Language of Game Authoring

Cat Kutay[1] and Rui Prada[2]

[1] CSE, The University of New South Wales, Sydney
[2] Instituto Superior Técnico and INESC-ID, Lisboa, Portugal

Abstract

With the growth of transmedia production, narrative and role-play games are sometimes reused in the production of films (as in Machinima), and their content can have a similar format. In this paper we look at reversing this production process, and developing games from films. The first step is using film characters and props in simulations, next expanding to story branching and then to emergent narratives through the development of agent models and authoring tools. Various simulation editors exist for writing teaching material using a limited set of characters, animations and environments. This paper describes a project that expands the functionality of such tools by using existing game resources developed for filming within the Unity 3D Game Engine. We present the language based on Game Description Language (GDL) for reusing resources to develop the models and animations for characters in a game.

Keywords: *Serious Games; Game Authoring; Scenario Development.*

1. Introduction

With the increase of gaming for teaching, there is a need for easy editing tools to create these games, and also for a pool of animated characters and scenery to develop an immersive experience. Also there is a need for formalisation of the game creation process so as to provide some certainty that the end product will be supportive of the learning goals intended (Gunther et al 2006). There have been various descriptive models for creating learning games (AR authoring tools by Klopfer & Sheldon 2010; Game Object Model by Amory, 2007; and the LM-GM Model by Arnab et al, 2014) and some prescriptive formulae, such as work by Est et al. (2010) who recommend the use of authoring tools to make learning scenarios more flexible to the learners situation.

In particular the difficulty is that authoring tools are usually highly specialise in focus or too technical for general use. Narrative and role-play games have many features in common with films, and the resources that go into creating animated characters and scenery in 3D format can be used to create games using the same themes. In many cases these games will have an educational focus, such as in re-creating historical events, animals or other cultures and allowing the user to explore these environments.

This work is part of the project to develop *Machinima Storyteller (*Kuffner et al. 2013*)*, which is an editing environment plugin for *Unity 3D Game Engine.* This provides the tools to select an actor (with their personality) and set up their character, appearance, signature animations, and the poses that characterise their interactions with scenery props. Finally a scripting tool provides the sequence of actions they use to follow the script. The film is then generated with audio samples or text to speech, and recorded using the game renderer. The final editing (sound and after effects) is done with external tools.

The first step in creating a game environment can be the re-use of the film-scripted scenarios as in-game cutscenes. We are working with Machinima Storyteller as it was developed to provide tools that define the actor's unconscious thought and acting style, all aspects that will enhance their character's presence and interactions in the game. That is, the controller tools proposed by Kuffner et al. (2013) for the director control of actor's style provide the initial parameters for the character's agent (ie the required characterisations for the actor portraying that character) and their interaction with the player. This paper describes the language for game editing as a plugin extension to Machinima Storyteller, using an approach based on the development of General Game Language for developing game players.

This research is in the domain of Serious Games (SGs); we look at the creation of games within an existing 3D environment used for film rendering. This environment provides a learning experience about the setting for those characters, and uses that environment to explore the world of the film. Given the close relation between SGs and simulations (Gentile et al., 2014) the film scripts provide a suitable foundation for this work. In working with SGs, we need to focus on effective educational outcomes, rather than presenting the entire complexity of reality in the game world. In making an SG the main purpose is to model the actors' decision-making processes in that character role and to convey the consequences to the player within social environment (the game) providing challenges arising from the complex nature of these systems [18].

The paper takes an agent-modeling based approach to generating the social environment of the player-agent interaction, using the concept that agent models provide generative model of the social environment (Epstein 1999) which the player can explore. Other approaches, such as conceptual models (Yuseff et al. 2009) and widgets design tools for modelling learning processes (Marfisi-Schottman et al. 2010) have been developed. This work is based on the design tool concept but makes use of agents-modelling techniques to avoid the need for synchronous student-tutor sessions. In particular we aim to blend the learning more with the gaming aspects. The most relevant work is the Agent-based Simulation System (ABSS) developed in the PNVP case study by Gentile et al. (2014).

2. Outline of Game Definition Language

GDL was developer to describe games to automated players, whereas we are looking at ways to define game features to be generated by semi-automated game-authoring tools. The basis for GDL is a conceptualization of games in terms of entities, actions, propositions, and players. The list of descriptors reserved for the formal relations are (Genesereth 2014; Thielschler 2010; Kulick 2009):

Table 1. Keywords and their descriptors for GDL

Keyword	**Descriptor**
role<role>	Defines the roles or players and their names
init <proposition>	Describes the initial game state or proposition
true<proposition>	Checks whether facts are in the game proposition or not
does<role><action>	Describes the last actions done
next<state>	Describes the game state in the next turn, depending on preconditions
legal<role><action>	Defines legal moves at a specific situation, depending on preconditions
goal<role><value>	Rates specific situations with values between 0 (worst) and 100 (best)
terminal	Defines terminal states of the game
base <proposition>	Defines base propositions in the game
input <role><action>	Defines action for player role
sees <role><percept>	Percepts seen by player role in next move from information given
random <role><value><event>	Defines the role that plays randomly
visible <role><expression>	Defines access to game knowledge
include<package><module>	Defines inclusions of smaller modules

These can be adapted to form the language to define the game play rules and the character interactions in SGs. However there are still more detailed features required for the description of the animations of the characters (NPC's) and the narrative flow in more detail in more complex games.

3. Starting Point

The assumption in this work is that the game production starts after the film has been developed. So we are aiming to re-use the resources existing in the film-editing environment that has been created with Machinima Storyteller. The components of the game environment that exist from within this environment are described in subsections here. But first we would like to compare to the multi-agent based simulation system (MABS) model proposed by Gentile et al. (2014):

The abstraction process of their proposed MABS involves the following stages:

- Model the environment, including relations between game entities, for example interaction with props, etc.
- Identify the agents of the system, and their characteristic. Agents are capable of perception, communication, and action.
- Specify the knowledge and behaviours of each agent, as rules or parameters of the model. Agents can model either physical or logical elements.
- Design the data distribution between different agents, the system state is thus distributed and mainly controlled by separate agents.

We also add a stage for developing the scenarios for the game cutscenes.

3.1. Location and theme – Modelling environment

This is the setting of story, possibly split across different scenes, including landscape and inanimate objects. To design a game we need to have a theme, and the film and its characters provide this for the game. The game can extend the theme through exploring new scenarios, while the location provides the map in which the game is set. We will then specify the visibility of the map at different levels.

For the game sequence is defined by the players movement between locations rather than the film-script timeline, The player may be able to move between time locations in the film plot as their motion between static locations in the map of the film now creates their timeline. So if the film has different time located scenarios in the one place, the player may encounter these together, unless only one scenario is visible at that level.

3.2. Characterization and animations – Identify the agents

The basic characters with their scripted interactions with props and scenery exist for the game and can be extended by re-using these in different combinations. Consider the different player learning experience when viewing a character juggling some balls alone for practise compared to when they are doing this in a crowded market place and in the latter case interacting with the actions used by crowd members.

Some level of rule definition for characters' response to other characters in terms of action tendencies are set as constants (eg mannerisms and character traits) in the filmmaking process while others are variable set in the scripted film scenarios (eg anger and joy). These will provide the basic agent support for interactions with the player. However this will have to be extended, as now we want the characterisation to respond to the unknown sequence of inputs by the player.

3.3. Status rules – Agent knowledge and behaviour model

The characters of the story can be provided with further agent encoding of patterns of behaviour (compare Comme il Faut by McCoy et al 2010; FAtiMA by Dias & Paiva 2005 and PyschSim by Marsella et al. 2004) for managing in-game behaviour. This will include certain criteria (modelling parameters) to set up characters, that specify how the agent will act that character, and how the character will interact with environment.

The type of rules used will depend on the game theme. For instance cultural or historical games may model cultural dimensions. Work-based simulation themes may model different types of team members or management styles.

3.4. Variable and goal sharing - knowledge of system

As in all games, the character agents and/or players may have incomplete knowledge of the system. In SGs it is very important that the designer have an idea of what data the agents have and how the player is to gain this knowledge. The knowledge of the system, which is distributed between messaging agents, is the domain of learning, and can be in the form of models of behaviour (eg cultural dimensions).

The characters' personal knowledge includes their specific rules (including variables, personal goals and tendencies or reactions). This allows the emergent nature of the gameplay. A major component to the game authoring at this stage is to provide challenges that lead the student to the desired learning.

3.5. Scenarios

Scripted scenes with character actions and interactions with other characters and props are created for the film, and this storyline will provide the outline of the sequence of events that the player will encounter. These will exist as scripted cutscenes that can be played in the game as they are for the filmmaking. Est et al (2010) point out the complexity of graphical authoring tools, and proposed a flow chart style scenario-scripting tool at a higher level to the gameplay level. In Machinima Storyteller we use scripting tools similar to movie scripts for sequential scenarios and state charts for the agent interaction rules. However, as in the work by Est et al, these scripting tools rely on underlying lower level descriptions and code, which have to be written. Hence, we are working on developing the language to describe such components to ensure that the authoring system is flexible, allowing new events or simulations to be scripted and interaction with the user provided.

4. For Game Making

When extending these resources to game making, we need to develop a language to handle the types of editing we will need. For this exercise we consider only one level of play; roles that include player and other roles carried out by NPC's; and focus on simulation games.

We can consider the film resources provide the states of the game, such as the animations, prop interactions and emotional responses. The game designer needs to develop the interactions with the player, and the mapping from player actions to in-game actions, including standard avatar controls.

Most crucially, the plot and timeline need to be extended to a branching story. This can include some aspects of the film as simulations, the actors emotional response to situations reflected in their interactions with the user, so that they stay 'in character' and new features for the user to discover in their play.

However, there are aspects to games that do not occur in films, such as the idea of challenges and rewards. These will be the learning aspects of the game and will relate to the climax and resolution of the film. Alternatively they can evolve from the ‘behind the scenes’ aspects of the film, the story that was not told in the film but comes to exist through adding depth to the location and characters.

4.1. For Serious Games

In serious game making, there are factors that effect learning effectiveness, such as situating the learning; minimize cognitive load and facilitating the learning task; engaging the player constructively or experimentally; and reusability of components and the knowledge associated with these (Catelano et al, 2014)

The linking of games to films provides the situated learning environment, and also provides for high reusability of concepts, characters and actions. The cognitive load is often in the user interface and the ability to control the system, so by developing in a standard game engine, a significant part of the interface commands will be known to young players.

The main areas to focus on are: creating a non-predictable and non-repetitive environment from the known (film) resources, such as introducing random elements, to ensure the player does not become complacent in their prediction of events and miss salient points; creating real-time feedback and self-evaluation opportunities; and structuring the learning to enable the player to link new concepts.

4.2. Situating the learning activities

While the film provides the story situation, the designer will still need to consider what part or location of the film is best for various learning features. For instance do we want the students to learn about some issue in a crowded room or one-to-one with a mentoring character.

4.3. Cognitive load - scaffolding and feedback

The process of using films in games gives the designer the advantage of having familiar characters and scenery within which the learner navigates their knowledge acquisition. However, the aim is to introduce gameplay and storyline that changes as a result of the users actions. The use of agents in simulations provides the opportunity to model quite complex behaviour, as between real humans. However, Gentile et al. (2014) note the point that the complexity of the agents can be difficult for students to understand. Also complex agents are difficult to test to verify the situation is instructive not confusing.

Other supporting features, such as visual cues, are important tools to incorporate in the design so that these can be set up to support the learning at each stage. The use of player modeling to record and react to player progress is an important aspect.

4.4. Engagement – cognitive and experimentally

Games provide an environment for allowing users to think through plans and then implement them in a safe and repeatable environment. Hence we want the situation to be real (and complex) while knowing the desired learning features will be clearly presented.

4.5. Reusability of components including knowledge links

The creation of games from artifacts would involve the inclusion of knowledge from one game implementation to another. That is we are aiming to ensure knowledge that is used to enhance game components, is reusable across new games created using same repository.

4.6. Assessment

The final aspect of the modeling, specifically of the player, is to record their interactions and apparently learning achievements. As with all learning deign it is important that this be incorporated in the initial design, as the assessment should match the learning environment. For multi-agent based simulation (MABS) SGs the learning is reflective, the player may be asked questions, but usually assessment has to be completed out-of-game.

5. Game Generation Language

This section describes the game generation language, a language to specify the game design, which will be implemented by the gameplay code. This design editor sits on top of the gameplay scripts and programming tools required to implement the game in 3D (cf. Shai scenario editor in Est et al., 2010). The language describes the features of the game editor and will be used to develop automation for editing or the editing workflows. Each term will be broken down into components within the module implementing that part of the automation.

5.1. Relation to Action Language

The language we have developed uses the following reserved forms based on action languages for agent control (where an agent can be artificial or player):

- map <location> – The layout and features included in a physical location as a subdivision of the film-scripted scenarios and film locations.
- level <location><characters><scenarios> - The set of objects, including 3D characters, interaction rules and existing scenarios in one level.
- branching <rule> - The rules governing branches in the story tree, linking the film-generated scenarios.
- action <input> - The input to the game from a player action, including dialogue selections, which a game character may react to.
- interaction <rule> <character> - The rules governing characterisation, or interactions through appearance, dialogue and animation reactions, including facial expressions.

- state <character> - The initial state of the character's agent based on their film character.
- status <rule> <character> - The character's changing status or persona that creates changes in interactions.
- goal <agent><value> - The types of player challenges and the reward of the challenge; and the rules for modeling of agent goals and outcomes.
- skill <agent> - The combination of actions and responses that form the acquisition of a skill.
- precondition<role><action> was legal<role><action>
- effect <state> was next<state>
- Script <interaction> - The combination of interactions in a scripted scene
- game = U_i Map_i – The combination of level maps.

In the following section we give some examples to follow which show how these terms can be used.

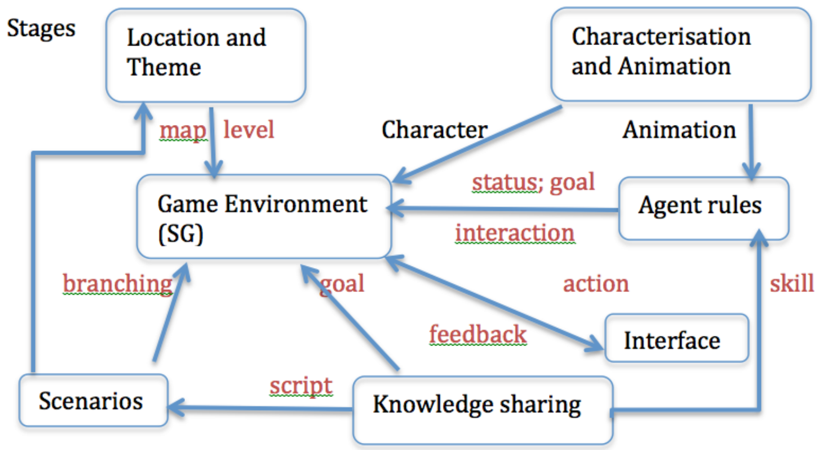

Figure 1 Showing Stages linked by language components

6. Examples

The example here is based on resources used to create cultural films about Australian Aboriginal people. The 3D models have been developed for animations and can provide the resources for cultural training. We show how the language supports this process.

6.1. Gaming Features

The language must support the generation of games that satisfy criteria for the user engagement in the game. The SGs need to provide more than just a series of cut scenes and independent agents. So while they are developed from the film reosurces, they further add to these with the specific gaols, interactions and status rules of the game. These features include:

- *Gameplay*

The gameplay is an overview of how the mechanics and environment combine to make the play enjoyable, challenging and informative. In this system the mechanics are the character animations guided by the agent models. The multi-agent system provides an interaction that produces the emergent narrative of the environment, a combination of the agent's characteristics (state and status) and agent modelling (goals) and the path chosen by the player (actions).
Also the player if provided with goals, which will include challenge, guidance and outcome. Another aspect of gameplay is the branching in the scenarios. It is possible to segment the cut scenes into a series of optional interactions, based on the player input.

- *Domain-specific features*

The map, levels, characters and their animations and the scenarios all come from the film resources and provide a thematic environment for learning. The creation of the game requires the development of goals and interaction rules that provide further representation fo the knowledge in the environment. For instance in the games in these examples we are looking at cultural modeling which requires the rules for social interaction be specified for the game.

- *Roles and Character Behaviours*

The roles for characters are created in the filming resources, but can be expanded in the game. These are the initial state of the characters that are then affected by the interactions, so that a new emotional and mental status is created within the character. The behaviour reflects the animation characterisation of the character, from the original filming resources and the outcome specified for the interaction rules.

6.2. Generic aspects

There are some features that are independent of the film content or generated game so we deal with these first. These are the aspects that deal with integrating the learning outcomes into the game and will be developed in future papers.

1. action <input> – this is the interface mapping from player controls to actions in the game, such as walking, talking to people, fighting. There will be a one to many correspondence with the members of the set of {interaction <rule>}.
2. goal <player><value> - This is one of the two main differences between games and films. As well as providing the player with tools to interact with the film characters and so become immersed in the story, games also provide challenges.

The main types of challenges (that fit within the film-game model rather than added as non-thematic tests) are solved by:

   a. Set of {does<player><action>} - Conflict within the story to be resolved, through a series of actions, including planning decisions.
   b. Set of {sees <player><percept>} - Problem solving, or the knowledge items to be extracted from the environment as part of the learning is achieved.

3. skill <player> = set of {action <input>, interaction <rule>, status <rule>}. To achieve challenges that involve gameplay the player needs to gain skills and these will be defined in terms of the interaction rules of the game the player has to initiate to get the desired response or access the desired information. While the challenge may be to achieve a certain goal, the goal will have to be achieved through a gameplay sequence, so this ensures the sequence exists in the game rules.
4. game as U Map$_i$ provides a sequence of levels, presumably linked by challenges that the player will proceed through. Different parts of the map can be made visible as the player achieves new skills or knowledge. This may not be in the order used in the film. In the following discussion of the formation of each Map, we do not deal with how to define the sequence of levels, however this is important in the learning model.

6.3. Traditional Story

The domain of interest is animated films being developed of Aboriginal tradition stories. These are about people moving around the country, interacting with the wildlife and with other people. In the stories people can turn into their totem and back into people. The story is about how to look after the totem and find the resources in the land, as well as how to treat other people

The story is first divided into different scenarios, or time/location divisions of the film material. Each scenario is complete in itself in traditional stories, although they are also linked by the main theme, so each scenario can form a level, or each location can form a level.

The player can go through the story in a different sequence to the original story. They may find that they are watching a group of people coming to a water hole to feed and the same group fighting and killing some of their members. These events are not consecutive in the traditional story. Hence the challenge will be to find out why the group then chose to fight. The death of the character will relate to a moral story or scenario. This can be taught to the player by letting them intervene in the fight and maybe prevent the death.

- map <location> is a division into locations with teleportation, or ‘travelling’ as in the traditional story, links these.
- level <location><characters><scenarios> is the set of stories enacted as scripted scenarios for the film, at one location in the map. Further scenarios can be scripted for the game in that location, to provide story branching.

- branching <rule> = set of {action <input>, interaction <rule>, status <rule>}. The branching rules are based on the result of player-character interactions
- interaction <rule> <character> - these rules will be in the form:
    a. goal <character><value>
    b. does <character><action> => next <state>

These rules will be dependant on the style of each character to be specified in terms of the characterisation attributes defined in Machinima Storyteller as role interpretation and unconscious behaviour, that provide social presence, acting quality and diversity.

- challenges – the story is full of moral aspects and survival techniques. These can be tested in the game as a series of interchanges with NPC or questions as part of a quest (using speech recognition system as in Mass Effect 3).

6.4. Characterisation and Status

The next phase is to develop resources related to social presence, acting quality and diversity. That is, to provide agent control plugins for non-verbal communication and actor's style. In the filming modules are called the Unconscious Module (UM) and Role Interpreter (RI) (Kuffner et al., 2013). Based on work on FAtiMA (Dias and Paiva, 2005) we are using two modules for the schematic and the coping process. The schematic action tendencies provide the base response of the character that is augmented by the problem solving and emotive reactions. Each module will consist of appraisal modules, using OCC parameters (Ortony et al, 1988) and values of Hofstede dimensions (Hofstede, 1991) to continually update the character affective modules to represent the characters state in the game world.

In the game version these components become the Characterisation Module (CM) and the Persona Module (PM). The CM with handle the interaction rules and the PM with deal with status values. The initial state of each characterisation rule will come from the UM parameters, and status updates from parameters from the RI. In this way these modules will incorporate the changes that occur between different film scenes.

Interactions define how the characters will react to each other, obstacles and the player. Once set up for the film version, this will form the basis of the characterisation within the game and provide the 'signature moves' that will encourage fans of one media to engage with the other. A single interaction rule is made of any of the elements:

- appearance ><prop><character- - character appearance plus any props including clothing
- dialogue <state><character> - dialogue to be made when a given state reached
- animations <state><character> - actions to be taken when given state reached including facial action
- rituals <state><character> - rules to be carried out when in a given state
- goals <character> - goals of the actions

Status initial values will be specified in the RI and these then can be varied for the game, according to the persona, to allow characters to 'grow'. These are cultural and emotional (group and individual) in origin and include

- values <character>
- emotions <character>

7. Further Research

The next stage is to break each language word or action into its separate components for designing the implementing modules. The concept is that each module will be a independent Unity plugin to provide the tools for designing that component of the game or learning system. For instance the map module would include tools to select locations to form a coherent part of the game. The film material within this map would then be available as selectable objects for designing the game levels. Also the map module would provide a visual display of the map for the player.

We consider here two areas where the development of learning material in such an environment can be improved through expanding the language of development.

7.1. Knowledge Representation

In any rule-bound, game-based environment of this kind, knowledge must be reconceived in a play-based digital form. (Castell and Jenson, 2006)

Understanding more about how the constraints and affordances of digital games work against traditional text-driven forms of knowledge representation, and coming seriously to terms with a 'ludic epistemology', a theory of play-based learning and knowledge, is fundamental for transforming and refining emerging conceptions and practices of education. (p. 38)

Game-based learning does not always fit well in the tradition of text-based forms of knowledge representation, however it provides an environment for a 'build it to understand it' approach to learning. Hence, this is an area to implement a visualisation of knowledge for learning in a new format.

The first step in this is developing a tool to author the interaction <rule> and status <rule> sequences. Agent models that rely on XML files containing the NPC's main goal and sub-interactions enact these. Also each new rule has to be tested for validity, or the may never be able to achieve a goal.

The process of developing these rules requires thorough understanding of the world in the game and how the player will experience it. However there are some limits put on this through the character 'acting' style and behavior as mentioned above, so the parameters affecting the interaction or status responses are well defined.

7.2. Knowledge searching

Serious Games provide an environment for students to search for knowledge, or generate knowledge through experiential learning. There are many ways the human mind searches for new knowledge and links this to prior knowledge and we need to develop new ways of immersing knowledge in the game environment.

Also there are the search needs of game authors, such as, using procedural content generation scripts to generate, or call up, suitable characters for a game (Mehm et al. 2012)

8. Conclusion

This work is an extension to the work done at INESC-ID on the development of Machinima Storyteller, a film-authoring tool developed as a plugin to Unity. The Australian project was set up to extend the transmedia capabilities to game authoring using the same resources. The game authoring process involves the creation of tools that can utilize and combine existing resources (cf. Est et al, 2010). Hence we designed here the higher-level interaction of components and developed a common design language for such tools.

The language is the design for the game-making plugin to Machinima Storyteller and provides the framework for integrating existing scenario scripting tools; multi-agent editing tools similar to Unity animator; map generation tools, story branching definitions and challenge construction tools.

9. Acknowledgement

This work was partially supported by national funds through FCT - Fundação para a Ciência e a Tecnologia, under project PEst-OE/EEI/LA0021/2013.

10. References

[1] Amory, A. (2007). Game object model version II: a theoretical framework for educational game development. Educational Technology Research and Development, 55, 51–77.

[2] Arnab S., Lim T., Carvalho M. B., Bellotti F., de Freitas S., Louchart S., Suttie N., Berta R. and De Gloria A. (2014) Mapping Learning and Game Mechanics for Serious Games Analysis, British Journal of Educational Technology,

[3] Catalano, C. E, Luccini, A. M, Mortara M. (2014) Best Practices for an Effective Design and Evaluation of Serious Games, International Journal of Serious Games, 1(1).

[4] Castell, S and Jenson J. (2006) Ludic Epistemology? Knowledge-representation in game-based learning. Proceedings of the ninth IASTED International Conference Computers and Advanced Technology in Education, Lima Peru

[5] De Gloria, A, Bellotti, F, Berta, R, Lavagnino E. (2014) Serious Games for education and training, in International Journal of Serious Games, 1 (1).

[6] Dias, J and Paiva, A., 2005. Feeling and Reasoning: a Computational Model. In Proceedings of the 12th Portuguese Conference on Artificial Intelligence, EPIA Springer. pp 127-140.

[7] Epstein, J. M. (2006). Generative Social Science: Studies in Agent-Based Computational Modelling, Princeton University Press, 2006

[8] Est, C. van, Poelman, R., Bidarra, R. (2010) High-Level Scenario Editing for Serious Games. In: Richard, P., Braz, J. (eds.) GRAPP 2011 - Proceedings of the International Conference on Computer Graphics Theory and Applications, pp. 339–346. SciTePress.
[9] Genesereth 2014 Game Definition Language. http://games.stanford.edu/index.php/intro-to-gdl.
[10] Gentile, M, La Guardia D, Dal Grande1, V, Ottaviano, S and Allegra M. (2014) An Agent Based Approach to designing Serious Game: the PNPV case study. In the International Journal of Serous Games, 1 (2), pp. 23-33.
[11] Gunter, G. A, Robert F. K, and Erik, H. V. (2006) A case for a formal design paradigm for serious games. The Journal of the International Digital Media and Arts Association 3.1 pp. 93-105.
[12] Hofstede G., 1991. Cultures and Organisations. McGraw-Hill, London
[13] Klopfer E, and Sheldon J. (2010) Augmenting your own reality: student authoring of science-based augmented reality games. New Directions for Youth Development. Winter, 128 pp. 85-94. doi: 10.1002/yd.378.
[14] Kuffner, R, Di Tullio E, and Prada R. (2013) Mappets: An Interactive Plugin for Transmedia Machinima on Unity3D", in Proceedings of the International Conference on Entertainment Computing--ICEC 2013, pp. 69--74, March 25-29, 2013.
[15] Kulick, J (2009) World Description Language - A logical Language for agent-based Systems and Games (Bachelor thesis) http://page.mi.fu-berlin.de/block/abschlussarbeiten/Bachelorarbeit-Kulick.pdf.
[16] Marsella, S. C., Pynadath, D. V., and Read, S. J. (2004) PsychSim: Agent-based modeling of social interactions and influence. In Proceedings of the International Conference on Cognitive Modelling, pp. 243-248.
[17] McCoy, J, Treanor, M, Samuel, B, Tearse, B, Mateas, M, Wardrip-Fruin, N, (2010) Authoring Game-based Interactive Narrative using Social Games and Comme il Faut, Proceedings of the 4th International Conference & Festival of the Electronic Literature Organization: Archive & Innovate (ELO 2010).
[18] Mehm, F, Reuter, C, Göbel, S and Steinmetz S (2012) Future Trends in Game Authoring Tools. Entertainment Computing - ICEC 2012 Lecture Notes in Computer Science Volume 7522, 2012, pp 536-541
[19] Ortony, A., Clore, G., and Collins, A., 1988. The Cognitive Structure of Emotions. New York, NY: Cambridge University Press.
[20] Thielschler 2010 A General Game Description Language for Incomplete Information Games. AAAI
[21] Theilscher, M (2011) General Game Playing in AI Research and Education. KI 2011: Advances in Artificial Intelligence Lecture Notes in Computer Science Volume 7006, 2011, pp 26-37.
[22] Yusoff, A, Crowder, R, Gilbert, L, Wills, G 2009. A Conceptual Framework for Serious Games. In *Proceedings of the 2009 Ninth IEEE International Conference on Advanced Learning Technologies* (ICALT '09). IEEE Computer Society, Washington, DC, USA, 21-23. DOI=10.1109/ICALT.2009.19 http://dx.doi.org/10.1109/ICALT.2009.19
[23] Marfisi-Schottman, I, George, S, Frank, T.-B, 2010. Tools and Methods for Efficiently Designing Serious Games. In Proceedings of ECGBL 2010 The 4th European Conference on Games Based Learning. p. 226-234. Danish School of Education Aarhus University, Copenhagen, Denmark.
[24]