2009 Second International Conferences on Advances in Computer-Human Interactions

FPS Extensions Modelling ESGs

Dr John R Rankin and Sandra Sampayo Vargas
Games Technology Laboratory
La Trobe University- Bundoora
Victoria, Australia
j.rankin@latrobe.edu.au; sandrasampayo@gmail.com

***Abstract*—In this paper we show that the FPS paradigm has metaphorical correspondences with a low level category of ESGs. Extensions of the FPS paradigm lead to simple concept learning lessons analogous to the needs of teachers for the development of low level ESGs for use in classroom teaching. The main FPS extension discussed in this paper is the FSM module. The teacher is not required to recompile any code but simply to provide the data representing the FSM corresponding to each concept learning lesson in the target ESG. The purpose of this research is not to develop the full ESG but rather to show that the FSM module coupled with other extensions to the FPS game development tools provides a means for building future ESG development tools.**

First Person Shooter; Serious Games; Finite State Machines; Game Engine.

I. INTRODUCTION

The world of video games is dominated by non-Serious Games – the so-called casual games [1]. The commercially successful Serious Games (SGs) are small in number and very high in development budget and this raises the question of why there is currently a lack of commercial value in general purpose SGs [2]. The main development emphasis has been on casual games, and correspondingly, the available game development tools have been predominantly by far in that market. The emphasis on casual games has also resulted in a lack of available tools for those game designers wanting to make SGs [3].

In this paper we want to investigate what tools are particularly appropriate to the development of SGs and find out what is needed in SG tools beyond the tools currently available for casual games. We further narrow the focus to consideration of just Educational Serious Games (ESGs) and in particular the development of ESGs for the classroom by teachers who have no skills or training in games technology programming or in digital art and animation. The question that was posed by Rankin et al [4] as the Teacher's Dilemma is: What software tools would a teacher need to rapidly create an ESG for his class in a two month time frame where the teacher has no games programming skills or digital artwork skills and a budget limited to no more than a couple of hundred dollars? Rankin et al [4] reported that such tools are not available to teachers at present though there is now a strong demand for such tools [5,6]. The current paper pursues this question and reports on software developments that attempt to fill the gaps where ESG tools are lacking in particular the development of an FSM module and coupling of the ESG to a database.

FPS And ESG Similarities

Rankin et al [4] showed that there are many ways to create casual games today. It seems that the most popular and easiest genre of video game to create with today's tools is the First Person Shooter (FPS) game [7]. In the typical and highly popular FPS game, the player moves around in a game world looking for "enemies", "bots" or "monsters" and tries to kill them all by firing directly at them using a choice of rifle-like weapons [8]. Anything that moves in these games is an enemy that must be shot at and killed. The typical FPS game world is in essence a 3D building that the player can explore and move around inside from room to room [7]. The player's goal is to hunt for enemies and kill them all. The more that are killed the higher the score that the player gains. (There is no score for only wounding a monster however.) Rankin et al [4] discussed the following 5 ways of creating such casual games: use of Game Engines, use of middleware software, use of world editors, modding with a scripting language, use of specialist game programming languages and low-end complete game development systems. Rankin et al [4] concluded that virtually none of these tools provided assistance for SG development or addressed the Teacher's Dilemma except for the low-end game development systems and that these however did not provide the capabilities for student learning rates and progress monitoring typically needed in ESGs.

We decided for this research to regard an FPS game as metaphorically representing simpler levels of an ESG game and to try to establish the correspondences in this metaphor. We used FPSs because of their prevalence in the world of video games and the great abundance of tools, support and information on how to build them. The FPS games are easy and quick to build and many tools are oriented specifically to provide FPS features [8]. For example the Irrlicht Game Engine is open source and freely available to all developers. It provides a range of features such as sky domes, maze design, particle systems, loading and handling of game objects as meshes in various formats, player controls, monster AI animations and sounds with an extra audio library [7,9].

Similar to an FPS, an ESG contains a game world which is equivalent to a building of corridors and rooms for the player to explore and move around inside. There can also be signs on the walls giving indications of where the player

978-0-7695-3529-6/09 $25.00 © 2009 IEEE
DOI 10.1109/ACHI.2009.70

should go next or what he should do next. In contrast to the FPS however, the actions that the player will take in an ESG are quite different. Instead of hunting for monsters and then killing them the player is confronted with course concepts in the form of exercises which the player must then satisfactorily perform. An example of this is the ChemLab ESG for teaching first year Chemistry, where in the game, the player enters into the chemistry lab room and takes various apparatuses out of cupboards, and chemicals out of other cupboards to lay them out on the bench for him to run a prescribed chemistry experiment [10]. The experiment may be as simple as dissolving two different salts in crystal form into water in a test tube and then pouring the solutions into a beaker and observing the precipitation that results from the chemical reaction. The player is required to note down the colour of the precipitant and write out its chemical formula. The player may have to enter the correct observations or answers to the exercise in the game itself or else into a special workbook provided for that purpose by the teacher to obtain a grade for that part of the ESG and then he can move on to the next stage in the game. This is clearly quite different from simply hunting and killing monsters and yet we can explore the correspondences where learning a concept in an ESG is a personal achievement just as much as hunting and killing is an achievement in an FPS game. Shooting at a monster is a single step process though the player may have to repeat it several times until it is done correctly. In contrast to shooting at a monster, learning a concept is a multistep process that can take longer to get right. In the Chemistry ESG example, the player must go to the cupboards, take out the right apparatus and chemicals, add water to them in the right vessels, mix the two solutions together into a beaker and record the results in order to accomplish the task correctly. This is a sequence of distinct actions by the player. The sequence is order dependent though some minor reordering is acceptable such as the order in which things are taken out of the cupboards as this does not make any difference to learning the concept.

We can mimic this to some extent in the FPS metaphor by requiring the player to kill the monster according to specific kill rules different for each type of monster. An example of such a kill rule could be: Use weapon A to shoot the monster in the right arm, then weapon B to shoot it in the left arm and then weapon C to shoot it twice through the torso. The player could also be required to visit certain sites in the FPS game world (with sound effect feedback) in order to be eligible to kill a specific monster. To complete the analogy with ESG learning exercises, the player might have to write down the dying words or last message of the monster he just killed to get the score for that monster. These rules can be made arbitrarily complex and detailed to try to imitate the complexity in learning concepts. However exactly the same rules must apply to all monsters of the same appearance so that once a rule is learned by the player the game will test and retest the player's learning and speed of recall of his learning by giving him repeated instances of encounters with that same type of monster (spread out amongst encounters with other types of monsters). In the FPS apart from weapons of different sorts, we also have different coloured bullets to fire. So the combination of bullet colour, weapon type, monster body parts as targets and eligibility visit triggers begins to provide a greater number of action choices for monster kill rules in analogy with the large number of possible actions for players in solving ESG learning concept exercises. The killing rules for each monster type can be represented as a finite state machine (FSM) and the player has to recognize the current state of the monster to know where he is at in the kill process [11]. The state of the monster can be reflected to the player in the monster's appearance or subtle change in its behaviour.

By using this metaphorical correspondence between FPSs and ESGs we can conduct research on ESG design and tools while using the available FPS technologies. The correspondences also enable us to use the standard game terminology in the didactic setting. Moreover, it highlights the difference between common FPS and educational standards in that the mentality level and level of thinking required in a FPS is well below that required for educational purposes. Of course the concocted monster kill rules are quite arbitrary and meaningless whereas the corresponding concept learning of the ESG provided by the teacher is purposeful and significant. However, arbitrarily complex monster kill rules may end up ruining the fun aspect of an FPS and thus indicate why ESGs (at the same metaphorical correspondence complexity level) may be less fun for young people to play than the simpler FPS games.

II. ESG REQUIREMENTS

According to the Teacher's Dilemma, the teacher wants to be able to write part of his teaching course material into the ESG without the need to write C++ source code or use a compiler. The teacher needs the flexibility to be able to change the game contents and to be able to edit – delete or change – nearly all of the game features. The following list gives the essential game features that should be editable through an ESG tool.

- Can you add new rooms to the game world?
- Can you edit/change/delete rooms?
- Can you change the wall textures?
- Can you edit what items are in the rooms and their positions?
- Can you change the signs on the walls?
- Can you add & delete signs?
- Can you change the narrations and the narration triggers?
- Can you add new monsters (same or different types)?
- Can you change/edit existing monsters to other (predefined) monster types?
- Can you set the monster kill rules?
- Can you add new weapons and bullet types?
- Can you distinguish hits to different body parts?

There are program features of ESGs that distinguish them quite markedly from all other game categories such as the FPS genre. For an ESG the teacher needs to be able to track

individual student performance and to know how effective the ESG is as a teaching tool for each of his students. To have this feature in the game the ESG software needs to have a login process at the start in order to identify the student. The ESG should be connected to a database in order to check for valid login details and to record individual student progress and assessment. The ESG needs to have functionality to measure students progress and assessments also [12], [13]. The ESG database should store the following data:

- Student name and password details
- Where has he been?
- How long has he taken?
- How many monsters has he killed and which types?
- Has he followed the correct order?
- How fast is he at recalling solutions?
- How much of the game world has he explored?
- Session dates, times and durations.
- Progress and scoring information.

Fig. 1 shows Game Engine extensions (asterisked) required for ESG engines. The simplified method of assessing student's learning in these ESGs is for the teacher to provide each student with a workbook. Every time the student completes a set exercise in the game, he is required to write down an observation or result into the allocated spot in his workbook. This prevents cheating since the student could not know the results before completing each exercise. After all the students have finished the ESG class, the teacher collects their workbooks for marking and recording of the results. Having the student break from the game to write a workbook is disruptive to his game immersion experience. It is superior to automate this process and the FSM module (see next section) does also enable us to have an automated assessment system in replacement of workbooks.

III. FSM MODULE

The FSM module is a class for declaring an FSM object for each learning exercise in the ESG [11]. It is integrated into the ESG engine according to the architecture shown in Fig. 1. Each state in an FSM has a flag to indicate whether the player has achieved that state or not yet in the game. A simple example of a concept learning lesson is taken from the ChemLab SG of Section 2 to demonstrate the purpose of the FSM module. In this experiment the pupil is required to perform the following experiment involving 7 steps or actions within the game:

1. Get containers 1, 2 and 3 from the cupboard
2. Get the chemicals 1 and 2 from another cupboard
3. Dissolve chemical 1 in water in container 1
4. Dissolve chemical 2 in water in container 2
5. Pour container 1 into container 3.
6. Pour container 2 into container 3.
7. Observe and note down the reaction.

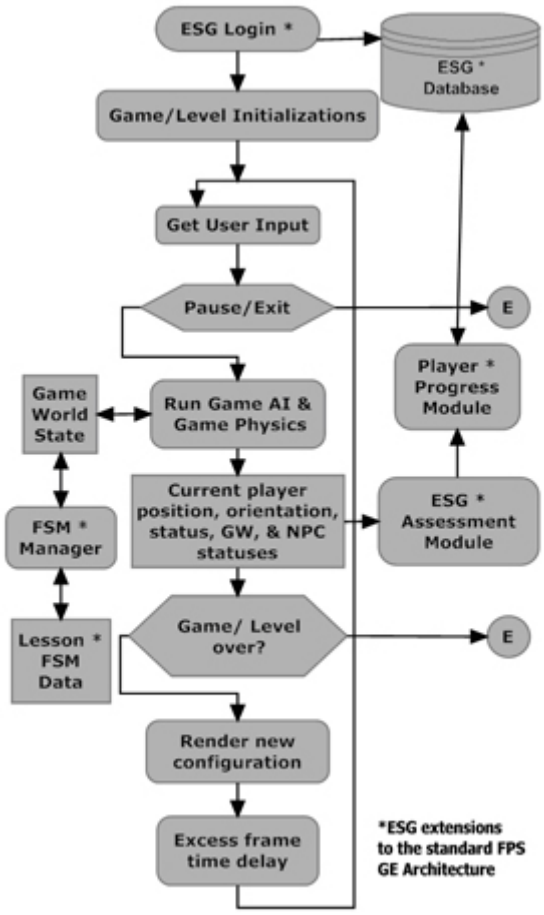

Figure 1. Game Engine Extensions required for ESG.

If these 7 steps have to be done in this order then each step becomes a state in the FSM and they are in a linear sequence. However, some of the steps within this learning sequence can be in a different order without changing the validity of the student's learning achievement. The order of steps 1 and 2 may be reversed. Likewise the order of steps 3 and 4 may be reversed and the order of steps 5 and 6 may be reversed. In this case extra states are required to say that steps 1 to 2 are done, steps 1 to 4 are done, and steps 1 to 6 are done. There are therefore 11 states in the FSM and the states do not have a linear order as shown in Fig. 2 which is the FSM representing achievement for this learning concept lesson. The player's actions can take him closer to accomplishing the learning exercise or further away from it.

The text file format for the FSM module to read in and create the necessary FSM objects for the ESG is as follows. The first line says the number of FSMs to create. The next line is the number of following lines defining the first FSM. Each line defining an FSM lists all the actions (comma separated) that can be done in any order. For the FSM above for example this data part is:

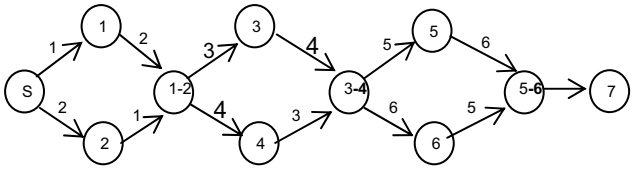

Figure 2. FSM for the 7 step ChemLab SG learning exercise.

4
1,2
3,4
5,6
7

The FSM module provides a scoring method for assessing the student's progress. The student can receive a non-zero score even if he only partially completes a set learning exercise in the game. The FSM module can assign a score to the student's work by the ratio of states achieved to total states required to be achieved to accomplish the learning exercise. With the assessment data stored in the ESG database, a separate program is required to allow the teacher to view all the students' results, compare and analyse them and this part is a straightforward coding problem. The teacher may well want to use these results to compare the rate of learning of his students through use of the ESG versus his standard teaching program.

IV. CONCLUSIONS

ESG development tools to address the Teacher's Dilemma are still not available. However this research shows what is necessary in these tools and our experimental prototype shows the viability of the concept. We can reduce and simplify the general ESG features so that the tool for creating an ESG of the simplified sort is less expensive and quicker to develop and still answers the Teacher's Dilemma. The ESG development tool can be built without the need for the teacher to write code or use a compiler by having the tool read in a game definition file [14] to set the teacher's ESG requirements such as the number of rooms, text labels for doors, locations of monsters etc and references to asset files such as wall textures, sound effects and monster meshes. The game definition file [14] is created in the correct format by a separate application that asks the teacher for all the required settings allowing for default values.

The FSM module described in this paper is an extension of the standard FPS code that is needed for a useful ESG. It could be built into the FPS Game Engine or the corresponding Game Engine for real ESGs. It provides a class whose objects are the FSMs corresponding to each learning exercise set by the teacher. The teacher must provide a text file that details the FSM for each learning exercise. This file is read in at the start of the game and the FSMs are then created. The FSMs also provide a more accurate and useful marking method than the FPS equivalent which is only either full marks for reaching the final state (killing a monster) or zero marks for not achieving the final state. The FSM in contrast provides a non-zero mark for partially completed learning exercises and therefore more accurately measures a student's learning progress.

Future research and development includes completion of the full ESG Engine and the creation of an ESG based on this engine using realistic teacher supplied learning exercises for a school class topic

ACKNOWLEDGMENT

This research was supported by La Trobe University grant and CONACyT Scholarship from the Government of Mexico.

REFERENCES

[1] W. Tinney, "Behind the Screens: The Technology of Casual Game Development". Casual Games Quarterly. The Technology Vol.1, Issue 1, 2005.

Available: http://www.igda.org/casual/quarterly/1_1/casual.php igda international game developers association.

[2] J. Rankin, S. Sampayo Vargas, "A Survey of Real-World Applications of Serious Games Technology". Proc. Simulation-maximising organisational benefits (SimTect2008), Melbourne, Australia. May 2008. pp. 305-311.

[3] K. White, "Casual Games get Serious". Casual Games Quarterly, Vol 2 Issue 3, 2007.

Available: http://www.igda.org/casual/quarterly/2_3/index.php?id=2 igda International game developers association.

[4] J. Rankin, S. Sampayo Vargas, "Survey of Development Tools for Low-End Serious Games". Proc. 5th International Conference on Information Technology and Applications. (ICITA 2008) Cairns, Australia, June 2008. pp. 462-466.

[5] M. Prensky. "Give Us 21st Centry Tools". Marc Prensky.com- Talk delivered at the Dept of Education's NCLB eLearning Summit. July 2004. Available: http://www.marcprensky.com/writing/Prensky%20-%2004-07-NCLB-post.ppt

[6] M. Prensky, "Engage Me or Enrage Me. What Today's Learners Demand". EDUCAUSE Vol.1, October 2005. Available: http://net.educause.edu/ir/library/pdf/erm0553.pdf

[7] B. Wünsche, B. Kot, A. Gits, J. Amor, J. Hosking, J. Grundy, "A Framework for Game Engine Based Visualisations". Available: http://www.cs.cornell.edu/~bjk45/publications/papers/WuenscheKotEtAl_GameEngines.pdf

[8] Young, V, "Programming a Multiplayer FPS in DirectX". USA: Charles River Media, Inc. 2005. pp. 389-478.

[9] N. Gebhardt. "Irrlicht Lightining Fast RealTime 3D Engine", 2003. Available: http://irrlicht.sourceforge.net/faq.html

[10] L, A. Annetta, "Serious Educational Games From Theory to Practice". USA: Sense Publishers. 2008. Ch.1. pp.5-7.

[11] M. DeLoura, "Game Programming Gems". Rockland, Massachusetts: Charles River Media. 2000. pp. 237-243.

[12] D. Michael, S. Chen, "Serious Games: Games That Educate, Train and Inform." USA, Boston: Thomson. 2006. Pp. 3-43.

[13] B. Bergeron. "Developing Serious Games". USA: Charles River Media. 2006. pp.227-286.

[14] J. Rankin, X. Ma, "Game Design Language." Proc. First International Conference on Information Technology & Applications (ICITA 2002), Bathurst, Australia, November 2002. pp. 1-3.