An Environment to Support Collaborative Learning by Modding

Sébastien George, Elise Lavoué, Baptiste Monterrat

HAL Id: hal-00869190
https://hal.archives-ouvertes.fr/hal-00869190

An Environment to Support
Collaborative Learning by Modding

Sébastien George[1,2], Élise Lavoué[1,3], Baptiste Monterrat[1,2]

[1]Université de Lyon, CNRS,
[2]INSA-Lyon, LIRIS, UMR5205, F-69621, France,
`{sebastien.george, baptiste.monterrat}@insa-lyon.fr`
[3]Université Jean Moulin Lyon 3, MAGELLAN, LIRIS, UMR5205,
`elise.lavoue@univ-lyon3.fr`

Abstract. In this paper, we propose an environment to support collaborative modding, as a new way to learn a subject. Modding can be defined as the activity to modify an existing game with dedicated tools. In a constructivist approach, we base our work on the assumption that modding a learning game can help learners to acquire the concepts of the subject concerned. We also think that modding in collaborative settings can help learners both to learn the subject and to learn to collaborate. We first propose a framework to support collaborative modding activities based on four components: the game, the Game Development Kit (GDK), contextual discussions and a knowledge map. We then propose an architecture that integrates these components on a unique platform. We finally present the results of a first exploratory study that demonstrates the feasibility and the interest of this approach for learning and the need for integrating collaborative tools in a unique environment.

1 Introduction

During the last decade, new tools have emerged in the field of video games in order to allow novice developers to create their own games. These tools are called "game editors", "game factories" or "Game Development Kit" (GDK). Meanwhile, playing learning games has proven educational benefits by immerging learners into a world where they have to invest themselves intellectually and mentally to progress, to face challenges or to accomplish quests. In this article, we are interested in a new way to learn the knowledge of a domain: modding learning games. Modding can be defined as the activity to modify an existing game with dedicated tools.

As we show in the state of the art, few studies have focused on this method of learning in education. Those who were interested in this method mainly used it to learn programming. The work presented in this paper explores new uses of learning by modding and proposes a generic framework to support collaborative learning ac-

tivities based on modding. The relative works on learning by modding is presented in Section 2. We explain in Section 3 the learning approaches supported by the activity of creating and modifying the content of a learning game in a collaborative way. We then present the framework and the tools we propose to support this activity (Section 4). As this research is exploratory, we finally present the results of a first study with 16 participants (Section 5 and 6). This study aims to test the interest of the approach and the proposed environment. We finally conclude with discussions and further work.

2 Learning by Modding: State of the Art

2.1 Users Becoming Programmers

At the beginning of gaming, the only way to modify a video game was to access its source code. Fortunately, the emergence of new types of editing tools provided users with easy access to the core of the game. The most basic way to modify a game for a user is to access its settings. Most games allow modifying for example the display mode or the level of difficulty. During the 80s, new types of tools arrived: level editors. They have contributed to the success of Lode Runner, one of the first games that allowed users to describe by themselves the initial state of a level (Djaouti, 2011). Nowadays, players have access to many different game engines allowing them to completely change the behavior of a game to convert it into a new one.

Tavares and Roque (2007) have highlighted the advantages for a game to be designed by a lot of players mixing a lot of ideas, instead of a few professionals. According to Volk (2008), "the roles of game designer and game player are obviously not a binary one, since every level of participation can be found in the modding movement". In the same thought, we understand that video games need both professional design and fans ideas. As both roles exist, modding tools have to be adapted to these different profiles.

Web 2.0 has grown thanks to the WYSIWYG ("What You See Is What You Get") editors. By analogy, the game 2.0 is becoming accessible to all thanks to a new generation of tools, providing users with a good representation of the game. Djaouti et al. (2011) define game 2.0 as any application allowing a user to create, share and play to a game content. Within this approach, some tools have been created to simplify programming (Moshirnia, 2007). These tools aim to increase the expressiveness of the mode of representation while reducing its complexity. They are often composed of two interfaces: one to edit the initial state of a scene or of a game level, and the other to modify the behavior of the game.

The drag and drop of objects from a library to the game is an example of possibility opening up level editors to a wider audience. Interfaces that offer the possibility to modify the behavior of a game are also becoming increasingly visual, as these examples show:

- Stencyl and Flip allow users to program a game without writing any code. Some blocks ("if", "then", "Boolean" ...) and existing functions can easily be dragged, dropped and ordered in a structure corresponding to the behavior of the game.
- Warcraft III editor offers further possibilities with a representation of the rules on the shape of triggers (composed of an event, conditions and actions). So programming concepts such as loops and functions are implicit and do not require to be known by the developer.
- Kodu Game Lab and Game Develop use an even simpler way to represent the behavior of a game: rules consisting only of conditions and actions. The game engine works as if the conditions were constantly tested, and corresponding actions triggered if necessary. This system is so simple that Kodu allows children to design their own games. Kodu also allows access to the behavior editor within the game itself: to select an object leads directly to the list of the rules that define the behavior of this object.

These editors are just a few examples of those developed recently. For more details, Djaouti, Alvarez and Jessel (2010) present a study of fifteen "gaming 2.0" tools and explain how they support the design of serious games. However, Postigo (2008) has shown the limits of modding proprietary games. In such games, modders do not have the right to share their creations. That is why we mainly studied free games and free GDK for educational purposes.

2.2 Experiences of Modding in Educational Contexts

Loh and Byun (2009) have created a serious game by modding the NeverWinter Nights 2 game. They created Saving Adryanee, a game whose "in-game" objective is to create a healing potion, and whose educational goal is to teach the concepts of chemistry. This complex work was conducted by a team of four persons in just two months, thanks to the experience of those who have conducted modding projects before them. As in this study, experienced developers and beginners usually communicate using Web collaboration tools (chat, forums), or sometimes tools integrated into the game itself (conversations between avatars in online games). Strong support exists within these communities, allowing new developers to quickly overcome the problems they meet. Based on this experience, the authors show that "instead of waiting for new serious games to be made available, the students and teachers could learn to use GDKs as learning tools. For instance, all high school freshmen may be required to learn the GDK, and to use it as one of the project presentation tools throughout high school."

In several studies, modding has been used as an activity for learning programming. Students were able to create mods themselves. For instance, McAtamney, O'Shea and Mtenzi (2005) make students use the Crytek engine to model the future of their university campus. Through this experience, students have learned how to use C++, direct X and the scripting language LUA, while developing their skills in mathematics, physics, 3D design and event-driven programming. In another study, El-Nasr and Smith (2006) showed that there are different modding tools adapted to different types

of learners. They experienced the Warcraft III Editor with high school students so that they learn the basics of algorithms. Students were able to create a game in just three days. They then used Web Driver and Unreal Engine 2.5 with students in Computer Science. In addition to the technical skills acquired, these experiences have enabled students to become familiar with the software development process. Indeed, the steps of a mod development (specifications, design, implementation, and test) and the steps of software development are roughly the same (Cignoni, 2001).

3   Learning Activities Based on Modding

3.1   Learning Content with a Constructivist Approach

Up to now, studies have shown that modding has mainly been used to teach computer science. We believe that modding can be used to learn in any domain. With modding, the content of a learning game could be the learning goal (Monterrat, Lavoué & George, 2012). In fact, if learning games help to acquire the knowledge contained in the game, we can assume that changing these games can be appropriate to deeply learn a subject. Oblinger (2006) explains that the educational potential of a learning game depends on the level of involvement of the player. The goal of our study is to create good conditions of involvement by allowing the player to change the game. A moddable learning game provides the user with a new way of learning with a game. This idea is clearly consistent with a situated approach of learning (Lave and Wenger, 1991), offering the user a way to build knowledge while designing part of a game.

Somehow, we want to allow a learner to play a teacher role by designing an educational game that others will use to learn. Teaching knowledge involves a deeper level of understanding, so it is a good way to learn. With modding, we can propose to the learner to play the role of the teacher to reach another level of expertise. As suggested by Loh and Byun (2009), students can be asked to create serious game mods for use by others in order to demonstrate their understanding in the selected subject. Furthermore, learners are led to seek further information by themselves, so they can acquire concepts that go beyond the knowledge of the teacher. According to the Magic Bullet model (Becker, 2011), the learners will increase the "external learning" (what learners will look outside of the game), and include it in the "mandatory learning" (knowledge which is necessary to have in order to progress in the game). Then, the evolution of the game will benefit the future players during a new learning session. In addition, this method of learning by modding allows the knowledge to evolve in order to be always up to date, like a wiki users keep faithful to the current world that is constantly evolving. Such a game can also be well adapted to different kinds of learners by being modified by the learners themselves.

3.2   Collaborating to Learn and Learning to Collaborate

According to Scacchi (2011), modding is a way to learn to work with others as a person who creates a mod trains to work in a team and to manage a group project. During the experiments of El-Nasr and Smith (2006), the students first learnt to divide the

tasks among groups of two and share their skills. They then went beyond, exchanging with other groups because they understood that communicating about their project and discovering others would be beneficial. In bigger projects, modding also teaches how to manage a team and sometimes how to resolve conflicts. For example, Loh and Byun (2009) have shown that, when they develop a mod, they cannot freely change the planned format of the game without offending the writer of the team. Moreover, when the number of people involved in a modding project grows, more errors and problems could appear but the quality of modding could be higher and a self-regulation mechanism may occur.

Ang, Zaphiris and Wilson (2005) have studied this "self-regulation mechanism" in communities that write wikis. We totally meet up with the idea that "by engaging learners to construct something meaningful and sharable with their peers, learners can learn by putting the knowledge into practice". The same phenomenon can occur during learning games modding, provided the game is designed as a collaborative media and modding is seen as a collaborative activity. In this way, we propose in this paper an approach to let the learners collaboratively modify the knowledge embedded in the game, as a wiki reader could participate to the document writing. In order to support this approach, we propose a particular framework in the next section.

4 An Environment to Support Learning by Modding

### 4.1 A Framework for Integrating Collaborative and Pedagogical Aspects

In this section, we propose a framework supporting collaborative modding in the context of learning activities (see Figure 1). This framework aims to help learners to acquire the knowledge of a specific domain predefined by a teacher. This framework consists of three parts: (1) the GDK associated with the game, (2) contextual discussions and (3) a knowledge map. The identification of these three main parts of the framework and their relationships are our main contribution for a computer supported collaborative learning by modding environment.

Game and GDK

Games are usually divided into several distinct areas called levels, worlds or cards, according to the game. We generically call them "scene". Other elements compose a video game: textures, sounds, characters, items, etc. We call them all by the generic term "object". Each scene of a game consists of a set of objects and a set of rules that describe their behavior. The GDK (Game Development Kit) is the tool associated with a game that allows modifying it. On the one hand, the GDK should to be simple enough for a quick start, with no programming skills required. Even if modding requires a learning step, this phase should be as short as possible so the user can work on the content of the game from the beginning. On the other hand, the GDK should be rich and powerful enough to allow the modder to modify the structure of the game in depth and to define new behaviors. A system of rules based on conditions and actions (as proposed in Game Develop) seems to be a good compromise. These rules allow

manipulating the objects of the game, each object type having conditions and actions associated with it. A rule can be composed of any number of conditions. If there are none, the action is triggered continuously. If a rule contains multiple conditions, they all must be checked so that the action is triggered, as if they were separated by the logical operator AND. The easiest way to perform the logical operator OR is then to put the conditions in different rules. Here are some examples:

‒ Condition for a sound: "Is the sound <victory sound> played now?"
‒ Condition for the keyboard: "Is the left arrow pressed at this time?"
‒ Action for a sound: "Play the sound <explosion>".
‒ Action for an image: "Move the image <Avatar> 10 pixels to the left".
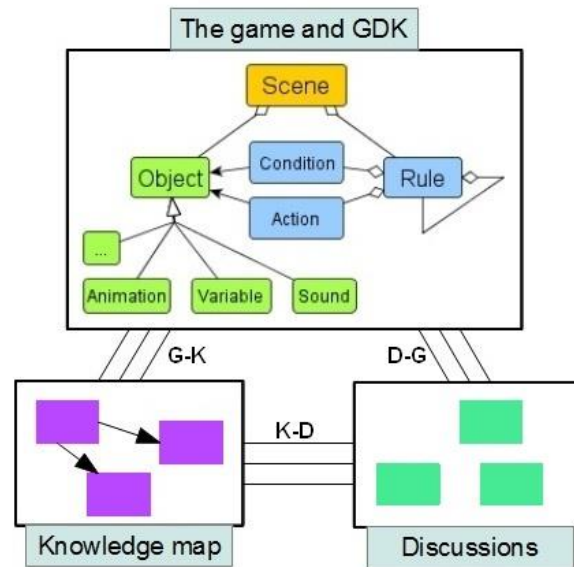
Fig. 1. Framework for a collaborative learning activity based on modding.

Contextual Discussions

To work together, modders should initially be able to share ideas and opinions; to make the game emerges from collective choices. They also need to share the elements of the game so that it is actually the result of a collective work. A social platform supporting discussions and sharing of elements seems necessary. Contextual discussions have proved to be useful in educational distance situations and could be implemented for instance with contextual forums (George, 2004). That is why we propose to link contextual discussions to each elements of the game (for instance scenes or object, D-G links on Figure 1). They help students to have structured discussions on the game elements on which they are working. In the context of a modding activity, these discussions can take place in virtual spaces in order to let the learners exchange

on knowledge. These contextual discussions can also be a way for the teacher to monitor, help and communicate with the students during the activity at a distance.

Knowledge Map

The knowledge map developed by the teacher contains the knowledge of the domain to learn. On the one hand, each knowledge element is linked to one or several elements of the game that allow learning it (G-K links on Figure 1). On the other hand, each knowledge element is also linked to a discussion (K-D links on Figure 1), allowing students that work on the same elements to discuss on it. Here are two examples of use in educational contexts:

— The knowledge "knowing how to apply the law of gravity" could be linked to a set of rules simulating this law in the game, and to a discussion between students about gravity relative to the size of the planet.
— The knowledge "knowing how to recognize a platypus" could be linked to the scene of a zoo and to a discussion of the differences between reptiles and mammals.

We suggest that the teacher is the supervisor of the learning activity by modding. The teacher is the expert of the domain knowledge and has to take into account the different student's profiles when preparing the learning by modding situation. With such a framework, the teachers' tasks are:

— To develop a learning scenario in accordance with the educational objectives.
— To create the first elements of the game that the players will modify.
— To manage and to monitor the modding activities to help learners (technically or cognitively), and ensure they meet the objectives.
— To evaluate the learners.

4.2   Integrating Tools to Support Collaborative Modding

We argue that the game and its corresponding GDK have to be integrated in a unique platform. Otherwise, switching from the game to the GDK, and from the GDK to the social platform could be a barrier for learning. That is why these 3 facets of game modding (game, GDK and discussions) have to be parts of a unique tool, as illustrated on Figure 2. Some studies have highlighted this need, for instance, McAtamney et al. (2005) noted that, in a modding activity, the students choose to use the Crytek engine particularly because it offers all the necessary tools (object database, scripting language and 3D tools) in a single environment. Furthermore that aspect matches to the definition of Game 2.0 given by Djaouti et al. (2011).

For the activity to be collaborative, it is essential that the game elements are stored on a server. Each user (client) must also have a local version of the game in order that individual tests do not affect the collaborative project. We propose to save scenes (S), objects (O) and rules (R) in a xml file that contains links to the files corresponding to the manipulated objects (images, sounds, ...). In this way, it is then easy to manage

this set of files with a version control system we have included in the platform (in our case, SVN). The two main SVN commands are "update" (update the local version of the game) and "commit" (apply our changes to the collective project). Because game development involves the possibility of errors, the system should also save on the server earlier versions of the game and provide access to all these versions. Finally, a version control system can also merge two projects when necessary, for example when two learners have made changes at the same time on the same game.
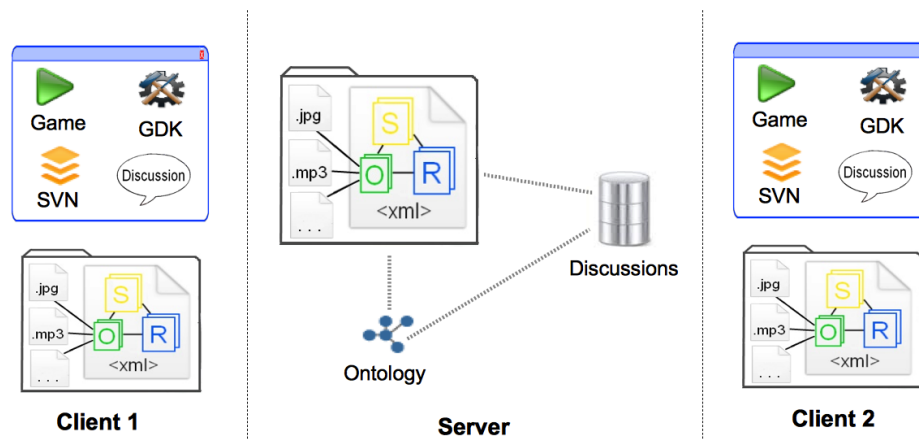

Fig. 2. Architecture of a collaborative learning by modding environment

5   Exploratory Study

5.1   Context and Participants

16 persons participated in this study. The participants were mainly students in higher education (n=14) and a few retired persons (n=2). 7 participants had prior knowledge on programming.

We proposed to the participants a prototype of modding environment. This prototype is an implementation of the system described in the previous section. According to an iterative approach, we focused this first experiment on the collaboration functionalities and we decided to not implement the knowledge map in this version. This prototype combines existing software, each addressing some of the requirements.

We chose a GDK called "Game Develop" to be the basis for our application. A wide range of conditions and actions included makes this editor expressive enough to create any kind of 2D games. In addition, the rules displayed in ordinary language makes it understandable by people without programming skills. The game was based on a central scene in which the player can find doors that give access to three other scenes. These three scenes were already created in the game, so that the students could modify them. The interface was pretty basic, in order to facilitate the use of the game by novices.

To support the discussions, we created a contextual forum available in a browser, using HTML, PHP and MySQL. In this way, each scene of the game has its own discussion thread. In parallel, the project teamwork was managed with SVN software, which allows multiple users to work remotely on common files while maintaining their versions up to date.

In order to unify the tools, a unique program, created in C language, started Game Develop (on the team project) and a browser (on the forum of the team) in one click. The participants had therefore to follow these steps at each use:

1. UPDATE (update of the local project with the contributions of other members).
2. Searching, modding, participating in discussions, etc.
3. COMMIT (update of the collective project on the server-side with the local version of the project).

5.2 Research Questions and Methodology

We used an exploratory approach in this study, as we still know little about the use of a modding environment for learning. In line with the objectives of the environment we propose, this study aims to study:

− The interest of a collaborative modding activity for learning.
− The effect of the collaborative tools we integrated in the learning activity.

The participants were asked to modify the game by using Game Develop so as to learn the Esperanto language. We divided the 16 participants into two groups: a test group and a control group. The test group was provided with the collaboration functionalities of the prototype (the contextual forum and the SVN). The members of the control group had to find by their own the means to communicate and exchange their parts of the project.

Each group was divided into two teams of 4 members (teams A and B in the control group, teams C and D in the test group). Each team has conducted the modding activity according to three main phases:

1. In presence course, by class:

- The students assisted a course on the basis of Esperanto during one hour.
- They then had a few minutes to play the game to be familiar with it.
- They finally were trained to use the GDK during an hour, with the help of the teacher.

2. During a week, the students were asked to collaborate at a distance (by group of 4) to modify the game according to the instructions. We advised them to "mod" during about 1:30 hour, with the possibility to do it in several times.
3. After this modding phase, the students of each team could play the game modded by another team in the same group (test or control).

We collected several data on this activity:

— The game that has been developed by each team. The teacher who made the course on Esperanto has evaluated these productions.
— The answers to a questionnaire about their opinion on the activity and the proposed environment. 15 participants answered the questionnaire at the end of the study.

6   Results

6.1   General Results on the Activity

We first report the observations made by the teacher who monitored the work of the participants during the study:

— The members of the team A (control group) allocated tasks and performed only the requested work for the activity.
— Only two members of the team B (control group) were involved in the project, the other two members have encountered technical problems.
— The team C (test group) has been the most active and has produced most advanced game.
— The members of the team D (test group) have not worked at home; they did not give any production.

We analyze in this section the responses of the students to the questionnaire with a global approach. We also report their opinion on the modding activity they carried out: the interest for the modding activity, the learning opportunity and the interest of playing the game modded.

According to Table 1, almost all participants were interested in the collaborative modding activity. Furthermore, most of them enjoyed the activity. Those who did not enjoy are the members of the team D who gave up the activity. These results confirm the potential of this type of activity for the involvement of the learners. As stated by a participant: "*Although I'm not a game player, I liked a lot this way to learn Esperanto*"[1].

Table 1. Responses to the questionnaire on the interest of the activity (n=15).

	Not at all	Not much	Yes a little	Yes
"Were you interested in the activity?"	0%	7%	13%	80%
"Did you enjoy the activity?"	0%	13%	33%	53%

We asked the participants about the pleasant aspect of this educational approach:
- "*Finally, we feel we have created something (sense of being useful), as opposed to a project whose outcome is simply a PowerPoint presentation or a report. I find it is rewarding*".

---

[1]   These responses in French have been translated into English in accordance with their original meaning.

- "*Creating a game that aims to teach Esperanto to someone else is a good idea because, firstly, we learn Esperanto content by creating this game and, secondly, we have the feeling that it could be useful for other learners!*".

- "*The fact that the learning activity leads to a concrete "object", in this case a game*".

As shown in Table 2, the results let us think that the activity helped participants to learn Esperanto. They not only learned during the course, but also by modding the game and by playing the game developed by the other participants. Those who did not learn Esperanto are the members of the team D who gave up the activity.

Table 2. Responses to the questionnaire on the way to learn Esperanto during the study (n=15).

	Not at all	Not much	Yes a little	Yes
"Did you learn notions of Esperanto by this activity?"	0%	7%	20%	73%
"Did you learn notions of Esperanto by trying to mod the game?"	13%	0%	60%	27%
"If you have tested the modifications made by the others, did you learn new vocabulary?"	13%	7%	33%	47%

We also asked the participants if the aim of the activity (to learn Esperanto in a ludic way) has been reached. The responses are encouraging (1):

- "*Yes, because I felt that I was playing so I had fun trying to mod the game. I realized once the experiment is finished that I remember very well what I have learned in a ludic manner*".

- "*Yes! For once this is a subtle way to tackle pedagogy, placing barriers to work a language in a clever manner, without thinking about it. A success in my opinion*".

We then asked the participants for comments about the interest of playing the game developed by other participants (1):

- "*It was a good idea, and we continue to learn when we try the other games*".

- "*This is very interesting, because it allows you to see other people's ideas and to test other Esperanto exercises*".

These comments show that modding can be a peer learning activity as learners can learn with the productions of the others.

6.2   Results on Modding with Integrated Collaboration Tools

In this section, we analyse the results per team, so as to study the impact of the use of integrated collaboration tools (test group) on the activity. We have first observed the games developed by the participants. We present in Table 3 the level of the productions of the four teams for the modding activity, according to two criteria:

─ The level of modifications made in the game: simple (copy of rules by changing the text, that was sufficient to follow the instructions) or advanced (modifications of rules, advanced modifications of the behaviour of the game).

— The level of integration of elements of the domain (Esperanto): only vocabulary taught during the course; also concepts taught during the course; or new concepts or vocabulary on Esperanto.

Concerning the accessibility of the modding activity, 4 participants that had no programming skills succeeded in making advanced modifications in the game. More generally, the results show that previous programming skills have no impact on the level of the productions delivered by the students and on their involvement in the activity.

Half of the participants have tested the modifications made by the other members of their team: 3 members of the team B and 4 members of the C. So the availability of integrated collaboration tools had no obvious impact on this result, as there is no difference between the control group and the test group.

Table 3. Productions of the participants according to the programming skills (n=16).
(P = Programming skills, M = Modifications in the game, U = Using vocabulary and concepts, T = Testing other games).

Group	Control group								Test group							
Name of the team	A				B				C				D			
Has programming skills		P	P					P	P				P		P	P
Made simple modifications	M	M	M	M	M	M			M	M	M	M				M
Made advanced modifications	M				M				M	M	M				M	
Used vocabulary taught during the course	U	U	U	U	U	U	U		U	U	U	U				U
Used concepts taught during the course	U				U		U		U	U	U	U				U
Used new concepts about Esperanto					U				U	U		U				
Tested the modifications made by the other members of his/her team		T				T	T	T	T	T	T	T				

While the results for the two control teams are rather similar, we observe that the results for the control teams are very different. While the team C has produced high-level deliverables, the team D gave up. As the members of the team D have gave up the activity, we cannot conclude on their way to carry out the activity. However, we observe that most of the members (3) of the team C have made advanced modifications in the game, against only one member in the teams A and B. Furthermore, all the 4 members of the group C have integrated concepts taught during the course, against only one in the group A and two in the group B. Furthermore, 3 members of the group C have integrated new concepts in the game, while only one people in the other

groups (group B) done this high-level activity. As a complement to these observations, 3 members of the team C declare in the questionnaire that the SVN tool helped them to share the productions within their team (one non-response). We can so conclude that, in this exploratory study, the use of integrated collaboration tools helped the team C to produce more high-level productions than the teams A and B. We can also notice that programming experience have no influence in the acceptance, neither in the ability to deeply modify the game (4 advanced modifications on 6 have been done by learners without programming skills).

The collaborative tools were also significant for the teacher, as they helped to manage the group. The forum was a way to send a message to all the members of a team, and the SVN was an easy way to access to their work when it was over. For instance, the production of team C was available on the SVN when it was time to give it back, whereas the control group teams have sent the work in several parts, so the teacher had to merge them manually.

7 Conclusion

In this paper, we have presented collaborative modding as a new way to explore learning games for educational purposes. Collaboratively modding games not only help learners to learn the content of the games, but also to learn to collaborate. Based on these assumptions, we proposed a software architecture to support such an activity. This architecture has been implemented in an environment composed of four elements: a game, a Game Development Kit, a contextual forum and a version control software. The results of an exploratory study show that this approach is accessible to non-programmers, that it is of interest to learners and that they can learn the concepts of a domain (the Esperanto language in this case). This study also shows the need for collaborative tools integrated into the modding environment to help learners to make advanced changes in the game. We are aware that such an activity requires more time than more traditional ways of teaching for both teachers and learners. Thus a teaching activity should not be based only on collaborative learning by modding, but it should be a part of it.

Although this first exploratory study was conducted with a rather few participants, it leads us to identify the potential of such an environment. On a short term, we should focus our work on the development of an environment that better integrates all the needed functionalities on a unique interface. We should also integrate the knowledge map to test its utility in a following experiment. In fact, we believe that this innovative learning environment requires a larger scale experiment that could provide more quantitative results.

Furthermore, we have proposed several tools for students but rather few for the teachers and/or tutors. One future issue of this work is the development of an assistant system for the teachers to prepare the modding environment. This system would help them to build the knowledge map, to link the knowledge with the elements of the game and then to monitor the students during the modding activity and to assess their productions.

References

1. Ang, C. S., Zaphiris, P., & Wilson, S.: Wiki-supported Collaborative Narrative Construction in Game Communities. *In:* ECSCW'05 workshop on Computer Games CSCW, Paris, France (2005).
2. Becker, K.: The Magic Bullet: A Tool for Assessing and Evaluating Learning Potential in Games. *International Journal of Game-Based Learning*, *1*(1), 19-31 (2011).
3. Cignoni, G.: Reporting about the Mod software process. *Software Process Technology*, 242–245 (2001)
4. Djaouti, D., Alvarez, J., & Jessel, J. P.: Can Gaming 2.0 help design Serious Games?: a comparative study. In: Proceedings of the 5th ACM SIGGRAPH Symposium on Video Games, pp. 11–18 (2010)
5. Djaouti, Damien, Alvarez, J., & Jessel, J.-P.: Classifying Serious Games: The G/P/S Model. *Handbook of Research on Improving Learning and Motivation through Educational Games: Multidisciplinary Approaches,* IGI Global, 118-136 (2011)
6. El-Nasr, M. S., & Smith, B. K.: Learning through game modding. *Computers in Entertainment (CIE)*, *4*(1), 1-20 (2006)
7. George, S.: Contextualizing Discussions in Distance Learning Systems, *Proceedings of the In: 4th IEEE International Conference on Advanced Learning Technologies,* pp. 226-230. Joensuu, Finland (2004)
8. Lave, J., & Wenger, E.: *Situated Learning. Legitimate Peripheral Participation*. Cambridge University Press (1991)
9. Loh, C. S., & Byun, J. H.: Modding Neverwinter Nights into serious games. Digital Simulations for Improving Education: Learning Through Artificial Teaching Environments, 408–426 (2009)
10. McAtamney, H., O'Shea, B., & Mtenzi, F.: Using the Crytek game engine in the Dublin Institute of Technology, In: Proceedings of the 7th International Conference on Computer Games, Angoulême, France (2005)
11. Monterrat, B., Lavoué, E., & George, S.: Learning Game 2.0: Support for Game Modding as a Learning Activity. In: *6th European Conference on Games Based Learning,* pp. 340-347. Cork, Ireland (2012)
12. Moshirnia, A.: The educational potential of modified video games. *Issues in informing science and information technology*, *4*, 511–521 (2007)
13. Oblinger.: Games and Learning, *Educase quarterly*, *29* (3), 1–7 (2006)
14. Postigo, H.: Video Game Appropriation through Modifications: Attitudes Concerning Intellectual Property among Modders and Fans. *Convergence: The International Journal of Research into New Media Technologies*, *14*(1), 59-74 (2008)
15. Scacchi, W.: Modding as a Basis for Developing Game Systems. In: Proceeding of the 1st international workshop on Games and software engineering, pp. 5-8, Waikiki, Honolulu, USA (2011)
16. Tavares, J. P., & Roque, L.: Games 2.0: Participatory Game Creation. In: Proceedings of the 6th Symposium on Computer Games and Digital Entertainment. São Leopoldo, Brazil (2007)
17. Volk, D.: Co-creative game development in a participatory Metaverse. In: Proceedings of the Tenth Anniversary Conference on Participatory Design 2008, pp. 262–265. Bloomington, IN, USA (2008)