Adding User Modelling to a Serious Game
The Methodology behind "My Dream Theatre"

João Santos Tiple
joao.tiple@ist.utl.pt
Instituto Superior Técnico, Lisboa, Portugal
November 2013

Abstract

Throughout the centuries education has evolved due to economical and cultural changes, leading to more complex requirements regarding societies needs. In the last decade gamification and incentive-centered design have been hot topics regarding the game and application industry in general. More recently, they have began being applied in medicine such as medical training, proper usage of medication, psychology evaluation and counselling. In the same context, the SIREN project aims to improve children's' conflict resolution capabilities in order to give them tools to better handle certain situations. This is done by giving them an open, friendly and safe environment to learn about conflict. The goal of this thesis is to improve a particular game of the SIREN project, "My Dream Theatre", making it an Intelligent Tutoring System. This is done with gamification techniques and procedural content generation, in order to better adapt the game during runtime and deliver a better and more personalized learning experience. The methodology applied to transform "My Dream Theatre" into an Intelligent Tutoring System and the conclusions and results drawn by this process are described in this document.

Keywords: Gamification, Learning, Experience Driven Procedural Content Generation, User Experience, SIREN, Serious Games

1. Introduction

Conflicts seem to arise in almost every context and developmental stage of human life. Social conflict can be observed in the form of scuffles in school-yards, to bullying in the workplace or even in international warfare. Whether conflicts are inevitable or not is disputed, but there is a common agreement that there is a necessity to educate conflict resolution in the early stages of life.

The SIREN project[1] is an international project that aims to create a new type of educational game, the conflict resolution game, which takes advantage of recent advances in serious games, social networks, computational intelligence and emotional modelling to create uniquely motivating and educating games that can help shape how children think about and handle conflict. The software developed by the project is able to automatically generate conflict scenarios that fit the teaching needs of particular groups of children with varying cultural backgrounds, maturity, and technical expertise, and the desired learning outcomes as specified by their teacher. By doing this, teachers, without specific technical training, can use the system in their classes and improve their students learning experience.

Currently the SIREN project is composed of two games: "Village Voices" and "My Dream Theatre". While Village Voices is a game that promotes experiential learning, social constructivism and problem solving, "My Dream Theatre" was designed to prepare and teach skills to children related to conflict situations by making them act as a mediator instead of being the protagonist in the conflict situation.

In "My Dream Theatre", the player takes the role of a theatre director that must manage a cast of several actors, each with their own personality perks, issues and acting skills. It is up to the director to assign each role of the play to an actor while keeping in mind that actors may become unhappy if they do not get the roles that they like. If an actor becomes disgruntled enough, a conflict situation will arise and actors may spread that conflict and/or leave the play. This will cause the performance to receive less points, in form of applause. In this serious game, it is expected that the player increases his or her vocabulary and meaning

[1] http://sirenproject.eu/

of many conflict related words such as, mediation, assertive, cooperative, etc. Also, it is expected that the game adapts to the needs of the player by providing conflict scenarios generated on the fly. These scenarios are not imposed but suggested and it is up to the teacher to decide if that scenario is indeed appropriate for the learning needs of the player.

This automatic generation of conflict scenarios is done by using techniques similar to the ones used procedural content generation that have been used on other games. These techniques were mostly developed by João Bertrand Cabral [2] in his thesis.

What was lacking on "My Dream Theatre", prior to this thesis, was a structure that could assess the knowledge of the player and adapt the game so that it suits their learning needs. Also, player motivation was solely tied to game mechanics. This caused some players to not be motivated to learn because the game did not encourage them to explore different conflict resolution techniques. To better motivate the player, gamification techniques were applied to the existing game design.

In order to do this it was necessary to define an approach, a methodology if we may, to apply to "My Dream Theatre". This methodology has to guarantee that the following characteristics: can be used on several games; keeps track of the players knowledge regarding several subjects; suggests a learning path and motivates the player.

This resulted in the definition of a Player Model, that accommodates the players experience, a Reward System that rewards the player when certain milestones are achieved, and a Knowledge evaluator that assesses the current knowledge of the player based on his actions and Rewards won. This was then implemented on "My Dream Theatre", with the game engine Unity3D[2] over the course of three months, followed by a one month evaluation of the system.

2. Related Work

The SIREN Project is very unique since there is no other project that puts together a serious game with the aim of teaching conflict resolution while adapting its content to the users needs. In order to properly use gamification and procedural generation of content techniques in an educational context, research had to be done in the fields of Gamification, Intelligent Tutoring Systems, and the work already done on the project in which "My Dream Theatre" by [3] is integrated, the SIREN Project.

[2]http://unity3d.com/

2.1. Gamification

Video games are today one of the main forms of entertainment in Europe[3] and the rest of the world[4] and are being used as a medium to convey ideas and sell brands[5]. With the continuous growth of the gaming industry[6] more and more people, from all ages, make contact with games on a daily basis. Such popularity makes people more receptive to game elements in non-game environments. Gamification uses game design techniques and game mechanics in non-game contexts to further engage users to learn and solve problems. A task can be Gamified in several ways but at it is core there is always a reward, as it was demonstrated by [10]. This reward can be a badge, a trophy, a medal or something of monetary or personal value. Sometimes a task can be Gamified in such a way that there is a competition between the people that are trying to complete the task or there is a usage of meaningful choices or the introduction of the notion of a difficulty level. All of these serve to amplify the information learning and retention of information by the player. Gamification techniques can, oddly, be applied to tasks inside a game that are cumbersome or hard to understand by the players. For example, a tutorial can be designed with the same principles that exist when gamifying a task.

2.2. Applying Gamification: Serious Games

When we Gamify a task and turn it into a game we may not want that the player merely gets an extra motivation. There are cases in which the gamification serves to instruct and educate the player in matters that are hard or costly to teach, such as piloting an air plane. In such cases when the game has a primary purpose to educate, that game is defined as a serious game.

The notion of serious games has been around for a while. In 1970, Abt wrote that "serious games in the sense that these games have an explicit and carefully thought-out educational purpose and are not intended to be played primarily for amusement" [20]. Since then Serious Games have been used in almost all areas, from education to aviation with varying degrees of success.

Although we can apply Gamification to a non-virtual task, by virtualizing the task we gain access to a controlled environment and to analysis tools that

[3]http://kotaku.com/5011072/study-video-games-mainstream-entertainment-in-europe

[4]http://www.esrb.org/about/video-game-industry-statistics.jsp

[5]http://www.gamasutra.com/view/feature/6287/video_games_as_media.php

[6]http://www.theesa.com/facts/pdfs/ESA_EF_2012.pdf

are impracticable or time-consuming to use in the real world.

2.3. Intelligent Tutoring System

Automation has had a notable impact in most industries since its inception in the manufacturing industry. Work posts that were deemed irreplaceable such as telephone operators have since been replaced largely by automated telephone switchboards and answering machines. In medicine, medical processes such as primary screening in electrocardiography or radiography are carried out at much greater speed and accuracy by automated systems. Automated teller machines have reduced the need for bank visits to obtain cash, perform payments and consult the balance. In general, automation has been responsible for the shift in the world economy from industrial jobs to service jobs in the 20th and 21st centuries.[7] Automating a teaching process can be trickier since each student has different needs and requirements. As such, the definition of an Intelligent Tutoring System was born with research done by [11]. The purpose of an Intelligent Tutoring System is to enable a meaningful and effective learning manner by using a variety of computing technologies. Unlike other automation processes this one does not aim to fully replace teachers but does aim to take care of adapting what is taught to the student according to its own needs, stated [11].

Following general consensus among researchers [18] the structure of an Intelligent Tutoring System is composed of the following: The Domain model; The Student model; The Tutoring model, and The User Interface model.

The Domain Model, developed by [1], is built on the cognitive architecture Adaptive Control of Thought-Rational – ACT-R – which tries to take into account all the possible steps required to solve a certain problem. [13] stated that this model "contains the concepts, rules, and problem-solving strategies of the domain to be learned. It can fulfil several roles: as a source of expert knowledge, a standard for evaluating the student's performance or for detecting errors, etc.".

The Student Model can be seen an overlay on the Domain Model. It is considered to be the core component of an ITS. It pays special attention to student's cognitive and affective states and their evolution as the learning process advances. As the student's proficiency in their problem solving increases, the system engages in a process called Model Tracing. Whenever the Student Model deviates from the Domain Model, the system identifies, or flags, that an error has occurred.

The Tutor Model accepts information from the domain and student models and makes choices about tutoring strategies and actions. This model should, whenever the learner requires, show what should be done next, relative to their current location in the model. In addition, the system should provide feedback to the student whenever he deviates from the Student Model. The tutor model must also calculate the odds that the student has learned certain skills and update his or her standing on the Student Model accordingly. This process is known as Knowledge tracing. Knowledge Tracing builds a profile of strengths and weaknesses relative to the Student Model regarding the actions that the student performed. This includes whenever the student requires for assistance.

The User Interface model is responsible for clearly stating what is being taught and provide means to the student to explore and solve the problems presented.

Intelligent Tutoring Systems architectures reflects the emphasis that is given to the level of intelligence of the different models. A given system may generate new problems so that students can always have new problems to work on, but it might only have simple methods for teaching those problems. In contrast, another system that concentrates on multiple ways of teaching a particular topic might use a less sophisticated way of presenting the content that is being taught.

2.4. SIREN and My Dream Theatre

The main goal of the SIREN project, headed by [8], is to create an intelligent interactive software system, composed of several serious games, which should aid the teachers role in educating young people on how to resolve conflicts. Currently the SIREN project is composed of two games: “Village Voices” and “My Dream Theatre” developed by [3]. While “Village Voices” is a game that promotes experiential learning, social constructivism and problem solving, “My Dream Theatre” – MDT – was designed to prepare and teach skills to children related to conflict situations by making them act as a mediator instead of being the protagonist in the conflict situation. In “My Dream Theatre”, the player takes the role of a theatre director that must manage a cast of several actors, each with their own personality perks, issues and acting skills. It is up to the director to assign each role of the play to an actor while keeping in mind that actors may become unhappy if they do not get the roles that they like. If an actor becomes disgruntled enough, a situation will arise and actors may spread that conflict and/or leave the play. This will cause the performance to receive less points. The objective of the game is to teach players about

[7] http://www.boston.com/bostonworks/galleries/30fast_declining_occupations/pg=10

how conflict appears and how it can be handled and solved. To achieve this goal, the game integrates conflict resolution skills into the actual gameplay. This is done in two ways, firstly by allowing the player to attribute roles and generate conflict and, secondly, by using the appropriate mediation strategy, taking into account the actors personality, to solve the conflict. This way, it is expected that the player increases his or her vocabulary and meaning of many conflict related words such as mediation, assertive, cooperative, etc.

Prior to the contributions of this thesis, the only adaptation that MDT had was regarding to the knowledge that the player regarding game mechanics. With what was added the game now adapts to the players' knowledge of conflict and, consequentially, also to game mechanics. This is done by providing a conflict scenario tailored to improve the players' knowledge, based on the work of [8].

3. A Methodology for Self-Adapting Serious Games

In order to make the necessary adaptations for "My Dream Theatre" to become an Intelligent Tutoring System, a methodology had to be defined before making changes in the game.

Since the SIREN project includes games that can teach children about conflict, and more may appear on the future, there was a need to formally define a general approach when designing the adaptation system, in order to guarantee that the game has the following properties: (1) it must successfully teach one or more subjects; (2) the game adapts itself according to the knowledge of the player; (3) the game rewards the player whenever he or she reaches certain milestones; (4) what is learned on one game is transported to other games.

Based on the research done on motivation, gamification techniques, and ITS development a methodology that guarantees that these four goals are respected is proposed in the following section. This methodology defines a set of steps to gamify a task: define a Skill Set that the player must be proficient with; characterize the Environment regarding its constraints, such as which components can be interacted with and what are the termination conditions; characterize the Player by having a Model that represents him or her accurately in the game; determine which preconditions must be met before the player is evaluated; how and when is the player rewarded; and finally how is all of this connected and how does it adapt itself to the players' needs.

In this section we will go through all of the steps that must be performed in order for the game to have the properties that were previously described. The steps are the following: definition of a Skill Set; Environment characterization; identification of Interactive Components on the Environment; the Scenarios and respective Termination Conditions of an Environment; the type of Environment; the Player Model and the Actions that can be performed; the Rewards and the Recurring Assessment Component that deals with the Adaptation of the game.

3.1. Defining the Skill Set

The first step that must be taken is to determine which Skills are meant to be taught to the Player, and evaluated by the system, thus defining the Skill Set. However, before determining the Skill Set we must determine what is a Skill in the scope of the proposed model. A Skill can be almost anything as simple as "digging holes" to something more abstract as "Empathy - Understanding other peoples emotions". A Skill should be identified by a sequence that is unique in the set of Skills. However, the identifier alone does not give us anything besides a context. In order for the Skill to be evaluated by the system it has to be able to be quantified.

Also, in order to determine the relevance of each Skill, it should have a function that returns the weight of the Skill based on player knowledge or Environment characteristics. To determine the weight of the Skill, this function can take into account the Rewards that the player has earned, the development of each other Skill, the current state of the game, etc. This way it is possible to order and filter Skills by relevance, taking into account the current knowledge of the player.

By increasing or decreasing the value that is associated to the identifier we are actively quantifying the player's proficiency in such skill and with the weight function it is also possible to know how relevant the Skill is to the player. After determining which skills do we want the player to learn we should have something similar to this Skill Set, A = {{Id1, Val1, WeightFunc1}, {Id2, Val2, WeightFunc2},...,{IdN, ValN, WeightFuncN}}. A more specific example could be the following Skill Set A = {{"Dig holes", 1, DigHolesEvaluator}, {"Pick up Rocks", 3, PickUpRocksEvaluator}} that tells us that the "Pick up Rocks" Skill is more developed than the "Dig holes" Skill.

3.2. Characterizing the Environment

After the Skill Set is determined, the Environment can be defined. This Environment can be one that

already exists or it can be fabricated in order to suit our needs, which are the teaching of the Skill Set that was previously defined. The Environment should at least be characterized by the following: components that can be interacted by the user; Scenarios that happen on the environment; the termination condition of each the Scenarios and the type of Environment to which the Model is going to be applied.

3.2.1 Identifying Interactive Components

If we want the player to learn something in the Environment that we chose or fabricated, we must allow him or her to interact and explore the Environment. This is done by determining all of the Elements of an Environment and deciding which of them are Interactive Components. When identifying Interactive Components in the Environment the following should be kept in mind:

- The player must be able to interact with the component during a play session;
- It is possible to know if the component has been interacted with;
- Not all elements of the Environment have to be interacted with;
- The element may or may not be related to a Skill;
- A set of elements can be a single interactive component;

For example, a Large Rock is an Element that and an Interactive Component. A Small Rock is an Element but not an Interactive Component. However, several Small Rocks make an Interactive Component. As it will be explained further in the chapter 3.3.1, the Interactive Components do not have to be directly related to a Skill but the Actions performed on them must.

3.2.2 Scenarios and Termination Conditions

On a single Environment several Scenarios may exist. A Scenario is composed of a set of well defined Interactive Components and Termination Conditions. The Termination Conditions of a Scenario are a set of rules that define when should the Scenario be terminated. A Scenario can have several of these and the Player should always be able to accomplish at least one of them. The Termination Conditions may be shared between Scenarios. The Termination Conditions of an Scenario should be used if they occur naturally (i.e. a car crosses the finishing line) in order to respect familiarity. If the Environment itself does not have a closing objective or some sort of milestone that indicates to the player that the scenario is over, then the Termination Conditions should be fabricated. For example, if the Environment is a city park where the Player can ride his or hers Bike freely, there could be a Time Limit for the ride or an area where the player goes to when he wants to finish the Scenario. By defining different Scenarios it is possible to expose the Player to some Interactive Components while disabling others thus allowing a richer learning experience.

3.2.3 Type of Environment

As stated before, what is evaluated, and is related to the Skill Set, are the Actions that are performed on the Interactive Components of an Environment. However it is important to correctly appraise and define the type of Environment since the way that the player interacts with the Components can be limited by the specificity of the Environment. An Environment can be characterized according to the following: type of input required, whether it is synchronous or asynchronous, the time constraints and repeatability.

Regarding Player Input, it may Require No Input if the Environment may change even without input from the player (i.e. a city full of people) or it may Require Input if the Environment may only change if there is player input (i.e build a chair);

The Environment is Asynchronous if events happen asynchronously on the environment. (i.e. a game of football) and Synchronous if the events happen at specific moments in time (i.e. a game of chess)

If there is no expiration date for the lifetime of the Environment (i.e. the universe) the Environment has No Time Constraints. If the Environment is terminated after a period of time. (i.e. a car race) it has Time Constraints.

When the Environment cannot be repeated with the exact same constraints as before (i.e. while learning to drive a car, the car crashes) it is Not Repeatable. If it can be reproduced any number of times with the same constraints (i.e. a tennis game between two players') it is Repeatable.

3.3. Player Model

As stated before there is a Skill Set composed of Skills that are meant to be learned by the Player, an Environment that has one or more Scenarios which are in turn composed of Interactive Components. Those Interactive Components are then acted upon by the

Player with Actions. Thus it is necessary to define the Actions that can be performed and learning Skills that are to be evaluated. The Player Model is the structure that keeps track of all the Actions performed by the player, the state of the Environment during the play sessions and the progress of the Player regarding the Skills that are being assessed.

3.3.1 Actions on the Environment

Actions can be performed by anyone in the Environment but only the actions performed by the Player should be assessed. Thus, there should be two separate categories of actions: Player Actions and Non-Player Actions. Non-Player actions should affect the state of the Environment in some way that the Player needs to react or acknowledge them. Player Actions, even though they might not always affect the state of the Environment, must be able to be assessed to determine if they have been correctly used or not. As it will be described on the following section, in order to know assess the knowledge of the player on a given Skill, it is necessary to validate his or hers actions.

3.3.2 Player Actions and Events

When the Player interacts with an Interactive Component he is either using correctly or incorrectly a Player Action. However, sometimes it may not be possible to certain that the player is learning just by doing a certain action at a certain point of the game, it may be necessary to perform additional actions or trigger a determined Event. Events are meant to give more freedom when defining milestones and may be composed of several player actions, consequences of certain actions, or certain characteristics of the component that was interacted with.

For example, for the Scenario S1 in Environment E1 there are three Interactive Components: "Large Rock", "Small Rock" and "Small Blue Rock"; the following Skills "Pick up rocks without hurting your back." and "Distinguish between coloured objects" are to be learned by the Player and the only Player Action that exists is "Pick Up Rock".

If the Player uses "Pick Up Rock" on a Large Rock he will hurt his back, however if he uses "Pick Up Rock on a Small Rock" the Skill "Pick up rocks without hurting your back." is increased. Both the correct and incorrect use of the Player Action should be logged in the system. However, we may want to check if the player only "Pick Up Rock" on blue colored rocks. Thus we may define the following Event "Use the Skill Pick Up Rock only on Small Blue Rocks".

3.4. Motivating and Rewarding the Player

A serious game can have several different education goals and multiple rewards. In this system each Scenario has its own goals in the form of Termination Conditions. This does not mean that by reaching a Goal the Player has learned something. It is necessary to create Intermediate Goals in order to keep player interest high and relate the Skills that we are trying to teach to the Player and the Tasks that he performs.

If we take into account the Expectancy Theory [7], we can describe player motivation with the following: Player Motivation (PM) = Expectancy(E) x Instrumentality(I) x Valence(V). Expectancy is the player's belief that his efforts (E) will result in the attainment of performance (P) goals. This is closely related to self-confidence and past experience. Instrumentality is the belief that if the performance goals (P) are attained, then he or she will get a reward. Finally, Valence is the value that the player places on the rewards based on his or hers needs, personal goals, values and sources of motivation.

With careful Game Design it is possible to keep the players' Expectancy in realistic values, Instrumentality can be guaranteed as long as the game rewards the player whenever he accomplishes something meaningful and Valence can be manipulated by offering carefully crafted Rewards that appeal to the Players. This also goes on par with the revised model that Lawler[14] suggested.

These Rewards have yet another meaning to the system, they offer an easy and immediate way to asses the Players proficiency on each Skill.

In order to keep the player engaged in the game and to monitor his or her progress there needs to be a system that keeps track of that progress. This is done by rewarding the player at certain milestones.

3.4.1 Defining Rewards

When defining the Rewards that are given to a player, there should be a special care to not define these randomly or on a whim. The Rewards should attend to the games' pedagogical needs.

As such, when designing which Rewards the player can earn, each one them has to be allocated to a different dimension. In addition, in order for the badges to respect both the pedagogical needs, game design and effectiveness, they must be subjected to the Kellers ARCS model[8] subcategories: Perceptual arousal; Inquiry arousal; Variability; Goal orientation; Motive matching; Familiarity; Learning requirements; Success

[8] http://www.arcsmodel.com/

opportunities; Personal control; Natural consequences; Positive consequences and Equity.

Applying Kellers model to games is not new. In 1998, Dempsey and Johnson proposed applying the ARCS model to select and analyse which games are most effectively used in the classroom. As such, a similar model can be applied while designing the Badges. By asking the following questions to ourselves regarding the Rewards, and answering them in a positive manner, we are assuring that they are meaningful to the game and to the players' learning experience.

While designing Rewards we should also keep in mind the following: does this Reward have any in influence on the players' learning experience and will it alter the game in any way? If all of this is respected then the Rewards will surely be tied to the game and the goals of the game.

3.4.2 When to Reward

In "The Role Of Expectancy In Delayed Reinforcement", by [12], demonstrated that if a person experiences a certain type of reward as consequence of a specific action, he or she is most likely to have a high degree of Expectancy regarding being rewarded under similar circumstances.

In modern games, most of these rewards follow a similar framework to the one suggested by [10], but there is no written rule of when should a player be rewarded in a game. However, in a serious game that wants to teach something, there are some aspects that we can keep in mind while defining when should the player be rewarded.

The player should be rewarded only if there is a high enough degree of certainty that the player has learned something. This degree of certainty can be predetermined by conducting user tests and agreeing on what is the most correct value or following the opinion of an expert on the subject. When this happens the player is eligible to be rewarded, the question that poses is, when should the player be rewarded?

This is a question not easily answered since, because of internal and external factors, it is not guaranteed that the player will understand what did he just do that justified a reward.

When deciding when to reward the player, the following should be taken into consideration:

- How long does a play session last?
- Is the reward fostering a positive reinforcement?
- Is the reward going to motivate the player?

After the Rewards are defined, the amount of times that a task needs to be performed should be defined by an expert on the area and then adjusted after playtesting is done. However, other factors may influence this decision. The Badges created for "My Dream Theatre" were defined taking into account the limited time that players' had to interact with the game. Also, the purpose of these tests was to evaluate the adaptation mechanism, thus the amount of times that a player had to perform a task in order to gain a Badge was relatively low.

3.4.3 How Skills are related to the Rewards

Relating Skills with Rewards is relatively easy. A Task is associated to a certain Skill and can be composed by any number of Player Actions and/or Events. These Events are certain conditions that are related to Player Actions but are only met if, for example, an NPC performs a Non-Player Action.

As such, we can define that a Task T is described by the following {S1, {A1, A2, Ev1}}, where S1 is a Skill and A1, A2 are the Player Actions that have to be performed by the Player, Ev1 is the event that must happen in order for him or her to complete the Task.

Now, if we define R as the following Reward, {T, 5, G1}, where T is the Task previously defined, 5 is the amount of times that the task must be performed in order for R to be earned and G1 is the Game where the player can complete this task, we are saying that "If player completes Task T in Game G1 at least five times, then he is proficient on Skill S1 and should be rewarded with R".

This way, by looking at the Rewards that the player has earned it is possible to determine which are the Skills that are least or more developed by the Player. If the Skills are transversal to the games, in each game we can know what are the proficiencies that the player already has and adapt the game accordingly.

3.5. Recurring Assessment Component

In order for the system to adapt to the Player, his or hers knowledge must be evaluated. Since the Player Model constantly keeps track of the progress of every Skill and the Scenarios have well-defined Termination Conditions, it is possible to evaluate the players' progress from the beginning of the Scenario until it ends. This evaluation is done by the Recurring Assessment Component – RAC.

This Recurring Assessment Component is composed of Tasks. As we have seen before, these Tasks are directly related to Skills. Tasks can be abstract such as

"Make sure everyone is happy", which may involve the use of several Player Actions to accomplish, or it can be more straightforward with something like "Cheer up Anna." which involves using the "Cheer Up" Player Action on the NPC "Anna". Also, as stated before, for consistency, all Player Actions that are required to be used to accomplish a Task should be related to the same Skill Set as the Skill Set of the Task. This component constantly evaluates the players' knowledge and rewards him or her if the conditions defined on the Rewards are met.

In this section we will explain the two identified types of assessment, how are Rewards won by the player and how does the game adapt itself.

3.5.1 Self-Contained and Two-Part evaluations

For the purpose of this system, the evaluation can be of two types, Self-Contained or composed of Two Parts, and assesses the progress of each Skill and rewards the player properly.

A Self-Contained evaluation means that the Player Actions performed by the Player during the Scenarios are enough to assess his or her knowledge on the Skills that are being developed. However, this implies that the evaluation can be repeated several times and that the player never "flunks". This Self-Contained evaluation can be, for example, composed of several Tasks, that are accomplished by performing Player Actions, that the player must complete. These Tasks can be fundamental for the game to progress or can be merely side-objectives to the game itself.

A Two-Part evaluation is used when the actions of the player on the Environment are not enough to assess the knowledge of the player on certain Skills. Thus it is necessary to complement it with another evaluation. This evaluation can be a final score or a more traditional component like a test. The result of this test should then be translated into Rewards so that the system can still keep track of the knowledge of the Player even though it did not evaluate his or hers Actions.

3.5.2 How are the Rewards related to the Recurring Assessment Component

As it was said before in 3.4.3, Rewards are related to Tasks that are in turn related to Skills. When the player completes Tasks he advances his or hers progress in a certain Skill thus gaining Rewards.

By looking at the earned Rewards, the RAC can determine which and how much the Skills are developed and suggest appropriate Tasks that are related to said Skills. In turn this will adapt the game so that these new Tasks can be accomplished. This process is done by double layer system that is explained in the following section 3.6.

3.6. Double Adaptive Layer

In order for the game to truly adapt to the player's needs, it does not only need to identify those needs but also to create situations in which the player is able to progress its Skill Set.

This is done by using a two-layer architecture in which one layer - Suggestion Layer - focuses on determining the player's knowledge and the second - Manipulation Layer - on manipulating game conditions and providing Tasks. The Suggestion layer is described on this thesis and encompasses the Skill Set, Environment, Player Actions, Tasks and Rewards. The Manipulation layer is the work of João Cabral[2] and it will be described in the following sections. Both layers are connected by the Skill Set which the Player is being taught.

3.6.1 Advantages of a Two-Layer Architecture

By using a two level architecture, the player's knowledge about the Skill Set is separated from the game system itself, meaning that the Player Model can be used throughout several games that educate for the same topic, i.e. "My Dream Theatre" and "Village Voices". This means that both layers can adapt at different paces: while the first layer can contain information and adapt to the needs of a Player regarding the Skill Set, the second layer can adapt the game rules taking into account the player's Skill Set and his or hers proficiency with the game. For example, a Player can be very proficient with the Empathy Skill because he learned it on Game A but he doesn't know how to play Game B which also teaches the Empathy Skill. This gives a very big advantage to the system: it can easily be scaled over several Games that focus on the same topic. In order for this to happen, each Game must implement the Reward system previously described so that there is a uniform evaluation. By doing this the Suggestion layer can collect all the Badges that the player has earned throughout the games and asses the proper level for each Skill. Each Game must also implement its own Manipulation layer so that it can consume the information that the Suggestion layer transmits. By having each game sharing the same Suggestion layer, information regarding the Skill Set of the player can be used by the system in order to make more accurate and adequate decisions over which competences should be to developed, thus enriching the education of the child.

3.6.2 Suggestion Layer

The main purpose of the Suggestion Layer is to determine the knowledge of the player. This is done by analysing Player Input and updating the Player Model. Then, with the information of the Player Model, the layer checks with a Reward Database to determine if the player has won any Rewards. By knowing which Rewards the player has won, the System can determine which Skill is the one that should be taught to the player, thus fulfilling the pedagogical needs. This process can be observed in Figure 1.

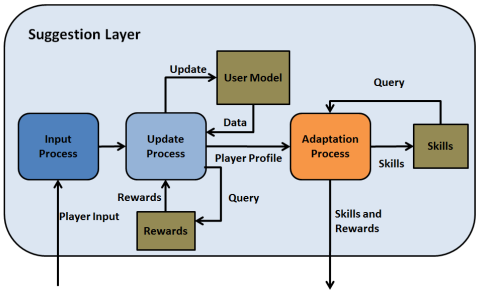

Figure 1. Suggestion Layer.

3.6.3 Manipulation Layer

The main purpose of the Manipulation Layer is to adapt the game content taking into account the information received from the Suggestion Layer. This adaptation must provide a Scenario that allows the Player to have contact with the Skill that is currently being developed. This adaptation can occur in several phases while the player is interacting with the system. If the Player has not yet chosen a Scenario to play, the Manipulation Layer should provide an adequate Scenario, however if the player is already completing a Scenario and the learning requirements change, the Manipulation Layer should adapt the Scenario accordingly so that the new learning requirements can be fulfilled.

This process can be observed in Figure 2.

3.6.4 Layer Communication

Layer communication is done by assessing which are the player's needs in terms of Skills in the Suggestion Layer and sending those needs to the Manipulation layer. As it was described before, these needs are determined with the Recurring Assessment Component, by looking at the Rewards that the player has learned, by assessing the Weight of each Skill, via the weight funtion, the Skills that has most weight is identified

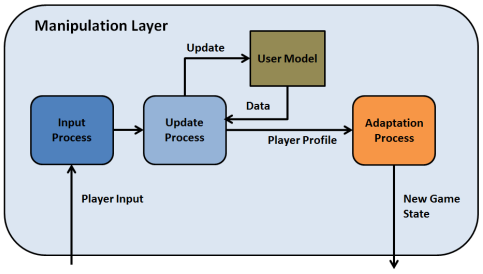

Figure 2. Manipulation Layer.

and signalled to the Manipulation layer. The Manipulation layer then adapts the game so that the Player can effectively complete Tasks that are related to the Skill with most weight. As the Player completes Tasks he earns Rewards thus altering the progression of each Skill and changing the weights of each Skill.

The way that layers communicate can be observed in Figure 3.

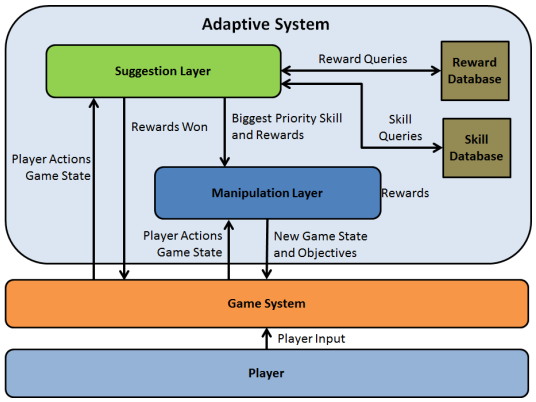

Figure 3. Layer communication.

3.7. Conclusions

All of the research that was done on Gamification, User Modelling and Intelligent Tutoring Systems, allowed us to elaborate and put into practice a methodology that, if properly used by game designers, guarantees that the result of applying said methodology will be able to teach to the player a certain Skill Set and adapt correspondingly to the players' successes and failures. Additionally, since the methodology is generic, it is possible to continue testing the validity of a personalized learning experience, via serious games that adapt to players', even if the "My Dream Theatre" project does not go through, with another game.

The work that was done with "My Dream Theatre" showed that it is possible for a serious game to adapt to the person that is playing it, even if those tests did not produce a complex adaptation. The gamification techniques that were applied on "My Dream Theatre" not only allowed us to add extra incentives to the game, but they also created a stable core that keeps track of each players' progress. With this information the game can then adapt correspondingly, thus making "My Dream Theatre" an Intelligent Tutoring System.

Without the work done by previous authors in the fields of Intelligent Tutoring Systems and Procedural Content Generation, it would have not been possible to create the Double Adaptive Layer structure that is the basis of the adaptation system of "My Dream Theatre". This system not only stores information about the player but also suggests which is the best learning path to a certain him or her.

Following the implementation of said system, the one month testing period, with children from ages 10 to 13, provided us with enough data to realize that, even though the children enjoyed the game, learned some vocabulary regarding conflict, and received a personalized learning path due to the adaptation system, it was not possible to determine if a personalized learning experience is better or worse than a traditional one.

References

[1] J. R. Anderson, M. D. Byrne, S. Douglass, C. Lebiere, and Y. Qin. An integrated theory of the mind. *Psychological Review*, 111(4):1036–1060, 2004.

[2] H. Campos, J. Campos, J. a. Cabral, C. Martinho, J. H. Nielsen, and A. Paiva. My dream theatre. In *Proceedings of the 2013 international conference on Autonomous agents and multi-agent systems*, AAMAS '13, pages 1357–1358, Richland, SC, 2013. International Foundation for Autonomous Agents and Multiagent Systems.

[3] H. Campos, J. Campos, J. a. Cabral, C. Martinho, J. H. Nielsen, and A. Paiva. My dream theatre. In *Proceedings of the 2013 international conference on Autonomous agents and multi-agent systems*, AAMAS '13, pages 1357–1358, Richland, SC, 2013. International Foundation for Autonomous Agents and Multiagent Systems.

[4] D. Crawford and R. Bodine. *Conflict Resolution Education: A Guide to Implementing Programs in Schools, Youth-Serving Organizations, and Community and Juvenile Justice Settings*. DIANE Publishing, 4 edition, 1996.

[5] P. W. N. Douglas Bernstein, Peggy Wright Nash. *Essentials of psychology*. Cengage Learning, 2005.

[6] M. W. Eysenck and M. T. Keane. *Cognitive Psychology: A Student's Handbook*. Psychology Press (UK), Aug. 2000.

[7] M. W. Eysenck and M. T. Keane. *Cognitive Psychology: A Student's Handbook*. Psychology Press (UK), 2000.

[8] R. K. A. J. K. K. A. P. A. V. G. N. Yannakakis, J. Togelius. Siren: Towards adaptive serious games for teaching conflict resolution. 2010.

[9] L. B. Gambrell. Creating classroom cultures that foster reading motivation. *The Reading Teacher*, 50(1):14–25, 1996.

[10] V. Hamari, Juho; Eranti. High-fidelity aerostructural design optimization of a supersonic business jet. *Proceedings of Digra 2011 Conference: Think Design Play*, pages 14–17, 2011.

[11] H. C. Lane. Intelligent tutoring systems: Prospects for guided practice and efficient learning, 2006.

[12] A. R. Mahrer. The role of expectancy in delayed reinforcement. *Journal of Experimental Psychology*, 52(2):101–106, 1956.

[13] R. Nkambou, R. Mizoguchi, and J. Bourdeau. *Advances in Intelligent Tutoring Systems*. Springer Publishing Company, Incorporated, 1st edition, 2010.

[14] L. Porter and E. Lawler. *Managerial attitudes and performance*. The Irwin-Dorsey series in behavioral science. Irwin, Homewood, Ill., 1968.

[15] D. E. L. Ryan, R. M. Promoting self-determined school engagement: Motivation, learning, and well-being. *K. R. Wentzel, A. Wigfield (Eds.), Handbook on motivation at school*, pages 171–196, 2009.

[16] N. Shaker, M. Nicolau, G. N. Yannakakis, J. Togelius, and M. O'Neill. Evolving levels for super mario bros using grammatical evolution. In *CIG*, pages 304–311, 2012.

[17] J. A. Shepperd and K. M. Taylor. Social Loafing and Expectancy-Value Theory. *Personality and Social Psychology Bulletin*, 25(9):1147–1158, Sept. 1999.

[18] D. H. Sleeman and J. S. Brown. Intelligent Tutoring Systems: an overview. In D. H. Sleeman and J. S. Brown, editors, *Intelligent Tutoring Systems*, pages 1–11. Academic Press, 1982.

[19] W. E. Ulrich. Expectancy theory and its application to edp professionals: a tentative view. *SIGCPR Comput. Pers.*, 5(3-4):2–4, Jan. 1976.

[20] J. Vogel. Serious games Clark C. Abt The Viking Press, $5.95. *Business Horizons*, 13(3):95+, June 1970.

[21] R. H. Whittaker. *Communities and ecosystems [by] Robert H. Whittaker*. Macmillan [New York], 1970.

[22] G. N. Yannakakis and J. Togelius. Experience-driven procedural content generation. *IEEE Transactions on Affective Computing*, 2(3):147–161, 2011.