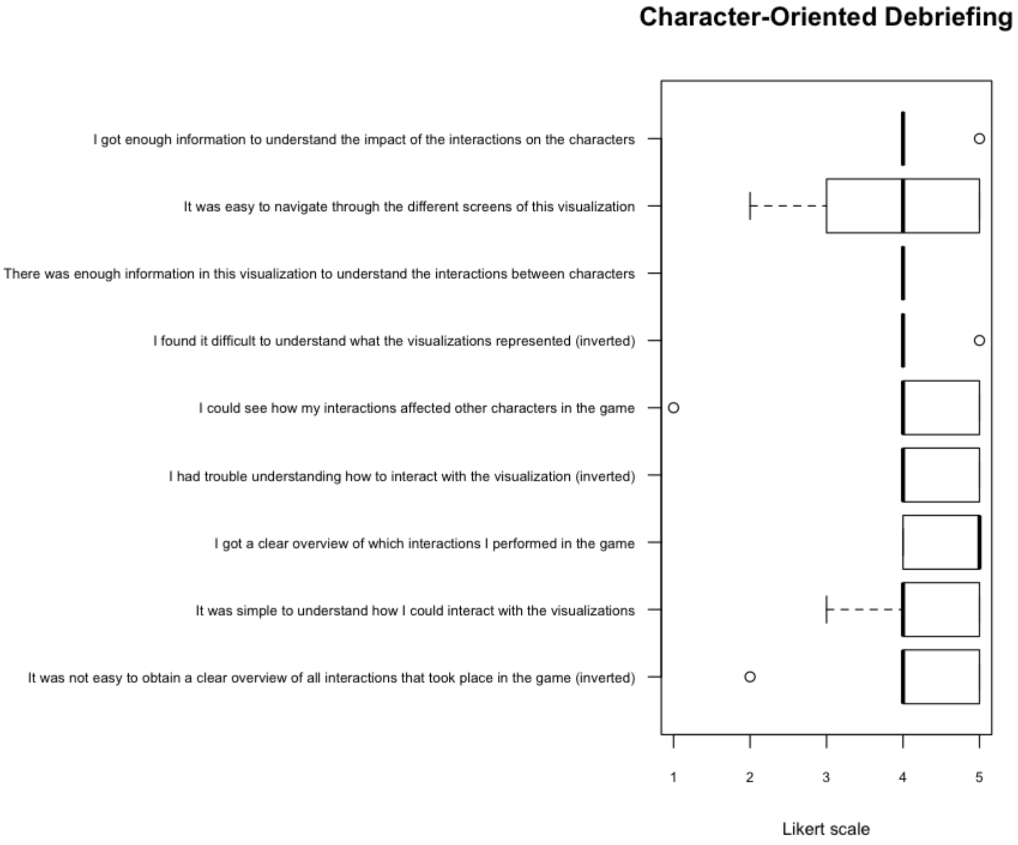


Figure 5.3: Box plot representation of the answers provided for the questionnaire about the Character-Oriented debriefing

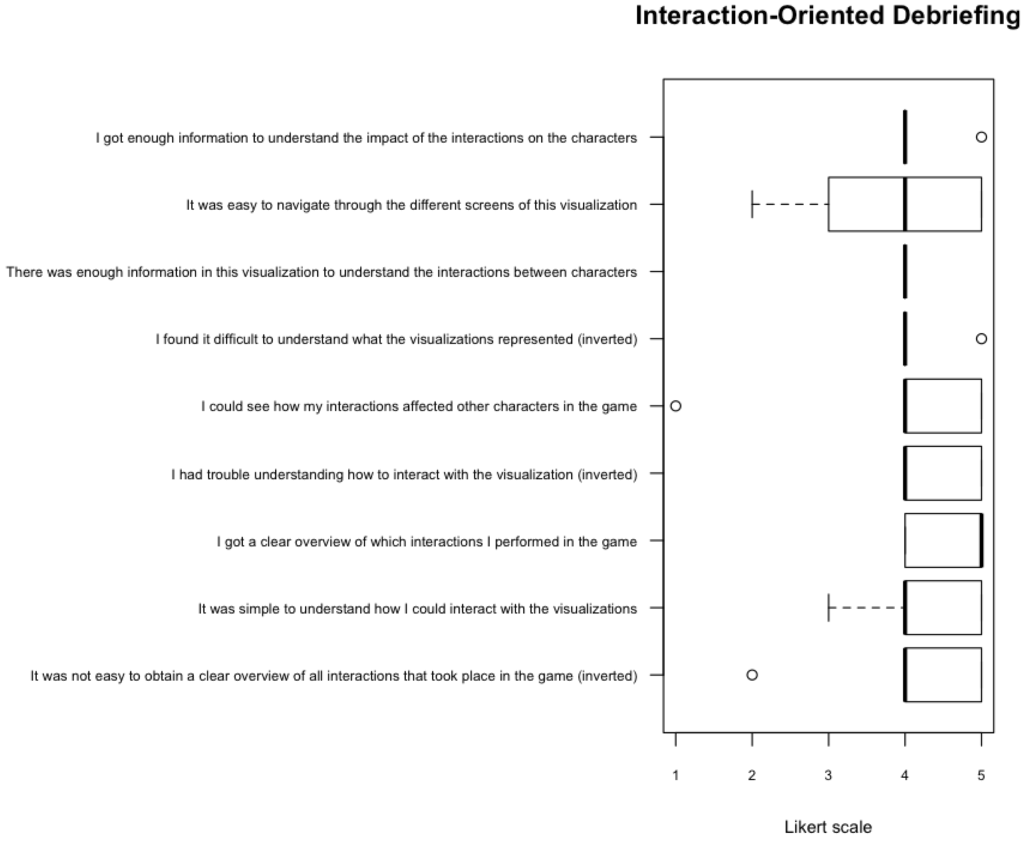


Figure 5.4: Box plot representation of the answers provided for the questionnaire about the Interaction-Oriented debriefing

For the evaluation of the usability of the whole system a System Usability Scale (SUS) was used. In this questionnaire and the boxplot depicted in figure 5.5 the answered values are not inverted because when using SUS the usability of system can be considered better than average when the calculated score [1] is higher than 68 (Brooke, 1996). In our case the total score is 78 which indicates that the usability of our system is not a problem.

The feedback provided in the debriefing system scored a 4,5 on average on the associated questionnaire (see figure 5.6).

Finally the last questionnaire (figure 5.7), regarding the understanding of

[1] This calculation converts the scores of each questions to a new number by either subtracting the score with 1 (for odd numbered questions) or subtracting the score from 5 in the even numbered questions. The total is then multiplied by 2.5.

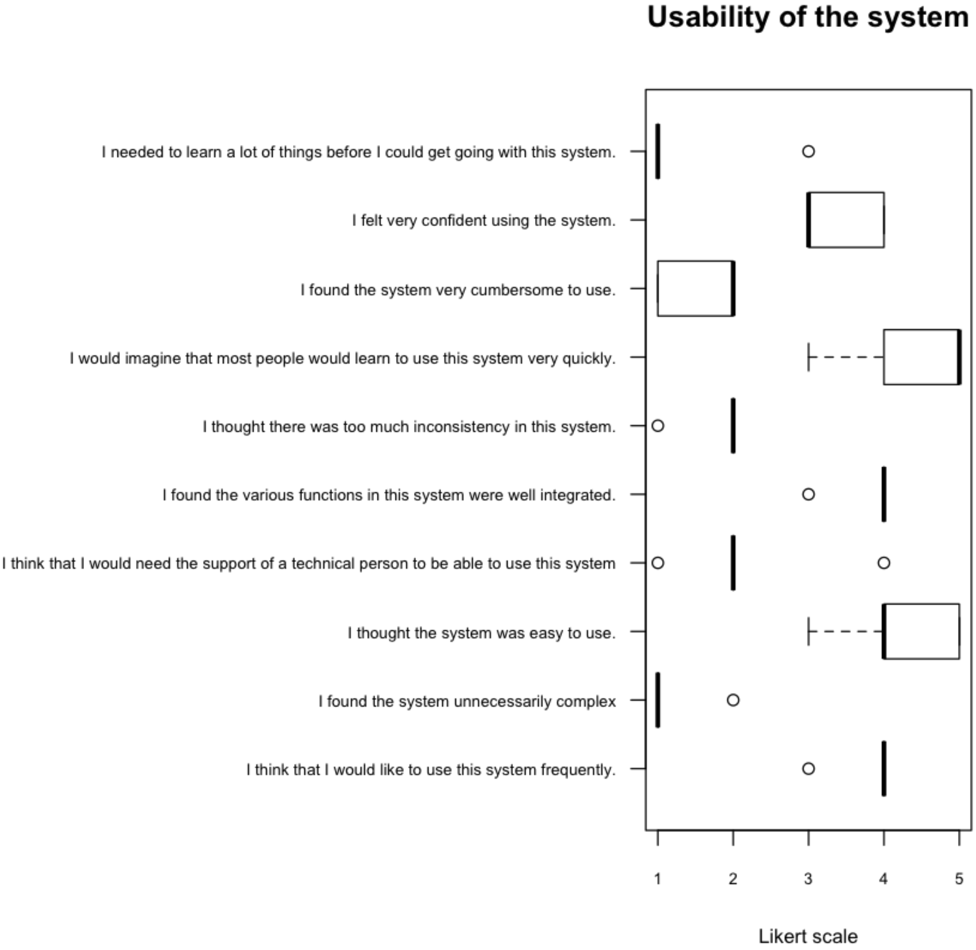


Figure 5.5: Box plot representation of the answers provided for the questionnaire about the usability of the system

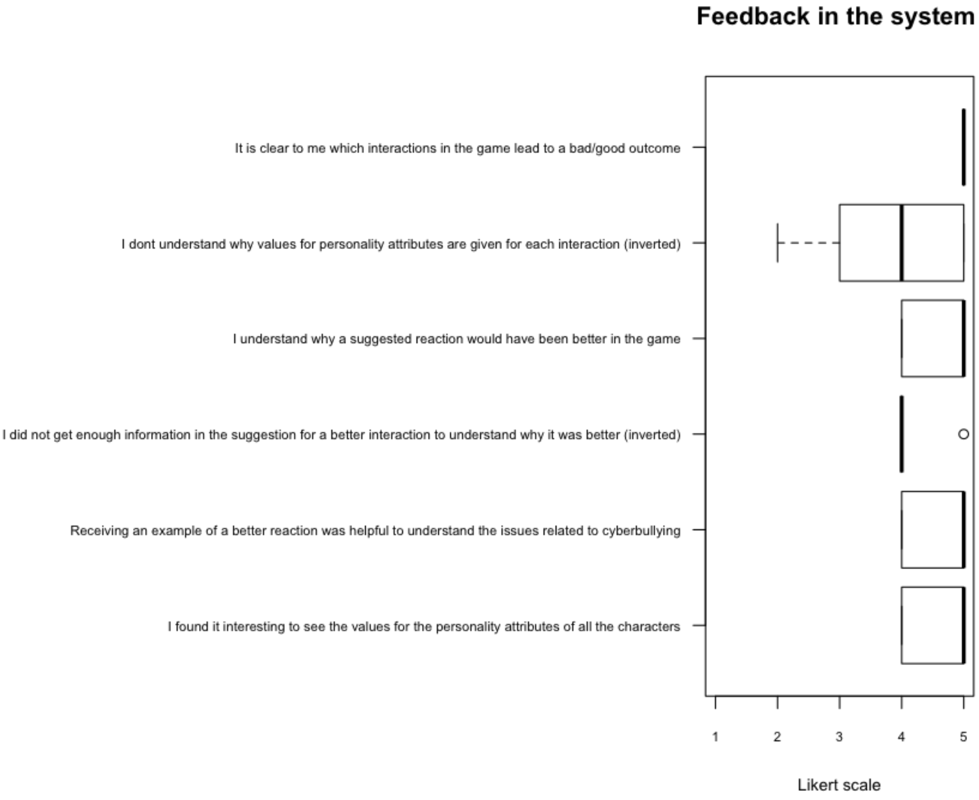


Figure 5.6: Box plot representation of the answers provided for the questionnaire about the feedback and textual feedback in the system

the game and the outcome of the interactions scored 4,3.

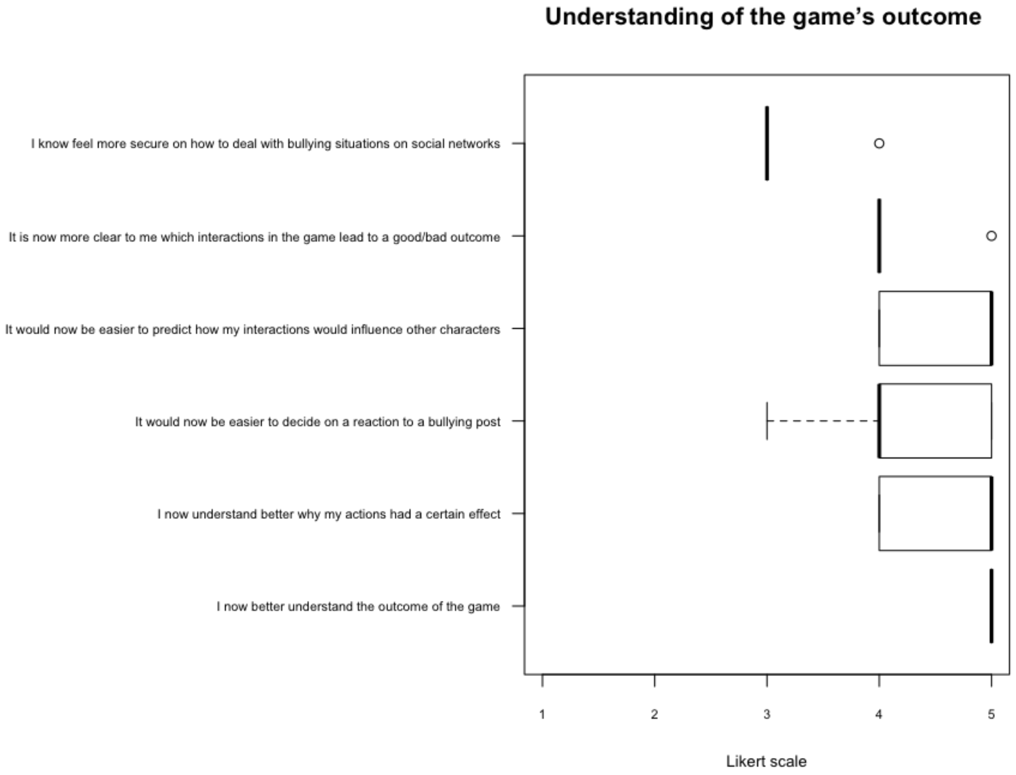


Figure 5.7: Box plot representation of the answers provided for the general questionnaire to evaluate the understanding of the game's outcome.

5.4 Discussion

The results of the questionnaire that was filled out after playing the game indicate that the understanding of the game's outcome was not optimal. Most participants had questions about how to proceed with the game and how to achieve the objective of the level. Although the results are not too bad, we have to consider that only one level is playable in the current implementation of BullyBook. This first level is simple and therefore it is likely that participants who had trouble understanding the outcome of the first level will have more trouble understanding the outcome of next levels. The simplicity of the first level is also reflected in the strategy used by the participants.

The Time-Oriented Debriefing (TOD) received a good score by the participants. We think that this is mainly due to its simple design. Two questions however didn't have such a good result. The first one is about the details showed in the timeline and the second one about the overview this visualization provided. Although the simplicity of the visualization made it easy to interact with it, getting useful information regarding the game was lacking according to the participants.

The Character-Oriented Debriefing received better scores than the TOD, which can be explained by the two-way visualization. The first screen only showed the relationships between characters, and only after clicking these relationships the interactions between the two characters are displayed. This made the interaction more structured and fun to use, according to one participant. Finally, as hoped, the Character-Oriented Debriefing had the best score of all 3 debriefing modules. This module contains a complete overview of the interaction and the filters added to the visualization avoided the undesired effect of overloading the screen with information. The questionnaire about the feedback that was given for each interaction in the system received good results, which shows that the information given about the interaction was found useful to understand the game's outcome. One participant, however, didn't like the use of progress bars for the personality model and would have preferred a more textual explanation of the impact of interactions on other characters.

The scores for the usability indicate (78 on the SUS) that the system is easy to use. Although this system is a prototype implementation, the result is important because it is an indication that the usability of the system didn't have a negative impact on the results obtained in the other questionnaires.

Finally the most satisfying result is that the participants reported to have a better understanding of the game's outcome after having used the debriefing module. While the post-game questionnaire resulted in the average of 3,1, the post-debriefing questionnaire about a better understanding of the outcome resulted in an average 4,3. This result is also encouraging because as the development of the game goes, the complexity of the levels will be raised and the debriefing could play a more important role in understanding the outcome of the game. On the question to measure if their awareness about cyberbullying was raised with this game and debriefing, the results are of not so significant. In fact in most cases the participant replied neutrally to this question.

6 Conclusion

6.1 Summary

This dissertation described the steps undertaken in the goal of creating an automatic debriefing system for a Serious Game, i.e. Bullybook. The first step consisted of investigating the related work in this domain. Unfortunately, very little existing automatic debriefing systems exist for serious games. However some elements were found to be interesting, such as a questionnaire to evaluate the user's understanding of the game and the underlying principles. Using the information we gathered during the research we made an analysis of the foundations on which we would build our debriefing strategy. To accomplish this we first looked at what information we would have to log in the course of the game to use it afterwards in the debriefing. This led us to the concept of game state, which captures all relevant data at a given moment in time. These game states can be saved every time an interaction occurs in the game. After this we looked at how we could represent the information related to these game states to the player in a way that would allow reflection and understanding of the outcome of the game. We opted to do this by using visualizations, because they allow us to use a small space to display a large amount of data. We centered our debriefing system around 3 different visualization modules. Each of these modules focuses on a different aspect of the game, i.e. on time, on the characters, and on the interactions. This analysis part was followed by the design of our debriefing system. The decision was made to use an ontology to represent the structure of our game. This was done with the purpose of offering a reusable solution for similar games: in theory for a different game, a similar ontology could be used, however in practice some adjustments will be required. We accompanied this ontology with rules that would also fit this purpose. Finally, we made a prototype implementation of the designed system, using various technologies (web pages, Stardog triple store, REST web service...). This system was then evaluated in a pilot study, with the goal of gathering a first feedback on the system. The results indicated that the debriefing system did help the participants in understanding the outcome of the game better. This can be concluded after comparison of the answers between the post-game and the post-debriefing questionnaires that were similar. The visualization based on the interactions was the best way to represent the game states, according to the evaluation. However, the debriefing system didn't raise the awareness about cyberbullying for the participants. A possible explanation for this is the fact that the current state of the game only contains one (simple) level.

6.2 Contributions

With this work we aimed at making a first step in the direction of automatic debriefing for serious games to answer our first research question, i.e. how to explain the outcome of the game to a casual player. The main purpose of debriefing is to allow a reflection on the game and understanding of the concepts that contributed to the outcome, which is not always easy for non-linear games. A formal debriefing with a facilitator, would be too costly to be considered a general applicable solution. This dissertation exposed our approach of making this process of reflection automatic and therefore such an automatic debriefing system wouldn't require the intervention of a facilitator. To do so, certain elements were captured in game states, which answer our research question regarding which elements to capture in the game. The representation of game states are displayed in visualizations that focus on different aspects of the game state. The results of the pilot evaluation study showed that this approach indeed could help understanding the outcome of the game. Another objective of this work was to allow our solution to be reusable in other serious games. This was achieved by using an Ontology with RDF standards. A set of rules was also created to manipulate and reason over the data.

6.3 Limitations and Future work

Although the debriefing system showed good results in the pilot study, these results have to be put in the context of the current development of the Bully-Book game. With only one level playable, it is difficult to evaluate the real contribution the debriefing did bring to the understanding of the game. To evaluate the actual effect that the debriefing system has on the understanding of the game, the game should contain more levels and complex situations that would challenge the player more. The further development of the debriefing system should go paired with the development of the game, to make sure the debriefing matches the game interactions for more complicated situations. With a more complete game, a more thorough evaluation can be performed.

Replay functionality

One element that was discussed but not integrated in the current implementation of the debriefing system is the replay functionality. This functionality can have a powerful effect on the reflection process, as it resituates the player in the context of his decisions. Unfortunately with the current state of the game, replaying situations would make little sense. But when more levels will be playable, this is certainly a functionality that should be considered.

Active questionnaire

Another extension that also relates to the further development of the game that was not included in this implementation is the active questionnaire. This interactive way of asking the debriefing system why a certain action occurred would certainly help in understanding the game. A participant confirmed this idea during the pilot study, expressing the difficulty of understanding the underlying meaning of the outcome without an elaborate textual feedback.

Integrated solution

The current implementation of the debriefing system is in the form of a web application that is consulted after having played the game. This is not ideal when playing a game. A better solution would be to have an integrated debriefing system that shows up after playing the game.

Fully RDF implementation

Although we aimed at providing a solution that would be as reusable as possible by creating a Debriefing Ontology, in the current implementation certain functionalities are accomplished using a relational database. Among the future works can be included the migration of the current implementation to support RDF completely.

A
Appendix

A.1 Questionnaires

Post-Game session

Name :
Age :
Gender :

- Male
- Female

I use social networks :

- Multiple times a day
- Once a day
- Multiple days a week
- Once a week
- Less frequently

	Strongly Disagree	Disagree	Neither Disagree Nor Agree	Agree	Strongly Agree
I understand the outcome of the game	**1**	**2**	**3**	**4**	**5**
My actions had the effect I expected they would have on other characters	**1**	**2**	**3**	**4**	**5**
Deciding on my reaction to a bullying post was easy	**1**	**2**	**3**	**4**	**5**
Predicting how my interactions would influence other characters was not easy	**1**	**2**	**3**	**4**	**5**
It is clear to me which interactions in the game lead to a good/bad outcome	**1**	**2**	**3**	**4**	**5**

I now feel more secure on how to deal with bullying situations on social networks	**1**	**2**	**3**	**4**	**5**

Open questions :

- Which strategy did you apply to complete the first level?
- Did you change your strategy in the course of the game?

Time-Oriented Visualization

Visualizations in the debriefing (in general)

	Strongly Disagree	Disagree	Neither Disagree Nor Agree	Agree	Strongly Agree
It was not easy to obtain a clear overview of all interactions that took place in the game	1	2	3	4	5
It was simple to understand how I could interact with the visualizations	1	2	3	4	5
I got a clear overview of which interactions I performed in the game	1	2	3	4	5
I had trouble understanding how to interact with the visualization	1	2	3	4	5
I could see how my interactions affected other characters in the game	1	2	3	4	5
I found it difficult to understand what the visualizations represented	1	2	3	4	5

Specific visualization

Showing the interactions based on time was a good way to recall my interactions	**1**	**2**	**3**	**4**	**5**
More details about an interaction on the timeline could have helped me to inspect my interactions	**1**	**2**	**3**	**4**	**5**
Using a timeline is not a good way to represent the course of the game.	**1**	**2**	**3**	**4**	**5**

Character-Oriented Visualization

Visualizations in the debriefing (in general)

	Strongly Disagree	Disagree	Neither Disagree Nor Agree	Agree	Strongly Agree
It was not easy to obtain a clear overview of all interactions that took place in the game	1	2	3	4	5
It was simple to understand how I could interact with the visualizations	1	2	3	4	5
I got a clear overview of which interactions I performed in the game	1	2	3	4	5
I had trouble understanding how to interact with the visualization	1	2	3	4	5
I could see how my interactions affected other characters in the game	1	2	3	4	5
I found it difficult to understand what the visualizations represented	1	2	3	4	5

Character-Oriented specific questions

a.	**There was enough information in this visualization to understand the interactions between characters**	**1**	**2**	**3**	**4**	**5**
b.	**It was easy to navigate through the different screens of this visualization**	**1**	**2**	**3**	**4**	**5**
c.	**I got enough information to understand the impact of the interactions on the characters**	**1**	**2**	**3**	**4**	**5**

Interaction-Oriented Visualization

Visualizations in the debriefing (in general)

	Strongly Disagree	Disagree	Neither Disagree Nor Agree	Agree	Strongly Agree
It was not easy to obtain a clear overview of all interactions that took place in the game	1	2	3	4	5
It was simple to understand how I could interact with the visualizations	1	2	3	4	5
I got a clear overview of which interactions I performed in the game	1	2	3	4	5
I had trouble understanding how to interact with the visualization	1	2	3	4	5
I could see how my interactions affected other characters in the game	1	2	3	4	5
I found it difficult to understand what the visualizations represented	1	2	3	4	5

Interaction-Oriented Debriefing (specific)

The visualization contained too many elements	**1**	**2**	**3**	**4**	**5**
This visualization was a good summary of my game session	**1**	**2**	**3**	**4**	**5**

Feedback in the Debriefing System

	Strongly Disagree	Disagree	Neither Disagree Nor Agree	Agree	Strongly Agree
I found it interesting to see the values for the personality attributes of all the characters	1	2	3	4	5
Receiving an example of a better reaction was helpful to understand the issues related to cyberbullying	1	2	3	4	5
I did not get enough information in the suggestion for a better interaction to understand why it was better	1	2	3	4	5
I understand why a suggested reaction would have been better in the game	1	2	3	4	5
I don't understand why values for personality attributes are given for each interaction	1	2	3	4	5
It is clear to me which interactions in the game lead to a bad/good outcome	1	2	3	4	5

Usability of the system

	Strongly Disagree	Disagree	Neither Disagree Nor Agree	Agree	Strongly Agree
Usability of the debriefing system					
I think that I would like to use this system frequently	1	2	3	4	5
I found the system unnecessarily complex.	1	2	3	4	5
I thought the system was easy to use.	1	2	3	4	5
I think that I would need the support of a technical person to be able to use this system.	1	2	3	4	5
I found the various functions in this system were well integrated.	1	2	3	4	5
I thought there was too much inconsistency in this system.	1	2	3	4	5
I would imagine that most people would learn to use this system very quickly.	1	2	3	4	5

I found the system very cumbersome to use.	**1**	**2**	**3**	**4**	**5**
I felt very confident using the system.	**1**	**2**	**3**	**4**	**5**
I needed to learn a lot of things before I could get going with this system.	**1**	**2**	**3**	**4**	**5**

Understanding of the game outcome

	Strongly Disagree	Disagree	Neither Disagree Nor Agree	Agree	Strongly Agree
I now better understand the outcome of the game	1	2	3	4	5
I now understand better why my actions had a certain effect	1	2	3	4	5
It would now be easier to decide on a reaction to a bullying post	1	2	3	4	5
It would now be easier to predict how my interactions would influence other characters	1	2	3	4	5
It is now more clear to me which interactions in the game lead to a good/bad outcome	1	2	3	4	5
I know feel more secure on how to deal with bullying situations on social networks	1	2	3	4	5

A.2 Ontology

```xml
<?xml version="1.0"?>
<rdf:RDF xmlns="http://www.semanticweb.org/metzodell/ontologies/2016/2/untitled-ontology-2#"
     xml:base="http://www.semanticweb.org/metzodell/ontologies/2016/2/untitled-ontology-2"
     xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
     xmlns:owl="http://www.w3.org/2002/07/owl#"
     xmlns:xml="http://www.w3.org/XML/1998/namespace"
     xmlns:xsd="http://www.w3.org/2001/XMLSchema#"
     xmlns:debriefing="http://www.semanticweb.org/metzodell/ontologies/2016/2/untitled-ontology-2#"
     xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#">
    <owl:Ontology rdf:about="http://www.semanticweb.org/metzodell/ontologies/2016/2/untitled-ontology-2"/>
    <!--
    ///////////////////////////////////////////////////////////////////////////////////////
    //
    // Object Properties
    //
    ///////////////////////////////////////////////////////////////////////////////////////
     -->

    <!-- debriefing:#affectsInitiator -->

    <owl:ObjectProperty rdf:about="debriefing:#affectsInitiator">
        <rdfs:domain rdf:resource="debriefing:#Gamestate"/>
        <rdfs:range rdf:resource="debriefing:#Character"/>
    </owl:ObjectProperty>

    <!-- debriefing:#belongsTo -->

    <owl:ObjectProperty rdf:about="debriefing:#belongsTo">
        <rdfs:domain rdf:resource="debriefing:#PersonnalityModel"/>
        <rdfs:range rdf:resource="debriefing:#Gamestate"/>
    </owl:ObjectProperty>

    <!-- debriefing:#belongsToCharacter -->

    <owl:ObjectProperty rdf:about="debriefing:#belongsToCharacter">
        <rdfs:domain rdf:resource="debriefing:#PersonnalityModel"/>
        <rdfs:range rdf:resource="debriefing:#Character"/>
    </owl:ObjectProperty>
```

```
<!-- debriefing:#hasAffectedInitiator -->
<owl:ObjectProperty rdf:about="debriefing:#hasAffectedInitiator">
  <rdfs:subPropertyOf rdf:resource="http://www.w3.org/2002/07/owl#topObjectProperty"/>
  <rdfs:domain rdf:resource="debriefing:#Gamestate"/>
  <rdfs:range rdf:resource="debriefing:#PersonnalityModel"/>
  <rdfs:comment>has affected the initiators personnality</rdfs:comment>
  <rdfs:label>affectsTarget</rdfs:label>
</owl:ObjectProperty>

<!-- debriefing:#hasAuthor -->
<owl:ObjectProperty rdf:about="debriefing:#hasAuthor">
  <rdfs:subPropertyOf rdf:resource="http://www.w3.org/2002/07/owl#topObjectProperty"/>
  <rdfs:domain rdf:resource="debriefing:#Message"/>
  <rdfs:range rdf:resource="debriefing:#Character"/>
</owl:ObjectProperty>

<!-- debriefing:#hasInitialPersonnality -->
<owl:ObjectProperty rdf:about="debriefing:#hasInitialPersonnality">
  <rdfs:domain rdf:resource="debriefing:#Character"/>
  <rdfs:range rdf:resource="debriefing:#PersonnalityModel"/>
  <rdfs:comment>had initial personnality model</rdfs:comment>
  <rdfs:label>hasInitialPersonality</rdfs:label>
</owl:ObjectProperty>
<!-- debriefing:#hasInitiator -->

<owl:ObjectProperty rdf:about="debriefing:#hasInitiator">
  <rdfs:subPropertyOf rdf:resource="http://www.w3.org/2002/07/owl#topObjectProperty"/>
  <rdfs:domain rdf:resource="debriefing:#Gamestate"/>
  <rdfs:range rdf:resource="debriefing:#Character"/>
</owl:ObjectProperty>
<!-- debriefing:#hasInteraction -->
<owl:ObjectProperty rdf:about="debriefing:#hasInteraction">
  <rdfs:subPropertyOf rdf:resource="http://www.w3.org/2002/07/owl#topObjectProperty"/>
  <rdfs:domain rdf:resource="debriefing:#Gamestate"/>
  <rdfs:range rdf:resource="debriefing:#Interaction"/>
</owl:ObjectProperty>
<!-- debriefing:#hasInteractionType -->

<owl:ObjectProperty rdf:about="debriefing:#hasInteractionType">
```

```
        <rdfs:subPropertyOf rdf:resource="http://www.w3.org/2002/07/owl#topObjectProperty"/>
        <rdfs:domain rdf:resource="debriefing:#Interaction"/>
        <rdfs:range rdf:resource="debriefing:#InteractionType"/>
        <rdfs:comment>was of type</rdfs:comment>
        <rdfs:label>was of type</rdfs:label>
    </owl:ObjectProperty>

    <!-- debriefing:#hasTarget -->

    <owl:ObjectProperty rdf:about="debriefing:#hasTarget">
        <rdfs:subPropertyOf rdf:resource="http://www.w3.org/2002/07/owl#topObjectProperty"/>
        <rdfs:domain rdf:resource="debriefing:#Gamestate"/>
        <rdfs:range rdf:resource="debriefing:#Character"/>
        <rdfs:comment>targeted</rdfs:comment>
        <rdfs:label>hasTarget</rdfs:label>
    </owl:ObjectProperty>

    <!-- debriefing:#isInitiatedBy -->

    <owl:ObjectProperty rdf:about="debriefing:#isInitiatedBy">
        <rdfs:domain rdf:resource="debriefing:#Interaction"/>
        <rdfs:range rdf:resource="debriefing:#Character"/>
        <rdfs:comment>was initiated by</rdfs:comment>
        <rdfs:label>was initiated by</rdfs:label>
    </owl:ObjectProperty>
        <!--
    ///////////////////////////////////////////////////////////////////////////////////////
    //
    // Data properties
    //
    ///////////////////////////////////////////////////////////////////////////////////////
     -->
<!-- debriefing:#CharacterName -->

    <owl:DatatypeProperty rdf:about="debriefing:#CharacterName">
        <rdfs:domain rdf:resource="debriefing:#Character"/>
        <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#string"/>
        <rdfs:comment>has character name</rdfs:comment>
        <rdfs:label>has character name</rdfs:label>
```

```
  </owl:DatatypeProperty>
  <!-- debriefing:#CharacterPicture --
  <owl:DatatypeProperty rdf:about="debriefing:#CharacterPicture">
    <rdfs:subPropertyOf rdf:resource="http://www.w3.org/2002/07/owl#topDataProperty"/>
    <rdfs:domain rdf:resource="debriefing:#Character"/>
    <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#string"/>
    <rdfs:comment>has picture</rdfs:comment>
    <rdfs:label>has picture</rdfs:label>
  </owl:DatatypeProperty>
    <!-- debriefing:#hasAcceptance -->
  <owl:DatatypeProperty rdf:about="debriefing:#hasAcceptance">
    <rdfs:subPropertyOf rdf:resource="debriefing:#hasPersonnalityAttribute"/>
    <rdfs:domain rdf:resource="debriefing:#Character"/>
    <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#string"/>
    <rdfs:comment>has acceptance</rdfs:comment>
    <rdfs:label>has acceptance</rdfs:label>
  </owl:DatatypeProperty>
  <!-- debriefing:#hasContent -->


  <owl:DatatypeProperty rdf:about="debriefing:#hasContent"/>
   <!-- debriefing:#hasCuriosity -->


  <owl:DatatypeProperty rdf:about="debriefing:#hasCuriosity">
    <rdfs:subPropertyOf rdf:resource="debriefing:#hasPersonnalityAttribute"/>
    <rdfs:domain rdf:resource="debriefing:#Character"/>
    <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#string"/>
    <rdfs:comment>has curiosity</rdfs:comment>
    <rdfs:label>has curiosity</rdfs:label>
  </owl:DatatypeProperty>
<!-- debriefing:#hasEating -->
  <owl:DatatypeProperty rdf:about="debriefing:#hasEating">
    <rdfs:subPropertyOf rdf:resource="debriefing:#hasPersonnalityAttribute"/>
    <rdfs:domain rdf:resource="debriefing:#Character"/>
    <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#string"/>
    <rdfs:comment>has personnality attribute</rdfs:comment>
    <rdfs:label>has personnality attribute</rdfs:label>
  </owl:DatatypeProperty>
  <!-- debriefing:#hasFamily -->
  <owl:DatatypeProperty rdf:about="debriefing:#hasFamily">
```

```
    <rdfs:subPropertyOf rdf:resource="debriefing:#hasPersonnalityAttribute"/>
    <rdfs:domain rdf:resource="debriefing:#Character"/>
    <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#string"/>
    <rdfs:comment>has family</rdfs:comment>
    <rdfs:label>has family</rdfs:label>
</owl:DatatypeProperty>
<!-- debriefing:#hasHonor -->
<owl:DatatypeProperty rdf:about="debriefing:#hasHonor">
    <rdfs:subPropertyOf rdf:resource="debriefing:#hasPersonnalityAttribute"/>
    <rdfs:domain rdf:resource="debriefing:#Character"/>
    <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#string"/>
    <rdfs:comment>has honor</rdfs:comment>
    <rdfs:label>has honor</rdfs:label>
</owl:DatatypeProperty>
<!-- debriefing:#hasIdealism -->
<owl:DatatypeProperty rdf:about="debriefing:#hasIdealism">
    <rdfs:subPropertyOf rdf:resource="debriefing:#hasPersonnalityAttribute"/>
    <rdfs:domain rdf:resource="debriefing:#Character"/>
    <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#string"/>
    <rdfs:comment>has idealism</rdfs:comment>
    <rdfs:label>has idealism</rdfs:label>
</owl:DatatypeProperty>
<!-- debriefing:#hasIndependence -->
<owl:DatatypeProperty rdf:about="debriefing:#hasIndependence">
    <rdfs:subPropertyOf rdf:resource="debriefing:#hasPersonnalityAttribute"/>
    <rdfs:domain rdf:resource="debriefing:#Character"/>
    <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#string"/>
    <rdfs:comment>has independence</rdfs:comment>
    <rdfs:label>has independence</rdfs:label>
</owl:DatatypeProperty>
<!-- debriefing:#hasInteractionProgress -->
<owl:DatatypeProperty rdf:about="debriefing:#hasInteractionProgress">
    <rdfs:domain rdf:resource="debriefing:#Interaction"/>
    <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#string"/>
    <rdfs:comment>has progress</rdfs:comment>
    <rdfs:label>has progress</rdfs:label>
</owl:DatatypeProperty>
<!-- debriefing:#hasInteractionTime -->
<owl:DatatypeProperty rdf:about="debriefing:#hasInteractionTime">
```

```
      <rdfs:domain rdf:resource="debriefing:#Interaction"/>
      <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#string"/>
      <rdfs:comment>occurred at</rdfs:comment>
      <rdfs:label>occurred at</rdfs:label>
  </owl:DatatypeProperty>

<!-- debriefing:#hasOrder -->
  <owl:DatatypeProperty rdf:about="debriefing:#hasOrder">
      <rdfs:subPropertyOf rdf:resource="debriefing:#hasPersonnalityAttribute"/>
      <rdfs:domain rdf:resource="debriefing:#Character"/>
      <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#string"/>
      <rdfs:comment>has order</rdfs:comment>
      <rdfs:label>has order</rdfs:label>
  </owl:DatatypeProperty>
  <!-- debriefing:#hasPersonnalityAttribute -->
  <owl:DatatypeProperty rdf:about="debriefing:#hasPersonnalityAttribute">
      <rdfs:domain rdf:resource="debriefing:#PersonnalityModel"/>
      <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#string"/>
      <rdfs:comment> has attribute </rdfs:comment>
      <rdfs:label>has attribute</rdfs:label>
  </owl:DatatypeProperty>
  <!-- debriefing:#hasPhysicalActivity -->
  <owl:DatatypeProperty rdf:about="debriefing:#hasPhysicalActivity">
      <rdfs:subPropertyOf rdf:resource="debriefing:#hasPersonnalityAttribute"/>
      <rdfs:domain rdf:resource="debriefing:#Character"/>
      <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#string"/>
      <rdfs:comment>has physical activity</rdfs:comment>
      <rdfs:label>has physical activity</rdfs:label>
  </owl:DatatypeProperty>



  <!-- debriefing:#hasPower -->
  <owl:DatatypeProperty rdf:about="debriefing:#hasPower">
      <rdfs:subPropertyOf rdf:resource="debriefing:#hasPersonnalityAttribute"/>
      <rdfs:domain rdf:resource="debriefing:#Character"/>
      <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#string"/>
      <rdfs:comment>has power</rdfs:comment>
      <rdfs:label>has power</rdfs:label>
```

```
</owl:DatatypeProperty>
<!-- debriefing:#hasRomance -->
<owl:DatatypeProperty rdf:about="debriefing:#hasRomance">
  <rdfs:subPropertyOf rdf:resource="debriefing:#hasPersonnalityAttribute"/>
  <rdfs:domain rdf:resource="debriefing:#Character"/>
  <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#string"/>
  <rdfs:comment>has romance</rdfs:comment>
  <rdfs:label>has romance</rdfs:label>
</owl:DatatypeProperty>
<!-- debriefing:#hasSaving -->
<owl:DatatypeProperty rdf:about="debriefing:#hasSaving">
  <rdfs:subPropertyOf rdf:resource="debriefing:#hasPersonnalityAttribute"/>
  <rdfs:domain rdf:resource="debriefing:#Character"/>
  <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#string"/>
  <rdfs:comment>has saving</rdfs:comment>
  <rdfs:label>has saving</rdfs:label>
</owl:DatatypeProperty>



<!-- debriefing:#hasSocialContact -->
<owl:DatatypeProperty rdf:about="debriefing:#hasSocialContact">
  <rdfs:subPropertyOf rdf:resource="debriefing:#hasPersonnalityAttribute"/>
  <rdfs:domain rdf:resource="debriefing:#Character"/>
  <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#string"/>
  <rdfs:comment>has social contact</rdfs:comment>
  <rdfs:label>has social contact</rdfs:label>
</owl:DatatypeProperty>

<!-- debriefing:#hasStatus -->
<owl:DatatypeProperty rdf:about="debriefing:#hasStatus">
  <rdfs:subPropertyOf rdf:resource="debriefing:#hasPersonnalityAttribute"/>
  <rdfs:domain rdf:resource="debriefing:#Character"/>
  <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#string"/>
  <rdfs:comment>has status</rdfs:comment>
  <rdfs:label>has status</rdfs:label>
</owl:DatatypeProperty>
<!-- debriefing:#hasTranquility -->
<owl:DatatypeProperty rdf:about="debriefing:#hasTranquility">
```

```
        <rdfs:subPropertyOf rdf:resource="debriefing:#hasPersonnalityAttribute"/>

        <rdfs:domain rdf:resource="debriefing:#Character"/>

        <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#string"/>

        <rdfs:comment>has tranquility</rdfs:comment>

        <rdfs:label>has tranquility</rdfs:label>

    </owl:DatatypeProperty>

    <!-- debriefing:#hasVengeance -->

    <owl:DatatypeProperty rdf:about="debriefing:#hasVengeance">

        <rdfs:subPropertyOf rdf:resource="debriefing:#hasPersonnalityAttribute"/>

        <rdfs:domain rdf:resource="debriefing:#Character"/>

        <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#string"/>

        <rdfs:comment>has vengeance</rdfs:comment>

        <rdfs:label>has vengeance</rdfs:label>

    </owl:DatatypeProperty>

    <!--

    ///////////////////////////////////////////////////////////////////////////////////////

    //

    // Classes

    //

    ///////////////////////////////////////////////////////////////////////////////////////

     -->

    <!-- debriefing:#Character -->

    <owl:Class rdf:about="debriefing:#Character">

        <rdfs:comment>This characters are as well non playable characters as the main player and come with a set of
attributes.</rdfs:comment>

        <rdfs:label>Character</rdfs:label>

    </owl:Class>




    <!-- debriefing:#Gamestate -->

 <owl:Class rdf:about="debriefing:#Gamestate"/>

<!-- debriefing:#Interaction -->

 <owl:Class rdf:about="debriefing:#Interaction">

        <rdfs:comment>The interaction can be of varied forms such as a Like or a Dialog.</rdfs:comment>

        <rdfs:label>Interaction</rdfs:label>

    </owl:Class>

    <!-- debriefing:#InteractionType -->

    <owl:Class rdf:about="debriefing:#InteractionType">

        <rdfs:comment>The type of interaction can be Greeting, Simple bullying, bullying or neutral dialog</rdfs:comment>
        <rdfs:label>InteractionType</rdfs:label>

    </owl:Class>

     <!-- debriefing:#Message -->

    <owl:Class rdf:about="debriefing:#Message">

        <rdfs:subClassOf rdf:resource="http://www.w3.org/2002/07/owl#Thing"/>

    </owl:Class>




    <!-- debriefing:#PersonnalityModel -->


    <owl:Class rdf:about="debriefing:#PersonnalityModel">

        <rdfs:comment>The personnality model refers to Reiss&apos; personnality models and consists of 16 values that human try to
reach.</rdfs:comment>

        <rdfs:label>PersonnalityModel</rdfs:label>

    </owl:Class>


    <!--

    ///////////////////////////////////////////////////////////////////////////////////////

    //

    // Individuals

    //

    ///////////////////////////////////////////////////////////////////////////////////////

     -->




    <!-- debriefing:#CharacterAngelo -->


    <owl:NamedIndividual rdf:about="debriefing:#CharacterAngelo">

        <rdf:type rdf:resource="debriefing:#Character"/>

        <CharacterName>Angelo</CharacterName>

    </owl:NamedIndividual>

 <!-- debriefing:#CharacterClara -->

    <owl:NamedIndividual rdf:about="debriefing:#CharacterClara">

        <rdf:type rdf:resource="debriefing:#Character"/>

        <CharacterName>Clara</CharacterName>

    </owl:NamedIndividual>

    <!-- debriefing:#CharacterFebe -->
    <owl:NamedIndividual rdf:about="debriefing:#CharacterFebe">
        <rdf:type rdf:resource="debriefing:#Character"/>
        <CharacterName>Febe</CharacterName>
    </owl:NamedIndividual>
    <!-- debriefing:#CharacterJuliette -->
    <owl:NamedIndividual rdf:about="debriefing:#CharacterJuliette">
        <rdf:type rdf:resource="debriefing:#Character"/>
        <CharacterName>Juliette</CharacterName>
    </owl:NamedIndividual>
    <!-- debriefing:#CharacterSepe -->
    <owl:NamedIndividual rdf:about="debriefing:#CharacterSepe">
        <rdf:type rdf:resource="debriefing:#Character"/>
    </owl:NamedIndividual>



    <!-- debriefing:#CharacterTim -->
    <owl:NamedIndividual rdf:about="debriefing:#CharacterTim">
        <rdf:type rdf:resource="debriefing:#Character"/>
    </owl:NamedIndividual>
    <!--
    ///////////////////////////////////////////////////////////////////////////////////////
    //
    // General axioms
    //
    ///////////////////////////////////////////////////////////////////////////////////////
     -->
    <rdf:Description>
        <rdf:type rdf:resource="http://www.w3.org/2002/07/owl#AllDisjointClasses"/>
        <owl:members rdf:parseType="Collection">
            <rdf:Description rdf:about="debriefing:#Character"/>
            <rdf:Description rdf:about="debriefing:#Interaction"/>
            <rdf:Description rdf:about="debriefing:#InteractionType"/>
            <rdf:Description rdf:about="debriefing:#PersonnalityModel"/>
        </owl:members>
    </rdf:Description>
</rdf:RDF>
<!-- Generated by the OWL API (version 4.2.1.20160306-0033) https://github.com/owlcs/owlapi -->
```

Bibliography

Abt, C. C. (1987). *Serious games.* University Press of America.

Barbosa, A. F., & Silva, F. G. (2011). Serious games: design and development of oxyblood. In *Proceedings of the 8th international conference on advances in computer entertainment technology* (p. 15).

Berne, E. (1961). *Transactional analysis in psychotherapy: A systematic individual and social psychiatry:(19. printing).* Grove Press.

Bostock, M. (2015). *D3.js.* Retrieved June 1, 2016, from D3.js, Data-Driven Documents: `https://d3js.org/`.

Brooke, J. (1996). *Sus: A 'quick and dirty' usability scale, usability evaluation in industry, jordan, pw, thomas, b., weerdmeester, ba and mcclelland, al.* London: Taylor and Francis.

Caillois, R. (2015). *Les jeux et les hommes. le masque et le vertige.* Editions Gallimard.

Cebolledo, E., & De Troyer, O. (2015). iattac: A system for autonomous agents and dynamic social interactions–the architecture. In *Serious games* (pp. 135–146). Springer.

Cleophas, C. (2012). Designing serious games for revenue management training and strategy development. In *Proceedings of the winter simulation conference* (p. 140).

Coenen, T., Mostmans, L., & Naessens, K. (2013). Museus: Case study of a pervasive cultural heritage serious game. *Journal on Computing and Cultural Heritage (JOCCH)*, *6*(2), 8.

Crawford, C. (2003). *Chris crawford on game design.* New Riders.

Crookall, D. (2010). Serious games, debriefing, and simulation/gaming as a discipline. *Simulation & gaming*, *41*(6), 898–920.

Duggan, M., & Brenner, J. (2013). *The demographics of social media users, 2012* (Vol. 14). Pew Research Center's Internet & American Life Project Washington, DC.

Fanning, R. M., & Gaba, D. M. (2007). The role of debriefing in simulation-based learning. *Simulation in healthcare*, *2*(2), 115–125.

Friendly attac. (2012). Retrieved June 1, 2016, from Friendly ATTAC: `http://www.friendlyattac.be/`.

Gane, C. P., & Sarson, T. (1979). *Structured systems analysis: tools and techniques.* Prentice Hall Professional Technical Reference.

Gutierrez, E. O. C., & De Troyer, O. (2015). Modelling social network interactions in games. In *Eleventh artificial intelligence and interactive digital entertainment conference.*

Hakulinen, L. (2011). Using serious games in computer science education. In *Proceedings of the 11th koli calling international conference on computing education research* (pp. 83–88).

Harpstead, E., Myers, B. A., & Aleven, V. (2013). In search of learning: facilitating data analysis in educational games. In *Proceedings of the sigchi conference on human factors in computing systems* (pp. 79–88).

Johnson, W. L. (1994). Agents that explain their own actions. *AD-A280 063*, *8*, 21.

Kay, R. (2002). Quickstudy: System development life cycle. *Computerworld*, *14*, 18.

Lazar, J., Feng, J. H., & Hochheiser, H. (2010). *Research methods in human-computer interaction.* John Wiley & Sons.

MacDonald, C. D., & Roberts-Pittman, B. (2010). Cyberbullying among college students: Prevalence and demographic differences. *Procedia-Social and Behavioral Sciences*, *9*, 2003–2009.

Nicholson, S. (2012). Completing the experience: Debriefing in experiential educational games. In *Proceedings of the 3rd international conference on society and information technologies* (pp. 117–121).

Pavlov, O. V., Saeed, K., & Robinson, L. W. (2015). Improving instructional simulation with structural debriefing. *Simulation & Gaming*, *46*(3-4), 383–403.

Rebolledo-Mendez, G., Avramides, K., de Freitas, S., & Memarzia, K. (2009). Societal impact of a serious game on raising public awareness: the case of floodsim. In *Proceedings of the 2009 acm siggraph symposium on video games* (pp. 15–22).

Reiss, S. (2008). The normal personality. *A new way of*.

Reynolds, D. (2014). *The organization ontology.* Retrieved June 1, 2016, from W3, The Organization Ontology: `https://www.w3.org/TR/vocab-org/`.

Shoukry, L., Göbel, S., & Steinmetz, R. (2014). Learning analytics and serious games: Trends and considerations. In *Proceedings of the 2014 acm international workshop on serious games* (pp. 21–26).

Squire, K. (2008). Open-ended video games: A model for developing learning for the interactive age. *The ecology of games: Connecting youth, games, and learning*, 167–198.

Stardog. (2016). Retrieved June 1, 2016, from Stardog, Enterprise Data Unification with Smart Graphs: `http://stardog.com/`.

Zyda, M. (2005). From visual simulation to virtual reality to games. *Computer*, *38*(9), 25–32.