

Live LEGO House v.01

Cristina Portalés Ricart

ABSTRACT

A house built with the LEGO blocks is enriched with multimedia files (sounds, videos and 3D animations) by means of the Mixed Reality technology (with the MXRToolKit libraries). Two dolls have to coexist, each of them having different characteristics and needs. Both users will need to agree in some aspects in order to rule the game properly.

1. INTRODUCTION

In the age of new technologies, coexistence is still an unresolved task. Different religions, culture or gender are sometimes the focus of misunderstandings and conflicts between human beings. Live LEGO house (abbreviated: LLH) is an interactive game in which the mutual coexistence matters. Through the Mixed Reality technology an interactive game is proposed, where gaming is used in this case as a tool to explore coexistence.

The proposed game is built in a collaborative scenario for two different users, each of them leading one of the dolls. It is also a game for all ages. The game can be divided in the following steps:

1. Construct your own house: use your imagination and creativity to build a house within the LEGO blocks.
2. Tell the program where the objects are: this is called the calibration process. Simply by pressing some buttons on a prepared keyboard you tell the pc how your house is arranged.
3. Explore the matters of coexistence and try to solve them wisely: you will discover how to play having into account your fellow's character.

2. THE MXR SCENARIO

2.1. SOFTWARE: MXRTOOLKIT

Mixed reality refers to the combination of computer graphics and real-world objects. This encompasses both augmented reality, which involves placing

computer graphics objects into the real world and augmented virtuality, which involves placing real-world objects into virtual environments.

As it is described in the MXRToolKit documentation, 'the MXR Software Development Kit (or MXRToolkit) consists of a library of routines to help with all aspects of building mixed reality applications. The philosophy of the SDK is to keep the interface extremely simple. All of the code is called using 'c' style function calls and 'c' style structures so that an understanding of C++ syntax is not required'. Therefore, one of the great advantages of the MXRToolKit compared with other marker based programs is the facility to build applications (for non-programmers experts) and mix different kind of media files (sound, videos, 3D objects, animations, etc) in a simple project. Thus this program was found to be suited to build the game.

Two characteristics that define the MXR applications are the collaborative scenarios and the tangible interaction. The following definitions are extracted from the MXRToolKit documentation:

- Collaborative reality: mixed reality allows users to interact with graphical objects while still allowing users to see the real world. This allows several users to naturally collaborate with a computer generated objects as if they were real world items. This provides a substrate for human-human interaction that is far more powerful than a number of people crowding around a computer screen.
- Tangible interaction: The tracking routines in this library allow computer graphical objects to be attached to real-world physical objects. By manipulating the position of the real object, we can correspondingly manipulate the position of the virtual object. This is known as "tangible interaction". It provides an extremely natural interface for manipulating three-dimensional objects which does not need to be learnt by the user.

2.2. THE INTERACTION SPACE: THE HOUSE

The house is the interactive space where the MXR happens (figure 1). The whole system consists on:

- A PC screen, where a TV is displayed showing the Augmented Reality;
- A WebCam pointing to the house;
- The LEGO blocks that form the house/furniture and represent the 'tangible' part of the application;
- The dolls that are also built with LEGO blocks and having the markers for the MXR, that represent the interface between user and computer and are also tangible;
- Two speakers where sounds are reproduced.

The camera pointing to the house is like an eye that registers the actions of the dolls, and acts accordingly. Thus, the house is 'alive', as different media files will be shown depending on doll's behavior, e.g. good actions (positive coexistence) will be rewarded, whereas bad actions (negative coexistence)

will lead to some kind of punishment (e.g. prohibition of listening to music while your partner is sleeping). Therefore, it can be said that rather than the users, the house is the one leading the game.



Figure 1: The interactive scenario.

2.3. THE INTERFACE BETWEEN USER-COMPUTER: THE DOLLS

If you are asked to communicate with your PC, it suddenly comes to your mind a mouse, a keyboard or a joystick as the 'normal' interfaces. You probably do not ever think about a 'doll' as an interface between you and your computer. But this is exactly what is happening with the LLH: the mouse is changed by a doll, making the interaction really easy as one only has to play around with the dolls to make everything happens.

Dolls are made with the LEGO blocks as well. The shape reminds to the LEGO dolls as well. Two MXR markers are placed in each doll, one at the front and one at the back. 3D models are the same as LEGO dolls, but within a small difference: note that a nose was added (original LEGO dolls do not have this characteristic). In this way, dolls are also able to 'smell', a characteristic that is not possible with the current version of the mixed reality application; note that users can hear, touch and taste, but not smell (or taste).

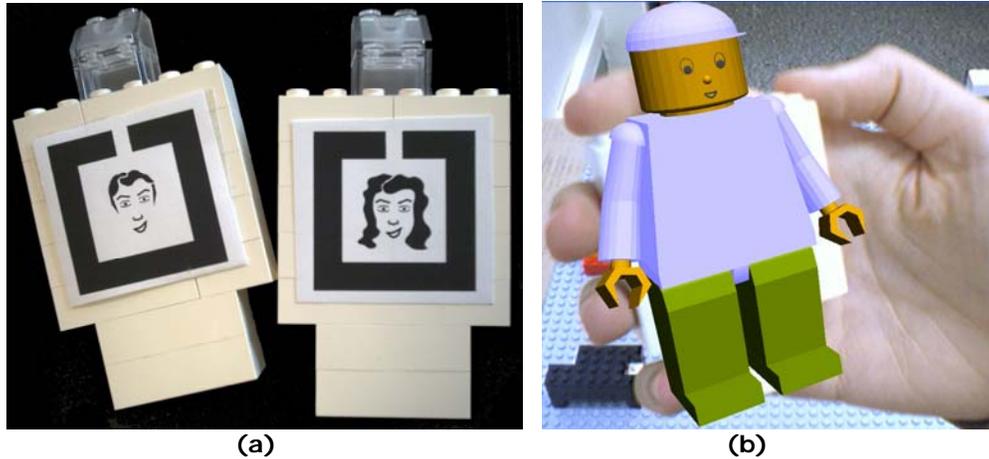


Figure 2. The dolls: (a) LEGOMan and LEGOWoman with attached markers; (b) User carrying the augmented LEGOMan.

2.4. MXR MARKERS

MXR markers are necessary in order to run the application. Here are the total amount of media files, frames and markers used in the current version of the LLH:

- Amount of media files: 65
 - 3D Objects: 14
 - Sounds: 45
 - Videos: 6
- Amount of defined frames: 16
 - Dolls: 2
 - Furniture: 14
- Amount of markers: 5
 - 4 real (dolls)
 - 1 virtual (rest of frames)

The most important part of this application is the amount and different kind of media files that are shown during the game. The total amount (65) compared with the two single markers that the user can see is quite unusual in this kind of applications. Normally, one media file is related to a single marker (or even to a single set of markers), but this has the disadvantage that only a reduced amount of media files can be used, as increasing the number of markers would slow down the system, or even would cause the system not to distinguish properly between markers (this is related to the camera resolution and the markers size). It would be quite hard to build an application with the MXRToolKit by using a total of 65 markers. Furthermore, for the LEGO Live House it was thought that using so many markers on the LEGO blocks would be unattractive. Also the problem of user hiding markers while playing would be a great problem.

Three steps were thought in order to reduce the amount of markers:

- The first step was to assign different media files to a same marker. These media files are rendered depending on the actions of the user. For example, when a user comes close to the radio, the radio is on and some music is played; user can change the music by a rotation (pitch value: rotation about Y axis).
- The second step was to ‘remove’ all the physical markers that remained still with respect to the camera. This is possible because the camera remains still (not HMD is used) and is based on the mathematical principle that this kind of markers would have a constant transformation (rotation & translation) with respect to the camera. The objects that remain also still are the furniture of the house (all out of the dolls), so the physical markers for them are not needed.
- The last step was to join all the furniture markers into a single marker. This can be done because the program is never going to ‘physically’ find the marker. This single marker is referred as a ‘virtual’ marker.

2.5. CALIBRATION PROCESS

Users can build the house as they wish. That means that furniture do not remain in the same position with respect to the camera depending on the users. But, at it was explained before, the transformation matrix of furniture should be known beforehand, as markers are not placed on the objects. That means that whenever the application is started, the user should ‘tell the program’ where the real objects are placed. This step is called ‘calibration procedure’, where the user should tell the program where the objects are by means of the keyboard. In the following table the keyboard buttons are shown for each doll. Note that the doll should be placed next to each of the objects and at that position the user should press the indicated button. In this way, the transformation matrix of the doll is passed to the furniture.

LEGO Man	Furniture	LEGO Woman
a	Washing machine	A
b	Radio	B
c	Chair	C
d	TV	D
e	Bed	E
f	Shower	F
g	Telephone	G

Table 1. Keyboard buttons related to furniture and dolls.

3. MEDIA FILES

3.1. INTERACTION

In this section, the way the interaction is produced is explained.

- Interaction is triggered by two different ways:
 - Proximity to the elements of the house: when dolls are close to different real elements of the house, some media files related to them are activated.
 - Dolls tilting (roll and pitch) also activate different media files.
- The sequence of actions by a single doll also conditions media files. This is technically managed with some kind of variables associated to the real objects and gives a 'surprise' element to the game. Note that users have the feeling that the media files do happen randomly, but in fact they are not.
- Actions of a doll condition other doll:
 - If both dolls are watching TV, a common program will be displayed.
 - If one of the dolls is sleeping, the sound is not allowed, that means that the other doll cannot listen to the radio or make a telephone call.
- Also a restricted action is introduced regarding to water: daily used amount of water is restricted, so dolls cannot have too many showers or put the washing machine too many times per day. If they are not allowed to use more water, they have to wait to next day (go to sleep!).

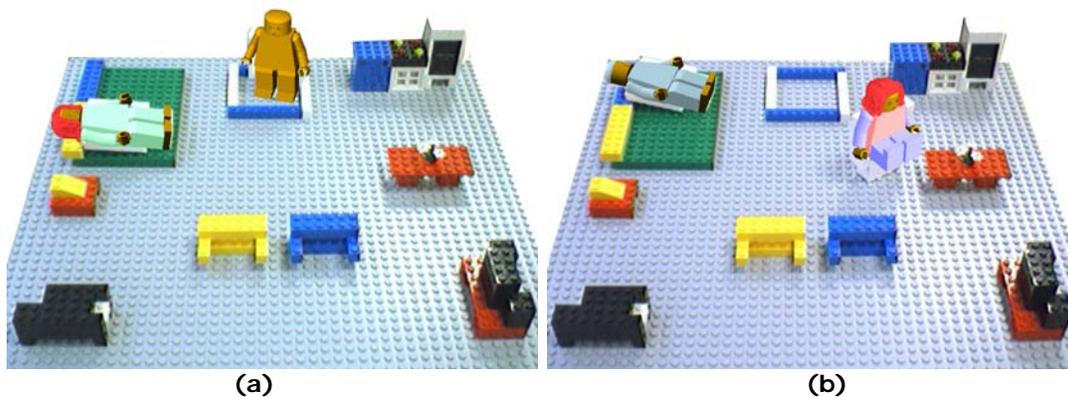


Figure 3: (a) Legoman is having a shower while legowoman is sleeping; (b) Legowoman is sitting quiet while legoman is sleeping.

3.2. SOUNDS

Sound media files are mainly taken from the web page:

<http://recursos.cnice.mec.es/bancoimagenes/sonidos/index.php>

In this web page, set of free media files from the Spanish Ministry of Education and Science thought for education purposes can be found. Between others, these sounds are included in the game: falling water, singing in the shower, tuning the radio, telephone ring, snoring, laughing, etc.

3.3. VIDEOS

One of the characteristics of Singapore is the achieved coexistence between people of different religions, for e.g.: a Hindu Temple can be found in Chinatown; all kind of food is offered at the stalls of a single hawker center; different traditions are respected; different languages are in use with the English as link; etc. Some video files showing these facts are shown in the LLH game within the TV.

The TV is maybe the central element in the game (figure 4). Users will pay attention to what is happening inside. This element is also interactive, and it will show the augmented dolls (TV is off) or some video files (TV is on), depending on users' actions:

- If at least one doll is sitting down in front of the TV:
 - The TV is on
 - A video file is rendered
- Otherwise:
 - The TV is off
 - The AR scenario is shown, as the house was reflected on a real TV-screen
- If both dolls are sitting, only the common program that delights both dolls is seen:

Doll 1	Doll 1&2	Doll 2
Program 1	Program 1	Program 1
Program 2		Program 4
Program 3		Program 5

Table 2. Programs associated to dolls.

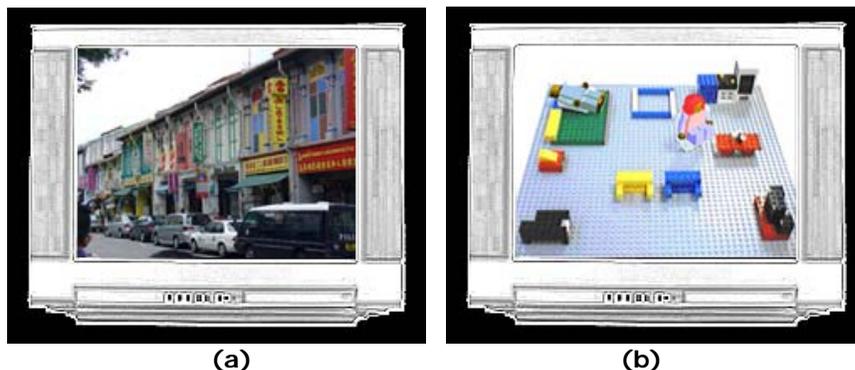


Figure 4. TV as displayed in the PC screen: (a) the TV is on; (b) the TV is off.

3.4. 3D OBJECTS

The 3D objects were modeled with the 3DStudioMax software. Within this, also some objects were animated (for example, when the dolls are sleeping). Afterwards, the objects were exported into the VRML format in order to be rendered by the MXRToolKit.

In the following figure the 3D objects are shown:

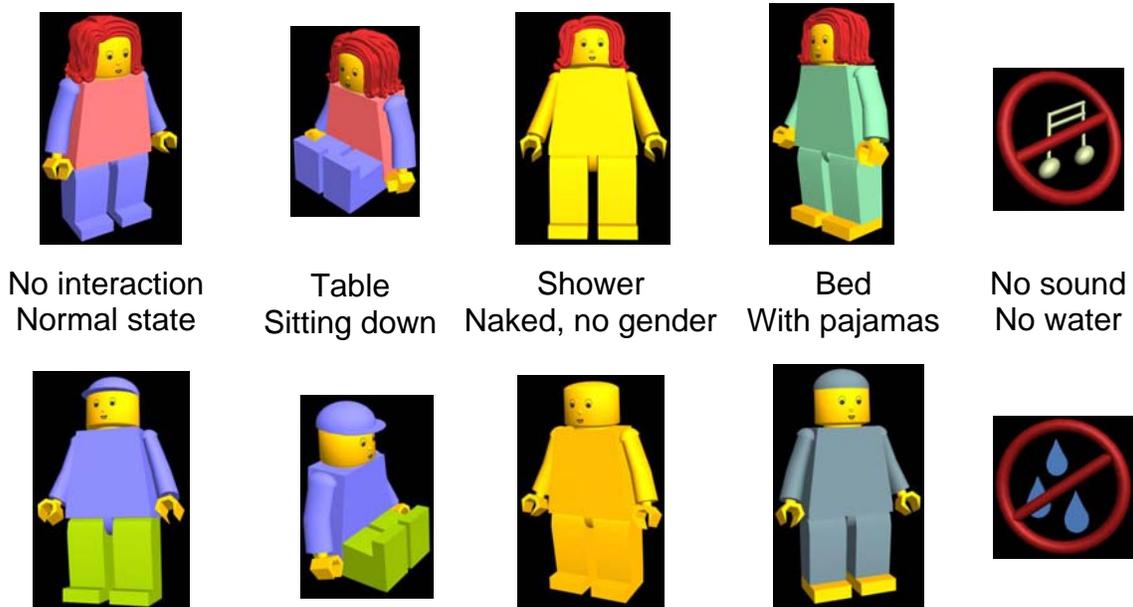


Figure 5. 3D models: dolls and restrictions.

Dolls have been modeled according to the original LEGO dolls (out of the nose, as explained before). That means that not extra 'bending' has been introduced: for example, when the dolls are sitting down, the entire leg is a single block. Also no gender has been added: when the dolls are having a shower, both bodies look like the same, and man/woman can only be distinguished for the hair.

4. NEW FUNCTIONS ADDED

Some new functions were implemented to the MXRToolKit libraries. These functions are:

- Distance between two markers:

$$\text{dist2} = \text{sqrt}(x*x+y*y+z*z);$$

- Roll, pitch & yaw between two markers:

```
roll = atan (T.r23/T.r33)*180/3.1416;  
pitch = asin (-T.r13)*180/3.1416;  
yaw = atan (T.r12/T.r11)*180/3.1416;
```

- 2 opposite markers working as a single marker (this is a modified function from the mxrFrameReadCube):

```
mxrFrameReadDoll(mxrFrame *frame, float dollWidth,  
float dpi, char *front, char*back);
```

- Mirror effect in the video files (flip image horizontally):

```
mxrImageFlipX(mxrImage *outImg, mxrImage *inImg);
```

5. PROBLEMS FOUND

Here, a brief summary of the problems found during the game execution is shown (some of them have already been solved):

- Reflective surfaces of the LEGO blocks: Reflected light is a problem in these kinds of applications, where marker tracking is based on image recognition. LEGO blocks are quite reflective, especially those with light colors.
- Field of view of the camera: As the markers on dolls are small (3.5x3.5 cm), camera should be close to the house. The closer the camera is, the less part of the house is seen. Therefore, a webcam of greater field of view is preferred. At the moment, a Fire-i webcam of Unibrain is in use.
- Occlusion: in the LLH the occlusion problem comes out. This problem arises when real objects are placed in between virtual objects and the camera. To avoid this, the position of the real objects should be known, but this would be quite difficult for the LLH as real objects can be moved. But there is a simple way to avoid occlusion: first of all, the camera should be placed at a higher position than the house (around 20 cm above), pointing down; secondly, the furniture of the house should be not that high (max. 3 blocks high); within this two simple considerations, the occlusion would be avoided. This fact is illustrated in the following figures, where the real doll is placed in between a table (in red, at the back) and an armchair (in blue, at the foreground). In the first case, placing the camera too low will make evident the occlusion problem; in the second case, with the camera at a higher position, the occlusion is avoided.

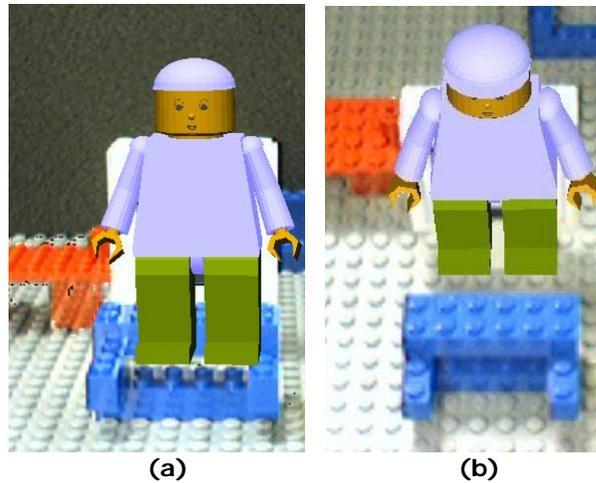


Figure 6. Image showing the occlusion problem: (a) with the camera at a lower position; (b) with the camera at a high position enough to avoid occlusion.

6. EVALUATION

This primer evaluation was performed mainly with staff of the lab (a total of six members participated). The purpose is to find the weak points of the application (in this first version) regarding to technical aspects, in order to improve the game before testing with children. The following table shows the asked questions, with the ratings being: 1. very bad; 2. bad; 3. could be improved; 4. good; 5. very good. Average values are shown.

	1	2	3	4	5
The role of the game: exploring coexistence				X	
The collaborative scenario: 2 users					X
The calibration procedure with the keyboard				X	
The 'real' dolls as interface					X
The idea of the house as a 'live' being					X
Media Files: Sound				X	
3D Models				X	
Videos				X	
The ambiguity in the doll's gender				X	
The restrictions: water/sound				X	
The apparent 'random' sequence of the media files				X	
The 'TV' as a display for the augmented reality					X
The way interaction is produced: distances & rotations				X	
The coexistence with your partner					X
The 'education' possibilities of the game				X	
General rating				X	

As it can be seen, the general rating of the game is 'good', but still some work is necessary in order to improve the interaction and enrich the game.

Also the calibration process will be changed in order to make it easier for children and also faster.

The participants were also asked to add some comments, focusing in what they would add/change to the game, and pointing to what they thought are the weak points of the system. Here are some of the comments:

- The coexistence part feels not so strong now. Maybe can add some effect to make it more evident. For example, put virtual food on the table when the person reaches the table; if two persons are there, it should be more food.
- Exploring coexistence not obvious without any explanation. Models can be talking/interacting with each other to make this obvious without explanation. For e.g., you try to turn on TV while partner sleeping and your partner tell you off.
- Calibration procedure slightly complicated for children use.
- TV seems confusing when it plays a video file and you can't see the second person.
- Add more interactivities between the dolls, e.g. eating together, dancing together. Now dolls can only do things separately; we can have a scenario when both are next to the radio, and they will dance together.
- More activities for dolls and more complicated interaction, e.g. when doll A dirties the bed, doll B will dirty the shower deliberately.

These comments are very valuable and in further versions of the LLH they will be taken into account.

REFERENCES

Mixed Reality Software Development Kit (Mxr Toolkit) Documentation. 2003
Mixed Reality Lab, Nus.