HORIZON2020 FRAMEWORK PROGRAMME

ICT – 21 -2014

Advanced digital gaming/gamification technologies

Gamification of Prosocial Learning for Increased Youth Inclusion and Academic Achievement

D4.1 Intelligent Adaptation and Personalization
This document describes algorithms for online and offline adaptation of games.

**Author(s)**
Kiriakos Stefanidis (CERTH), Konstantinos Apostolakis (CERTH), Athanasios Psaltis (CERTH), Kosmas Dimitropoulos (CERTH), Petros Daras (CERTH)

**Reviewer(s)**
Kam Star (PLAYGEN), Eric Robertson (Redikod)

**Dissemination level**
- [ ] internal
- [x] public
- [ ] confidential

**Document Control Page**

<table>
<thead>
<tr>
<th>Version</th>
<th>Date</th>
<th>Modified by</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.0</td>
<td>01/06/2016</td>
<td>Kiriakos Stefanidis (CERTH)</td>
<td>First version draft</td>
</tr>
<tr>
<td>2.0</td>
<td>08/06/2016</td>
<td>Kiriakos Stefanidis (CERTH)</td>
<td>Ready for internal review</td>
</tr>
<tr>
<td>3.0</td>
<td>21/06/2016</td>
<td>Kiriakos Stefanidis (CERTH)</td>
<td>Final Version</td>
</tr>
<tr>
<td>4.0</td>
<td>22/06/2016</td>
<td>Pilar Pérez (ATOS)</td>
<td>Format review and final version</td>
</tr>
<tr>
<td>Abbreviation</td>
<td>Description</td>
<td></td>
<td></td>
</tr>
<tr>
<td>--------------</td>
<td>----------------------------------</td>
<td></td>
<td></td>
</tr>
<tr>
<td>PsL</td>
<td>ProsocialLearn</td>
<td></td>
<td></td>
</tr>
<tr>
<td>PAM</td>
<td>ProsocialLearn Adaptation Manager</td>
<td></td>
<td></td>
</tr>
<tr>
<td>PLO</td>
<td>Prosocial Learning Objective</td>
<td></td>
<td></td>
</tr>
<tr>
<td>API</td>
<td>Application Programming Interface</td>
<td></td>
<td></td>
</tr>
<tr>
<td>OL</td>
<td>Online Learning</td>
<td></td>
<td></td>
</tr>
<tr>
<td>XML</td>
<td>Extensible Markup Language</td>
<td></td>
<td></td>
</tr>
<tr>
<td>AU</td>
<td>Action Unit</td>
<td></td>
<td></td>
</tr>
<tr>
<td>NN</td>
<td>Neural Network</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Executive summary

This deliverable is part of work package four WP4: Dynamic and Personalized Game Elements for Prosocial Learning. It is a public document focusing on the Deliverable 4.1: 1st Intelligent adaptation and personalization. This document will be made available on the project website for external parties interested in adaptive and personalized games for optimized learning. It is hoped that the report will assist interested parties in understanding intelligent configuration of prosocial games, and dynamic recommendation of game adaptations for personalized learning based on player profiles.
4.2 Enhancing Prosocial Behavior through Game Scenarios ....................................................... 24
  4.2.1 Game Scenario Factors and the role of computational models .................................... 25
4.3 Measuring Prosocial Ability & Building Prosocial Profiles ............................................... 26
  4.3.1 System Description ........................................................................................................ 26
  4.3.2 Player’s Prosocial Ability ............................................................................................. 28
  4.3.3 Ability Rankings ............................................................................................................. 29
  4.3.4 Prosocial Profiles .......................................................................................................... 30
4.4 Matching Players to Scenarios .............................................................................................. 31

5 Online Adaptation ............................................................................................................. 33
  5.1 User Engagement & Prosocial Behavior ............................................................................. 33
    5.1.1 Need satisfaction in prosocial behaviour ................................................................. 33
    5.1.2 Need satisfaction in video games ................................................................................. 34
  5.2 Input from Multimodal Fusion ............................................................................................. 37
    5.2.1 Engagement Types ......................................................................................................... 37
    5.2.2 Fusion Architecture ........................................................................................................ 38
  5.3 Enhancing User Engagement through Positive Reinforcement ......................................... 39
  5.4 Supporting player engagement and mastery through corrective feedback ....................... 40
  5.5 Learning User’s Engagement Profile .................................................................................... 40
  5.6 Adjusting Game Elements for Positive Reinforcement and Corrective Feedback ............. 43

6 Conclusions .......................................................................................................................... 46

7 References ............................................................................................................................ 47

Appendix A - Running a simulation .......................................................................................... 50
Appendix B – Example game scenario adaptable factors based on computational model of Trust .... 51
1 Introduction

This section provides detailed information about the purpose of the Prosocial Adaptation Manager in general, as well as the scope and structure of the document, which set the tone for the intended audience and interested readers.

1.1 Purpose of the document

This document presents the role, system architecture, and constituent adaptation approaches of the ProsocialLearn Adaptation Manager (PAM). As part of the activities carried out in Task 4.1 and leading up to this deliverable, we explain how the Adaptation Manager module is built following the guidelines presented in document D2.6 “Prosocial Game Design Methodology”.

As has been described early on within the project, the adaptation manager should support adaptation triggered both online, i.e. during gameplay, to increase player engagement and achieve maximum potential for setting up learning outcomes, as well as offline matching of players to specific game scenarios, providing insight to game developers on how to introduce computational factors to generate different sets of circumstances that aim at balancing player ability to game context. In this respect, the adaptation system could be seen as a matching mechanism that adjusts game elements and receives feedback from the user about his/her experience with that element. This feedback will be either retrieved in the form of player engagement indicators (WP3), as well as the prosociality level of the player, measured through means described in this deliverable. Specifically towards this goal, we explore, and proceed to test methodologies from the acknowledged field of recommender systems.

In this deliverable, we will describe the algorithms and processes used for online and offline adaptation, underlining the theoretical foundations where needed. Towards this end, the scientific background staging the foundation of each approach, as well as related references on the appropriate techniques, are provided in the text to solidify our approach with regards to each adaptation technique.

1.2 Scope and Audience of the document

The dissemination level of this document is public. The members of the consortium but also other third parties and interested audiences can have access to the ProsocialLearn Adaptation Manager and architecture approach.

1.3 Structure of the document

This document contains the following key sections, conveniently detailed in the list below:

- **Section 1: Introduction** – an introductory section, i.e. this present section, which describes the WP as a whole, as well as the main purpose of the Task that generated this document.
- **Section 2: About the Prosocial Adaptation Manager** – In this Section, the role and purpose of the PAM are presented.
- **Section 3: Architecture** – this section outlines the architecture and organization of Adaptation Manager’s components.
- **Section 4: Offline Adaptation** – this Section delves into the scientific literature presenting the theoretical foundations and a description of the offline adaptation modules of the PAM.
• **Section 5: Online Adaptation** – Similarly to Section 4, this Section provides an analytical description of the purpose, as well as the functionality of the PAM online adaptation components.

• **Section 6: Conclusion** – this section presents the conclusion of the work described in this document.

Finally, Appendices A and B provide some insight on the simulations and methods proposed within this document, providing some examples on proper use of the described structures, as well as an overview on the positioning of the PAM within the overall PsL architecture.
2 About the ProsocialLearn Adaptation Manager

2.1 Purpose

The purpose of the ProsocialLearn Adaptation Manager (PAM) is to provide personalization capabilities to the ProsocialLearn project and enhance the process of instructing prosocial behavior to students via gamification. A detailed explanation of its importance can be given along two different perspectives. On the one hand, personalization is currently an essential part in the development process of most software applications and serious games. This is due to the fact that personalization provides convenience to the user of modern software applications, compared to old software systems. On the other hand, in the process of teaching a concept to a student, personalization allows the effective transmission and acquisition of knowledge. Every student has his/her own individualised way of learning. The role of a teacher is to try and learn the individual preferences of students, and make students’ learning as effective as possible. Therefore, considering the benefits gained from both expressions of personalization, the development of an adaptation manager that matches the preferences of players with the proper game conditions (realising teaching conditions), can be of great importance for the effectiveness of a gamification platform targeting students.

2.2 Psychological & Pedagogical Approach

Prosociality can be interpreted in many ways, and people are exposed to prosocial behaviours in many occasions during their everyday lives. In the Oxford English Dictionary (Oxford, 1989), “prosocial” is defined as behaviour which is positive, helpful and intended to promote social acceptance and friendship, while adhering to the moral standards accepted by an established social group. A more simple way to explain prosociality, is to define the concept as the behaviour of helping others (Penner et al, 2005). Prosocial acts therefore include helping, sharing, donating, cooperation, as well as conforming to socially acceptable behaviour. Prosocial actions have been known to be motivated by a genuine concern for the well-being and rights of others, as well as for personal or practical concerns. Examples of the latter include one’s social status or reputation, hoping for a direct or indirect positive outcome, or behaving in harmony to one’s personal values or moral code.

2.2.1 Psychological Approach

Prosociality is in itself a complex concept and is comprised of many core domains, which include empathy, trust, fairness, compassion, generosity and cooperation (refer to D2.1 for more details on the core prosociality domains).

• **Empathy** corresponds to the ability to understand one’s own, as well as other people’s feelings, in a way that allows someone to feel someone else’s feelings.

• **Compassion** is the ability to take actions to help someone who is in need out of a genuine intent to improve one’s well-being, including the act of being kind to oneself (i.e. self-compassion).

• **Trust** corresponds to the ability to trust and be trusted, i.e. identifying one’s genuine or malicious intentions in order to engage in cooperation, as well as communicating to others that one can make a good cooperation partner.
• **Fairness** is the ability to understand when a situation’s outcomes correspond to disproportionate rewards (i.e. unfair treatment of oneself or others), and taking actions to distribute these rewards in a manner that is perceived as being just.

• **Generosity** is the ability of being able to part with one’s own resources out of pure volition to benefit someone else by handing over ownership of said resources.

• **Cooperation** is the ability to work in a group towards a common goal.

### 2.2.2 Pedagogical Approach

Although the domains are useful in explaining the different types of prosocial concepts, the high level nature of these concepts makes it somewhat difficult to translate prosocial domains to computational models, which in turn will support the design of games with successful learning outcomes. Although such models have been developed within the scope of the PsL project (e.g. the computational model for Trust described in D3.2, which also carries over to this document in several occasions), a more practical way of defining prosociality for game developers is through the description of skills or competencies. As explained in D2.6, these skills are based on the CASEL approach, and have led to the definition of a non-exhaustive list of the 42 prosocial skills that fit into 3 families of skills, namely skills for friendship, skills for feelings, and skills for collaboration.

In order to systematically teach prosocial skills, a skillstreaming process has been developed, which focuses on the following sequence of learning strategies:

i. Instruction/description: Includes both verbal and written description of a skill along with steps to perform the behaviour. The description should be highlighting the benefits to be gained by engaging in that behaviour, as well as short- and long-term outcomes attached to the act of that behaviour.

ii. Modelling: This includes a step-by-step demonstration of the behaviour or skill, prior to the actual re-enactment either in-game or as part of role-playing (see below) in the physical world.

iii. Role-playing: As mentioned in the bullet above, in this activity the children imitate the modelled behaviour in a variety of themes and contexts, attempting at re-enacting the ordered steps presented during the instruction phase.

iv. Performance feedback: This phase corresponds to the provision of in-game or physical performance rewards linked to the actor’s performance of role-playing a specific prosocial skill. Feedback in response to action consists of both positive reinforcement and corrective feedback.

v. Generalisation (trying the skill in different context): Generalization meant to help players identify where and when to use the skill, as well as how to apply it in variety of circumstances.

This particular sequence has proven robust in cases where children are lacking the actual know-how of a specific behavioural skill (i.e. skill-deficit) as well as children who fail to reproduce the skill in a variety of circumstances despite being aware of the correct behaviour (i.e. performance-deficit). Therefore, it makes sense for prosocial games to emulate the above skillstreaming approach in order to increase the chances of in-game modelled prosocial behaviour having an impact on children acquiring and learning when and how to use these skills. In terms of adaptable game elements, the
automatization of the above process would require both real-time, online, as well as offline, post-session adaptation, which is driven by the player’s in-game performance.

On-line adaptation in this respect, includes processes that regulate the provision of performance feedback (phase iv), while ensuring children are fully engaged and immersed in the game’s narrative, i.e., being emotionally invested in the well-being of other characters (whether player-controlled or NPC), and generally care for what happens to them, as a means to maximise the potential of an actual skill being learnt. Positive reinforcement, can be argued, are a main staple of games. In applying positive reinforcement, or positive rewards in games, a range of rewards must be incorporated for the adaptation mechanism to determine the proper one in each specific situation, including material rewards (i.e., a significant ability or rare and specific objects), activity rewards (free-for-all bonus levels, or otherwise parts of the game the player really enjoyed playing are made available for re-play for a specific amount of time), token rewards (i.e. through an in-game currency system that allows an exchange of a specific number of tokens for something more tangible, like material or activity rewards, as described above) as well as social rewards including both verbal and nonverbal praise, triggered by in-game mechanisms. All praise generated from the adaptation mechanism should be immediate, explicit and characterized with contingency, in that it relates to the action that the player has completed, in order to adequately communicate what the child is doing correctly.

Off-line adaptation processes on the other hand are based on the accumulation of ratings in order to determine a player’s true ability in performing a specific set of prosocial skills, and using these ratings tailor future game experiences in an appropriate manner that fits that particular player’s needs. In this respect, adaptation links to the Generalization phase (v), being primarily responsible for generating a substantial amount of circumstances with a varying level of challenge in terms of identifying where and when to use the skill. Towards this end, game designers should consider introducing parameters that affect game scenario outcomes, player incentives and narrative elements.

2.3 Place in PsL Architecture

As has been described in D2.3, the adaptation manager is a core component of the ProsocialLearn platform. The main task of the manager is to relate multiple indicators of the player’s state (i.e. player’s prosociality level, engagement scores and ability) to a set of game parameters that the manager can adjust in order to tailor the in-game experience in a way that is projected to increase those measures. As described later in this document, the adaptation manager must analyse information coming from both the multi-modal fusion module, as well as the prosocial games themselves, aggregating user behavioral responses occurring in both the physical world (i.e., player facial expressions, voice features, body movements), as well as the virtual game world (i.e. in-game actions, time to complete, etc.). After calculating engagement scores and determining player prosocial skill ranking updates, the data from the manager will be sent to the PsL Dashboard for visualization by teachers. Throughout the rest of this document, we describe optimization methodologies, which fit the requirements of a generic adaptation mechanism that can work independently and propose adaptations to any game it’s linking to.
Figure 1: ProsocialLearn System Architecture
Architecture

3.1 Offline & Online Mechanism

The Adaptation Manager is divided into two parts, namely the offline and the online adaptation mechanisms. The distinction concerns whether the processing takes place during gameplay or loading phase of the game. These two mechanisms aim to personalize the PsL games towards maximizing the players’ prosocial behavior. The reason for having two distinct mechanisms is to provide our platform with better personalization capabilities. Each mechanism processes different information about the player and concerns different types of factors affecting prosociality. Offline adaptation is based on historical player information that concerns their in-game performance, as well as, specific constraints of the Prosocial Learning Objectives (PLOs). While online adaptation considers real-time player data concerning player’s engagement estimation during specific time intervals in the game. In the paragraphs that follow, more detailed information about the two mechanisms is provided.

3.1.1 Off-line Adaptation

Offline adaptation is realised in the loading phase of the game. Its purpose is to select game conditions that are expected to drive the player towards expressing prosocial behaviour. These conditions are referred to as game scenarios. Every scenario belongs to a certain Prosocial Learning Objective (PLO) that is determined by the teacher before the game starts. These PLOs concern the prosocial ability that the teacher intends to train the student at. Each game is capable of utilizing one or more PLOs and for each of them contains a pool of game scenarios offered for offline adaptation.

![Figure 2: Schematic representation for the offline adaptation mechanism in PsL](image)

Before a game begins, the teacher selects the learning objective for training a specific prosocial skill and a registered PsL game that supports it. At the start of each game, the PsL platform exchanges information about the scenarios IDs for the specific game and communicates with the Adaptation Manager which in turn, checks if stored data concerning the active player and the game exist. If data
exist, then the game loads the information and fills-in the data structures needed for both offline and online adaptation mechanisms. In cases where there is no existing data, the data structures are created for the specific game and player. The offline adaptation executes the ability ranking system (Section 4.3) and the player is matched with a scenario based on his or hers ability level. This scenario is then used to initialize the game.

### 3.1.2 Online Adaptation

Online adaptation is realised during the actual gameplay of a PsL game. Its purpose is to select the proper game elements that contribute to the enhancement of the player’s engagement in a prosocial objective. These game elements concern the standard pedagogical practice of expressing *positive reinforcement* on a student’s performance. As it will be described in Section 4.1, the player’s engagement estimation is vital within the context of prosociality, due to the concept’s gamification as in-game tasks for building up a set of predefined abilities. Similar to offline adaptation, these game elements are contained in each PsL game.

![Schematic representation for the online adaptation mechanism in PsL](image)

During gameplay, the engagement of the player may reach values beneath acceptable levels. In such cases, adjustments must be made to the game in order to restore the engagement of the player. The online adaptation mechanism monitors the engagement estimations of the players, given as input by the multimodal fusion module (Section 5.2), and approximates the preference of the player for the positive reinforcement elements offered by the game. The element that is considered to be the player’s top preference, is then selected to adjust the game in real-time.
3.1.3 User Modelling

In order to estimate the preferences of each student, both adaptation mechanisms must maintain user models for the students. A typical adaptation procedure usually involves collecting data concerning the user’s state, updating a user model, and making personalized recommendations. The collection of the data can be either explicit by querying the user (e.g. by filling-up questionnaires) or implicit by estimating the users’ state using a computational model that is hard-coded within the game. The updating procedure of the model depends on how the model is represented in memory. A user model can be represented by a single value, a weighting scheme, a summary of the input data, or the actual input data. Finally, the personalization process refers to adapting content toward the user’s preferences.

There are several types of user models. The most basic types are:

- Static
- Dynamic
- Demographic

A static user model is one that does not change throughout the adaptation procedure. Therefore, such a model is incapable of capturing alterations on user's behaviour. On the contrary, the dynamic user models can be updated with new information concerning the user, and thus, are appropriate for monitoring behavioral shifts. The demographic user model, is also a common modeling type and is used when one wants to represent demographical information about the user.
For the purposes of the Adaptation Manager, we use dynamic user profiles in order to account for players’ behavioral shifts over time.

![Figure 5: A typical feedback loop for adaptation](image)

### 3.2 Online Learning Approach

In this section, a brief description of the machine learning approaches concerning the implementation of PAM’s algorithms is given.

#### 3.2.1 Generic Feedback Loop

A typical machine learning procedure relies on statistical learning and involves building a computational model for prediction problems, using knowledge acquired from data samples encountered in the environment of the agent.

![Figure 6: Machine learning feedback loop](image)

The generic feedback loop used in machine learning re-adjusts the prediction model for every data sample given as input. For a certain problem in study, this loop concerns making a prediction using the current state of the model, receiving feedback about the true outcome, computing the error (i.e.,
the difference between estimated and expected), and re-adjusting the parameters of the model. For a sufficient number of samples, this procedure will eventually converge to a model capable of making correct predictions in a satisfactory rate.

### 3.2.2 Online vs Offline Learning

Except categorising the method for adaptation to offline or online, another distinction must be made concerning whether the learning process of algorithms utilizing adaptation has been performed before the platform’s release to public or relies upon real-time user feedback.

The two main principles in machine learning are the batch and the online learning (OL) approach. In the former approach, the data samples reside in a database and are processed in an offline setting. In the latter approach, the data are gathered and processed in real-time during the actual execution of the application.

A summary about the appropriateness of use for each approach is outlined below

**Online Learning**
- When we have a continuous stream of data
- When it is important to update the algorithm in real time – can hit a moving target
- When training speed is important
- When parameters are “jumpy” around the optimal values

**Batch**
- When it is very important to get the exact optimal values
- When data can fit in memory
- When training time is not of the essence

In general, online learning algorithms have the capacity to make personalisation more evident to the user, due to their ability to handle his or hers behavioral shifts. Also, an appealing feature of OL approaches is their independence from the context of the learning task. All these features make the online learning approach a suitable candidate for PAM’s algorithm implementation.

### 3.2.3 About OL Convergence

Similar to all machine learning methods, the convergence of online learning algorithms depends mainly on the number of samples given as input to the algorithm.
Figure 7 shows the convergence of several online learning algorithms. The simulations were performed using the LIBOL library for online learning (ref). Each algorithm starts its execution having an initial error and in every iteration tries to gradually minimize it by self-adjusting its parameters using samples from its environment. An algorithm is said to reach convergence when the prediction error cannot be furtherly decreased (i.e., it remains constant). As it is evident from the diagram, all algorithms need approximately 2000-3000 iterations in order to reach convergence.

3.3 Components & Organization

The internal structure of the Adaptation Manager consists of a set of interrelating components that exemplify the adaptation algorithms described in the latter sections of this document.

The table below summarizes the Adaptation Manager’s supporting functionality for ProsocialLearn and the corresponding implementation components.

<table>
<thead>
<tr>
<th>Functionality</th>
<th>Implementation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Data Representation</td>
<td>PsLAdaptationBase</td>
</tr>
<tr>
<td>Learning Algorithms</td>
<td>PsLAdaptationCore</td>
</tr>
<tr>
<td>External Communication</td>
<td>PsLAdaptationModule</td>
</tr>
<tr>
<td>Information Update</td>
<td>PsLAdaptationUpdate</td>
</tr>
<tr>
<td>Information Storing</td>
<td>PsLAdaptationXML</td>
</tr>
</tbody>
</table>

A brief description of each of the functionalities follows.

3.3.1 Data Representation

PAM preserves its own data representation scheme containing all necessary data types for storing information in memory. These data structures include: i) the game information structures that keep information about the scenarios and elements that a game supports, ii) the player prosociality profile
that contains the ability estimates for all PLOs, and iii) the player engagement profile that contains information about the player’s engagement recordings for the game’s elements for positive reinforcement and corrective feedback.

### 3.3.2 Learning Algorithms

Two different online learning algorithms are utilized in PAM. The one serves the purposes of offline adaptation while the other concerns online adaptation. For offline adaptation, an ability ranking system is being included that matches the ability of the player for a certain task, to the appropriate scenario that is expected to enhance his or hers prosocial behaviour. For online adaptation, the engagement of the player is monitored and the game element expected to maximize his or hers engagement measurements is selected by analysis of the most recent recordings.

### 3.3.3 External Communication

Communication of the PAM with the PsL platform must be established in order to exchange information with games. This functionality is provided through a separate software component in PAM, ensuring flexibility on any changes in communication protocols.

### 3.3.4 Information update

The core algorithms of PAM construct and exploit user profiles in order to provide game adaptations. Both the prosocial and the engagement profile of a user are updated at specific moments within the game. More specifically, the engagement profile is updated during the actual gameplay, while the prosocial profile is updated at the end of each game.

### 3.3.5 Information Storing

All data collected for a user are stored into files. The memory data structures are converted to text and stored in XML form. A more analytical description of the information storing procedure is given in Section 3.4.

### 3.3.6 Component Organization

The functionality described above, is organized into software components. The organization of the Adaptation Manager components is given in Figure 8.
More specifically, all components make use of \textit{PsLAdaptationBase} in order to define and use the data types of PAM. The \textit{PsLAdaptationModule}, which provides external communication with the PsL platform, uses the \textit{PsLAdaptationCore}'s functions to transfer the requests of the game for adaptation. The \textit{PsLAdaptationCore} uses \textit{PsLAdaptationXML} to load the data at the start of each game, executes the learning algorithms, and uses the \textit{PsLAdaptationUpdate} in order to update the player profiles, as well as, other auxiliary data structures. Finally, at the end of each game, the \textit{PsLAdaptationCore} uses the \textit{PsLAdaptationXML} to store the data into XML format.

3.4 Information Exchange

At the start of each game, information is exchanged between the game, the PsL platform, and the Adaptation Manager. Initially, the game informs the platform about the IDs for the game scenarios and positive reinforcement and corrective feedback elements that it supports. The platform forwards this information to the Adaptation Manager (PAM). In the following, it should be clear that all communications between the game and the PAM is realised through the PsL platform. The PAM either loads information about the game and the player, or in the case that data files do not exist (i.e., first game), creates the necessary structures that hold that information in memory. Then it waits for receiving a signal about executing offline adaptation and initializing the game.

During the loading phase of the game, the game sends a signal to the PAM in order to determine the scenario that is considered to maximize the player’s prosocial ability. The PAM receives the signal and executes the ability ranking system that is outlined in Section 4. The result of the algorithm is the ID of the scenario that matches the ability of the player and is returned as a result to the game.

During gameplay, the online adaptation receives input from the multimodal fusion module and updates the user model for engagement. Additionally the mechanism may detect the need for
adjusting the game towards maximizing the player’s engagement level. In that case, PAM will replace certain game elements with those that are considered to be more suited to the specific user. The PAM accepts the request and using the methodology outlined in Section 5, responds with the proper adjustment.

Finally, at the end of the game, the outcome of the PLO is transferred from the game to the PAM and the ability of the player for the current PLO is updated according to the method described in Section 8.

3.5 Data Storage

ProsocialLearn’s Adaptation Manager (PAM) needs to store player profile data in order to make personalized adaptations to games. This data consist the knowledge base of the system. Profile data include the prosocial profiles composed of the players’ abilities for each of the PLOs, and the engagement profiles composed by the engagement estimates for each of the positive reinforcement adjustments. For the initial version of PAM, the data are not stored in a database, but stored into XML files. The XML schema adopted, is organized under three categories: Game Data, Prosocial Data, and Engagement Data. An example of that schema is given in Figure 10. In subsequent versions this may be replaced with JSON and within more easily scalable database.
Figure 10: XML schema for storing player information in Adaptation Manager

Game Info contains information about the game such as the scenarios IDs and positive reinforcement and corrective feedback element IDs. It also contains the game scenarios’ ratings for prosocial ability (see Section 4.3).

Prosocial Info contains information about the player’s prosocial profile. A player’s prosocial profile consists of ability estimations for every prosocial learning objective (PLO).

Engagement Info contains information about the player’s engagement profile. A player’s engagement profile consists of estimations from the multimodal fusion module (Section 5.2). For a specific game, the player’s engagement profile is composed of engagement measurements for elements that the player has experienced.
4 Offline Adaptation

In this Section we describe a framework for offline adaptation of prosocial games. The process includes the definition of a measurable representation of Prosocial Learning Objectives (PLOs), and means by which a number of game scenarios can be automatically generated through computational models, allowing game developers to create a plethora of different circumstances in which prosocial behaviour needs to be re-enacted by players. A way for measuring player prosocial ability for each PLO, based on their performance in following through the appropriate steps leading to a proper use of the skill in question within the context of a specific scenario is also presented. Also, means by which the offline adaptation mechanism will match each player to the appropriate scenario, after taking into consideration both player prosocial ability as well as the scenario prosociality rating with respect to the PLO being are covered. The methodology described in this Section therefore accounts for both game developers’ requirements of automatically generating in-game situations under different sets of circumstances governed by a number of Factors, as well as intelligent adaptation that ensures players are faced with proper (prosocial) challenge in identifying the “when”, “how” and “where” to apply their knowledge of the skill, and how to achieve the PLO being awarded.

4.1 The Role of Prosocial Learning Objectives (PLOs)

As described in D2.3, Prosocial Learning Objectives (PLOs) are an intrinsic part of the pedagogic process supported by the ProsocialLearn project, defined as a zone within the prosocial measurable space, which can be reached through the modeling (by game developers) and pursuit (by players) of in-game prosocial behavior. These achievements need to be evaluated in terms of interactions and emotions that players exchange with one another and with other entities of the game (i.e. NPCs) at run-time. Game developers are thus required to map their specific game logic to descriptions of prosocial related activities, such that game adaptation recommendations to support student prosocial learning are generated. As is further described in D4.3, a Prosocial Skill Game Model (PGSM) is detailed, where functional attributes constitute the game prosocial world data model, in order to define interactions required to help players attain and improve upon their current arsenal of prosocial skills. Game mechanics can be linked to this prosocial world data model in order to develop a functional representation of the model into a game, where each mechanic maps to varying skills and states of skill performance. A PLO within the prosocial games based approach requires not only in-game modelling of prosocial skills and reinforcement mechanisms that provide corrective feedback, but also includes firm and explicit grounding of the players about the skills to be practiced in the game, both prior as well as after play time. The significance of the player’s in-game experience can reach its maximum potential when it is further enhanced through the provision of real-world rewards, as fit for the specific situation.

In order to attain a specific PLO, players are required to pursue and engage in prosocial behaviour in a manner that allows the basic prosocial skill to be performed in a contextually meaningful and behaviourally proper manner. Regulating conditions for achieving a PLO within the context of an in-game situation can be defined in accordance to the Prosocial World Data Model (D4.3). With respect to the individual playing the game, for example, an activity log of all past player interactions with other characters and in-game entities underlining the frequency a prosocial skill is performed can dictate what prosocial skill should be performed, and to what extent the action should be supported by the player throughout the duration of the game session. For example, players’ trust in other agents can be calculated by means of a computational model (refer to D3.2, as well as Section 4.2 in this document for more details) and estimates of the player’s general trusting disposition can be...
drawn. Such estimates can help identify whether a PLO needs to be defined for trust, i.e. in cases where players do not exhibit enough trust to engage in cooperating behaviour within an in-game society of characters, or whether the semantics of the PLO should be regulated to avert excessive behaviour (like for example, learning when to say “no”, being naïvely dispositioned to trust in strangers, etc.). Additionally, regulations can be made in accordance to the actual performance of the skill, i.e. determining whether, and to what extent the skill has to be done within a specific situation in order to award PLO achievement.

In the remainder of this Section, we will describe methods for offline adaptation that link to a maximised potential of understanding and thus achieving a set of PLOs in a prosocial game world. We define a PLO skill ranking system that matches players to in-game situations that better suit their past demonstrated prosocial ability, therefore ensuring that appropriately moderated challenges are presented to each player. In this context, the notion of Game Scenarios and game scenario Factors are introduced, taking particular care to address the conceptual Prosocial Skill Game Model through a quantifiable description of the Prosocial World Data Model (D4.3).

### 4.2 Enhancing Prosocial Behavior through Game Scenarios

As mentioned in Section 3.1.1, players are ranked according to their in-game performance in terms of behaving according to the designated PLO across a variety of game situations occurring in the majority of the planned prosocial games. The setup for each PLO should be delivered to the player within the context of a scenario. This scenario can be designed to unfold in a finite state machine manner, i.e. the game developers should limit player potential responses to in-game events in context, in order to have them choose between two possible actions: Acting in accordance to the PLO or acting contrary to it. Elements of the game such as the storyline, setting and even the game mechanics governing the scenario gameplay are designed and incorporated within the context of this scenario by the actual game developer. For example, if the ‘Helping Others’ PLO is to be incorporated into a prosocial game, the scenario should challenge the player to identify when and why someone (i.e. an NPC character in the game) needs help, and learning how and when to help them. The act of ‘Help’, i.e. the actual prosocial behavior intended to be adopted by the player as a means of reaching the PLO, can be modelled using any game mechanic that the game developer considers to be proper for the case, i.e. ‘Social Dilemma’, ‘Transfer of Ownership’, ‘Role Reversal’, etc. (for a list of identified prosocial game mechanics and a full description, readers are encouraged to refer to document D4.3).

The game storyline and setting are also a matter of preference, as the game could be focused on children cooperating to solve a difficult math problem in a school playground, or even a fantasy-themed Role Playing Game where the player’s hero encounters a starving NPC during an otherwise common “dungeon raid” quest. The player can then choose to either engage in in-game behavior that models the prosocial skill (i.e. propose the correct mathematical formula to solve the problem, or hand over some potions in each respective example preset above) or not. A time-limit can also be defined, as means to force players to decide within a specific time frame, after which a specific case for receiving ‘no input’ by the player can be modelled (a reference to a case in which the player has neither exhibited clear prosocial or contradicting behavior).

As described in D4.3, a performed prosocial act can also be distinct on a base of sufficiency, i.e. players can regulate whether they’ll behave in a prosocial manner excessively or insufficiently. The developers, as directors, are put in charge of the game’s scenarios for modeling prosocial behavior, i.e., when the game makes an offer to the player to act prosocially (the scenario is triggered), and should be able to determine the outcomes in order to award the aforementioned scores. The scenario outcomes in each case should also be clear to the players after engaging in a specific in-
game behavior, i.e. acting according to the PLO will give the player a substantial amount of feedback: They either get points, advance to a new level, or at any rate find themselves in a better place in terms of inventory assets, knowledge or self-esteem, than they were when first presented the choice. Acting contrary to the PLO should similarly cause a substantial setback to the player’s progress, triggering appropriate feedback to take corrective action in future endeavors. Messages in written text may display on-screen in an attempt to correct player action in future similar situations. The player’s PLO rating will be updated using the aforementioned scoring scheme after the scenario has unfolded, or after any suitable rating period defined by the developers.

4.2.1 Game Scenario Factors and the role of computational models

Each PLO (and in extent, each scenario) needs to be governed by a suitable computational model, that utilizes a number of Factors to determine in general the difficulty of the scenario. These factors should be parameterizable, in order to allow online/offline adaptation mechanisms to make adjustments to tailor the game experience to player preferences and current skill ranking. Each PLO should therefore clearly link to (a subset of) these factors in order to ensure adaptation is possible. For example, the computational model of Trust defined in D3.2, which treats the aforementioned scenarios as situations, uses the following set of factors, which can be adapted by the adaptation manager:

- The potential Utility of a situation a for player x, \( U_a(x) \). This is a measurement of the mean overall outcome of all possible outcomes of that specific situation. We accept through prosocial theory, that behaviors are credited as ‘prosocial’ if they are expressed with intent on helping others (Penner et al, 2005), rather than serving personal gain, and therefore, a zero-utility situation should trigger prosocial behavior on the part of the child, i.e. help someone in need even if there’s no gain in it. Therefore, such cases where \( U_a(x) = 0 \) should always lean towards prosociality.

- The Perceived Competence of a non-player character y for situation a, \( C_y(a) \). Competence of a character involves making a judgement about that character’s ability in a specific situation (i.e. if I trust in y, will he actually do the task entrusted to him?).

- The Importance of situation a for player x, \( I_a(x) \). By definition, the Importance \( I_a(x) \) of a situation a, \( a \in A \), for player x is a subjective measure of the benefits to be gained from the situation. Therefore, applying mechanisms to increase or decrease the importance of a require careful consideration and are not as straightforward as increasing or decreasing a’s potential utility.

- The Perceived Risk inherent in situation a not occurring, \( R_a(x) \). Determining risk in mathematical notation involves differing methods for differing situations, namely, three states for x considering the risks involved in a particular situation a.

For the prosocial behavior of Trust skills, these factors are used to calculate the so-called Cooperation Threshold \( W_a(x) \), which is an estimate of a barrier for overcoming trust issues. More specifically, the cooperation threshold is calculated for each situation and resembles an estimate of how much trust the player is asked to exhibit in order to achieve the situation’s PLO. Having defined a mathematical model for estimating player Situational Trust \( T(x, a) \) in a
foreign character \( y^i \), the player \( x \) can consider whether or not to engage in cooperating behavior (otherwise engage in trusting behavior, i.e. act according to the PLOs for Trust). If the Situational Trust estimated for a specific situation \( \alpha \) is found to be above the situation’s estimated cooperation threshold \( W_\alpha(a)^f \), cooperation \( f(x,y,a) \) has to occur in order to reach the scenario PLO. In Appendix B, we will describe candidate game adaptation based on a computational model of trust.

### 4.3 Measuring Prosocial Ability & Building Prosocial Profiles

A central issue for the ProsocialLearn project is how to measure the player’s prosocial behavior in technical terms, that is, within the context of a game. As observed in most gamification frameworks, ProsocialLearn adopts the notion of tasks or learning objectives to model the concept of interest, in our case, prosocial behaviour i.e. the display of a particular prosocial skill within the game. In order to realise that definition, the ability of a player to accomplish a certain prosocial task can be considered to be an estimate of his or hers prosociality level.

An ability ranking system can be used to estimate the skill level of each player of the PsL platform. The famous Elo system was developed in 1959 by Arpad Elo and was initially used to measure the ability of chess players (Elo & Sloan, 1978). Through the years, it has been incorporated by several organizations that wanted an estimation of their participants’ skill level. The system learns the players’ ability iteratively by utilizing pairwise comparisons (David, 1988) between them and has the potential to match each player with another one of similar ability level. Recently, the system was extended by Microsoft to model players’ ability using Gaussian distributions and to include matchmaking of teams of players (Herbrich et al., 2007). The concept of matchmaking was further altered in (Klinkenberg et al., 2011) where the one player in Elo’s system was replaced by an item, thus matching players to items. This method was incorporated in RAGE which is a recent European project for serious games (RAGE Project, 2015). In that project, game scenarios were matched to players according to their competence. Therefore, a similar ability matching methodology can be applied to ProsocialLearn to model the players’ prosocial ability.

#### 4.3.1 System Description

The Elo rating system learns the players’ ability iteratively by utilizing pairwise comparisons (David, 1988) between them and is based on the following update rule

\[
\hat{\theta}_j = \theta_j + K \left( S_j - E(S_j) \right),
\]

where, \( \theta_j \) is the ability of player \( j \), \( K \) is a weight for rating the uncertainty for \( j \), \( S_j \) is \( j \)'s score, and \( E(S_j) \) is \( j \)'s expected probability of winning. The uncertainty \( K \) controls the rate of convergence of the system. That is, small values of \( K \) make the system learn the ability of a player slowly, while a large \( K \) would mean that the system relies its estimation only on recent events and for that it is inaccurate.

The expected score \( E(S_j) \) for a player \( j \) concerns what the system expects for an outcome of a certain match, considering the ratings for both players. Its definition is given by

---

\( ^2 \) Situational Trust refers to the amount of trust the player \( x \) has in character \( y \) in situation \( \alpha \), and is the most important aspect of trust in cooperative situations, as it provides a measure of trust in another to engage in cooperation to resolve a situation.
where, $N$ is the total games played among players $j$ and $i$. As we see from the above definition, the expected value is a function of the difference of the two components ratings. That is, if the difference is negligible, the expected value will tend towards $0.5$, while large differences force the result towards $0$ or $1$ according to which player preserved the largest rating. By definition, expected values follow an S-like curve, granting for smoothness upon the ability ratings of players, as it is depicted in Figure 11. As a consequence of the above definition for the expected scores, the following equality must hold for players $i$ and $j$,

$$E(S_j) = E(S_i) = 1,$$

In some cases, the theoretical scores of the Elo expectations present a notable constant difference from the actual scores of players. Specifically, an analysis of the FIDE rating database has shown that for some player categories the Elo system provides more accurate score predictions than for the players of other categories. These findings are depicted in Figure 11.

![Figure 11: Curves for the actual and expected scores of player categories in FIDE rating database](image)

In Figure 11, the blue dots refer to the average of only those games where both players were rated 2200+, the yellow dots refer to the games where both players scored below 2200, and the red dots refer to the rest of the games, where one player scored above 2200 and the other below 2200. We can see that the blue dots are definitely closer to the prediction of 76%. Also, the red dots are quite close as well, whereas the yellow dots (i.e., games between weaker players) decline the most, showing down around 71%.

As it is evident from the figure, the blue dots (both players 2200+) are the closest to the white Elo prediction, whereas the yellow dots (both players sub-2200) are the furthest. This means that the Elo system is working better for the top players than for the weaker ones. We must therefore investigate whether most of the games will be played between weaker players. In that case, any overall average
across all three types of games will get skewed by those yellow games. Thus, it is better to examine the results via those groupings, rather than the overall average.

4.3.2 Player’s Prosocial Ability

While the original form of the Elo system has been used for two-player competitive games, variations of it can also be used to match players to items. In the specific implementation of such variation by Klinkenberg et al., the concept of uncertainty is included in the calculation to account for the recency and frequency of the player’s experience with a certain task. Also, the speed that a player solves a task is taken into account and the computation of the expected score changed. For the purpose of ProsocialLearn, these changes do not pose a necessary requirement for computing the player’s ability on prosocial tasks, and for that reason they may not be incorporated in our implementation of the system.

Considering its main purpose to be used for player rankings in two-player competitive games, the Elo system uses a simple methodology to evaluate the outcomes for each game. Specifically, it consists of scores for the three distinct states that a competitive game can come into, namely, winning, losing, and tie. The numerical values adopted to model these states are usually, 1 for winning, 0 for losing, and 0.5 for tie, respectively. In the same sense, and following the works of (Klinkenberg et al., 2011) and (RAGE Project, 2015), we can model the outcomes of a prosocial objective using the above scheme for competitive games. The mapping of values can be done as follows: 1 for accomplishing a PLO, 0 for failure, and 0.5 for any other state. As we will see below, this convention simplifies the procedure of modelling prosocial outcomes and allows the calculation of players’ prosocial ability and the formation of player rankings in a manner similar to the original Elo system.

The description of the different outcomes for a specific PLO can be represented through the notion of a payoff matrix. Commonly encountered in the area of Game Theory (see Weisstein in references), a payoff matrix is a simple structure that is used to outline the different states that a cooperative game can end up to. Each state is the result of the actions that the two players have taken, and is listed for each of them as a label on the columns and the rows of the table, respectively. For every outcome, the payoff for each of the players is given in the cells of the matrix. In the example given in Figure 12, the payoff matrix for the PLO of ‘Healing’ contains the payoffs for the healer and the scenario in each of its four cells. These payoffs are the scores that will be used by the Elo system to update the player prosocial ability and scenario rating for the current PLO.

By definition, the Elo computation for prosociality is based on pairwise comparisons between players and scenarios. This means that in order to have an accurate estimation of player ability and scenario rating for a PLO, the system must be updated for many players and scenarios for that specific PLO.
This is a common requirement for all online learning algorithms, since usually they need a large number of samples in order to converge to a precise estimation.

### 4.3.3 Ability Rankings

The ability estimates calculated for the players, form rankings that outline which player performs better on the predefined prosocial objectives. These rankings may provide the teachers with useful information concerning students’ prosocial behavior. For example, examination of a ranking list for a certain PLO, might indicate those students that may be experiencing difficulties in expressing prosocial behavior under specific circumstances. Also, a clustering analysis of the rankings across all PLOs, might reveal interesting patterns related to personality characteristics of the students. In any case, these measurements could be used as indicators for possible deflects in students’ prosocial behavior, but need to be correlated with certified socio/psychological tests in order to reach a definite conclusion.

<table>
<thead>
<tr>
<th>Rank</th>
<th>Player</th>
<th>Prosocial Ability</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Eleni</td>
<td>2110</td>
</tr>
<tr>
<td>2</td>
<td>Sofia</td>
<td>2043</td>
</tr>
<tr>
<td>3</td>
<td>Kostas</td>
<td>1876</td>
</tr>
<tr>
<td>4</td>
<td>Stergios</td>
<td>1808</td>
</tr>
<tr>
<td>5</td>
<td>Dimitris</td>
<td>1737</td>
</tr>
<tr>
<td>6</td>
<td>Pavlos</td>
<td>1598</td>
</tr>
<tr>
<td>7</td>
<td>Mirto</td>
<td>1555</td>
</tr>
<tr>
<td>8</td>
<td>Thanasis</td>
<td>1528</td>
</tr>
<tr>
<td>9</td>
<td>Giannis</td>
<td>1498</td>
</tr>
<tr>
<td>10</td>
<td>Anna</td>
<td>1451</td>
</tr>
</tbody>
</table>

**Figure 13: Example of Player Ability Rankings for a PLO**

Game scenarios also form rankings due to the ratings collected from all players that have experienced them. For a specific PLO, such a ranking list consists of scenarios across all registered PsL games realizing that objective. An examination of these rankings could provide information about the scenario features that lead to the enhancement of prosocial behavior. Another interesting finding might concern the types of games that lead to the students expression of prosocial behavior.
Once enough data for the performance of a population of players are collected, the probability distribution of prosocial ability can be estimated. From that, we can draw conclusions about the prosociality level of different groups of students (e.g. classrooms or schools), or even the whole population of PsL players. We can also assign discrete categories to the continuous value ranges of the distribution, in order to provide a more qualitative description of the prosociality state of the group in study. Figure 15, shows such a discrete categorization for players according to their Elo ability estimation in a game. In this case, the ability of the population of players follows a Gaussian distribution and most players preserve an ability ranking near 1500.

### 4.3.4 Prosocial Profiles

The prosocial profile of each user consists of an aggregation of his/hers ability estimations for all of the ProsocialLearn’s learning objectives. For providing a reliable measurement, the ability of a player for accomplishing a certain PLO should be computed throughout experiencing many games and gaming sessions. Each game contains a pool of game scenarios offered for the purposes of offline adaptation. In turn, each game scenario corresponds to a specific PLO. When the teacher decides to use the PsL platform, he or she initially specifies the desired PLO for training. Next, a selection of a PsL game that supports that PLO follows. The student plays the game one or more times and his/hers prosocial ability is built through the experience a series of scenarios.
Figure 16: Example of a player’s prosocial profile

Figure 16 depicts an example of a prosocial profile for a PsL player. The profile consists of ability measurements for the 36 prosocial learning objects mentioned in Section 4.1. The values of the PLOs are colored using a color gradient, aiming to show those PLOs that the student performs similar compared to the majority of the population, those ones that he or she excels at, and those that might be correlated to deficits in prosocial behavior. As we have discussed in Section 4.1, the PLOs are divided into three distinct classes, namely, friendship skills, cooperation skills and feelings skills.

Similarly to a prosocial profile for a player, the prosocial profile of a game is the aggregation of the ratings for all the scenarios that it supports. In the same manner as the player profile updates, the scenario ratings are computed by the experience (outcomes) that players have with the game’s scenarios.

An important issue to consider is the method of validation for the constructed prosocial profiles. In the simplest case, the teacher’s overall experience with the students can be used as a criterion for the correctness of the PsL profiles. Another method of validation could be using a certified questionnaire that is applied in psychological, pedagogical or social sciences in order to detect problems in the prosocial behavior of children. If the outcomes of the validation process are in line with the PsL profiles, then (as we already mentioned in the previous paragraph) a student’s prosocial profile may also be used for the same task of indicating problems, patterns, or progress concerning his or hers prosocial behavior.

4.4 Matching Players to Scenarios

Before each game begins, the offline adaptation mechanism can determine the proper scenario for the player, namely, the one that preserves those conditions that maximize his or hers prosocial ability for the current PLO. As we mentioned in the previous sections, the new rating for a player depends both on his hers past rating and the score that corresponds to the outcome for the current prosocial task. Similar to the typical use of the Elo system for providing interesting matchmakings between contestants, it is our intention to match a player of a certain prosocial ability to a scenario of similar rating for a PLO. This is intended for the following reason: we want the player to experience a scenario that is not too difficult, nor too easy to accomplish. That way, we aim to achieve true enhancement of prosocial behavior, compelling the player to accomplish a task that lies on the limit of his or hers prosocial ability.
Following (Klinkenberg et al., 2011), we can determine the proper scenario for the player in real-time by selecting the one which rating has the minimum distance to the player’s rating. In mathematical notation, the rule that gives the id for the next scenario reads

\[ id = \arg \min_s |P_s - P_{est}|, \]

where \( P_s \) is the rating of a scenario and \( P_{est} \) is a stochastic estimate of the prosocial ability of the player, given by

\[ P_{est} = P_p + \ln \frac{r}{1-r}, \]

where \( P_p \) is the true prosocial ability of the player, and \( r \) is a real number drawn from a normal distribution. The reason that the matching between players and scenarios is done stochastically, is solely to avoid repetitions of the same scenario over a short period of time.

As a final statement, we should note that adaptation accuracy is directly proportional to profile quality. This means that if the profile that is being built is representative of the player, then the adaptation is guaranteed to make the right adjustments to the game, i.e. the adjustments that better suit the user’s preferences. This correlation between adaptation effectiveness and player modelling is well acknowledged in the context of adaptive hypermedia (Brusilovsky, 2001). Concerning the modelling scheme that we have adopted, where each user profile is composed of ability estimates, the profile quality depends on the level of convergence of the ability ranking algorithm. Hence, having properly defined the learning speed of the algorithm, it will eventually reach a representative user profile in a certain period of time, after which the adaptation mechanism can give accurate scenario selections.
5 Online Adaptation

5.1 User Engagement & Prosocial Behavior

Prosocial behaviour can be motivated through various means. Each prosocial behaviours may have unique characteristics, however all involve intentional actions that help contribute towards the benefit of others. Some early studies in the 80s have suggested that motivations for prosocial behaviour arise out of a pursuit of personal gain or at the least, an attempt to avoid loss. Clary & Snyder (1991) have stated that individuals will engage in prosocial behaviour to the extent that the latter will provide them with motives to continue to do so. According to (Clary et al, 1998), certain motives are essential for satisfaction and enjoyment to be derived from prosocial behaviour.

In order to investigate in more detail the effects of volition and autonomy on activities and behaviour, such as helping others, donating or other forms of prosocial behaviour, as well as identify the motivating factors that will drive individuals to engage in these activities (i.e. spend substantial time and energy engaging in these behaviours), the theoretical framework of self-determination theory (SDT; Deci & Ryan, 2000) has been applied (Weinstein & Ryan, 2010). SDT is a theory of human motivation that is principally concerned with the potential of social contexts to provide experiences that satisfy universal human needs (Deci & Ryan, 2000). Within SDT, personal well-being is believed to be enhanced when one’s actions and interactions satisfy the fundamental psychological needs of competence (sense of efficacy), relatedness (social connectedness) and autonomy (volition and personal agency). The three are defined as basic psychological needs that are cross-developmentally and cross-culturally required for psychological growth, integrity and subsequently, well-being (Deci & Ryan, 2000). Activities can therefore be perceived in terms of so-called psychological nourishments, which characterize these activities as being inherently enjoyable or fun. These nourishments then proceed to influence the effects these activities have on motivation and well-being.

5.1.1 Need satisfaction in prosocial behaviour

The SDT distinguishes motivations for engaging in prosocial behaviour in autonomous and controlled motivations (Deci & Ryan, 2000). Motivation for engaging in actions that are experienced as emanating from personal choice or volition for acting are characteristic of the former category, while engaging in activities or behaviour from a desire to maintain self-esteem, please others or obey demands, among other reasons are prime examples of the latter. SDT’s distinction between autonomous and controlled motives may be particularly relevant to prosocial behaviours, which can stem either from personal values and initiatives, or from external pressures or rewards, and thus might be expected to vary in their autonomous versus controlled motives. Additionally, SDT distinguishes activities into intrinsically and extrinsically motivated. The act of engaging in activities for their own sake, their inherent satisfaction are classified as intrinsically motivated, while pursuing activities or behaviours to access desired end states or avoid aversive ones constitute extrinsic motivation. Activities can be attributed with properties of both categorizations, for example, when activities are characterized by salient rewards, punishments and self-esteem pressures; they foster controlled forms of extrinsic motivation.

Throughout the psychological literature, advocating the use of external controls to prompt prosocial action has both been encouraged (Krehbiel & MacKay, 1988; Sobus, 1995) and discouraged (Frey, 1997; Frey & Jegen, 2001), mainly as factors that can enhance extrinsic motivation, such as rewards, pressures or evaluations, typically undermine intrinsic motivation. The latter opinion therefore
argues that the use of rewards or the imposition of requirements can undermine prosocial engagement. When compared with individuals pursuing an activity for extrinsic reasons, intrinsically motivated actors have been found to enjoy an activity more, to be more creative, demonstrate superior cognitive flexibility, process information more carefully and incur greater psychological and physical benefits (Ryan & Deci, 2000). A similar trend has been observed in autonomous vs controlled motivation with respect to prosocial behaviour: when prosocial behaviours are controlled, i.e. their causality is externally perceived, the satisfaction of the basic psychological needs is diminished. The controlled actor does not feel that he/she “owns” the act (deCharms, 1968; Deci & Ryan, 1985b), therefore increasing saliency to the fact that his/her autonomy is being undermined. In other words, while good is being done through the action, the actor does not feel responsible for what is being achieved. Findings support the idea that autonomous motivation for engaging in prosocial behaviour yields benefits for both helper and recipient through greater need satisfaction, i.e., when volitional or autonomous, prosocial behaviours have the capacity to facilitate satisfaction of each of these basic needs (Gagné, 2003).

In conclusion, the above paragraphs highlight the importance of distinguishing autonomous from controlled motivations when examining prosocial behaviours and their outcomes. Autonomous engagement in prosocial behaviour contributes to the satisfaction of all three basic needs, therefore yielding greater well-being benefits than controlled motivations, which may impede these needs satisfactions. Prosocial actors autonomously motivated to engage in prosocial behaviour experience a greater sense of personal volition and therefore identify more personally meaningful reasons for engaging in the prosocial act. This will in turn motivate these actors to potentially put greater effort in carrying out the act, be more enthusiastic about it, express more care and respond in more appropriately to the prosocial behaviour recipient’s wishes (Weinstein & Ryan, 2010). Therefore, in modelling prosocial acts as in-game responses reproducible by game players in video games, game developers should keep in mind the following conclusions, in order to ensure that the actual act of prosocial behaviour (regardless of game mechanics, input and the theme of the game) are perceived as enjoyable, and will therefore motivate players to engage in these acts out of volitional choice in the future:

1. Prosocial acts foster well-being in the actor when these acts are autonomous, but not when they are controlled.
2. These positive effects of autonomous versus controlled helping on well-being would be mediated by the satisfaction of basic psychological needs.
3. Recipients would benefit most, when helpers are motivated autonomously rather than by control.

After raising the issue of when prosocial behaviour is perceived more enjoyable and therefore increases its chances of being volitionally chosen by players as a means to advance the gameplay, we will examine actual game play under the theoretical framework of the SDT in the following paragraphs

### 5.1.2 Need satisfaction in video games

Video games can positively influence both psychological and physical well-being, as has been demonstrated in recent related scientific literature (Baranowski et al., 2008). Video games motivate a remarkable amount of goal-oriented behaviour. SDT’s distinction between intrinsically and extrinsically motivated behaviours can be particularly relevant to video games, as the latter are better suited to be characterized by this distinction, and carry the potential to enhance intrinsic
motivation and short-term well-being for as long as they provide players with experiences that satisfy universal psychological needs. This means that games are generally more or less appealing, and have a greater or lesser influence on player well-being based on the in-game experiences they provide, and how these fulfil fundamental psychological needs. Need-satisfying experiences within video game play would contribute to (Przybilski & Rigby, 2010):

- Intrinsic motivation for play;
- Immersion in gaming environments; and
- Positive short-term shifts in player well-being.

The motivational affordances provided by games are first and foremost factor that relates to their appeal, over and above the content of games and individual differences. This means that, contrary to popular belief, violent game content (i.e. the re-enactment of in-game behaviour that is primarily antagonizing prosocial behaviour in the related scientific literature; Anderson et al, 2010) is not, on average a significant motivator to play games (Weinstein & Ryan, 2010). The broader appeal of games is rather based on the psychological need satisfaction game play can provide, and motivational processes have proven to be robust predictors over and above differences in player demographics, applying across game genres and content. Need satisfaction in video games can be tracked down to the individual fundamental psychological needs described in the SDT:

**Competence need:** Every successful video game excels at supporting the fundamental human need for competence. The pacing of challenges present in the game must be designed so players can continually experience enhanced competence as they progress in the game, with challenges becoming increasingly more difficult as player ability increases. This process of balancing player skill against game challenges is key to maintaining interest and loyalty of players, as too much of a challenge will almost certainly lead to frustration, while too less of a challenge will lead to boredom. Therefore, key elements to keep in mind when designing in-game activities for players to carry out are skill-graded challenges, positive reinforcement and corrective feedback, features that primarily satisfy player experiences of competence need satisfaction, and thus important elements in motivating play, whilst supporting them in identifying areas they can improve upon. In the context of a prosocial game, engaging in prosocial behaviour can foster competence need satisfaction when designed in a way that the players, as helpers, are acting on the game world in ways that directly result in positive changes (Weinstein & Ryan, 2010).

**Autonomy need:** Satisfying the fundamental need for autonomy can be achieved in games that empower players to shape the game’s narrative. Such games present players with multiple routes to an end in terms of overall achievement; provide players with a wide range of options over multiple game elements; allow players to decide which missions they will pursue and in what order, customize the skills they acquire over the course of the game and how their characters appears. In this respect, modelling prosocial behaviour that is freely chosen as a player character course of action, and is therefore an expression of well-internalized values, will provide opportunities to experience autonomy need satisfaction during the course of gameplay, and in continuation lead to the positive states that follow from it, therefore relating to a positive change in layer affect and enjoyment of the game.

**Relatedness need:** Social interaction has always been an important part of video game play. The online game market currently constitutes one of the largest in the video gaming industry. Online features enable players to develop social bonds, allowing them to cooperate or compete with a
friend sitting next to them, or even tens of thousands of geographically distributed players from all over the world. Again, prosocial behaviour is relevant in that helping another is inherently interpersonal and thus impacts relatedness by directly promoting closeness to others, positive responses from others and cohesiveness or intimacy. Therefore, online prosocial games where players develop bonds from helping one another advance in the game can lay the foundation for satisfying the fundamental human need of relatedness. Care has to be taken to allow in-game helping behaviour to be autonomously motivated (as described in the previous Section), rather than controlled (i.e. having the player be explicitly told to help another in order to advance, promise of reward or otherwise threaten the player with unfortunate consequences for not helping), as the experience of relatedness is likely to be undermined. The player’s actions will be attributed less to connection, or caring for another player’s well-being in the shared game world and instead to controls that brought about the actions.

The above paragraphs make it clear that all of the three basic SDT needs make independent contributions to game enjoyment, while providing robust estimations of future engagement (Ryan et al, 2006). Additionally, these needs have been found to contribute to the sense of immersion within the virtual game world. Game immersion can be decomposed into three basic subcomponents (Ryan et al, 2006):

(a) Physical presence, i.e. the feeling as if one is actually present in the game world;
(b) Emotional presence, i.e. the feeling as if game events carry real emotional weight; and
(c) Narrative presence, i.e. being personally engaged and invested in the game’s story,

Video game play that satisfies the needs for competence, autonomy and relatedness has been found to robustly increase the players’ sense of immersion, across different game types (Ryan et al, 2006) and game contents (Przybylski et al 2009). When the game invests substantial effort into satisfying player needs, players are in turn more prone to be embedded in the emotional, physical and narrative elements of the game world. Particularly, experiences of competence and autonomy need satisfaction have been found to relate to greater immersion in the game world, experience greater game enjoyment, increase self-esteem and thus also the likelihood of reengaging the game in the future. It is because of this link between competence and autonomy to post-play well-being, which relate these factors to more hours of play each week (Weinstein & Ryan, 2010).

Finally, mastery over the game controls and mechanics has been found to be necessary to access these need-satisfying opportunities provided by games. As players don’t have a fully intuitive sense of orientation and action in virtual environments, they must invest time and energy to master the game control interface, all while learning the mechanics of the game. This carries practical implications to the engagement and motivation in the game, as mastery of controls plays an important role, largely as a necessary, but not sufficient condition for achieving psychologically need-satisfying game play (Weinstein & Ryan, 2010). Achieving an intuitive grasp of game interfaces is not in itself sufficient for enhancing player motivation, or well-being, and players generally do not enjoy games that are characterized by what is known as a ‘steep learning curve’. However, player preservation is motivated by the potential to unlock what is hoped will be future fulfilling game experiences that meet player’s psychological needs. Therefore, mastery of the game’s control scheme does indeed relate to the enjoyment to be had playing the game, but it does not account for unique variance in player motivation and well-being when in-game need satisfaction is considered. Mastery over a particular prosocial skill or behaviour may require corrective feedback, which the player can utilise to support their development.
In order to summarize all of the above, game developers seeking to create engaging prosocial games that motivate players to invest more time playing and volitionally return to the game world for more action, should focus on incorporating features that cater for the satisfaction of the three fundamental psychological needs of competence, autonomy and relatedness. The first two are especially important in contributing to the immersive components that will keep players fully embedded into the game’s narrative theme. In terms of the prosocial actions modelled as in-game behaviours, advocating the use of external controls, such as rewards, punishments and self-esteem pressures, to prompt prosocial action is generally discouraged, as players must be allowed to act on their own volition, in order to experience greater enjoyment out of carrying out the act itself. This directly relates to a successful achievement of game immersion, thus keeping players emotionally invested in the well-being of other characters in the game world. Finally, the actual game controls and mechanics should be designed in analogy to the extent that the game attempts to satisfy player psychological needs, i.e. the greater the eventual rewards, the more players will appreciate in retrospect their time spent on achieving an intuitive grasp of the game controls.

5.2 Input from Multimodal Fusion

Engagement is an essential element of the player experience, and the concept is described in various ways in the literature. Terms like engagement, immersion, fun and flow are commonly used to motivate the development of serious games both for education and more generally. By providing greater levels of engagement and enjoyment, players will become immersed and thus more committed to the learning experience. In education, for example, this suggests that greater levels of engagement will provide a significant educational advantage over more traditional educational approaches. The focus is thus to investigate the factors that trigger player engagement and support players’ aspiration to continue playing, what they do while they have the desire to continue, and which emotions, affect and experiences characterise engagement. A player could, for instance, be motivated to begin playing due to boredom, but it’s when the player becomes excited and wants to continue playing that engagement is experienced.

Overall, engagement is a complex concept, encompassing varying components. In the following section we briefly describe key concepts related to the general notion of engagement, namely cognitive engagement, behavioural engagement and affective engagement. We suggest that differentiating engagement into these components, “Behavioural Engagement”, “Affective Engagement” and “Cognitive Engagement” allows for more targeted measurement of the different aspects of engagement and also suggests a variety of approaches that might be required to better quantifying the various components of engagement.

5.2.1 Engagement Types

- **Behavioural engagement** is defined as focused activity on a task, with a typical measurement being time on task (e.g. measure the responsiveness from the moment that guide suggests direction and muscle makes the decision). Behavioural engagement might thus be influenced by both motivational characteristics of situation and personality.

- **Cognitive engagement** is defined as mental activity associated with presented content, and is measured by successfully achieving the desired goal of the game (e.g. 6 out of 6 diamonds collected), or by pre and post testing of outcomes. Other, more objective, tests related to physiology, such as eye-blink and scanning, have also been suggested to measure this aspect of engagement. Once again individual characteristics might impact on this type of engagement and dynamic difficulty or adaptive stair casing approaches have been used to try
and ensure that challenges meet the players changing ability levels (skill levels). This a key relationship often attributed to players being in a high zone of performance.

- **Affective engagement** relates to emotional responses of players to game content. Affect is in itself a complex concept that can be further differentiated into emotion, cognitive and affective processing elements. Affective processing might be measured in response to a simple emotional cue, and can be positive or negative.

### 5.2.2 Fusion Architecture

We developed a multimodal fusion architecture that uses stacked generalization on augmented noisy datasets and provides enhanced accuracy as well as robustness in the absence of one of the input modalities. Moreover, we designed a list of body actions and facial expressions commonly encountered in a typical game, eliciting emotions based on Ekman’s discrete categorization theory (Ekman & Friesen, 1978). Based on this list of emotions, a bimodal database was created using Microsoft’s Kinect sensor, containing feature vectors extracted from users’ facial expressions and body gestures.

#### A. Facial Expression Stream

As described in D3.1, we distinct our extracted AU features in two categories, mainly upper face and lower face AUs. More specifically, we employ two three-layer neural networks with one hidden layer to recognize AUs through a number of parameters defined by low-level features extracted for the upper and lower face regions. The ultimate goal of identifying and extracting AUs is to classify expressions under a certain emotion category. We further concatenate the two neural network posteriors in a unified representation, and train an extra layer on top of them as shown in the upper left part of Fig.17.

#### B. Body Motion Stream

In order to combine information extracted from body stream, we propose a two-layered network in which we have stacked seven NNs, six at the first layer and one at the second layer. Each layer is trained separately, starting from base layer and moving up to the second, with no feedback from the higher layer to the lower layer. Each NN of the first layer receives as input the features of a different group of features. Then, the output probabilities of the first layer are fed as input to the second one and a separate NN is trained. The output probabilities of the second layer constitute the classification result of the body motion analysis mono-modal classifier as shown in the bottom left part of Fig.17.

#### C. Dynamic Fusion

In previous deliverable D3.2, we have presented that deep learning networks can be applied at feature level as well as at decision level, being trained directly on raw data or decisions accordingly. In this direction, we employ a late fusion scheme, where each intermediate classifier is trained to provide a local decision. In terms of affect, local classifiers return a confidence as a probability in the range of [0, 1] in a set of predefined classes. The local decisions are then combined into a single semantic representation, which is further analysed to provide the final decision about the task. The aforementioned scheme for late fusion is illustrated in Fig.16.
5.3 Enhancing User Engagement through Positive Reinforcement

Engagement refers to the behavioral intensity and emotional quality of a person’s active involvement during a task (Reeve et al., 2004). According to Welborn, in school settings, engagement is important because it functions as a behavioral pathway by which students’ motivational processes contribute to their subsequent learning and development (Welborn, 1991). In fact, teachers often look to students’ preferences and engagement signals to gain the perspective they need to adjust instruction. During instruction, if students display strong and consistent signs of engagement, this confirms that what the students are presently doing aligns well with their inner motivational resources (Reeve, 2015). This means that if the student expresses a consistent level of engagement, ideally, he or she can perform the task of interest without the need of external pressure or desire for rewards. As also stated in (Reeve, 2015), in some situations, teachers can use a disengagement signal as a trigger to change the flow of instruction away from that which neglects a student’s motivation and toward that which involves and vitalizes it.

A very simple form of intervention that can be used in order to motivate students towards completing their task, is the common practice of positive reinforcement. In general, reinforcement is the process of strengthening a person’s behaviour as a consequence of applying an antecedent stimulus, and can either be positive or negative, depending on whether stimulus is added or removed following correct behaviour. Positive reinforcement occurs when an event or stimulus is presented (positive) as a consequence of a behaviour and the behaviour increases (reinforcement) (Flora, 2004). Types of reinforcers include:

- Social Reinforcers
• Activity Reinforcers
• Tangible Reinforcers
• Token Reinforcers

Social reinforcers are those that express approval and praise for appropriate behaviour. Activity reinforcers are those that allow students to participate in preferred activities. Tangible reinforcers include toys and awards. Finally, token reinforcers are those involving awarding points or tokens.

We should mention here that, positive reinforcement can increase the probability of not only desirable behaviour but also undesirable behaviour. For example, if a student whines in order to get attention and is successful in getting it, the attention serves as positive reinforcement which increases the likelihood that the student will continue to whine. In order to avoid such cases, an effective intervention should follow specific guidelines for the appropriate use of reinforcers. Some guidelines are given below.

1. Reinforcement must be consistently delivered
2. Reinforcement must be delivered immediately
3. Improvement (and not perfection) should be reinforced
4. Social reinforcement can be combined with other types of reinforcement
5. Social reinforcers must not be ambiguous
6. Reinforcement must be age-appropriate

5.4 Supporting player engagement and mastery through corrective feedback

As described in detail in D2.6 and D4.3, corrective feedback forms an integral part of the behavioural model of teaching prosocial skills. It provides specific, often textually represented, instructions to players to correct their behaviour and supports them in identifying and paying more attention to the outcome of their actions and what they ought to do to be more successful. Reddy(2012) cites a general rule for a ratio of 5:1 between positive reinforcement and corrective feedback, that is for approximately every five instances of positive reinforcements, a corrective feedback element can be given. This is primarily to ensure the player is not discourages. In section 4 of D4.3, a more sophisticated and adaptive model for this ratio is provided which takes into account the frequency of a prosocial skill being displayed by the player, based on whether or not the skill is being demonstrated ‘just right’, ‘too much’, ‘too little’ – or if the ‘opposite’ skill is being displayed (for example instead of cooperating too little, the player deliberately takes a competitive and uncooperative stance.

5.5 Learning User’s Engagement Profile

As we described in Section 5.2, in a typical duration of a game, real-time engagement estimates are computed by the multimodal fusion mechanism. These estimates concern the experience that the player has with a positive reinforcement element. In a typical duration of a game, the player may encounter many of these elements, thus, recordings for the player’s engagement measurements must be kept for each element in order to infer the user’s preferences and execute online adaptation.
For a certain PsL game, the recordings of user engagement for his/her experience with each positive reinforcement element encountered, are kept into buffers of equal size. These buffers have a fixed size and are circular, meaning that when the buffer is filled up and a new engagement measurement occurs, then the oldest stored value is discarded and the other values are shifted towards the end in order to create an empty space in the beginning for the new value. The positions in the buffer account for the order in time in which the measurements occurred at. Thus, by shifting positions towards the end of the buffer, we keep only the most recent recordings to represent the user’s engagement state. The collection of the engagement recordings for every element of the game, consist the player’s engagement profile for that game.

In order to learn the user’s engagement profile, the real-time adaptation module must provide an online learning mechanism. This mechanism will gradually learn the user’s engagement profile, considering both exploiting a user profile in order to satisfy the need for personalization, but also supplying the user with new elements to experience in order to take into account the case of behavioral shifts over time. This situation is known as the exploration-exploitation strategy. A special case for utilizing such as strategy is the multi-armed bandit problem. Having its roots in probability theory, the multi-armed bandit problem describes the case in which a gambler has to decide among a number of slot machines (i.e., one-armed bandits) which of them to play, how many times to play each, and in which order to play them. After each play, each machine provides a reward draw from a probability distribution that represents the machine’s functionality. The problem’s objective is to maximize the sum of rewards earned through a sequence of lever pulls. Thus, the gambler at each trial must decide about exploitation of the machine that has the highest expected payoff and exploration to get more information about the expected payoffs of the other machines. This trade-off between exploration and exploitation is also faced in reinforcement learning. This similar problem consists of an artificial agent that simultaneously attempts to acquire new knowledge (called “exploration”) and optimize his or her decisions based on existing knowledge (called “exploitation”). The agent attempts to balance these competing processes in order to maximize its total reward over a time period.
Epsilon-greedy or $\varepsilon$-greedy, is probably the simplest action–selection strategy to solve the multi-armed bandit problem. It is mainly used in reinforcement learning for online learning tasks. At each iteration, the algorithm selects a random solution from the population with a certain probability $1 - \varepsilon$, while with the remaining probability $\varepsilon$ selects the solution with the highest ranking. The evaluation process ends after a number of evaluations are completed.

```
1: procedure SELECT(P)
   $\triangleright$ $P$ is the population, $p$ is an individual of $P$
   $\triangleright$ $e(p)$ is the number of evaluations of $p$
   $\triangleright$ $f(p)$ is the fitness of $p$
2: best = arg max$_p \ f(p)$:
   $\triangleright$ If the best individual is not good enough
      $\triangleright$ return an individual never
      $\triangleright$ evaluated (explore)
3: if ($f$(best) $\leq$ $\theta_{best}$) then
4:    return $p|e(p) = 0$;
5: end if
   $\triangleright$ If all the individuals have been evaluated
      $\triangleright$ return the best individual (exploit)
6: if ($\exists p \in P|e(p) = 0$) $\land$ ($e$(best) $< \theta_{eval}$) then
7:    return best;
8: end if
9: if (rand() $< \varepsilon$) then
10:   return $p|e(p) = 0$;
   $\triangleright$ Return an individual never evaluated (explore)
11: else
12:   return best;
   $\triangleright$ Return the best individual (exploit)
13: end if
14: end procedure
```

**Figure 19:** A variation of $\varepsilon$-greedy used for online neuroevolution in a racing game

Many variations of the original $\varepsilon$-greedy algorithm have been adopted by research studies for solving online learning tasks. Tokic developed “Value-Difference Based Exploration” or VDBE, for balancing the exploration-exploitation tradeoff in reinforcement learning tasks (Tokic, 2010). Bouneffouf et al., created a context-aware version of the algorithm for use in mobile recommender systems (Bouneffouf et al., 2012). In their implementation, $\varepsilon$ is evaluated based on the situation that the learning task is performed in. Cardamone et al., developed an extension called $\varepsilon$-greedy-improved, and used it for boosting the performance of an online neuroevolution process in order to build up a driver behavior for a racing game (Cardamone et al., 2010).

An $\varepsilon$-greedy variation that has recently attracted a lot of interest, is the $\varepsilon$-decreasing (ED) strategy. The technique is similar to the $\varepsilon$-greedy strategy except that the value of $\varepsilon$ decreases as the experiment progresses, resulting in highly explorative behavior at the start and highly exploitative behavior at the finish. Generally, the $\varepsilon$ in ED method gets decreased according to a monotonously decreasing function (Takahashi, 2011). An example of such a function is given below

$$e(t) = 0.5/(1.0 + rt)$$
where $r$ is the algorithm's parameter for controlling the rate of decrease.

The $\varepsilon$-decreasing strategy fits the requirements of the PsL online adaptation for learning players' engagement profiles. Initially, there are no engagement profile data for the algorithm to consult for a real-time selection of a positive reinforcement element. During that time period, the $\varepsilon$-decreasing algorithm expresses highly explorative behavior in order for the player to experience as many different elements as possible. In this stage, the algorithm tries to maximize the coverage of the search space (Shani, 2011) (i.e., experience of available game elements). When few engagement data occur in the player's profile, the algorithm continues to keep an explorative behavior in order to increase its confidence for the profile's appropriateness. When the profile is filled with engagement values (i.e., engagement buffers for each positive reinforcement element are filled), the $\varepsilon$ value favors mainly exploitation, but does not terminate its exploration phase. The focus on exploitation in that stage, is of major significance in order to provide reliable results for predicting the actual state of the user. Additionally, the retention of the exploration phase guarantees adaptivity to user behavioral shifts over time. That way, the personalization capabilities of the system are enhanced.

### 5.6 Adjusting Game Elements for Positive Reinforcement and Corrective Feedback

When the player’s engagement profile reaches a level of maturity (i.e., when enough engagement estimates have been collected), the online adaptation module must analyse it and infer the preference of the player for a game adjustment that is considered to enhance his or hers engagement level. As we mentioned in the previous section, for every game, the user’s engagement profile is composed of estimates from multimodal fusion, which are stored into buffers according to the positive reinforcement game element that the user experienced. When these buffers are filled up with values, a mathematical measure should process them and derive a score or ranking for each of them. These rankings would imply the elements that mostly contribute to the process of enhancing the user’s motivation. Therefore, it is our intention to incorporate such a measure for analysing a player's engagement profile and inferring the player’s preference for motivation regarding the accomplishment of a particular task.

Many techniques for analysing signals and time-series have been adopted in a variety of scientific areas, including electrical engineering, physics, bioinformatics, economics, neurophysiology and psychology. More specifically, the simple measures of local minima and maxima, the time between them, as well as inter-spike intervals, have been used by Kougioumtzis et al. in order to discriminate preictal stages (Kougioumtzis et al., 2007). Long-range correlation and other measures of stride-interval dynamics have shown to be effective in detecting neurological abnormalities (such as Huntington’s disease (HD), Parkinson’s disease (PD) and ALS) and in the quantification of their severity (Gavrishchaka et al., 2015). Trend, seasonality, cycles and autocorrelation have been used for detecting behavioral patterns in psychological research (Jebb et al., 2015).

A simple assumption that we can adopt and realises Reeve’s view of correlating the student’s engagement with his/her inner motivation, is that consistency is expressed through analysing engagement history. For that reason, any valid measure for exploring the user’s engagement profile, must take into account historical data about user’s engagement measurements. Also, we should consider the fact that user interests might change over time. Thus, in order to account for user’s behavioral shifts, it is necessary to consider that the most important estimates of user engagement are the last ones recorded. To further extend and improve that statement, we can consider a weighting scheme that assigns the largest weights to the most recent recordings and proportionally decreases the weights while time passes. Thus, such a scheme sets the focus mainly on present
recordings, but also, does not neglect the player's engagement history. That way we ensure that our measure will preserve both robustness to our estimations and capability for properly handling user's behavioral shifts.

A well acknowledged weighting scheme that utilizes the above requirements can be found in the context of Reinforcement Learning, where the computation of the overall reward of a learning process relies on gradually discounting rewards over time. In such a case, one considers immediate rewards to be more important than later rewards and expresses this fact with the notion of a discount factor (van Hasselt, 2010). The value for this factor must be set to be lower than 1. Using such a definition, we ensure that the value of the reward is finite at any step of the learning process. Therefore, this definition bounds our computation and ensures that the resulting value will not become very large (i.e., the variable will not get out of range). Another interesting characteristic for the factor is that it controls the convergence of the learning process. For example, in some cases, a lower discount factor might imply faster learning. In any case, agents with short term objectives might use lower discount factors than agents with long term objectives.

More specifically, for non-episodic tasks, i.e. tasks that do not have an ending state, the overall reward over time for an agent is given by the following rule

$$ R = \sum_{t=0}^{\infty} y^t r_{t+1}, \quad 0 \leq y \leq 1 $$

where $y$ is the discount factor and weights the recordings in decreasing order over time.

Using the above measure on player's engagement estimates, we can therefore create rankings of game elements that the player has experienced over a period of time, through receiving positive reinforcement and corrective feedback. According to Reeve, if those values show signs of strong and consistent engagement on a certain task, then they may be considered to express the player’s inner motivations for the task. By additionally setting the main focus on present recordings, we can assume these estimates to implicitly express the player’s current preferences for motivation. As a result, a ranking of game elements can be formed. Therefore, we can adapt the game properly, by selecting the game element with the highest ranking.

An example of the proposed approach for adjusting the reinforcement game elements is given in the following table. A player's engagement values are recorded throughout a series of game sessions and the engagement profile built is the one presented in Figure 18. The player's preference estimation is then computed according to the formula above, using a discount factor of 0.4. The bold row in the table shows the element that is considered to produce the highest engagement experience to the user. For that reason, it will be selected by the online adaptation mechanism as the next element for expressing positive reinforcement.

<table>
<thead>
<tr>
<th>Positive Reinforcement</th>
<th>Player's Preference Estimation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Typical Well-Done Message</td>
<td>0.932</td>
</tr>
<tr>
<td><strong>Fireworks Exploding</strong></td>
<td><strong>1.426</strong></td>
</tr>
<tr>
<td>Blinking Congratulations Message &amp; Player's Score Increase</td>
<td>1.064</td>
</tr>
</tbody>
</table>
This model is compatible with the computation model of positive reinforcement and corrective feedback provided in D4.3. Whereas player preference estimation thresholds can be set against the expected ranges of displaying the prosocial behaviour for the specific individual ranging from ‘too little’, ‘just right’, ‘too much’ and the opposite.
6 Conclusions

This deliverable has provided an initial description of the ProsocialLearn Adaptation Manager (PAM). It presented two distinct mechanisms, namely, offline and online adaptation, aiming to enhance prosocial behaviour in children using the ProsocialLearn platform. The theoretical justification and technical analysis for both mechanisms has been explained in detail throughout the sections of this document.

More specifically, we explained how the offline mechanism tries to enhance prosocial behavior through game scenarios exemplifying prosocial learning objectives (PLOs). The diversity of scenarios concerns certain conditions or factors that favour the expression of prosocial behavior within a gaming experience. These factors might include among others, utility and importance of a situation outcome, NPC competence and risks inherent in a situation unfolding in a specific manner. Such factors can directly relate to the provision of appropriate corrective feedback, positive reinforcement and scenario rewards, as dictated by the actual game design. Whilst having a set of predefined scenarios, the offline mechanism uses an ability ranking system in order to assign rankings to both players and scenarios. For the players, these rankings correspond to their ability in expressing prosocial behavior. A matching between players and scenarios, according to their ranking, is used to personalize the game for each player.

On the other hand, the online mechanism aims to complement the personalization process by enhancing the players’ engagement through the provision of positive reinforcement. More specifically, this mechanism monitors the player’s engagement estimates (given as input from T3.2) and constructs player profiles containing previous recordings of their engagement across multiple prosocial games. A measure for analysing those signals is used to resemble the preference of the user for specific game elements and as a result, adjust the game accordingly in order to maintain engagement levels high, in order to maximize the potential of achieving prosocial learning outcomes.

The next step for the task of Intelligent Adaptation and Personalization is to use the adaptation mechanisms in real-life/near-operational settings where real students will play prosocial games while being monitored by the observations acquisition platform developed as part of activities carried out in WP3. The purpose will be to validate the PAM’s efficacy for personalization and behavior enhancement.
7 References


[http://rageproject.eu/](http://rageproject.eu/)
Appendix A - Running a simulation

In order to design and prepare the components of the Adaptation Manager at a significant pace to closely follow developments of the entire PsL platform, and also determine the manager’s positioning within the PsL architecture, we have created a simulation consisting of a dedicated, approximation of the PsL service that will communicate with the game, the proposed Adaptation Manager module, and a basic definition of the user profiles to store data. Within this simulated environment, the communication between the three prototype modules is done via a server-client protocol that utilizes a web sockets interface. Initially, the external game connects to our dedicated PsL service. Immediately after the handshake, all the necessary information about the game elements supported is sent, along with the initial scores of the engagement-elements used for triggering online adaptation. The PsL service then proceeds to deliver the information to the Adaptation Manager, also sending the active user’s profile in the process. After that, the Adaptation Manager is initialized by constructing the main data structures for supporting both offline and online algorithms.

After accumulating all required information, and determining the proper data structures, the Adaptation Manager proceeds to make the first adjustment (offline adaptation) for determining the game scenario, as well as proper scenario factors, utilizing the internal ability ranking algorithm. The communication continues with the game triggering adaptation signals and the Adaptation Manager adjusting game elements that target ways to enhance the user’s engagement in ways that will increase his/her motivation to continue playing. Care has been taken to allow updates to take place immediately after each user engagement measurement by the fusion module. All communications between the game and the Adaptation Manager use the PsL service as an intermediary. The architecture of the simulation modules is displayed in Figure 20.

Figure 20: Simulating the adaptation procedure in PsL
Appendix B – Example game scenario adaptable factors based on computational model of Trust

Trust computational model

Having determined the situational trust $T_x(y, a_x)\uparrow$ in $y$, $x$ can consider whether or not to cooperate, or otherwise engage in trusting behavior. To determine the answer, threshold values for trust must be defined. Marsh (Marsh, 1994) defines the cooperation threshold as a barrier for overcoming trust issues. If the situational trust is above a cooperation threshold $W_x(a)\uparrow$ cooperation will occur, otherwise cooperation will not occur. This is demonstrated in the following Equation:

$$T_x(y, a)\uparrow > W_x(a)\uparrow \rightarrow f(x, y, a) \quad (1)$$

where $f(x, y, a)$ denotes the process of $x$ trusting $y$ to cooperate for situation $a$. Marsh models the cooperation threshold as a subjective measure tempered by objective beliefs. More specifically, the threshold depends on the risk $x$ perceives on taking by trusting in $y$, i.e. $R_x(y, a)$, as well as $y$’s competence, perceived by $x$ with regard to situation $a$, i.e. ($C_x(y, a)$). We drop the temporal notation $t$ in the following formulas, as it is implied:

$$W_x(a) = \frac{R_x(a)}{C_x(a) + T_x(y)} \times I_x(a) \quad (2)$$

$T_x(y)$ is defined as $x$’s general trust estimate of $y$, after taking into account all the past relevant data with respect to situational trust values $T_x(y, a')\uparrow$ in which $x$ had a common experience with $y$ in situations similar to $a$.

It is therefore important to note that training children in trust through prosocial gameplay becomes a matter of teaching them to regulate their cooperation thresholds, i.e. know when it is a good idea to cooperate and who to trust. Children who tend to set up very high cooperation thresholds will find it difficult to trust someone to engage in cooperation, and should therefore be taught to lower their threshold, especially when the trusting candidate has proven to be worthy of their trust. Similarly, children with very low cooperation thresholds risk the danger of being manipulated by untrustworthy agents, and should therefore learn to raise their defences with regard to the situation and trusting candidate. Marsh & Dibben (Marsh & Dibben, 2005) define two distinct cases where $W_x(a)$ is not satisfied, namely:

a) when $x$ doesn’t trust $y$ enough in a situation to cooperate; and
b) If $x$’s situational trust in $y$ is negative.

These are the cases of untrust and distrust, respectively. Mathematically, these concepts are noted as follows:

$$0 < T_x(y, a) < W_x(a) \rightarrow \neg f(x, y, a) \quad (3)$$

$$T_x(y, a) < 0 \rightarrow \neg f(x, y, a) \quad (4)$$

Eq. (3) refers to untrust, i.e. when $x$’s trust in $y$ with regard to situation $a$ is less than the cooperation threshold $W_x(a)$, but still larger than 0. Eq. (4) formalizes distrust, i.e. an active judgement in the negative intentions of another, or that $x$ believes that $y$ will act contrary to their best interests in the

---

2 As described in D3.2, General Trust refers to trust in other agents, therefore $T_x(y)\uparrow$ is a measure of how much $x$ trusts in $y$ at a specific time $t$
situation $a$. To deal with these trust issues, several options can be examined (dual options are listed, handling cases of untrust/blind trust respectively):

a) Decrease/Increase the potential utility of $a$ for $x$, $U_x(a)$.

b) Increase/Decrease the perceived competence of $y$, $C_x(a)$.

c) Decrease/Increase the importance of $a$, $I_x(a)$.

d) Decrease/Increase the perceived risk inherent in $a$, $R_x(a)$.

Therefore, in order to train children in maintaining what has been called “moderate levels of trust” and regulating their cooperation thresholds in a situation $a$, they need to learn how to perceive the risk $R_x(a)$ of trusting an agent $y$, as well as how to evaluate $y$’s competence $C_x(y, a)$ with regard to the situation at hand. Also, the perception of importance $I_x(a)$ and utility $U_x(a)$ of a situation can be examined to increase or decrease trust in $y$. For developers to know that the training is working, $x$’s trust in $y$ should follow the rules below:

1. If $x$ helped $y$ in the past, and $y$ responded at this time $t$ by defecting, the trust $x$ has in $y$ will reduce by a significant amount:

   \[ \text{Helped}(x, y, a)^{t-\delta} \land \text{Defected}(y, \beta)^t \Rightarrow T_x(y)^{t+1} \ll T_x(y)^t \quad (5) \]

2. If $x$ helped $y$ in the past, and $y$ responded at this time $t$ with cooperation, then the trust $x$ has in $y$ will remain the same or increase by a small amount:

   \[ \text{Helped}(x, y, a)^{t-\delta} \land \text{Cooperated}(y, \beta)^t \Rightarrow T_x(y)^{t+1} \geq T_x(y)^t \quad (6) \]

In other words, the amount of trust $x$ has in $y$ should substantially decrease following $y$ not reciprocating. However, $y$’s reciprocation merely confirms to $x$ that it was right in helping $y$ in the first place. A graphical representation of the continuous values of trust and where definitions are placed, is shown in Figure 20. A flow diagram covering these two cases is shown in Figure 8.

![Figure 21: Trust continuum, from Distrust to Trust and Untrust. In this diagram, $K_x(y) = 0$ denotes $x$ has not yet met $y$ (refer to D3.2 for more information on the computational model of Trust derived for use with the project’s prosocial games platform).](image-url)
Figure 22: Flow diagram displaying evolution of trust x has in y in general according to y’s reciprocation to a situation.

We discuss strategies on dealing with trust issues in training through online adaptation in the following several paragraphs. These strategies are presented in order of relative straightforwardness, ranging from aspects which are generally measurable to more vague concepts presenting more difficult challenges in estimating a specific measure. Examples are presented where needed, in an attempt to formulate adaptation methodologies into core game design. Offline adaptation procedures should be able to generate scenarios with clear intention on the player following a particular behaviour in order to advance in the game (for example, change the characters in a story to make the process of picking a cooperation partner more difficult). Additional factors may also be discovered during game design for other prosocial skills, and a complete list will be delivered in a future edition of this deliverable.

We note that there is no robust evidence yet, that proves that these regulations will definitely improve children’s assessments of trustworthiness. This is a topic to be explored within the ProsocialLearn project, and evidence is expected to be reported in the next edition of this deliverable.

Decrease/Increase the potential utility of a for x, $U_x(a)$.

Increasing or decreasing the utility of a situation is the most straightforward approach to adapt a particular game’s mechanics to create a variety of situations to present to players. This stems from the fact that utility is not a subjective measure taken from x’s perspective on the situation, but rather the mean overall outcome of all possible outcomes of that specific situation. In terms of the Prosocial World Data Model, Utility clearly relates to the Outcome and Narrative elements, as adjustments made to a situations utility would require a debrief of relevant details of the in-game narrative to help determine an action’s outcome. Since these outcomes should be defined by the game developer beforehand when designing the situation, providing a means to adapt this value could lead to players reconsidering their trust: when a student chooses not to trust in y, while in
truth, y should have been trusted, game-based adaptation has to intervene by reducing the situation’s potential utility, in hopes of making x less skeptical about y’s manipulative intent (i.e. “if I’ve got nothing to lose, there’s no harm in helping out”). We accept through prosocial theory, that behaviours are credited as ‘prosocial’ if they are expressed with intent on helping others (Penner et al., 2005) rather than serving personal gain, and therefore, a zero-utility situation should trigger prosocial behaviour on the part of the child, i.e. help someone in need even if there’s no gain in it. Therefore, such cases where \( U_x(a) = 0 \) should always lean towards prosociality. It is the developer’s task, to clearly communicate the situation’s potential utility to the player before the player is called upon to make a decision on whether to cooperate. A flow diagram showcasing Elo-based PLO skill-based scenario ranking based on the scenario’s Utility factor is presented in Figure 8.

**Increase/Decrease the perceived competence of y, \( C_x(a) \).**

Competence of an agent involves making a judgement about that agent’s ability in a specific situation. It is therefore strongly linked to the NPC and Group elements of the *Prosocial World Data Model*, as a characteristic of these characters’ capacity for carrying out actions and displaying emotions. There are three distinct states of knowledge regarding the competence of an agent y under consideration as a trustworthy party. The first is when y is not known to x, i.e. \( K_x(y) = 0 \) which holds for every situation a. In this case, the perceived competence \( C_x(y, a) \) is a measure of x’s general trusting disposition \( T_x \) moderated by the situations importance to x, \( I_x(a) \). In the following notations, the temporal superscript \( t \) is implied:

\[
C_x(y, a) = T_x I_x(a) \quad (7)
\]

The second case is when x knows y, i.e \( K_x(y) = 1 \), but not in this, or similar situations. In this case, it is impossible to use past actual competence values in similar situations for estimating the competence of y. However, x does by definition share some past competence experiences in other situations (lest we roll back to case 1). This knowledge is moderated by the amount of trust x has in y in general:

\[
C_x(y, a) = \frac{1}{A} \sum_{\beta \in B} C_x^\prime(y, \beta)^T \times T_x(y) \quad (8)
\]

where \( C_x^\prime(y, \beta)^T \) is the experienced competence with y in all situations \( \beta \in B \) in times \( \tau < t \) (the fact that competence value is taken after the situation is concluded), where \( B \neq A \) while considering situation \( a \in A \).

Finally the third case is when x knows y, i.e. \( K_x(y) = 1 \) and y has been trusted in the past in similar/identical situations. Therefore, x has to simply consider all similar situations in the past and assess the outcome to estimate competence. This means, taking all the resultant competence values from past situations and gain an estimate of the competence from these:

\[
C_x(y, a) = \frac{1}{A} \sum_{a \in A} C_x^\prime(y, a)^T \quad (9)
\]

All of the previous Equations end up in the conclusion that the competence value can only be zero or greater.
Figure 23: UML diagram for calculating PLO scores based on the Utility Factor in Trust-themed game scenarios
These metrics can give us, as game developers, some baseline on where the child’s opinion on \( y \)'s competence for situation \( a \) should be, not what the child actually perceives as \( y \)'s competence in the situation. According to this metric however, the child’s choices of trust and distrust can be evaluated by means of accuracy. For example, if the child chooses to cooperate with \( y \) on a situation \( a \), where the estimated \( C_x(a) \) is below an acceptable level of cooperation (set by an external script programmed for that specific agent in the game), we can create an Action Point (AP) for the adaptation mechanism to trigger a future scenario in which \( y \)'s untrustworthy nature becomes more obvious. Furthermore, the impending negative outcome of the task which the child chose to cooperate with \( y \) in order to complete, should be clearly visible with respect to the child’s longed-for benefits from situation \( a \) (i.e. the child won’t get any feedback that may have been anticipated for completing the task). Figure 9 presents a flow diagram showcasing Elo-based PLO skill-based scenario ranking based on the scenario’s NPC-character Competence factor.

![Figure 24: UML diagram for calculating PLO scores based on the Competence Factor of an NPC character in Trust-themed game scenarios. In cases such as this, Competence of \( y \) can be regulated in accordance to Equation 2, i.e. the larger \( y \)'s competence is perceived, the less of a barrier the cooperation threshold \( W_x(a) \) will impose.

Decrease/Increase the Importance of \( a \), \( I_x(a) \).](image)

By definition, the Importance \( I_x(a) \) of a situation \( a \), \( a \in A \), for agent \( x \) is a subjective measure of the benefits to be gained from the situation. Therefore, applying mechanisms to increase or decrease the importance of \( a \) require careful consideration and are not as straightforward as increasing or decreasing \( a \)'s potential utility, as described in Section 4.2.2.2. Importance plays a role in both the
computation of situational trust $T_x(y,a)^I$, as well as the cooperation threshold $W_x(a)$, as is evident by Eq. (2). Increasing the importance of a situation, both cooperation threshold as well as situational trust should increase. This simulates the fact that, the more important a situation is, the more $x$ needs to trust in $y$, as the situation is too important to be done badly, or not at all. Similarly, if the importance is reduced to zero, the cooperation threshold is also led to zero, as is situational trust, and therefore this signifies an indifference (on the part of $x$) as to the outcome of the situation ($x$ will be more likely to cooperate when the importance is set to 0, since $W_x(a)$ will also be zero and $T_x(y,a)^I$ will therefore be passing the threshold, according to Eq. 1)). Again, as was the case with utility $U_x(a) = 0$, this $I_x(a) = 0$ is a sensible matter in which to promote prosocial, rather than antisocial behavior. Figure 10 demonstrates an example flow diagram showcasing Elo-based PLO skill-based scenario ranking based on the scenario’s Utility factor.

Of course, the Importance of a situation closely relates to the Utility, i.e. how important it is for player $x$ that the situation $a$ reaches the best possible outcome as seen from $x$’s perspective. Again this is a subjective measure: in trusting someone to guide $x$ to the whereabouts of a unique item in the game, the utility for the situation unfolding in the best possible result for $x$ is potentially high (i.e. receiving an item otherwise impossible to get), but the importance is depending on the player’s disposition towards the outcome: some players might consider it unworthy of the trouble (i.e. “I already own a similar item, so I could embark on this quest to potentially get a useless item”, or “I don’t have enough resources to undertake this quest at the moment”). In this context, there is a strong link between the Importance factor and the Outcome, Narrative (expected, due to similarity with the Utility Factor) and Resource elements of the Prosocial World Data Model, i.e. the perceived estimation of the direct outcome of performing a prosocial skill can be moderated by the player’s active judgement on expenditure of energy and/or other resources.

An example for better grasping how Importance plays a role in player in-game behaviour and activities pursued, and how this factor is different from a game scenario’s utility, are the highly popular ‘achievement’ structures made popular by the Xbox 360 game console, and have since been adopted by almost every conceivable gaming platform. Achievements are simply unlockable icons which players collect as an indicator of their accomplishments in the game. Although these achievements do not contribute any direct benefit in the game (i.e. utility is zero), some players are known to go to extreme lengths to acquire these icons (including playing the game on the hardest setting, replaying missions to uncover hidden collectables, otherwise insignificant to the game’s narrative, or trying to finish a challenging game sequence in one sitting over multiple tries) as a sense of completing the game. If players like pursuing activities in the game because it will unlock an achievement or trophy icon for their collection, we can estimate the player’s assessment of importance by offering means by which players can unlock difficult achievements more easily and monitor player reaction to those offers. We can fairly expect players, who tend to go through a lot of trouble in order to acquire these achievements, to regard a situation as a reward, as highly important. Game designers can therefore formulate the importance of a situation into a measurable concept, by monitoring gameplay patterns and extracting information on player most pursued habits. Importance is therefore inscribed in player patterns, stemming out of autonomous volition, and players do not have to be explicitly told about it (e.g. “complete this task because it is important”).
Figure 25: UML diagram for calculating PLO scores based on the Importance Factor in Trust-themed game scenarios.
Decrease/Increase the perceived risk inherent in a, $R_x(a)$.

The perceived risk of trusting y in situation a is a more vague concept than estimating y’s competence in that situation. Even the wisest people may not be completely aware of all the risks attached to a situation. Zeckhauser & Viscusi (Zeckhauser & Viscusi, 1990) note that decisions involving risks illustrate the limits of human rationale. In the same work they differentiate between risk, uncertainty and ignorance. In the first case, it is assumed x knows the precise probabilities of each outcome of a situation a. In the second, these probabilities are unknown. In the third, x might not even be in state to clearly define the outcomes. Therefore, determining risk in mathematical notation involves differing methods for differing situations, namely, three states for x considering the risks involved in a particular situation a. In the first case, the player has no knowledge or experience of situation a, i.e., x is acting under a state of ignorance. This is definitely the most problematic case, and is usually resolved through initiative taken on the part of x, i.e., the amount of trust x has in itself within the situation of making a decision. In the second case of uncertainty, the outcomes of a situation are known, however their probabilities (or some of them) are not. Again, the player is called upon to take initiative in self-assessment of its capability in taking a decision, with a better chance of determining the outcome. In the third case of risk assessment, x has to consider the mean assessment of the probability of each outcome.

Computing perceived risk and regulating action to increase or reduce the perception is the vaguest of the strategies proposed in this Section for adaptation of game scenarios. Perceived risks can only be greater than, or equal to zero, as negative risks are not sensible. The following pointers are provided on the effect of perceived risk on $W_x(a)$, as is evident from Equation (2):

- If no risk is involved ($R_x(a) = 0$), the cooperation threshold can be effectively ignored, as $W_x(a) = 0$. This effectively boils down to a zero-utility or zero-importance case, and prosocial behaviour on the part of the child should be designed to be the desired outcome in these situations.
- A high risk $R_x(a)$, in conjunction with low competence $C_x(a)$ lead to a high cooperation threshold.
- As trust $T_x(y)$ and competence $C_x(a)$ increase while risk $R_x(a)$ remains static, the cooperation threshold decreases.
- As competence $C_x(a)$ and trust $T_x(y)$ remain static and risk $R_x(a)$ increases, so does the cooperation threshold.

It is safe to assume that, in the state of risk with complete knowledge (as is the case in designing a game scenario), we should be able to know how to estimate a risk (i.e. take the mean assessment of the probability of each outcome). If the perceived risk is a value that is known to the developer and can be adjusted according to the scenario requirements, the computation of its value and mapping to game outcomes should be straightforward for the developer. However the child’s perspective state (i.e. one of ignorance or uncertainty, at any rate being sure that the child clearly understands the risks presented to it) is not a given, although we accept that any mistakes made will be learnt from (and a child realizing a mistake will be observable through the child’s behavioral response audio/visual cues, as described in the work delivered as part of activities carried out in ProsocialLearn WP3). It is primarily through experience that the information necessary to assess risks and one’s own capabilities with regards to that assessment (Marsh, 1994). Adapting a situation with respect to risk,
we will know if the training is working by monitoring the student’s cooperation threshold over time through varying scales of risky situations.