Prototyping games for training and education in Second Life:

Time2Play and TREG

K. Vega A. Pereira G. Carvalho A. Raposo H. Fuks

Pontifical Catholic University of Rio de Janeiro (PUC-Rio), Department of Informatics, Brazil

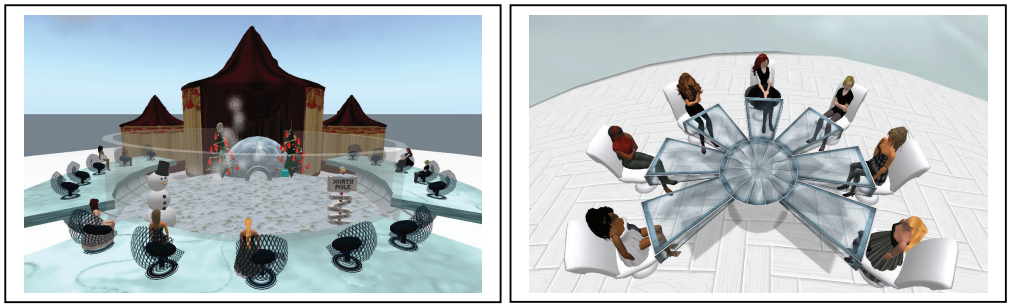

Figure 1: Educational Games in Second Life: Time2Play (left) and TREG (right).

Abstract

The purpose of this paper is to report the experience in prototyping 2 games for education and training in Second Life, Time2Play and TREG. Starting from a prototyping process, it was adapted for getting better results in the development of the games. Based on our experience, Second Life provides a sound platform for the step-by-step prototyping evolution.

Keywords: prototyping, prototyping process, game development, Second Life.

Authors' contact:
{kvega, andreiapereira, guga, abraposo, hugo} @inf.puc-rio.br

1. Introduction

Virtual environments have been successfully used in several contexts like educational, social, gaming and commercial. A virtual world is an environment simulated by a computer provided for some specific goals. Second Life is a 3D virtual world opened to the public since 2003 where users have the possibility of creating their part of that world [Linden Labs 2009]. Gartner reports that 80% of the active users will have a "Second Life" by the end of the 2011 [Pettey 2007]. This offers new challenges and opportunities for educators.

Educational simulations seem to be the new paradigm of knowledge transfer and social training. Moreover, current studies indicate that students are becoming more pragmatic, visual and computer-savvy [Aldrich 2005].

The Time2Play and TREG games were created in Second Life (SL) using its building blocks and scripting possibilities. In this paper we will discuss our experience in prototyping the Time2Play and TREG games using Second Life as the development platform.

This work is organized in eight sections. Section 2 introduces Second Life and its possibilities for creating educational games. Section 3 presents some related work of serious games created in Second Life. Section 4 shows overviews of Time2Play and TREG: a storytelling and training game designed for Second Life. Section 5 discusses the prototype process followed by its development. Section 6 and 7 describe the prototyping methods used and our experience prototyping Time2Play and TREG. Section 8 concludes the work.

2. Second Life as a development platform

Virtual worlds are an interactive multi-user environments simulated by a computer. They are also called Massive-Multiplayer Online Role-Playing Games (MMORPGs) and have these common features [Book 2009]:

1. Shared Space: Many users are simultaneously sharing the same world;
2. Graphical User Interface: They have visual environments from a 2D style to a more immersive 3D world;
3. Immediacy: Interaction with the world takes place in real time;
4. Interactivity: Virtual worlds use to allow users to make changes to it like alter, develop, build, or submit customized content;

5. Persistence: the environment continuous existing and being developed internally even if there are no users interacting in it; and

6. Socialization/Community: the world allows and encourages the formation of in-world social groups like teams, guilds, clubs, cliques, housemates, neighborhoods, etc.

The online 3D virtual world Second Life was launched by Linden Lab in 2003 [2009]. Users, represented by avatars, do real life activities - they interact, play, build and do business purchasing and selling the virtual currency Linden Dollars - building most of the existing content in Second Life. Their raw materials are prims that are the basic building 3D geometric blocks for creating objects. These objects needs scripts for "getting alive" in order to interact with other objects, avatars and Non-Player Characters. Scripting is done using the environment's event oriented language: Linden Scripting Language (LSL) which is familiar for C programmers.

Below we present some of SL building and scripting features that are relevant for this work.

- Appearance and motion. Avatars can change the body and clothes using configured ones or they can configure their own body and clothes using the appearance editor. The face and body motion is made by animations and gestures.
- Textures. It is possible to use textures applied to objects and clothes, obtained out-world, shared among avatars or as in-world snapshots.
- Freebies. There are elements that are shared with the SL community.
- Permissions. These are configured for a collaborative development process. The SL land owner or administrator sets permissions for a group or an avatar to build in their lands. An avatar gives permissions to specific avatars for moving, copying or editing their in-world objects. And an object or script owner gives the permission to the next owner to copy, modify or resell it.
- Teleporting. An avatar is teleported to specific places by other avatars invitations, Landmarks or scripts.
- Voice chat. Avatars could communicate and coordinate enabling this feature.
- Machinima. A filming technique which was used for making videos with avatars, NPCs and objects in virtual worlds.

All that features can be used for the creation of these games in Second Life. But, it was necessary to consider some Second Life constrains in order to develop them.

- The hardware requirements must be the minimal considered by Linden Labs [2009]. It is recommended having a powerful video card. This will facilitate the load in the Client of textures and sculpted objects like the NPCs and some delaying in scripts like listen running scripts or rezzing objects.
- Second Life allows permissions configuration for building and collaborative editing. However there are some constrains for developing in-world like there is no version control system, the LSL editor in-world doesn't have a debugger and compiler and Second Life doesn't have a database repository in-world.

3. Related Work

Oblinger [2004] calls this new students' generation the Net Generation (NetGen). NetGeners tend to be experiential learners, community-oriented and their learning preferences include teamwork and technology use. Games and simulations are a potential learning environment to create educational engagement for them. Virtual worlds can be an effective environment for educational games [Cunha at el. 2008]. Second Life gives the possibility to create different educational content like classes, discussion panels and games [Klunge and Riley].

Kidz Connect is a program created by ZoomLab that connects young people in different countries via media art, performance and collaborative creation in virtual worlds like Second Life (Figure 2). Guided by artists and educators from theatre and digital arts, students learned skills like playback theatre, digital storytelling, and 3D modeling. Students from two different countries write, create and perform a live show. In Second Life, the students met and collaborated to build a hybrid virtual city combining aspects of both countries and in that common space, they created a performance that occurred both live and online simultaneously [Kidz Connect 2009].

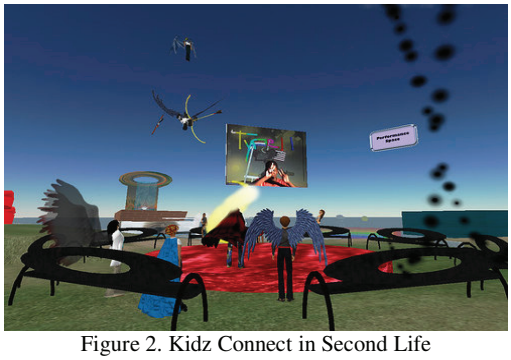

Figure 2. Kidz Connect in Second Life

Training by playing is a great way to improve people skills and have fun at the same time. Nowadays, companies and universities are researching and investing in Second Life for training in several fields.

Idaho State University created a large scale training game to simulate top exercises in health care and emergency preparedness [Ramloll et al. 2006]. Figure 3 shows a hospital scenario where a pandemic flu training exercise was simulated [Idaho 2009].

Our research found games developed in Second Life for computer science. The Ohio University developed two multi-player games in Second Life for software engineering education [Ye 2007]. The first one based on Groupthink Software Specification Exercise developed at the M.I.T. [Ernst 2006a]. Groupthink aims to teach in a game show style how to write effective specifications. This game is available to use in-world [Ernst 2006b]. The other game was developed at the UC Irvine based on the SimSE game where students manage a simulated software project [Navarro et al. 2007].

Figure 3. Hospital Scenario in Play2Train

IBM developed a series of games called Open Encounters of z Virtual Kind for challenging skills in IBM technologies and Open Source like IBM System z mainframe, Service Oriented Architecture (SOA), the Cell/B.E. processor, Grid computing, Linux, Java, and a host of other technologies [IBM 2009].

4. Overview of Time2Play and TREG

In this section we give an overview of the two games that were prototyped using Second Life as the development platform. This choice was supported because of the immersive and collaborative features inherent to Second Life that are needed for the games Time2Play and TREG.

4.1 Time2Play

Time2Play enables the creation of stories and the re-creation of well-known stories in a 3D environment. Each learner has an avatar for enacting her part in the story, role playing that way with other learns, allowing the socialization of knowledge.

After logging in Second Life, the avatar is teleported to the auditorium shown in Figure 4. It is a theatre that enables avatars to watch and participate in the story being enacted. Sitting avatars may at any moment grab clothes or objects, entering this way into the performance of the play, moving from lurkers to players bringing new life into the stage.

The backstage auditorium comprises 3 rooms loaded with different features and functions for supporting learners' performances: main room, dressing room and animations room. In the Main Room, there is a panel that provides scenarios based on themes such as a beach, a forest, a snowy park, among others. These scenarios play environmental sounds based on the theme they represent. There are also two panels that provide objects and special effects to complement the scenarios, offering other possibilities for the players. The Dressing Room has panels that provide clothes, hair, makeup and accessories that allow the avatar's characterization according to the stories that will be performed. There is also the Characters panel that enables learners to transform their avatars into non-human characters by acquiring an alternative form such as a robot, cat, witch, etc. Finally, there is the Animations Room that provides different animations that engage the avatar in a sequence of movements such as swimming, running, or dancing, that could be activated in their performances.

Figure 4. Time2Play auditorium.

Learners divided their work into two activities: story creation and story enactment. They coordinated these activities using the voice chat featured available within SL. Firstly, they came up with the story idea, and then, they selected and modified the scenarios using the panels and, finally, chose their characters' appearances and costumes. During story enactment, each learner played a character in the story, that had no rules or actions pre-defined by the game.

4.2 TREG

The Training in Requirements Engineering Game (TREG) is a 3D online game which aims to teach requirements engineering techniques using simulations based on collaboration. The expected audience is a stakeholder involved in requirements elicitation (students, customers, users or software suppliers) that wants to be trained in this topic. In this phase of the project, we are focusing on training in the workshop

technique, a collaborative way for gathering and analyzing requirements.

There were used the building capabilities in Second Life to create the main building, metaphor rooms, simulation rooms, NPCs and heads-up-display (HUD). Scripts were used to move the avatars by teleporting between the metaphor rooms, to program the HUD with the game states, to handle the NPCs' functionalities in the simulation rooms, to communicate objects, and to play videos. The main building includes a reception area, teleporters that transports the trainees to the metaphor rooms and a NPC, a guide in-world. Figure 5 shows Miss Workshop, a specific NPC implemented for guiding the trainee in the playing of the game.

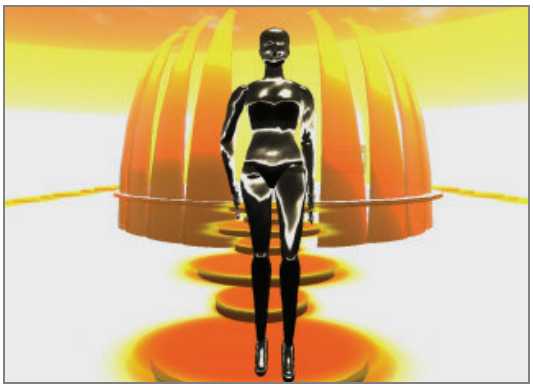

Figure 5. Miss Workshop, the NPC implemented for guiding

The HUD created for the TREG game controls the scores of the game: main score, investment, time, mission and technique. One can score more points, loose investment and time and move to a next level. This information is registered for following the trainee's participation in the game.

The NPC is an object that resembles an avatar but it is controlled by scripts. They are normally used in fictional simulations or role-playing games for interacting with avatars in pre-articulated situations where human beings are dispensable or not available [Bartle 2004]. TREG makes use of NPCs for representing simulations, guiding and shooting machinima videos. The "Making Workshops" Room in Figure 6 shows a workshop session populated by NPCs.

Figure 6. The Cooking Metaphor and the "Making Workshops" Room

Machinima is a technique that relies on the use of 3D game engines to generate a recorded performance in virtual worlds and uses in-world film techniques where characters and events can be controlled by humans, scripts or artificial intelligence [Nitsche, 2005]. Machinima was used to film some problematic workshop situations. There, the trainee will watch a machinima showing the consequences of choosing the "Right People", for example, learning this way a new workshop technique based on a collaboration pattern, if it applies.

5. Prototyping Process

An incremental and iterative prototyping process was used for the creation of these games. Second Life facilitates the prototyping process as it shows the elements in-world as-is which makes it possible to have an early vision of the game and figure out the necessity of additional features and functions.

Figure 7 shows the Prototyping Process of the book Effective Prototyping for Software Makers [Arnowitz et al. 2007]. An iteration consists of the four phases of this process. They will iterate until the software validate all the requirements established.

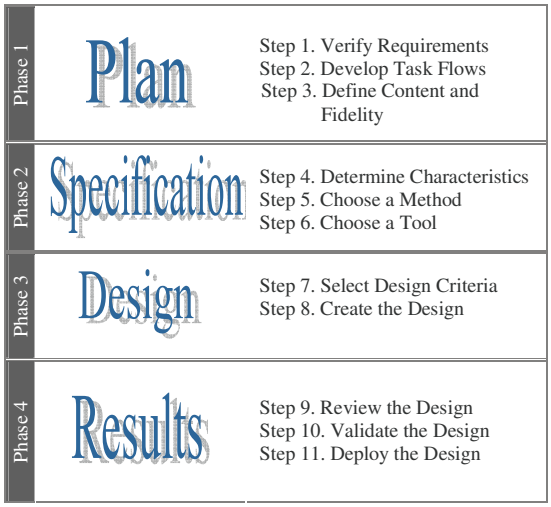

Figure 7. The Effective Prototyping Process [Arnowitz et al. 2007].

The following steps describe how the process had been customized for implemented the game in Second Life.

Step 1. Verify Requirements

The software requirements were discovered from assumptions taking into account that the audience to each game is orientated and their goals. They were gathered, inventoried and prioritized.

Step 2. Develop Task Flows

The task flows depicts the steps that the learner has to follow to complete an activity. In this step, it was defined these tasks and the narration of the scenarios.

A list of the tasks was used for identified the learners' actions in Time2Play. In TREG, branching Stories were used as the simulation games genre for mapping the scenarios of the gameplay. The Branching Stories Graph gives an overview of the game and guides the interaction of the different scenarios given to the trainee. A scenario template was used for getting a fine-detailed specification of the gameplay.

Step 3. Define Content and Fidelity

It was decided to use Second Life as the main prototyping tool for a high fidelity representation. The level of detailed depended on the iteration milestone, from a high level visualization perspective of the game to a coded behavior of the game.

Step 4. Determine Characteristics

There were identified the characteristics proposed by Arnowitz [2007] to determine the prototype method to applied in each iteration. These characteristics are audience, stage, speed, longevity, expression, style and medium. The next sections shows the characteristics found in the games. As it was a prototype driven development, a prototype was created in all the iterations.

Step 5. Choose a Method

A prototyping method was chosen depending on the iteration characteristics. Arnowitz [2007] proposes the following prototyping methods: card sorting, wireframe, storyboard, paper, digital, blank model, video, Wizard-of-Oz and code. Sections 6.1 and 7.1 expose the chosen methods in the games.

Step 6. Choose a Tool

Although it was decided since the beginning of the projects that Second Life would be the prototyping tool, it was necessary the use of other tools for clarifying concepts or specifying functionalities such as office or CASE ones.

Step 7. Select Design Criteria

Despite Second Life promotes a freedom content creation, there are some building and scripting constrains that must be taken into account. In addition, the prototype characteristics had to be considered for each iteration. A design guideline was used to minimize user's memory load. Thus, the game objects don't get overlooked or trivialized. In Time2Play, panels have the same measures and navigation functionality. In TREG, a specific NPC for guiding the trainee is located in the rooms.

Step 8. Create the Design

This step applied all the design rationale into the prototype. In the games was important to prioritize the elements to develop. In TREG a top-down strategy was applied whereas Time2Play applied the opposite strategy, bottom-up. In addition, Time2Play, as it was a collaborative development, required to preset the environment.

Step 9. Review the Design

The prototypes of the games were reviewed by an internal audience of researchers involved in the projects. A teacher was required as the subject matter expert in Time2Play and an expert in software development process for TREG.

Step 10. Validate the Design

After the design revision, it became necessary to validate the prototypes with external stakeholders. Time2Play used usability tests for validating the user experience and ensuring its usability. Section 5.2 specifies the validation steps.

Step 11. Deploy the Design

The development environments for each project were specific sandboxes located at the land in Second Life where the prototypes were developed. Then, these prototypes were deployed to the production environment taking into account the characteristics and requirements of each game.

6. Prototyping Time2Play

Time2Play prototype characteristics were pre-defined for choosing the prototyping method for each iteration. These characteristics were based on the Effective Prototyping Process [Arnowitz et al. 2007].

- Audience: Internal when the prototype was showed to the internal team and external when the audience was a subject matter expert or the children.
- Speed: As it was a reusable strategy, the speed of prototyping was rapid. No more than 2 weeks each prototype.
- Longevity: Long. The prototype was persistent and all its elements continued existing from the beginning of the prototyping process.
- Expression: Conceptual in a card sorting prototype for determining the concept of the project. And it was experiential using Second Life in all the other prototypes.
- Style: It was interactive as the audience could actively explore it in Second Life. Card sorting prototype used a narrative style for getting the conceptual design.
- Medium: It was digital. But in the Card Sorting prototype a physical medium was used.

6.1 Time2Play Prototyping Methods

Time2Play makes use of a reutilization strategy that joins a storytelling game, a 3D auditorium and different tools for creating a storytelling environment. This strategy takes into account the idea of a 2D

environment called Legal in which children learn by reading and writing stories [Pereira and Lopes]. Time2Play combines an auditorium and other elements developed in Second Life for offering the theater environment and tools like clothes or animations for enacting the stories.

- **Iteration 1. -** ***From Idea to Card Sorting.***

The idea of Time2Play comes from Legal. A Card Sorting prototype was used by the internal team to determine the conceptual overview of the game. Cards with the required features were sorted to define the high level functionality of the scenario panel, the clothes panels and the on-demand auditorium.

- **Iteration 2 -** ***From Card Sorting to Low-coded Prototype***

Card sorting prototype was used to get an overview of the conceptual model of the game. Then, a low-coded prototype was developed in Second Life. Several freebies were used to implement the scenarios and to offer clothes for enacting the play. It also reused the on-demand auditorium and transformed it to looks like a theater.

- **Iteration 3 -** ***From Low-coded to High-coded Prototype***

After the review session of the low-coded prototype, new functionalities were required: characters, hair and skins, special effects, animations and objects. All these features were added to the storytelling environment. A teacher was invited as a subject matter expert to validate the prototype.

- **Iteration 4 -** ***From High-coded Prototype to Product Version***

Finally usability tests validated the software requirements. Section 6.2 describes the process used in these tests.

6.2 Prototyping Time2Play Experience

Being a storytelling based game, Time2Play need resources for empowering their users to act as players in a play. Therefore, they need costumes, make-up, scenarios, story elements and special effects for enacting their roles.

In SL, costumes, make-up and story elements maybe bought, built or collected as freebies, the latter being the case in Time2Play. Scenarios were modeled and assembled using the appropriated story elements combined with animations and scripts. Special effects had to be programmed.

As the prototyping of the game was done in-world, all the changes applied to the visible objects were noticed by the members. The scripts were coded in a collaborative programming way. Due to the lack of an in-world version control system, developers were assigned different objects and tasks. The coordination was done using the local chat. They also used Instant Messages for asynchronous collaboration when they were not logged in SL.

Prototyping proved itself very useful in the interaction with the young users. They had all sort of desires that we wanted to accommodate in the game. In each prototyping cycle, a few of their whims were included. These cycles were even more important for the developers themselves. They used this interplay in order to add new features and functionalities for offering new enacting possibilities. For example, while performing Snow White and the Seven Dwarfs, when the fairy tells the prince to kiss Snow White, the kids loved to trigger the bubbly-butterfly-fairy-hearts-stars special effect, carefully scripted for their contentment.

After the third iteration in the prototyping process, 3 'proof of concept' sessions took place with 8 learners from age 7 to 12, divided into groups of 2 or 3. The usability test that was used [Dumas and Redish] consists in installing the prototype and evaluating its overall quality. It was observed the learner's navigation, understanding and interaction with the environment. Each learner used a pre-created avatar having navigation limitations on account of their age (younger than 18). Four phases of the usability test were executed: Profiling Questionnaire to figure out learners' profile, Training in Second Life and Time2Play, Main Task which is a challenge for collaborative building and enacting a story using Time2Play and Final Interview for giving learners the chance to express their views on the game and the technology. During the "Validate the Design" step in the last iteration (Section 5), the demand for additional characters, imagined or requested, brought evidence of the need to create non-player characters for this game.

7. Prototyping TREG

This section shows how the prototyping methods changed throughout the iterations in the TREG project. Also it explains our experience in the prototyping process.

TREG prototype characteristics were pre-defined for choosing the prototyping method for each iteration. These characteristics were based on the Effective Prototyping Process [Arnowitz et al. 2007].

- Audience: Internal. The researchers participated in the revisions of the prototypes.
- Speed: Rapid at the early iterations for obtaining a faster feedback and diligent at the last iterations for having more code time to get a high-fidelity and best quality prototype.
- Longevity: Short as the first prototypes were used for clarifying the project main idea. The last iterations took place having in mind a long longevity life as some of the objects were supposed to continue existing in the project.

- Expression: TREG focused on an experiential expression for testing the look and feel of the game. However, during the early iterations, a conceptual expression was needed for understanding the game unfolding using slides.
- Style: A narrative style was used in the early iterations of the process. The scenarios were specified in a template and shown using slides. Also there was a low-coded prototype in Second Life for narrating the branching stories.
- Medium: A digital medium was used for getting a true game representation.

7.1 TREG Prototyping Methods

TREG prototyping used a trial and error strategy, taking all together 5 iterations for refining the requirements and prototyping the workshop technique. There were several revision meetings during each iteration where the researchers compromised with a look and feel of the game. Second Life was used as the main tool for prototyping. Figure 9 shows the 5 iterations and the improvements of the lobby using the prototyping process.

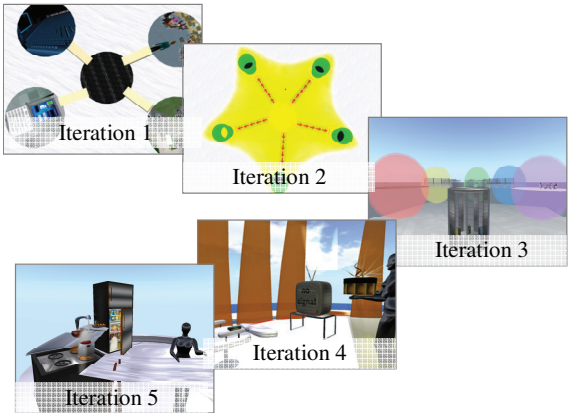

Figure 8.- Iterations improvements with the Effective Prototyping Process.

- **Iteration 1 -** ***From Idea to Quick Wireframe.***

At the very beginning of the prototyping process, the main idea was the creation of a training environment in Requirements Engineering. In the first prototype iteration, an idea of the game was conceived. A low-fidelity model presenting the first visualizations: the main reception, monoplayer games, a maze game, classrooms/discussion groups' rooms and an auditorium. It was instrumental to clarify that the goal was the creation of a classical environment for training.

- **Iteration 2 -** ***From Quick Wireframe to Wireframe***

Requirements process, techniques, management, analysis and validation were initially selected as tentative activities areas to be part of the game. A quick wireframe of a main lobby leading to 5 teleporting pods for going to these activities areas was prototype. Based on this high level structure wireframe, it was decided to narrow the prototyping to only one area, namely, the workshop technique.

- **Iteration 3 -** ***From Wireframe to Low-coded Prototype***

A new main lobby more aligned to the expected audience was prototype. Teleporting scripts were reused. In this iteration fleshed out the need of a simulation game genre, namely, branching stories. Storyboards prototyping method was not used. Detailed specifications were created using scenario templates.

- **Iteration 4 -** ***From Low-coded to High-coded Prototype***

A new high-fidelity and non-realistic main lobby was refined, including a NPC for guiding and giving the basic game resources for the trainee. A new communication feature was scripted in the NPCs and a metaphor relating cooking to making workshops was implemented.

As the simulations were defined and specified using branching stories and scenarios, the inclusion of state machine diagrams was an attractive option. Despite of the fact that they don't represent the look and feel of the game, these diagrams gave a new perspective of the game and improved the implementation in LSL.

- **Iteration 5 -** ***From High-coded prototype to Product Version.***

Finally, an advanced version for training in workshops was delivered. The main lobby, the HUD, the cooking metaphor and NPCs carried on in the following iterations for prototyping the other requirements techniques.

7.2 Prototyping TREG Experience

In order to keep the trainees immersed for enhancing their skills through a playful activity, real life metaphors are used for joining a common real life task together with a requirements engineering situation. For example, when the trainee is being trained in the workshop technique, a kitchen metaphor is proposed. The trainee enacts a *chef* role-play that must find the ingredients for the "making workshops" recipe. Figure 6 shows part of the environment implemented for this specific technique.

In order to prototype the game, first, the content was conceived taking into account requirements engineering concepts originated from Gottesdiener's book: "Requirements by Collaboration: Workshops for Defining Needs" [Gottesdiener 2002]. There she suggests 14 ingredients for accomplishing a successful requirements workshop. All these ingredients were reorganized using a structure based on the workshop process framework phases: plan, do check and act.

When the trainee chooses one of the ingredients for starting the training session, a situation related to the chosen ingredient is shown to the trainee revealing multiple paths to follow. Most paths lead to "making workshops" that don't really work. Then, collaboration patterns are offered to the trainee for choosing paths that do work. Figure 9 shows the TREG Design Process Box for the training process in the workshop technique. It takes the ingredients as input and the collaboration patterns as output.

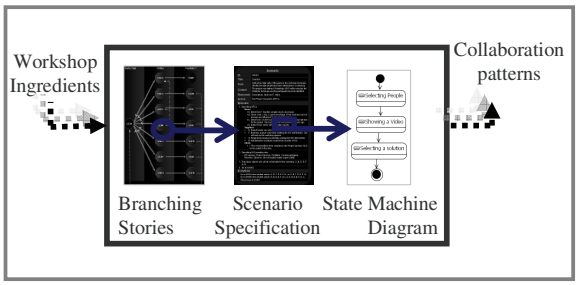

Figure 9. The TREG Design Process Box

Then, three implementation strategies were applied to design TREG. The game makes use of the branching stories genre for simulating situations that will immerse the trainee in a workshop scenario that must be planned, conducted and finished. Each node is described using the scenario template for fine tuning the situations specified in the branching stories graph. State machines use these diagrams, i.e., the specifications and transitions annotated in these templates, for modeling the system behavior.

The game makes use of the branching stories strategy for simulating different situations that will help the trainee to understand how a workshop must be planned, conducted and finished. Each state will be described using a Scenario template [Leite et al. 2000]. State machines diagrams model the game elements behavior.

Branching stories is one of the computer-based simulations genres defined by Aldrich [2005]. A graph with all the possible scenarios is created using the branching stories strategy for connecting and following the sequence of scenarios. The first trials were done using Storyboards. Unfortunately, this technique didn't. It consumed too much time for making the images to illustrate the game, instead of generating the content and sequences needed for the unfolding of the game. Branching stories, on the other hand, proved itself as a practical way to do the job.

Scenario is a description technique useful for depicting situations in an environment suitable for elicitation and specification of software requirements [Leite et al. 1997]. Each node of the branching stories graph will have a description based on a Scenario Template [Leite et al. 2000]. Moreover, exceptions in the Scenario Template lead to paths that clarify misunderstandings that are normally associated to trainees' choices.

State Machine Diagrams are used for modeling the dynamic perspective of the system. These diagrams define the states and transitions for each game object. Given that LSL is a state based language, from the early stages of the prototyping process, the relevance of these diagrams for implementing the system was clear.

All the objects and scripts were created from scratch for getting a more realistic and personalized setting. In the prototyping process, objects are built and shown as-is in-world in order to choose from the set of available objects those that will remain and those that will not be used in the next prototype. For example, 3 versions of the main building were discarded and a NPC walking capability was ruled out.

In the game, NPCs perform 3 specific functions: simulation, filming and guiding. They are used to simulate a specific workshop situation and interact with the trainee. They were filmed using the Machinima technique to create videos with several scenarios showed in-world to the trainee. Finally, some NPCs are presented to the trainees as in-world guides.

8. Conclusion

Prototyping helped to design and evaluate some aspects of the game in Second Life, and provided feedback for improving the quality of the game. The Second Life platform accommodates different development strategies as shown in Subsections 5.1 and 6.1. The prototyping process was customized for their development.

Time2Play and TREG are educational games with different audience and objectives. Time2Play is a storytelling game created for children. TREG is a game for training in software requirements engineering. Thus, they were developed in different ways:

- While Time2Play makes use of a reutilization strategy that joins a storytelling game, a 3D auditorium and different tools for creating a storytelling environment, the strategy adopted for TREG was a trial and error one where the requirements were refined.
- Time2Play was prototyped in a collaborative way; hence, some SL features had to be customized for this purpose.
- In Time2Play, the users participated in the "Validate the Design" step of the last iteration of the prototyping process. While they were playing the game, they asked for new features and some were included in the next iterations.
- Learners in Time2play had the permissions for building in the land and rezzing any objects during their performances or creating

another element for the scenario. TREG, on the other hand, was built and scripted from scratch. It demanded more time and investment for getting better resolution and definition of the objects, particles and scripts.

- TREG design was divided into two phases: Content creation and the combination of various implementation techniques. The content is based on the book: "Requirements by Collaboration: Workshops for Defining Needs" [Gottesdiener 2002]. Prototyping was instrumental in finding out the 3 techniques used in the implementation: branching stories for the unfolding of the game, scenarios for defining the game specification and state machine diagrams for modeling behavior.

There are some hardware and software specificities that were considered in these developments. They are related to bandwidth and rendering capabilities on the client side for preserving the quality of the play experience.

References

LINDEN LABS, 2009. Second Life website [online]. Available from: http://secondlife.com [Accessed 20 July 2009].

C. PETTEY, 2007. Gartner Says 80 Percent of Active Internet Users Will Have A "Second Life" in the Virtual World by the End of 2011, *Gartner Press*.

ALDRICH, C., 2005. Learning by doing; *Pfeiffer*, USA, 2005.

B. BOOK, 2009. What is a Virtual World. [online] Available from: http://www.virtualworldsreview.com. [Accessed 20 July 2009].

D. OBLINGER, 2004. The next Generation of Educational Engagement, *Journal of Interactive Media in Education,* SSN:1365-893X, May 21st, 2004.

M. CUNHA, A. RAPOSO, H. FUKS, 2008. Educational Technology for Collaborative Virtual Environments, *In Proceedings. of 12th International Conference on CSCW in Design,* China, April 16-18, 2008.

S. KLUGE, L. RILEY, 2008. Teaching in Virtual Worlds: Opportunities and Challenges, *Issues in Informing Science and Information Technology*, Informing Science Institute, 2008.

KIDZ CONNECT, 2009. Kidz Connect: Connecting cultures through creative collaboration [online]. Available from: http://www.kidzconnect.org/». [Accessed 20 July 2009].

RAMLOLL, R., BEEDASY, J., HUDNALL STAMM, B., PILAND, N., CUNNINGHAM, B., KIRKWOOD, A., MASSAD, P., SPEARMAN, R., PATEL, A., TIVIS, R. AND C. KELCHNER, 2006. Distance Learning and Simulation Technologies to Support Bioterrorism Preparedness Education, *In Proceedings of the ISCA 21st International Conference*, ISBN: 1-880843-58-7, pp 235-241.

IDAHO STATE UNIVERSITY, 2009. Virtual training at Bingham Memorial Hospital [online], Available from: http://play2train.us, [Accessed 20 July 2009].

EN YE, CHANG LIU, J. POLACK-WAHL, 2007. Enhancing Software Engineering Education Using Teaching Aids in 3-D Online Virtual Worlds", *37th ASEE/IEEE Frontiers in Education Conference T1E-8*, Milwaukee, WI, October 2007.

M. ERNST, 2006a. The Groupthink Specification Exercise, In *Software Engineering Education in the Modern Age: Challenges and Possibilities*, Lecture Notes in Computer Science vol. 4309, Dec. 2006, pp. 89-107.

ERNST, 2006B Groupthink exercise at Ohio University in Second Life. [in-world]. Available from: http://slurl.com/secondlife/OHIO%20Outreach/144/156/32 [Accessed 20 July 2009].

EMILY OH NAVARRO, ANDRE VAN DER HOEK, 2007. Comprehensive Evaluation of an Educational Software Engineering Simulation Environment, *20th Conference on Software Engineering Education & Training (CSEET'07).* Pp.195-202.

IBM, 2009. Open Encounters of z Virtual Kind skills challenge [online]. Available from: http://www-304.ibm.com/jct01005c/university/students/contests/SecondLife/index.html [Accessed 20 July 2009].

R. BARTLE, 2004. Designing Virtual Worlds. New Riders Games.

M. NITSCHE, 2005. Film live: And Excursion into Machinima", In: Developing Interactive Narrative Content: sagas_sagasnet_reader, *Brunhild Bushoff, Munich*, 210-243, 2005.

Arnowitz, J., M. Arent and N. Berger, 2007. Effective Prototyping for Software Makers, Morgan Kaufmann, Elsevier, Inc.

PEREIRA, A. R. ; LOPES, R. D. 2005 . Legal: Ambiente de Autoria para Educação Infantil apoiada em Meios Eletrônicos Interativos.. *In: Simpósio Brasileiro de Informática na Educação*, Juiz de Fora.

DUMAS, J., REDISH, J., 1999 A practical guide to usability testing *Intellect Books*.

E. GOTTESDIENER, 2002 Requirements by Collaboration: Workshops for Defining Needs, Addison Wesley, 2002.

LEITE, G. HADAD, J. DOORN, G. KAPLAN, 2000.A Scenario Construction Process. *In Proceeding: Requirements Engineering,* 2000; pags 38 - 61.

J. LEITE, G. ROSSI, F. BALAGUER, V. MAIORANA, G. KAPLAN, G. HADAD, A. OLIVEROS, 1997; Enhancing a Requirements Baseline with Scenarios. *In Proceedings: Requirements Engineering, Springer*; pags 84-198.