Master Thesis

MACe the Adventure

Exploring Game Development with Half-Life 2 and the Source SDK

Computer Science, specialization Multimedia

Faculty of Sciences
VU University of Amsterdam, Netherlands

S.V. Bhikharie

August 2007

Supervisor: Dr. Anton Eliëns

Introductory remarks

Version 1, published in August 2007

Website: http://www.few.vu.nl/~mace/

Abstract

Developing a videogame can be seen as finding a balance between verification and validation that determines the overall quality of the game. The discussion of the game development of MACe the Adventure, using Half-Life 2 and the Source SDK, serves as an example of how this balance works. The climate game Clima Futura provides a second example of how the balance between verification and validation works.

Acknowledgments

I gratefully acknowledge the contribution of the following people:

Amit Bhikharie (Adventure game expert, game tester)

Marco Bouterse (Original developer of custom shader applied in MACe)

Kin Hung Cheng (Co-creator of MACe concept and prototypes, game tester)

DaGoon (Creator of custom Alyx texture)

Anton Eliëns (Game development mentor and master project supervisor)

Johan Hoorn (Second evaluator)

Cheng Hsu (Game tester)

Mikhail Zouskov (Technical support for Steam)

Table of contents

1 Introduction

When developing a videogame, you should always strive to make it the best you can. In this regard, two important aspects of videogame development projects to always keep in mind are the validation and verification of a game. You should always strive to verify your game, to make the final product right. But it is equally important to validate, to make the right product (Larsen, 2002). In other words, verification talks about the game on the technical level, while validation does this on a conceptual level.

From this perspective, making the best game possible is finding a balance between verification and validation. This leads to my main question:

- How does the balance between verification and validation leads to making the best game possible?

To answer this question, I will discuss the development of my game, MACe the Adventure. To start out, some background is provided on the history of adventure games. This background is followed by an overview of the game aspects of MACe. Next, I will look back at the development history of MACe the Adventure. An important discussion topic here is how various technologies were used in developing MACe and moreover, how this affected the use of Half-Life 2 and the Source SDK. Finally, I will give an evaluation of the release and reception that is based on feedback and statistics.

As an additional example, I will discuss the climate game Clima Futura. Moreover, I will discuss how the experience with developing games like MACe the Adventure helped to decide upon a game architecture for Clima Futura.

In the conclusion, I will answer the main question based on my findings, followed by some afterthoughts about game development and the balance between verification and validation.

2 History of adventure games

The first adventure game was created in the 1970s. This game, Colossal Cave Adventure by William Crowther, was a text adventure as it used a text interface to create an interactive adventure (Adams, 2004). It quickly became very popular as it spread through the ARPANET, the predecessor of the Internet. Many text adventures followed after the success of Colossal Cave Adventure. Initially, these were variations and expansions of Colossal Cave Adventure, but later new and original text adventures were created. As technology advanced, so did the text adventure. The first change was the inclusion of graphics to illustrate the text descriptions. The next change proved to be a very important one for the adventure game genre.

At the beginning of the 1980s, graphics became functional as they showed players the actual location instead of describing them through text (Granade, 2007), along with showing the controlled character from a third person perspective. From here on, the graphical adventure was established. Ken and Roberta Williams created the first graphical adventures (Mystery House and King's Quest) and proved to be very successful with their company Sierra Online over the years. Besides Sierra, another major player soon established themselves, namely LucasArts. In 1987 they introduced the point-and-click icon interface with the game Maniac Mansion. It was further refined in LucasArts' Secret of Monkey Island. The biggest innovation of the point-and-click system was that it made adventure games more accessible. It eventually became a new standard for adventure games, as other companies started using a similar point-and-click system.

The next change for adventure games was once again brought forth through the advancement of technology. This time around, the introduction of the CD-ROM provided more capacity for games and their assets. Myst (1993) became famous for using this increased capacity to display pre-rendered 3D graphics and create detailed and more realistic worlds, which were viewed from a first person perspective. Myst also put a greater emphasis on exploration and mechanical puzzles instead of personal and object interaction.

Today's adventure games mostly have 3D graphics, as the whole game industry more or less embraced it as the de facto standard. Apart from this, there is no clear direction towards where the adventure game genre is going, as it had been

in the past. It is obvious that the whole adventure game genre has changed a lot over the years. Furthermore, with the introduction of so-called hybrid adventure games, which contain elements from other game genres, it is useful to know what characterizes the adventure game genre. Granade (2007) has made a well-formulated list of common characteristics for (graphic) adventure games:

- They involve a story
- They have puzzles to slow you down and engage you
- They involve exploration
- They employ graphics to tell their story

The variation within the adventure game genre is the result of different emphasis put on these characteristics. This is what makes adventure games different from (or the same as) one another. This list can also help to distinguish borrowed elements from other game genres in hybrid adventure games.

An implicit characteristic of adventure games is that they are games of progression (Juul, 2005). This means that the progression structure (in the form of a sequence of game events) is predetermined by the game designer.

3 Overview of MACe the Adventure

In this chapter I will go into detail about the game MACe the Adventure. First, I will explain the origin of the name MACe. Then I will take a look at what kind of game it is by analyzing its adventure game characteristics. This leads to a discussion of how the basic gameplay of MACe works. Finally, I will give a description of the MACe levels to get an impression of the actual game.

3.1 The name MACe explained

The name MACe the Adventure is derived from the name of the arcade center in the game. The owner of the aforementioned arcade center is Minos, and the name of his arcade center is an acronym for Minos Arcade Center (MACe). Since the adventure starts from this arcade center, the game is called MACe the Adventure.

3.2 Adventure game characteristics

With the adventure game characteristics from the previous chapter in mind, let us take a look at what kind of adventure game MACe is.

Story

The premise for MACe is as follows: "You take on the role of our heroine, who decides to visit the local arcade center, owned by the eccentric Minos. Unknown to her, this is the start of a strange journey...". Besides this initial premise, the story does not play any major role. Instead, it serves as the initial motivation to play the game.

Puzzles

Each level in MACe contains one or more puzzles. In order to progress to the next level, the player must solve them. Most puzzles are logic-based, supplemented by some search-and-retrieve and skill-based challenges.

Exploration

In order to solve the puzzles, exploration is needed. The levels are built in such a way that exploration helps the player with solving the puzzles. Through

interaction with the world, hints are given or inventory items are gathered for use.

Graphics

MACe is a 3D game that can be played from both a first and third person perspective. The player controls the main character directly and thus interacts with the world through this character. Furthermore, cut scenes are used to introduce and wrap up the game.

In short, MACe the Adventure heavily leans on puzzle solving and exploration. It can also be seen as a hybrid adventure game, since it contains some skill-based challenges.

3.3 Gameplay information

The player can explore the game world by directly controlling the main character. Through interaction with persons and objects, the player gains information. The heads-up display (HUD), which is located at the left bottom corner of the screen, displays the following information:

- Message field. Displays text. When a new message is added to the messages field, older messages are grayed out and eventually removed.
- Item field. Displays items. When an item is picked up or used, it is automatically added to or removed from the item field.
- Description picture field. In certain situations, a description picture of a person or object will appear in this field to give additional information about the current situation, action or dialog.
- HUD state. The HUD can be in the maximized or the minimized state. Whenever a new message is added to the message field, the HUD will automatically maximize itself.

Additionally, whenever a choice needs to be made, a pop up will appear. There always are two possible answers to choose from.

For a more detailed description about how to play MACe the Adventure, refer to the game manual in Appendix A.

3.4 The levels

MACe the Adventure consists of five levels. Each level requires one or more puzzles to be solved.

Picture 1: Minos Arcade Center

The first level starts in the arcade center. After a short intro, the player walks inside the arcade center. After talking to the owner of the arcade center, the player is asked to search for an arcade game poster.

Picture 2: Maze

The second level puts the player in a maze. The goal is to find and retrieve the key that opens the exit door. However, in order to get the key, the player will have to search for additional hints in the maze.

Picture 3: Insanity room

The third level is the insanity room. The player is placed in a small room that contains four teleport tiles. By stepping on the tiles, the player is teleported inside the insanity room. The question is how to escape this room.

Picture 4: Hall of light

The fourth level takes the player to the hall of light. After looking around, a teleporter can be seen. Unfortunately, this teleporter is unreachable at first. However, there seems to be a switch that is guarded by a large number of laser beams that provides the solution to reaching the teleporter.

Picture 5: Balance shrine

The fifth and final level brings the player to the balance shrine. An old wall scroll tells about the rules for obtaining balance. By applying this knowledge, the player is challenged to solve a mechanical puzzle that requires three rings to be positioned in the right way.

When the player successfully solves the final puzzle, a cut scene starts to wrap up the game.

4 The development history of MACe the Adventure

MACe the Adventure was developed in three steps. In each of these steps a certain aspect of game development was explored. These steps can be characterized by the technology that was used.

Game aspects	Technology used
Prototyping a game	VRML + JavaScript
Character animation	DLP + STEP
Implementation of prototype	Source + Half-Life 2

Table 1: Game development aspects explored mapped against technology used

By analyzing how the technologies contributed to the development of MACe, a trade-off can be determined for each technology concerning its usefulness for game development.

4.1 Game prototyping with VRML and JavaScript

The idea for MACe the Adventure was developed as part of the final project for the multimedia course Multimedia Authoring I (MMA1)[1]. This course is focused on getting familiar with 3D worlds by using the Virtual Reality Markup Language (VRML)[2]. The description for the final project for MMA1 contained rough ideas for a game:

> "It all begins in the arcade hall. Here you have all sorts of arcade machines that can be viewed. There is information available for each machine. When the visitor talks to the people in the hall, the adventure begins...
>
> We want to make an adventure game. The player has to looks for hints to get further in the adventure. It will be possible to 'talk' to people in the world, as far as interaction is concerned. This interaction is of importance, because you will get clues that will help the player advance. Video and audio will be used to support the interaction.

[1] http://www.cs.vu.nl/~eliens/onderwijs/multimedia/mma1/

[2] http://www.web3d.org/x3d/specifications/vrml/ISO-IEC-14772-VRML97/

> The goal of this game is to complete all levels by gathering information from the different characters and solving puzzles. It is possible to talk to the characters in the game by approaching them and clicking on them with your mouse. In most cases a character asks you a question after which you can reply by selecting one of the answers appearing in a pop up window. Answers, questions and remarks can be read on your HUD."

From this idea, three game concepts were derived:

- HUD: A box on the screen that displays text information and items.
- Director: The script that handles the game logic
- Levels: The game worlds that the player explores

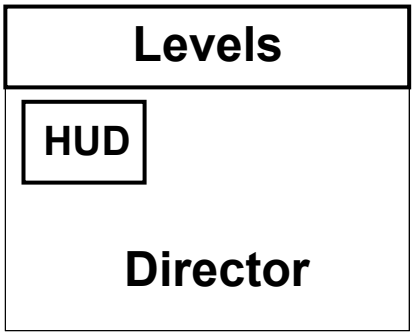
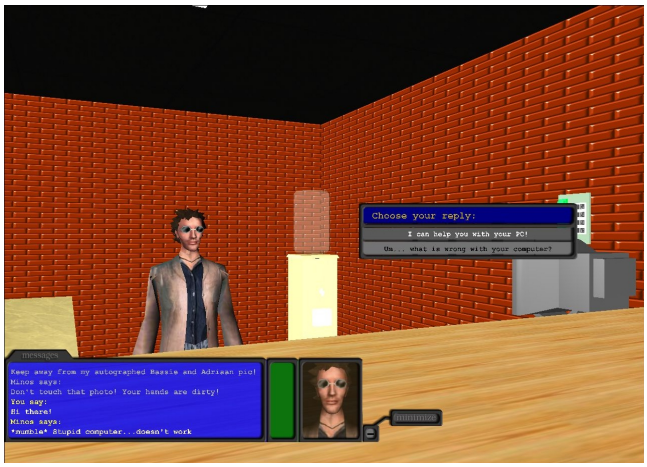


Figure 1: Game architecture of VRML/JavaScript prototype

Director is the main VRML file that handles the game logic through JavaScript. Some game logic examples are loading the worlds, responding to interaction with persons and objects and controlling the HUD. The advantage of using Director was that the levels could be developed without having to worry about how to apply the game logic for each separate level. This made creating levels more transparent. Together with the relative ease of creating 3D worlds using VRML[3], it allowed for rapid prototyping. The final version of the prototype consists of five levels (arcade center, maze, hall, insanity room, final room) plus an additional start and credits screen.

[3]http://www.devarticles.com/c/a/Web-Graphic-Design/3D-Graphics-Technology-VRML-Part-I-Introduction/



Picture 6: Minos Arcade Center (VRML/JavaScript prototype version)

4.1.1 Game development trade-off

Concluding, VRML with JavaScript provides the following trade-off for game development:

Advantages

- Well-suited for rapid prototyping of levels, since the threshold for learning and using VRML/JavaScript is relatively low
- Freely available for use by anyone

Disadvantages

- It provides a basic appearance that is a bit rough on the edges

4.2 Character animation with DLP and STEP

Adding character animation to the prototype of MACe was realized with Distributed Logic Programming (DLP) (Eliëns, 2002) and Scripting Technology for Embodied Persona (STEP) (Huang, 2004) through the follow-up course of MMA1, Multimedia Authoring 2 (MMA2)[4].

[4]http://wasp.cs.vu.nl/mma2/

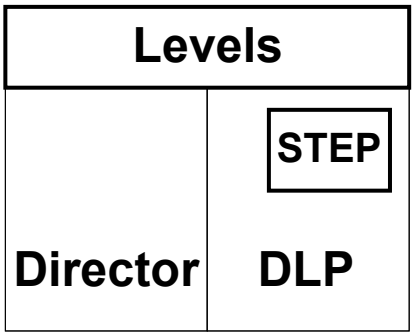


Figure 2: Game architecture of VRML/DLP prototype

To add character animations to MACe, the character avatars needed to be H-Anim[5] compliant. Unfortunately, this was not the case with the used avatars. Therefore, they were replaced with avatars that were H-Anim compliant. These were created as variations of one base avatars, to ensure consistency and uniformity for all the avatars.

To incorporate the H-Anim avatars and animation into the VRML prototype of MACe, a DLP script was implemented to independently run alongside Director, the core VRML/JavaScript program developed in MMA1. The DLP script checks where the player is (in which world) and which avatar the player is interacting with. Depending on the avatar and the selected dialog, a certain animation will be displayed. By using the recursive properties of DLP a so-called game loop was created that repeatedly checks parameters and executes the appropriate actions when needed. The final step was creating animation libraries using the STEP tool[6]. These libraries were made accessible for the DLP script to call when needed.

[5]http://www.h-anim.org/

[6]http://wasp.cs.vu.nl/step/tool.html

Picture 7: DLP animated characters in the VRML prototype

4.2.1 Game development trade-off

Concluding, DLP and STEP provide the following trade-off for game development:

Advantages

- Well-suited for rapid development of character animations with the STEP tool
- Freely available for use by anyone

Disadvantages

- Animation is basic
- Execution of animation is slow due to overhead of using a combination of different technologies
- Basic DLP knowledge is necessary to use the animations in combination with VRML

4.3 Prototype implementation with Source and Half-Life 2

The prototype implementation of MACe the Adventure was realized by developing it as a mod(ification) of the single player game of Half-Life 2. The technology behind Half-Life 2, Source, has proven to be state of the art (Valve, 2007) and therefore provides a solid foundation for creating games. This made it a very interesting and also suitable technology to use for recreating MACe.

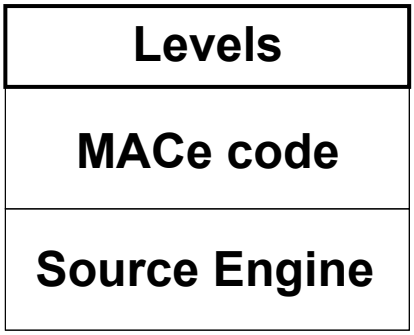


Figure 3: Game architecture of MACe the Adventure Source

To use Source and Half-Life 2 you first must have (bought and installed) a copy of the game Half-Life 2. The Source technology is made accessible through a number of tools and utilities, collectively referred to as the Source SDK, which can be downloaded freely. From here, the Half-Life 2 Single player C++ game code is available to modify. Additional software is needed to work with this code. As an example, I used Microsoft Visual Studio 2003 to edit and compile the code. This is also the case when you want to create additional game objects and character models, which all require additional software. The tools included with the Source SDK are a level editor (Hammer), a model viewer, a face animation tool (Face Poser), a model compiler (studiomdl), map compilation tools and 3D model exporters.

To make the most out of the Source SDK, you must learn how to work with a various number of tools and additional software. This makes using it intimidating at first, since the learning curve is quite steep. In my case, I had some prior experience with the Source SDK through my involvement with the VU game project[7], which made it easier to use the second time around.

[7] http://www.few.vu.nl/~vugame/

I started the implementation of the MACe prototype based on the following working plan:

Phase	Description
1. Coding basic game features	Recreate the prototype game engine
2. Level design & building	Recreate and redesign the levels plus additional coding for specific level features
3. Customization (look and feel)	Create logo and customize menu
4. Testing & revising	Playtesting by selected players and correcting the errors/problems found by the test players in order to improve the game code and levels

Table 2: Working plan for implementation of MACe prototype

4.3.1 Phase 1: Coding basic game features

With the prototype and the Half-Life 2 single player code as starting point, I made a list of modifications/features for MACe the Adventure:

- Remove violence. Since Half-Life 2 is a first person shooter and the violent aspects are not used in MACe, they need to be removed. Besides making it impossible for the player to access weapons, the crosshair is removed and weapon cheats are disabled. Furthermore, the player is made invincible and can never die.
- Third person view. In the prototype the player can switch between first and third person view. Since Half-Life 2 only uses a first person view for the player, a third person view needs to be added. By modifying the camera, a third person viewpoint is introduced. Furthermore, a keyboard shortcut is added that allows the player to toggle between first and third person view. By introducing a third person view, a custom player model must also be set, along with player model animations that will be visible when you are in third person view.
- Movement. The general player movement is tweaked to adjust to the style and gameplay of MACe.

- HUD. The HUD has to be built up from the ground, since it contains specific features not available in any default HUD in the Half-Life 2 code. Besides creating all the on-screen areas of the HUD, it is also necessary to create HUD entities. These entities are defined in the game code and are accessible through a game script. By making calls to the game engine, they can perform some action. In this case, they can manipulate the HUD. Three custom entities were defined: hudmonitor_text, item_setter and description_picture_setter.
- Pop up. The pop up is a custom coded object. To use a pop up in-game, an instance of the custom coded entity answer_popup must be called upon.

With this list at hand, I started coding to create the first version of the MACe engine. Additional coding was done in the next phases. For a full list of coded modifications and features, see Appendix B.

4.3.2 Phase 2: Level design & building

Creating levels for MACe was done by looking at the prototype levels and deciding how they should be adapted for the remake using the Hammer level editor. Another change that was made concerned the order of the levels. This was changed from arcade-maze-hall-insanity-balance-arcade to arcade-maze-insanity-hall-balance-arcade.

The arcade center level was changed quite a bit. The main reason for changing it was because I only wanted to include animated characters. Since this was not possible due to time restrictions, I needed to remove the characters as well as change the puzzle, since it was heavily based on interaction with the characters in the arcade center. However, Minos remained as the only character in the arcade center, although with limited animation. The level itself was also redesigned to be bigger and have a different lay-out for the arcade center.

The maze is nearly identical to the original. The only change is that the original side quest was scrapped in favour of a more functional side quest, namely obtaining a map of the maze, something that was not available in the prototype maze level. The radar from the prototype was recreated using a radar that was already available in the Half-Life 2 game code.

The insanity level puzzle stayed the same. Scrapped were two non-functional characters. The design of the level was drastically changed from a open, space themed area, to a cramped room.

The hall level puzzle was changed, because there were functional lasers available. The original puzzle also used a laser puzzle, but this was far easier to solve. The design of the level remained the same. Scrapped were three non-functional characters.

The balance level was changed in two ways. Firstly, the original level did not contain a puzzle. This was changed to include the balance puzzle. Secondly, the design of the level was improved to enhance the ambiance.

In the prototype, after the balance shrine level, the player was brought back to the arcade center where some actions needed to be taken to get to the credits screen. This was replaced in the Source version by adding a cut scene that wraps up the game and shows the credits, without the need for any player interaction.

Aside from specific changes for each level, all scripting now needed to be included in each level separately. This scripting includes game text, HUD description pictures and player interaction and is handled through the use of entities. In addition to the existing entities, some additional were coded to perform a specific function:

- trigger_teleport_remote_target_setter (insanity room)
- momentary_rot_button_extension (balance shrine)
- func_wall_toggle_nonsolid (hall of light)

4.3.3 Phase 3: Customization (look and feel)

To give MACe its own look, I designed a logo for the game. Furthermore, additional modifications/features were coded to created an overlay of the logo in the pause menu screen, along with a custom backdrop for the main menu.

4.3.4 Phase 4: Testing & revising

After the game was playable in the intended form (levels accessible in the right order, menu updated), and sufficiently tested by myself, I asked some people to testplay it. This brought up some level design issues.

The first issue was in the maze. By walking on top of the maze walls, it was possible to get stuck in the robot. This issue was resolved by adding an invisible box above the robot, which prevents the player from getting stuck.

The second issue was the difficulty of the final puzzle in the Balance Room. Of all the testers, only one was able to successfully solve it, who also happened to be an expert adventure game player. Although I had my doubts about the difficulty of the puzzle, I decided to leave the puzzle as it was. I felt that since it was the final level of the game, it was the appropriate place for this type of challenge.

4.3.5 Game development trade-off

Concluding, Source and Half-Life 2 provide the following trade-off for game development:

Advantages

- Well-suited for the development of high-end games

Disadvantages

- Steep learning curve, meaning that it takes a lot of time to get to know the provided Source tools and understand how to start coding
- Additional software is needed to modify the game code and to create new game objects and characters
- You need to buy the game Half-Life 2 to use the Source SDK

5 The reception of MACe the Adventure

MACe the Adventure was officially released on September 5 2006 with the launch of the game website[8]. It contains general information, the game manual, screenshots, a download for the game (an executable installer of about 34 megabytes) and contact information.

To further promote the game, I registered a profile for MACe on two websites that list game mods. The first mod website, Mod DB[9], is a well-known mod community website that lists and additionally hosts game mods. The Mod DB community consists of both mod players and developers. The second mod website, Half-Life 2 mods[10], is a website dedicated to listing Half-Life 2 mods.

5.1 Reception

To get an impression of the reception of MACe the Adventure, I looked at the ratings received at the profile websites. In addition, I received a review in German from Half-Life Portal[11]. An English translation of this review is provided in Appendix C.

Website	Rating	Original rating
Half-Life 2 mods	62 %	3.1/5 (91 votes)
Mod DB	83 %	8.3/10 (7 votes)
Half-Life Portal	70 %	70% (1 review)

Table 3: Ratings from various websites for MACe the Adventure, converted to a scale that ranges from 0 to 100 percent

Also, the number of downloads have been monitored since release.

[8] http://www.few.vu.nl/~mace/

[9] http://moddb.com

[10] http://www.hl2mods.co.uk

[11] http://www.hlportal.de

Period	Total number of downloads	Number of downloads per day
First month after release (September 2006)	1000	40
Eleven months after release (September 2006 - August 2007)	5000	14

Table 4: Selected download statistics for MACe the Adventure

By analyzing the ratings from these sources, I can say that MACe the Adventure has been received fairly positive, with ratings above average. This is confirmed by the download statistics, which indicate that people have downloaded MACe to play it.

5.2 Feedback

Besides the ratings and download statistics, I also collected feedback via the Mod DB profile and e-mail. I have compiled a list with issues along with the provided solutions:

- Technical issues with custom shader. In the hall of light level, there are platforms that appear at a certain point. These platforms make use of a custom shader I used. However, some players could not see these platforms. This is probably a video card related problem. To solve the problem, I provided a workaround that replaces the used shader by a standard texture (Appendix D).
- Difficulty of the Balance puzzle. I received a lot of comments about the Balance puzzle in the final level. After analyzing the received comments, I came to the conclusion that the hints to solve the puzzle were too abstract. Therefore, I provided some additional hints by e-mail to those who requested it (Appendix E).

  On a side note, some players were inventive enough to realize that this puzzle was based on the Ba-Qua sign and went beyond the game to find a reference picture of this sign to solve the puzzle.

6 Towards a game architecture for Clima Futura

Clima Futura[12] is a game about climate change and is currently under development at the VU University of Amsterdam. It is targeted at a young audience, ranging in the age group of 12 to 26 years. The primary goals are to create involvement with the climate issue and provide information by allowing the player to explore cause and effect relations, using models based on scientific research in a continuously evolving field of knowledge.

The development of Clima Futura is a multi-disciplinary undertaking, bringing together climate experts from a variety of backgrounds with multimedia/game development researchers. I was personally involved with this project to help with the game design. Based on the goals and concept of Clima Futura, the task at hand is to find a suitable game architecture.

6.1 Clima Futura concept

Clima Futura is a turn-based game, with 20 rounds spanning a 100-year period. In each turn, the player has the option to set parameters for the climate simulation model. The game is centered around the so-called climate star, which gives a subdivision of topics in climate research, as indicated below.

- Climate strategies
    1. Emission reduction
    2. Adaptation
- Climate systems
    3. Feedback monitoring
    4. Investment in research
    5. Climate response
- Energy and CO^2
    6. Investment in efficiency
    7. Investment in green technology

[12] http://www.climafutura.nl

8. Government rules

- Regional development
    9. Campaign for awareness
    10. Securing food and water
- Adaptation measures
    11. Public space
    12. Water management
    13. Use of natural resources
- International relations
    14. CO^2 emission trade
    15. European negotiations
    16. International covenants

Of the topics mentioned, not all are immediately represented in the simulation model underlying Clima Futura. Instead, they may be addressed in explorative interactive video. The climate star is actually used by the VU Climate centre[13] as an organizational framework to bring together researchers from the various climate disciplines. In Clima Futura it is used as a toolkit to present the options in manipulating the climate simulation model to the player. Advisors may be consulted to gain information about any of the topics of the climate star.

The result parameters of the climate simulation model are visible for the player in the values People, Profit and Planet, which can be characterized as:

- People - How is the policy judged by the people?
- Profit - What is the influence on the (national) economy?
- Planet - What are the effects for environment?

A generally acknowledged uncertainty within climate research surrounds the notion of climate sensitivity, which is the extent to which the climate and climate change is actually dependent on human activity. In Clima Futura, climate sensitivity is used as a parameter for setting the level of difficulty of the gameplay, where difficulty increases with the value for climate sensitivity.

[13] http://www.climatecentre.vu.nl

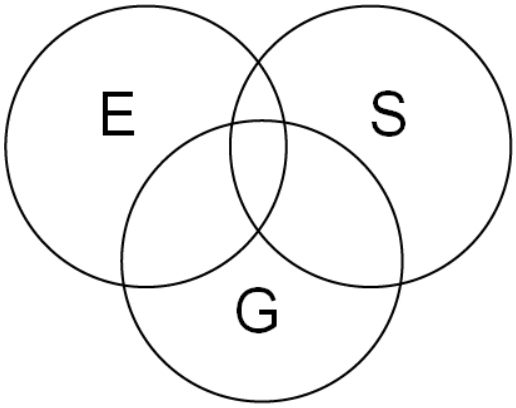


Figure 4: Gameplay, Simulation and Exploration

In summary, the gameplay of Clima Futura can be characterized as a combination of three processes:

- Game cycle - turns in subsequent rounds (G)
- Simulation(s) - based on climate model (S)
- Exploration - by means of interactive video (E)

Each of the three elements is essentially cyclic in nature and can trigger game events. For example, game events can be triggered by taking turns after 5-year periods, due to alarming situations in the climate simulation, such as danger of flooding an urban area, or accidental access to confidential information in the exploration of video material. In addition, Clima Futura features mini-games that can be selected on the occurrences of a game event, to acquire additional information, gain bonus points or just for entertainment. Examples of mini-games are negotiations with world leaders or a climate-related variant of Tetris.

6.2 Game architecture

Various technologies were explored in order to find a suitable one that would be compatible with the goals of Clima Futura. This exploration took place in parallel with the development of the earlier described concept. Prior experience with Source and Half-Life 2, in the form of the VU game project, MACe the Adventure and the Climate Control game[14] provided a concept for a first-person role-playing climate game in a 3D immersive environment. However, the use of such a platform would require far too much work, given the complexity of the game design. In other words, a trade-off has to be made concerning the complexity of

[14]http://www.few.vu.nl/~tlouisse/klimaat/

the game and the technology. Since the goal of Clima Futura is to give insight in the complexity of the climate issue, reducing the complexity of the game is no option. Therefore, the complexity of the technology must be brought down.

In this case, Adobe Flash[15] offers a nice trade-off. Instead of totally giving up on immersion, Flash video[16] can be used as a poor man's substitute for real 3D immersion. Furthermore, it offers good support for the integration of various assets, like photos, videos and existing flash games, into one coherent end product. Since Flash is supported by all major web browsers, it also provides an excellent platform for creating online games. Together with the Flex 2 SDK[17], which recently became open source, Flash offers a rich Internet application (RIA) toolkit that is sufficiently versatile for creating (online) games that require, in relation to high-end, a comparatively moderate development effort.

Based on Flash with the Flex 2 SDK and the Clima Futura concept, a modular architecture was chosen, which allows for component-wise development, with four basic modules and three (variants) of integration modules:

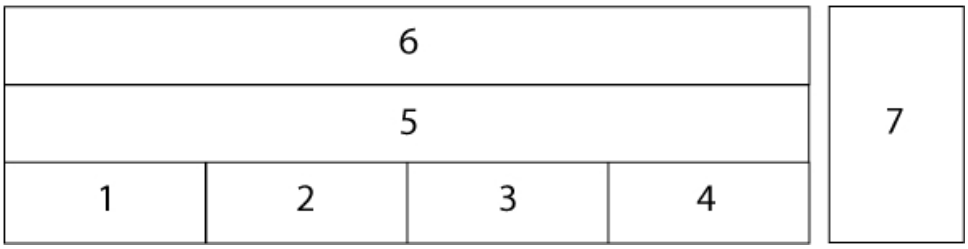


Figure 5: Clima Futura architecture

1. Climate model(s) - action script module(s)
2. Gameplay interaction - event handler per game event
3. Video content module - video fragment(s) and interaction overlays
4. Mini-games - Flash module(s) with actionscript interface
5. Clima Futura - integration of modules 1-4, plus server-side ranking
6. Adapted versions - educational, commercial
7. Multi-user version - with server-side support

[15]http://ww.adobe.com/devnet/flash

[16]http://www.adobe.com/products/flash/video

[17]http://www.adobe.com/products/flex/sdk

7 Conclusion

The balance between the verification (technical level) and validation (conceptual level) determines the overall quality of the game. The development of MACe the Adventure can be used to illustrate this statement.

The low technical complexity of VRML/JavaScript made it relatively easy to develop a prototype for MACe. This allowed MACe to grow on a conceptual level into a full-fledged game, which is exactly the purpose of a prototype. DLP/STEP provided a slightly more complex technology, but offered the possibility to expand the VRML prototype with character animations, fleshing out the prototype even more.

For the implementation of the MACe prototype, the Source technology was used, which has a high technical complexity. Prior experience with Source through the VU game project reduced this technical complexity a bit. Overall, the high technical complexity of Source was more or less evened out by the conceptual level of the prototype. This made it easier to focus on the technical development of MACe.

In the end, MACe the Adventure can be considered as a successful adventure game, with ratings above average, overall positive response and over 5000 downloads in 11 months.

The climate game Clima Futura provides a second example of how the balance between verification and validation works. In this case, the conceptual level for Clima Futura is high. To even this out, Flash with the Flex SDK is recommended as the preferred technology, since it provides sufficient technical complexity to create a compelling game at the high conceptual level Clima Futura is aiming at.

It should be clear from these examples that the balance between verification and validation is not like an equilibrium, but instead more like a mathematical chaotic function that somehow determines the quality of the game.

7.1 Afterthoughts

When looking at the balance between verification and validation, it is interesting to look at the extreme cases. On the one hand, there is a low conceptual level and a high technical complexity. This can be useful in the case of a technical

demo, which focuses on one particular game aspect. On the other hand, there is a high conceptual level and a low technical complexity. This provides an combination that can occur when games are developed for platforms that do not necessarily use the latest game technology, for instance hand-held devices.

By looking at these extreme cases, it becomes clear that games can cover the whole spectrum of the balance, especially when considering that it is also possible to adjust the quality for the game (think about budget games).

In any case, knowing about the balance between verification and validation is nice, but by no means enough to obtain it. This is realized with proper project management (Larsen, 2002) and is an equally interesting topic. In other words, game development is a process of both management and craft, which are equally important for the creation and success of a videogame.

References

Adams, R. (2004). *A history of 'Adventure'*. Retrieved from http://www.rickadams.org/adventure/a_history.html

Eliëns, A. (2002). *Distributed Logic Programming*. Retrieved July 17, 2007 from http://www.cs.vu.nl/~eliens/projects/logic/index.html

Granade, S. *A Beginner's Guide to Graphic Adventure Games*. Retrieved July 17, 2007 from http://brasslantern.org/beginners/intrographics.html

Half-Life 2 Mods. *MACe the Adventure | Half-Life 2 Mods*. Retrieved August 4, 2007 from http://www.hl2mods.co.uk/modsdetail.php?id=188

Half-Life Portal. *Half-Life Portal | Mods | MACe the Adventure*. Retrieved August 4, 2007 from http://www.hlportal.de/?sec=mods&site=mods&do=showmod&mod_id=354

Huang, Z. (2004). *STEP: A Scripting Language for Embodied Agents*. Retrieved July 17, 2007 from http://wasp.cs.vu.nl/step/index.html

Juul, J. (2005). *Half-Real: Video Games between Real Rules and Fictional Worlds*. Cambridge: MIT Press.

Larsen, S. (2002). *Playing the game - Managing Computer Game Development (International edition v1.11)*. Retrieved from http://blackwood.dk/PDF/PlayingTheGame-IE.pdf

Mod DB. MACe the Adventure for Half-Life 2 - Mod DB. Retrieved August 4, 2007 from http://mods.moddb.com/7816/mace-the-adventure/

Valve. Engine Licensing - information sheet. Retrieved July 17, 2007 from http://www.valvesoftware.com/files/SOURCE_InfoSheet.pdf

Appendix A - Manual for MACe the Adventure

Controls

Below are the default controls. You can redefine the keys/buttons in the OPTIONS.

Movement / Interaction

Move forward	w
Move back	s
Move left (strafe)	a
Move right (strafe)	d
Action Key	MOUSE BUTTON 1
Super Jump (if enabled)	SPACE
Duck	CTRL
Turn left	LEFT ARROW
Turn right	RIGHT ARROW

Miscellaneous

Take screen shot	F5
Pause game	PAUSE
Toggle 3rd/1st Person View	NUMERIC PAD PLUS
Minimize/Maximize the HUD	NUMERIC PAD MINUS

Other (non-customizable)

Look around	Mouse
Bring up the game menu	ESC

Starting the game

After starting up MACe the Adventure, the main game menu is shown. You have the following options:

NEW GAME - Select a chapter and begin playing. Depending on the progress you have made in the game, one or more chapters will be unlocked.

OPTIONS - Change game parameters. You can modify the game controls (keyboard), tweak the mouse handling and configure the audio and video settings.

QUIT - Exit MACe the Adventure.

Playing the game

MACe the Adventure is a puzzle-oriented adventure game. You take on the role of our heroine, who decides to visit the local arcade center, owned by the eccentric Minos. Unknown to her, this is the start of a strange journey...

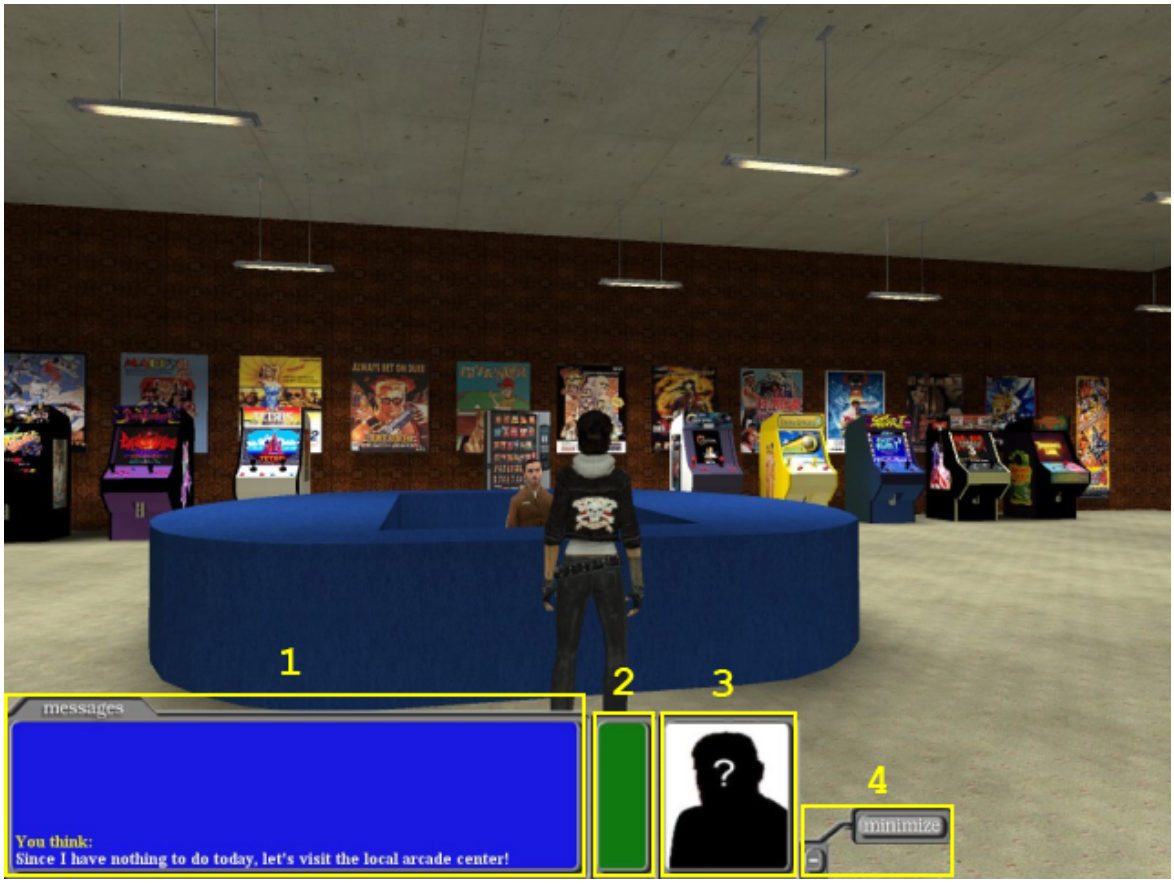
World interaction

To interact with a person or object, use the "Action Key". Depending on the situation, this will result in a feedback message. When using the "Action Key" keep the following general guidelines in mind:

- Make sure you are standing in front of the person/object. Sometimes it can be useful to switch to first person view to see if you are correctly positioned.
- Not all objects give feedback messages. Depending on (visual) hints given, it is made clear that you can interact with certain objects.

It is also possible to interact with an object by standing on it or touching it. This should be made clear immediately, in the form of a feedback message.

HUD



The HUD (Heads-Up Display) is displayed in the left bottom corner of your screen and gives the following information:

- Message field. Displays text. When a new message is added to the messages field, older messages are grayed out and eventually removed.
- Item field. Displays items. When an item is picked up or used, it is automatically added to or removed from the item field.
- Description picture field. In certain situations, a description picture of a person or object will appear in this field to give additional information about the current situation, action or dialog.
- HUD state. The HUD can be in the maximized or the minimized state. Whenever a new message is added to the message field, the HUD will automatically maximize itself.

Pop ups

Whenever a choice needs to be made, a pop up will appear. There are always two possible answers to choose from. Depending on your choice, this may or may not have consequences, so choose your answer wisely!

Game progress

To progress in the game, you must solve the puzzle(s) in the current level to advance to the next one. Each time you advance to the next level, it becomes available as a chapter that can be selected from the NEW GAME option in both the main game menu and the pause menu.

Pause menu

If you press ESC during the game, the pause menu is brought up, with the following options:

RESUME GAME - Continue playing the current game. You can also press ESC again to resume the game.

NEW GAME - Select a chapter and restart your game. Note that the current game is lost if you restart a chapter. Depending on the progress you have made in the game, one or more chapters will be unlocked.

OPTIONS - Change game parameters. You can modify the game controls (keyboard), tweak the mouse handling and configure the audio and video settings.

QUIT - Exit MACe the Adventure. Note that your current game is not saved.

Appendix B - List of coded modifications & features for MACe

Player

- Remove violence
    - No crosshair
    - Invincibility
    - Weapon cheats
- Third person view
    - Camera
    - Keyboard shortcut to toggle between first and third person view
    - Animations
    - Custom player model
- Movement
    - Sprinting
    - Jumping
    - Stop walking movement

Gameplay

- HUD
    - Drawing the HUD
    - Keyboard shortcut to minimize/maximize the HUD
- Radar
- Custom entities
    - answer_popup
    - hudmonitor_text
    - item_setter
    - description_picture_setter
    - trigger_teleport_remote_target_setter

- momentary_rot_button_extension
- func_wall_toggle_nonsolid

Menu

- Logo overlay
- Menu customization

Miscellaneous

- Disable node rebuilding message

Appendix C - Review Half-Life Portal

The original review in German can be found at:
http://www.hlportal.de/?sec=mods&site=mods&do=showmod&mod_id=354

MACe the Adventure

Half-Life 2 / Singleplayer

Mod-Info

Website: http://www.few.vu.nl/~mace/

Developer: S.V. Bhikharie

Released: Yes. Available since 05.09.2006

HLP-rating: 70 % (Place 78)

User-rating: no rating

Description

"MACe the Adventure" is an Adventure, that particularly consists of puzzles. It was developed by a student of the "Vrije Universiteit" in Amsterdam.

The story

On a trip to the arcade center the player is asked by the owner "Minos" whether he would like to solve a small puzzle. Thus a journey begins to different places, in which one must prove its abilities to solve puzzles...

The conversion

You control with the WASD keys; the left mouse button is used for the dialogues and items. In addition, the player can switch between first person and third person view. The large (minimizable) HUD provides clarity. You can immediately see what is in the inventory is, with whom one converses and reread discussions and events. Depending on the level you can also have a radar.

Five levels are playable nevertheless. Every level In principle offers a new puzzle. A story, which connects the maps logically, unfortunately is missing. The maps have simple designs. It is a little disturbing that only Half-Life 2 content was used. The main character for example is a simply changed Alyx.

Conclusion

An interesting mod, especially for puzzle fans. On the downside, the maps are not very detailed. This is probably because of the fact that MACe was developed by only one person.

Appendix D - Workaround for problem with custom shader

The textures for the platforms in the hall of light level use a custom shader, which could cause problems with certain video cards. Use the following workaround to solve this problem:

Open the following file in a simple text editor (notepad):

```
[Drive letter] : [your path]
/Steam/SteamApps/SourceMods/MACe/materials/sdk/show_gb.vmt
```

The file "show_gb.vmt" should look as follows:

```
"ColorShaderGB"
{
}
```

Change this to:

```
"LightmappedGeneric"
{
        $baseTexture "brick/brickwall045a"
}
```

Save the file and restart MACe the Adventure. If everything went well, the platforms will now use a standard brick texture. Note that once triggered, the platforms appear and disappear sequentially.

Appendix E - Additional hints for solving the Balance puzzle

The first step is realizing that each symbol corresponds to a binary number. To be more precise, each ring that can be moved represents a certain power of two (inner ring = 2^0 = 1, middle ring = 2^1 = 2, outer ring = 2^2 = 4). Furthermore, one long line represents a zero and two short lines represent a one. For example, the binary number for heaven (three long lines) is 0*1 + 0*2 + 0*4 = 0.

The second step is realizing that the symbols must be placed on the board in such a way that balance is build up gradually. To be more specific, you must count the lines for each symbol and make sure that adjacent symbols have a distance of 0 lines or 1 line. Since heaven has three lines and earth has six lines, this means that the number of lines a symbol has increases as it gets closer to earth.

The third step is to look at both sides of the balance shrine and count the number of objects that represent life and death for each side. You will notice that the death side has 9 graves, whilst the life side has 1 tree and 1 bird. In other words, the even numbers are placed on the life side (1 tree + 1 bird = 2 living beings = even number) and the odd numbers are placed on the death side (9 graves = odd number). This is one of the opposite pairs that must be balanced.

Finally, make sure that each symbol (binary number) is facing its opposite. In other words, the opposite of a long line is two short lines. For example the opposite for 0 (0*1 + 0*2 + 0*4) is 7 (1*1 + 1*2 + 1*4).

By combining all aforementioned hints, only one solution is possible.