MOLDE – a Methodology for Serious Games Measure-Oriented Level DEsign

Emanuel Henrique Farias Helder Cognaco de Oliveira Marcelo da Silva Hounsell
Gabriel Mesquita Rossito

LAboratory for Research on Visual Applications (LARVA),
Computer Science Department (DCC), Santa Catarina State University (UDESC)
Joinville, Santa Catarina, Brazil

Abstract

Serious Games (SGs) offer an interesting and motivating tool for education, rehabilitation, and many other areas. However, there seems to be no consensus on a methodology for designing such games, not even for parts of this process, such as the Level Design. Level Design is a critical analysis on the game construction to guarantee its flow and is highly dependent on the specialists that will be using the SG as an accessory tool. Some SGs where game mechanics relate directly to people´s (motor, cognitive, etc.) abilities (or disabilities), designing meaningful and adaptable levels reflect on the game's feasibility as a serious tool. This paper presents a Methodology, called MOLDE – a Measure-Oriented Level DEsign - that has been used to develop Educational and Health SGs aimed to a variety of population. The methodology guides a designer step-by-step to specify phases/missions and levels. The architecture allows the SG to be tailored to a wide variety of users. The paper also present experiences in using the MOLDE for three games to audiences as wide as Down´s syndrome children, frail elderly and stroke patients.

Keywords: Serious Games, Methodology, Level Design, Game Architecture.

Authors' contact:
emanuelfarias@outlook.com
marcelo.hounsell@udesc.br
{heldercdo,gabrielmesquit}@gmail.com

1. Introduction

Games are considered to be the most appropriate media to the new educational paradigm the generation of digital native [Prensky, 2001; Deshpande and Huang 2009] but its advantages goes beyond youngsters. In a Serious Game (SG) the main objective of the game is not only entertainment but something more serious, such as physiotherapy, education, skill acquisition, behavior change, cognitive training, etc. SGs emphasize different aspects such as sociability, physical activity, development of attention, memory and logical reasoning (among others) allowing a playful mechanism for promoting quality of life. Games as a whole have been considered of great acceptability by children because it is not perceived as a "lesson" carrier (Noble et al., 2000)

Health Games are one type of SGs that focus on helping people who have suffered from physical issues [Bruckheimer, Hounsell and Soares 2011]. Besides the serious part, these games are, of course, fun. The entertainment is very important, because it makes the player feel motivated and engaged with the game, thus, helps the player heal faster. After all, such game-based approach is a "simple and effective way of transmitting positive health messages" (Noble et al., 2000).

SGs do require to be fun, after all it is a game, but "developing and evaluating games for specific purposes whether learning of behavior change is a very complicated enterprise requiring an understanding of the design of games, knowledge of the relevant content or subject area, knowledge of motivation, pedagogy and behavior change […]" [Boyle, Connolly and Hainey, 2011].

It must be said thus that the clients of SGs are, at least, two-folded: first, are the specialists that summed the game to be used as a tool for their activities and expect it to comply with certain specific (and serious) features, and; second, there are the end-users, the players that were offered an alternative approach to their otherwise boring activity and expect it to be more fun and motivating. Level design is the part of a game design methodology that bridges these two expectations where the performance aimed by the former should be felt as pleasant to the later.

But there seems to be as many game design methodologies as there are designers [Prensky 2001] and, this is still the case, as pointed out by Boyle, Connolly and Hainey (2011): "no agreed organizational framework for serious games and little structured guidance about how to develop a serious game in a particular domain". The lack of methodologies to guide the development is a pitfall that holds back further developments in the area as well as new applications.

This paper condenses the experiences of LARVA (Laboratory for Research on Visual Applications) on developing a number of Educational and Health Serious Games as a roadmap methodology, called

MOLDE – a Measure-Oriented Level DEsign - to help guide others on similar paths. No creativity of aesthetic issues will be discussed but technical strategies on how to go about linking specialists' expectations on the game to users´ abilities and enjoyability.

The paper will comment on similar research in Section 2, then present MOLDE as a level design methodology, in Section 3. Afterwards, chapter 4 will present how MOLDE has been applied on the development of three SG as well as their variations. In Section 5 there is a discussion and a conclusion is presented in Section 6.

2. Related Work

In software engineering, development methodologies are simplified models of software development processes [Sommerville 2011]. They describe activities and their sequence without necessarily going into details, such as roles and best practices, for each activity. Therefore, development methodologies can support the whole software life-cycle, from the early requirements analysis to the proper implementation and maintenance of the resulting product.

Recent methodologies for SG development have been mostly based on well established traditional software engineering development models, namely iterative and sequential models. These methodologies can be roughly divided into: (i) initial analysis, (ii) development and (iii) evaluation, with a few tweaks from author to author.

As an example of the sequential model approach, Marcos and Zagallo [2011] proposed a methodology heavily based on digital art creative design process, therefore bringing more focus on aesthetic musing instead of concept and narrative design.

By proposing the Eight Phase Methodology (EPM), Lazarou [2011] managed to incorporate a sequential design process with the principles of Activity Theory and Human-Computer Interaction. EPM accommodates the participation of domain experts and end users while treating activities as "a systemic and collective unit driven by an object" [Lazarou 2011].

Some authors focus their work on methodologies and frameworks for more specific aspects of serious games, like Ibrahim and Jaafar [2009], who proposed the Adaptive Digital Game-Based Learning Framework (ADGBL), an iterative methodology that focuses on the design and pedagogical needs of a game, based on ISO 9241 concept of usability.

On a more practical level, Alamri et al. [2013] defined a framework for cloud-based pervasive SGs. Through their architecture, domain experts can interact remotely with end users while the actual game being executed and maintained in a cloud.

These methodologies and frameworks often focus too much on pedagogical aspects, narrative and conceptual design, which are all an important concerns, but they fall short on the aspect of Level Design, hence why MOLDE is of great benefit to SGs in general.

3. MOLDE – A Measure-Oriented Level DEsign

MOLDE is an iterative methodology (depicted in Figure 1), where the main purpose is to translate specialists' expected functionalities into game variables and then, using these variables to control the game difficulty levels either as significant major steps, called here as Phases, or interactive variables that keep the flow of the game, called here as Levels. Architecture is also proposed that allows the game to adapt to each player over the series of sessions of use as well as to a completely different population that do not share the same parameters.

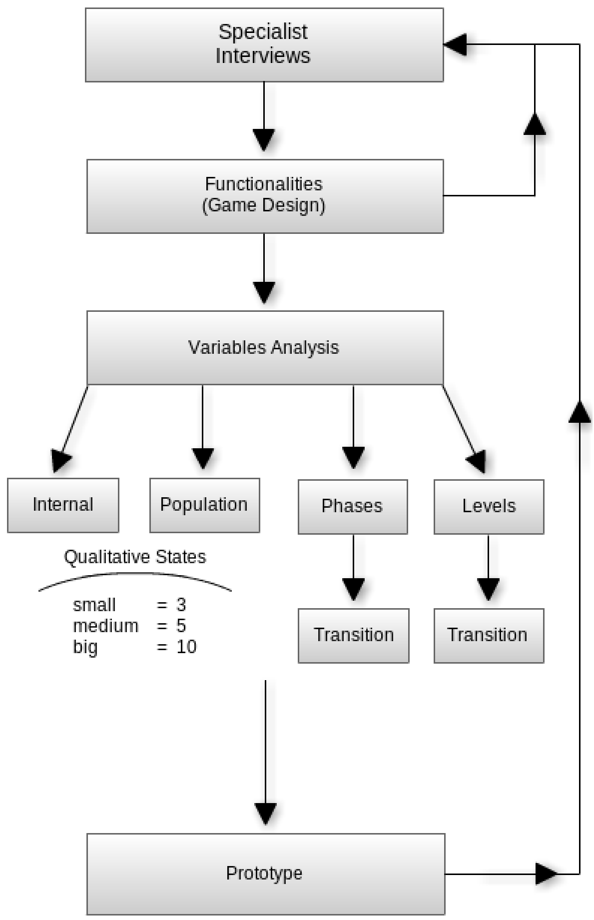

Figure 1: A Diagram representing MOLDE

As the name suggests MOLDE was conceived for SG that are able to capture motor-related measure data from the players and use this data to guide the game play regardless if the data is gathered using a RGB-D camera or OpenCV image-based functionalities.

3.1 First Step: Functionalities

Quite often a field specialist (teacher, physiotherapist, geriatrics doctor, and so on) comes to a game designer and requests a game that helps their patients or students achieve something. They usually come either with a lot of over the top ideas about the game looks or, without the faintest clue whatsoever on how the game would be. Therefore, the first thing a game designer should do is to perform a series of unstructured interviews to gather from the specialist what s/he is really trying to achieve with the game (for instance, a posture, a movement, a decision, a behavior change, and so on).

The game designer should let the specialist expose all their ideas, expectations and understandings about what the game can do for them. Afterwards, the game designer should gradually educate the field specialist about the opportunities and limitations that the game technology can bring to the process.

A list of required and desirable functionalities should be agreed upon after clarifying ideas over the game mechanics and interaction and aesthetic decisions.

With the list of functionalities the game designer can concentrate on analyzing game engines, input and output devices to see if the hardware is up to the required functionalities. These are the major functionalities that will make the job of the specialist easier and the reason the specialist would buy this game in the future. Along the process, the game designer should come up with ideas for the look and feel as well as gameplay.

3.2 Second Step: Variables Analysis

Once the functionalities are settled there is a list of requirements to meet. The game designer should analyze game's variables and separate them in on of the following categories:

- Internal;
- Population;
- Phase/Session;
- Level.

Identifying the meaningful variables that relate to users capabilities also help construct a scoring system that correlate closer to the user's overall performance.

Internal Variables

Internal variables are used just for internal purposes of conducting the game coding. Sometimes these variables are derived from others (more basic), which helps to control the game execution and has no relation to any specific functionality of the game. For instance, if a certain number of objects are to be drawn in the display, a spacing variable (derived internal) among them must be calculated considering also the display resolution (basic internal variable).

Internal variables account for any useful purposes that should not be bothered by the specialist neither the player. For instance, variables to convert radians to degrees, producing seeds for a randomizer function and, so on.

Phase/Session Variables

These variables represent the stages of the game whereas the user experiences noticeable differences regarding his/her targeted abilities. They can account for missions where the player should feel (and/or see) significant changes in the game difficulty. These changes should happen between successive sessions of playing the game which most often occur on successive days of use.

These variables should reflect the most important targeted abilities of the SG and represent patient/student major stages of development.

Level Variables

These variables are the ones that maintain the game's interactivity and entertainment acceptable for each person in a respective target group. Therefore, Level variables are responsible to keep the flow of the game at acceptable levels of difficulty. These can be modified during the same playing session where the user is in a specific phase/mission. These variables change more frequently than the phase variables. Level variables change the game mechanics subtly to adjust to patient/player capabilities.

Population's Variables

These variables determine specific values any game controls. Example, a variable X could have been associated to values such as MAX, MED, MIN which describes how the game progresses. However, from the implementation point of view, MAX needs to be set to the value 15, wherever MED could be set to 5 and MIN to 3. These actual values depend on the specific population and can vary accordingly. For instance, a Level Variable VEL (which stands for velocity) could be set to MIN but this value could be 12 if the game is to be used with Down´s Syndrome children or could be set to 20 if it is to be used for juveniles, or even set to 8 if it is intended to be used by people who suffered a stroke.

As one can see, the minimum velocity is the entry level for using a game but that could mean very different values depending on the targeted population. Even the number of grades for each variable (its granularity) is also defined as a modifiable parameter. For instance, one can find Small, Medium and Large enough for velocity but Tiny, Small, Medium, Large and Huge could also be used to quantify game elements. It should be added that once the number of grades are set, the values for each grade must be defined as well.

3.1.3 Third Step: Transitions

Once the game designer has established how the game would function in terms of phases and levels to a specific population, s/he should define how the game difficulty actually changes. This is done by defining Phases and Levels Transition Rules. Transition rules should define how and when the game should behave in a different way.

Transition rules basically assess player's performance and adjust Phase and Level accordingly. To do this, there is the need to define:

- The periodicity the game will analyze players performance;
- The thresholds that are applicable to decide on the transitions, and;
- The rules that compose successive phases and levels.

For instance, it is reasonable to expect that if a patient played very well one day (and topped the level on a particular phase), the game should start off the next day at the following phase but at the lower level. The game designer is responsible to set what is the level that corresponds to "very well".

In addition, it is not reasonable to assess player's performance after every single action because this could vary a bit due to distractions, tiredness, and other factors. Therefore, obtaining the average performance for the last couple of plays could be better.

Thresholds are limits to change upwards, downwards or keeping the player at the present phase/level. Transition Thresholds are variables that work different for different population once the flow of the game is set at different levels.

One reasonable way to go about the transitions of phases and levels is to consider changing one difficulty level ahead to "one" of the variables at a time. Therefore, guaranteeing that the game will not become over the top suddenly.

For instance: take Table 1 and imagine that each level (1 to 4 in the Table) depends on three variables and each variable has three grade states.

Table 1: Level transition changes only one variable at a time.

Level	Object Size	Amount of Objects	Speed
1	Large	Small	Low
2	Medium	Small	Low
3	Medium	Medium	Low
4	Medium	Medium	Mid

The first level should be the easiest to play. Notice that depending on the variable, this could be set to grades either Large or Small at the beginning of the level. So, the game designer should be careful to assign it. It seems reasonable that each time the game increases the level; only one variable should change its value. Thus, the player will feel that the game is getting harder in a smooth way.

Notice that if a level depends on a huge number of variables this approach will result in a big table. Generalizing, if a level depends of a number N of variables and each variable a number of M qualitative states, then the table will have:

$$N*(M-1)+1 \quad \text{rows/levels} \qquad \text{(eq. 1)}$$

There is no problem on changing the level of difficulty of two variables (Large to Medium and, Low to Medium) at once but the specialists should be consulted upon because this is not a decision solely from the gameplay point of view. The final number of phases and levels should have a say from both game designer and specialist.

These tables are divided in levels and phases. They can be stored in two external files, one for levels and another one for phases. Storing these tables out of the game code facilitates the changing process since there is no need to recompile the game after every modification. These tables store only qualitative variables, for example, speed: low. This "low" attribute needs to turn into a number. To define this quantitative variable it's recommended to create one external file for each target. For example, if the game will reach more than one population, each file will contain the respective values for that specific group.

3.1.4. Architecture Prototype

For the final product, it is important to consider storing the main variables outside of the game in external files in a coherent architecture. This approach will help the developers and the specialists change the value of a specific variable without recompiling the game.

An important feature of SGs is the possibility of reporting valuable information regarding players' performances to specialists. One way to show information to the specialist is to store the results from gaming sessions from each player. A good way to save this information is to store it in Comma Separated Value (CSV) files because they are easy to organize, to parse and to understand. The game will produce a big amount of data, thus, it's important to separate phase reports from level reports or even more specific reports.

MOLDE is focused on the development of SG that is used by specialists to help their work. They benefit from the games´ reports for later diagnosis. The prototype is an important step for confirming how field specialists will benefit from the gameplay and reports otherwise they won't use it at all.

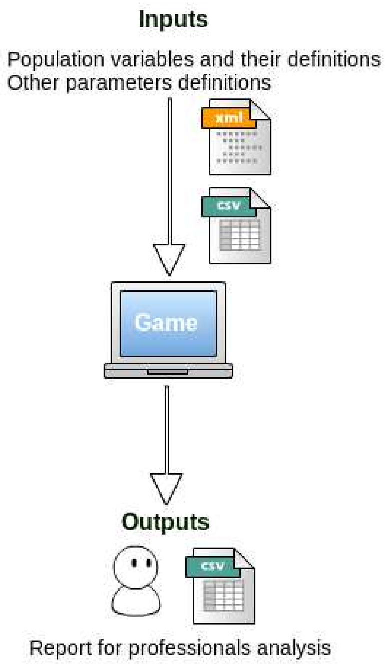

Figure 2: Relationship between inputs, outputs and the game.

4. Application

LARVA has been molding MOLDE to develop SGs with social impact. Each game has been implemented using different technologies (from Panda3D to Unity and OpenCV) and using a variety of input devices (from simple webcams to Kinect) but all games benefited from the MOLDE.

The Health SGs developed using MOLDE were:

- D2R2D [Bruckheimer, Hounsell and Soares 2011];
- D2R3D [Bruckheimer, Hounsell and Soares 2012];
- MoviLetrando [Farias et al. 2013];
- SIRTET [Rossito, 2013].

These SGs are at various stages of maturity. Some have been tried out with target populations and proved their effectiveness [Bruckheimer et al. 2011][Bruckheimer et al. 2012]. One is undergoing user experiments at the moment [Soares, 2014] and other is expecting approval from the Ethics Board to begin user trials (Farias et al. 2013). Also, they all implemented MOLDE a little bit different because of their specifics – which will be mentioned when appropriate.

5.1 Dance2Rehab

Dance2Rehab [Bruckheimer, Hounsell and Soares 2011], also called D2R, is a SG with the goal of stimulating the motor and cognitive functions of stroke patients. It is a software tool that has been incorporated in clinics daily use at NUPEN (Núcleo de Pesquisas em Neuroreabilitação) in Joinville, Brazil.

The game was developed with the assistance of physiotherapists. MOLDE helped to create an easy way for these specialists to change some behaviors of the game by simply changing values in a Comma Separated Value (CSV) file that holds parameters for the target population. CSV files could easily be opened by spreadsheets and are simple to be edited.

D2R uses a webcam to capture the image of the player. The player makes movements with his arms and hands to interact with the game by "touching" virtual objects (which are rain drops) that appear in a specific region on the screen. Figure 3 shows a stroke patient playing D2R where scores and time can be seen on the screen (see the cloud shaped white area at the top). Also, a grid was set visible to show all regions of interest that are dealt with in the game.

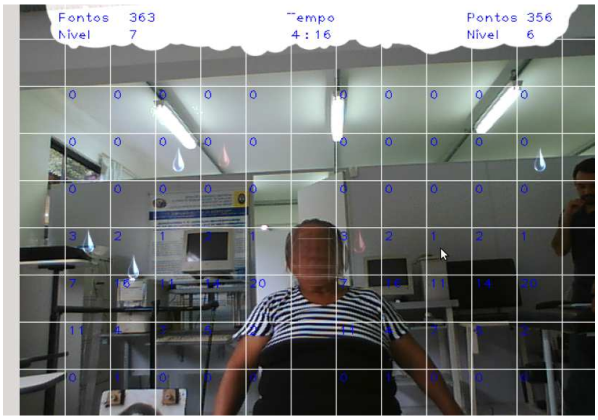

Figure 3: Dance2Rehab in action

These regions of interest can be easily weighted in the CSV file by health specialists. Table 2 shows how one side of the screen is divided and weighted when players touch the virtual object in that square.

Table 2 is the weighting distribution for the most demanding phase of the game where players are required to reach as high as possible. Two other CSV configuration files set the intermediate and easier distribution of spots to be used according to patients' limitations. Besides, different levels can be assigned to different sides as stroke patients are mostly impaired into one of her/his side.

Another result of using MOLDE to design D2R was the goal of producing reports for the health specialists.

Table 2: Points received by touching the correct virtual object in each region.

10	10	10	0	0
8	9	9	10	0
7	8	8	9	10
0	5	5	6	7
0	1	3	4	4
0	0	1	3	4
0	0	0	3	0

At the end of each session, a CSV file is generated containing how many objects the player "touched" in each specific region. Table 3 shows the number of times the patient touched the virtual object in each region of interest. The specialists can take advantage of these numbers to understand how far patients could reach.

Table 3: Numbers of object touched at each region

0	0	0	0	0
0	0	0	2	0
0	4	5	6	0
0	5	7	3	0
0	2	1	0	0
0	0	0	0	0
0	0	0	0	0

D2R has a configuration menu where at each session the specialist could set the game difficulty. Regardless of the values chosen in this menu, levels are automatically changed in the game depending on how the player is doing.

- If the player made a lot of mistakes, the game will decrease the level;
- If the player didn't make a lot of mistakes, the game will increase the level;
- Else, player will stay in the same level.

D2R don't use phases, only levels. Table 4 shows all the related variables and their meanings.

Range sets the range of space where weighted regions of interest are valued; Speed indicates how fast the rain drops fall and; side indicates the initial side to start the game (usually it should start with the impaired one).

Table 4: D2R's main variables

Acronym	Meaning	Type
LRG	Left Range	Level
RRG	Right Range	Level
LSP	Left Speed	Level
RSP	Right Speed	Level
SD	Side	Level

The player interacts with virtual objects at one side at a time. After a certain stage, the player will play with both arms. As can be seen in the Table 5 which shows how variables configuration evolves as the levels grow in number.

Table 5: D2R's level transition

LVL	LRG	LSP	RRG	RSP	SD
1	Bottom	Very slow	Bottom	Slow	Left
2	Bottom	Slow	Bottom	Slow	Left
3	Middle	Normal	Middle	Normal	Right
4	Middle	Normal	Middle	Normal	Right
5	Middle	Fast	Middle	Fast	Both
6	Top	Fast	Top	Fast	Both
7	Top	Very_fast	Top	Very_fast	Both
8	Top	Insane	Top	Insane	Both

5.2 MoviLetrando

MoviLetrando is a SG focused on the literacy of children with Down syndrome [Farias et al. 2013]. Figure 4 depicts the interface of MoviLetrando where a reference element (an alphanumerical character) is shown at the top of the screen (such as the number 3 in Figure 4).

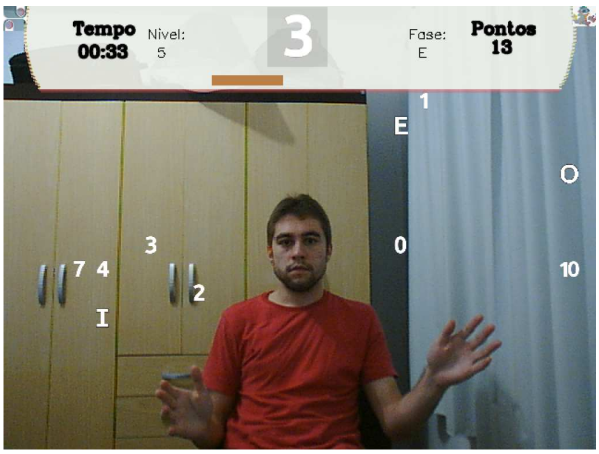

Figure 4: A screenshot of MoviLetrando.

After a while, many elements appear in the screen where the player can see her/himself into. The time starts to click and the child/player has to identify where the target reference is in her/his part of the screen and then plan a movement to reach it in time. The faster the touch occurs, the greater the number of points earned.

For MoviLetrando, sub-sets of the alphabet were divided as follows: at the early stages children play with vowels only (5 elements) then evolves to numbers (10 elements) and then consonants (17 elements). Advanced stages mix these sets into sets of greater number of elements (say 15, 22, 27 and 32).

Sets can be reconfigured by altering a CSV file. For instance, one can define a set of the first 15 letters of the alphabet. This file only defines the elements (letters and numbers) each set contains. In another CSV file the image file that represents each element is established. In this way, the game can be modified for different font types and group of elements. For instance, the same set can be chosen to be shown in black and white capital letters or, alternatively, in a rounded colored sort of letters. The pedagogical assistant is the one who assesses the child needs and selects which set and type should be used in the gaming session.

MoviLetrando implements both phase and level variables. Each phase has several levels. Table 6 lists the main variables, their meaning and type. It can be seen that some variables controls the Phases division and others the Levels.

Table 6: MoviLetrando's main variables

Acronym	Meaning	Type
WKS	Working Set (numbers, vowels, ...)	Phase
STI	Stimulus	Phase
NCS	Number of Concurrent Symbols	Phase
SZS	Size of Symbols	Level
ETS	Exposure Time of the Symbol	Level
SES	Side Exposure of the Symbol	Level
RSZ	Reference Symbol Size	Level

MoviLetrando also uses sounds (name or phoneme) as stimuli together with the image of the element. Therefore there is a Phase with images only, with images and sounds and, with sounds only (no reference image is shown). This led to the use of another two CSV files. One for the set of available sounds (letter A can be associated with the sound of the letter A being spoken by a male voice, female voice, with is phoneme spoken by his teacher or his own voice) and the other to assign which sound file will be used for each element.

During each phase, three variables change (as shown in Table 7): WKS, NCS and STI. For instance Phase A stands for vowels (WKS=1) being as reference elements, a small number of alternative elements in the scene (NCS=S) and both images and sounds being used (STI=2). In addition, Phase F stands for any number or letter being used as reference element shown using only sound stimuli (no image as a helping reference) and a large amount of alternatives presented in the scene.

These Phase variables represent significant evolution on the learning curve of alphabetization. Thus, for the player to advance to the next phase she has to perform well in the previous session.

Table 7: ML's phases transition.

Phase	WKS	NCS	STI
A	1	Small	2
B	2	Small	2
C	3	Medium	2
D	4	Medium	1
E	5	Large	1
F	6	Large	0

Four variables change along the way which generates seven possible levels, as presented in Table 8. For instance, Level 2 stands for a medium size of the images exposed at a long period of time at the right side of the screen using a large sized reference images (if any). Level 6 stands for small size of the images, exposed for a small period of time on both sides using a small sized reference image (if any).

Table 8: Movileletrando´s levels transition.

Level	SZS	ETS	SES	RSZ
1	large	large	left	large
2	medium	large	right	large
3	medium	medium	left	large
4	medium	medium	right	medium
5	small	medium	both	medium
6	small	small	both	medium
7	small	small	both	small

MOLDE could be adjusted for the specifics of this SG. For instance, MoviLetrando required extra files to accommodate the database sets related to images and sounds. Nevertheless the whole principle of dividing the interactions based on measurable and meaningful variables remained the same.

5.3 SIRTET

SIRTET [Rossito 2013] was inspired by Tetris. Its visuals and gameplay were reversed (as its name). The game places the player in a tunnel and objects come from the bottom of the tunnel and goes toward the player. Some objects needs to be touched (the targets), others need to be avoided (the obstacles). This game requires a Kinect that captures the image of the person and s/he must make moves with her/his body to interact with the game.

Figure 5 depicts SIRTET-K3D interface where a player can be seen touching a target object in the middle where his arm becomes green and some points are assigned to it.

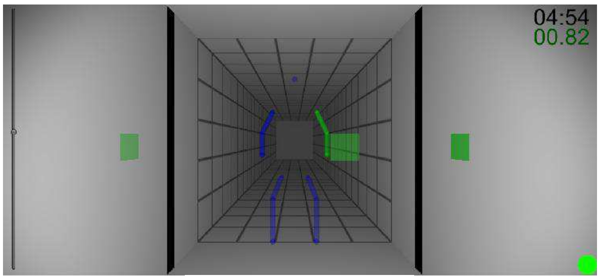

Figure 5: SIRTET-K3D

SIRTET is another example of using MOLDE. However, some adjustments needed to be performed regarding the variables for Phases and Levels.

Phases were set manually as a loop (sequence) of “challenges” that are composed of targets (green objects to be touched) and obstacles (red objects to be dodged). These cannot be created randomly because one should respect the capabilities of the target audience and, a specific sequence has to be incremental in terms of physical demand. To compose the challenges, a template is used that defines positioning for targets and obstacles.

Figure 6 shows the template that indicates to the game to generate an obstacle at an specific position (at waist level close to the person) within the hand reach (see vertical dotted lines) and at lower altitude (see horizontal dotted lines) that will require the player to deviate a bit.

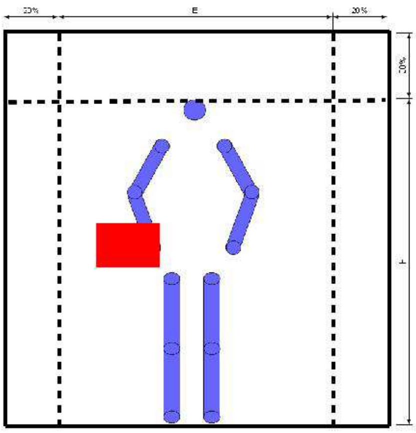

Figure 6: Generating an obstacle at a specific position.

Therefore, Phases are set as A=“Easy Targets”; B= “Easy Obstacles”; C = A + B; D = A + “Difficult Targets”, and; E = B + “Difficult Obstacles” and; Finally F = D + E. Physiotherapists can control for each Phase: the number of challenges, the composition of each challenge (targets only, obstacles only or a combination of it), the size of each object, and its type (target or obstacle).

For SIRTET, the most important programming variables were used for controlling level mechanics. These variables are presented in Table 9. Level mechanics include the SPO, regardless if they are targets or obstacles objects, IBC and PST (the bigger the persistence, the longer the player is required to sustain a specific position while touching an object).

Table 9: SIRTET's main variables.

Acronym	Meaning
SPO	Speed of objects
IBC	Interval between challenges
PST	Persistence time

The former is divided into Slow, Medium and Fast and; the latters are divided into Long, Medium and, Short interval.

Table 10 shows how the variables change from one level to the next. Notice, again, that only one variable changes its difficulty level at a time.

Table 10: SIRTET level transitions.

Level	SPO	IBC	PST
1	Slow	Long	Short
2	Slow	Medium	Short
3	Medium	Medium	Short
4	Medium	Medium	Medium
5	Medium	Short	Medium
6	Fast	Short	Medium
7	Fast	Short	Short

Physiotherapists prepare challenges according to the target population that could vary from children, to adolescents and, elderly people; and set specific values for Fast, Slow, Long and Short.

5. Discussion

MOLDE can be adjusted according to the needs of designers and content specialists. For instance, if the number of phases is reasoned to be too big, the designer should consult with the specialist in order to make a transition with more than one variable at a time. Nevertheless, the major idea of reasoning over the importance of the variables and their relationship with steps scaffolding upwards in big leaps (which define Phase variables) and small leaps (which define Level variables) still remained for all games implemented. Certainly the content specialist is the one capable of defining which aspect represents a big leap or a minor improvement.

Turns out that Level variables are changed more frequently during the game than the Phase variables. This is a good tip to reason about how to classify variable to become Phase or Level variables. Big leaps suggest Phase variables and changes on the fly suggest Level variables.

Tests with end-user are paramount. Computer scientists and specialists build a stereotype of the user that sometimes is far from the reality. For instance, MoviLetrando has been designed to start off very easy and slow but initial tests with target population have shown that it was not easy enough, it needed to be slowed down even further.

Quite often the specialist is not able to suggest variables. The designer has a difficult task ahead because he is the one responsible for translating specialists' intents into variables to be used in the game algorithm.

Using MOLDE made it much easier to adjust SGs for new populations. For instance, D2R was later adapted to be used with Down´s syndrome children as well which was achieved by modifying the populations parameters. Also, SIRTET is undergoing a slight change to be used not by frail elderly people but by children. This will be achieved by mainly changing the challenge sequence and the populations file since the whole idea of the game remains the same.

Not all kinds of SG can benefit from this MOLDE. So far, we have dealt with Educational and Health SGs. Health SGs that deals with posture, gesture and movements seem to be well suited. Educational SGs that deal with a quantifiable set of elements or features also seem to be good candidates.

Analyzing these SGs it can be said that MOLDE was suitable to them because these Educational and Health SGs share at least three aspects:

a) They deal with a handful of measures. But, if the target knowledge, behavior or skill it presented in a multifaceted way, the number of Phases and Levels could grow exponentially;

b) The measures used are all quantifiable. They can be considered as objectively verifiable by the game;

c) Because most measurements under consideration are closely related to the targeted objectives of the SG and then, they should be incorporated in the game mechanics.

As long as MOLDE has evolved over a variety of SGs, it can be said that MOLDE is a mature and tested methodology for SGs that uses measurable variables to bridge specialist purposes and game mechanics.

6. Conclusion

Serious Games (SGs) have grown in importance over the last decade because almost everyone is keen in technology nowadays. Games are a sort of technology that motivate and engage children. Therefore, using games for reaching children is a promising idea. However, there is a lack of methodologies that give some insight over how to go about translating the needs of a content specialist into a game.

This paper presented MOLDE, a Measure-Oriented Level DEsign that draws a road map for SGs Level Design. Once the SG has been developed using the proposed methodology, it was quite easy to comply with specialists demands to adjust the game for one or another aspect.

MOLDE has been compiled over the experiences of designing successful SGs (some that have been proved effective). We do not intend MOLDE to be a static methodology. On the contrary, we intend to push MOLDE features further on a wider variety of SGs in order to make it evolve into a complete Serious Game Design methodology. To achieve this we urge designers to use MOLDE as much as possible and report on their process of SG Level Design.

Acknowledgement

The authors would like to thanks CNPq for the Grant from the PIBIC program.

References

ALAMRI, A. et al. *A cloud-based pervasive serious game framework to support obesity treatment*. Computer Science and Information Systems/ComSIS, v. 10, n. 3, p. 1229-1246, 2013.

BOYLE, E.; CONNOLLY, T. M.; HAINEY, T. "*The role of psychology in understanding the impact of computer games*". Entertainment Computing 2(2011):69-74. 2011. DOI:10.1016/j.entcom.2010.12.002.

BRUCKHEIMER, A. D.; HOUNSELL, M. S.; SOARES, A. V. *The Design of a Virtual Rehabilitation Game*. In: International Association for Development of Information Society - Applied Computing, 2011, Rio de Janeiro - RJ. IADIS-AC. Porto-PT: IADIS, 2011. p. 339-346.

BRUCKHEIMER, A. D; HOUNSELL, M. S; SOARES, A. V. Dance2Rehab3D: *A 3D Virtual Rehabilitation Game*. In: 2012 14th Symposium on Virtual and Augmented Reality (SVR), 2012, Rio de Janiero. 2012 14th Symposium on Virtual and Augmented Reality. p. 182-10.

DESHPANDE, A. A. and HUANG, S. H. *Simulation Games in Engineering Education: A State- of-the-Art Review*.

Computer Application in Engineering Education. DOI 10.1002/cae.20323. 2009.

FARIAS, E. H.; HOUNSELL, M. S.; BLUME, L. B.; OTT, F. R.; CORDOVIL, F. V. P. *MoviLetrando: Jogo de Movimentos para Alfabetizar Crianças com Down*. In: XXIV Simpósio Brasileiro de Informática na Educação, p. 5-10, 2013. (In Portuguese).

IBRAHIM, R.; JAAFAR, A.. *Educational games (EG) design framework: Combination of game design, pedagogy and content modeling*. In:Electrical Engineering and Informatics, 2009. ICEEI'09. International Conference on. IEEE, 2009. p. 293-298.

LAZAROU, D. *Using Cultural□Historical Activity Theory to design and evaluate an educational game in science education*. Journal of Computer Assisted Learning, v. 27, n. 5, p. 424-439, 2011.

MARCOS, A.; ZAGALO, N.. *Instantiating the creation process in digital art for serious games design*. Entertainment Computing, v. 2, n. 2, p. 143-148, 2011.

NOBLE, A.; BEST, D.; SIDWELL, C.; STRANG, J. "*Is an Arcade-style Computer Game an Effective Medium for Providing Drug Education to Schoolchildren ?*". Education for Helath, 13(23):404-4-6.

PRENSKY, M. (ed.) Digital Natives, digital immigrants. In the Horizon, MCB University Press. 9, 2001, pp. 1-6.

ROSSITO, G. M.. *SIRTET-K3D: um jogo sério para auxiliar no equilíbrio de idosos*. 136 f. Trabalho de Conclusão de Curso (graduação) - Universidade do Estado de Santa Catarina, Curso de Ciência da Computação, Joinville, 2013. (In Portuguese).

SOARES, A. V. Utilização da Realidade Virtual na Reabilitação de Idosos Frágeis. Qualificação de Doutoramento. Programa de Pós-Graduação em Ciência do Movimento Humano. Universidade do Estado de Santa Catarina. 2014 (Restricted Acess, In Portuguese).

SOMMERVILLE, I. Software Engineering. 9ª Edição, Editora Addison-Wesley, 2011.