

Virtools & Maya Game Development



Multimedia Authoring II
mBMI
Vrije Universiteit Amsterdam

Paul Ivan
1502204
july 2006

Contents

1. INTRODUCTION	3
2. MAYA	4
3. VIRTOOLS	5
THE INTERFACE	5
INITIAL CONDITIONS	8
PHYSICS PACK	9
PITFALLS	10
4. DISCUSSION	11
LITERATURE	
APPENDIX A	

1 INTRODUCTION

This paper is a report about my project for the course Multimedia Authoring II at the Vrije Universiteit Amsterdam. The goal of this project was to develop an application with the use of Alias Maya and Virtools and to write a small report about my findings, especially considering Virtools. I started this project during the Winter Olympics of 2006. While surfing the net for a speedskating game, I came to the conclusion there weren't any. So I decided to build it myself, with my hometown as the décor. Before I could start building the game in Virtools, I had to model the skater and the environment in Alias Maya.

Maya is a huge 3d Modeling tool comparable to the equally known 3D Max. In this project Maya was used to build all the 3D objects of the game and all the animations of the skater. Although learning Maya takes a long time, you can achieve fast gains by doing some tutorials from the internet. There are a lot of (free) tutorials on the internet about how to model objects in Maya, how to rig characters or how to render objects (See appendix A for a list of useful websites).

For Virtools, on the other hand, there are practically no tutorials available. Therefore I will focus on Virtools in this paper, more specific this report will discuss some aspects previously not described by other students that followed this course ([1], [2]). Virtools is a quick and easy to use (after you get to know it) application/game development tool, in which you can use your modeled objects. There are a couple things about Virtools that, in my opinion, are essential to know before you can start really using it.

First how to use Maya to model objects for export to Virtools. Second the interface of Virtools, what the different windows are used for and especially the way 'building blocks' work. Third how to use the 'Initial Conditions' and in the last part I will present some information about how the Physics Pack works.

2 MAYA

As stated in the introduction, this report will only briefly discuss Maya. Since free tutorials for Maya are extensively available I will give some pointer for modeling objects in Maya for export to Virtools. The most important being, that Virtools is only capable of importing polygonal objects (So no Nurbs Surfaces or Subdivisions).

Second, it is important to keep your objects as simple as possible. This means not too much polygons. I've learned this the hard way, when I build the environment for Amsterdam City Speedskating in a way to detailed manner. For example, the building I modeled for the environment had to much more polygons when I first exported them to Virtools. This seemed to be a hard constrain on the CPU and thus on the frame rate.

So I had to find another way to make the building look real. An easy and effective way of doing this is using textures with an alpha component. That way I could use simple cubes for the building, making them transparent at places where needed, and still make them look quite complex. Virtools support this, at least in the *.png format (for the textures). This is illustrated in figure 1.

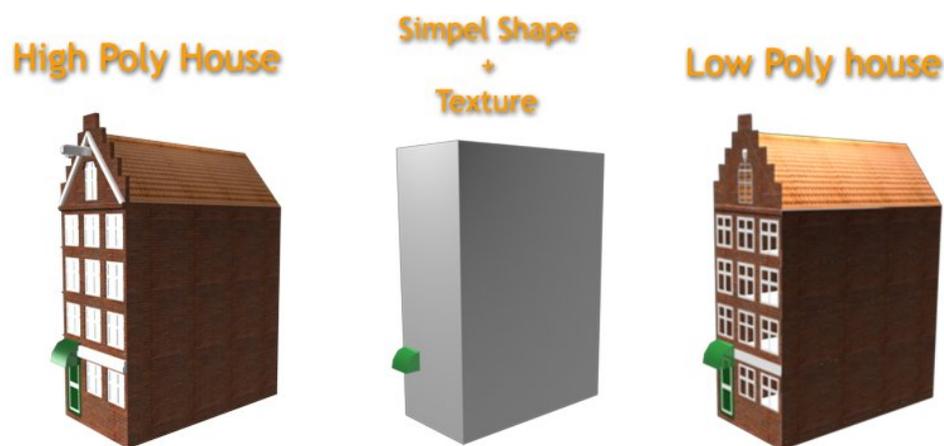


Figure 1. High Polygon House vs. Low Polygon House

3 VIRTOOLS

3.1 THE INTERFACE

The interface of Virtools consists of a lot of different windows and editors for different purposes. Here I will discuss those that are in my opinion the most useful.

3D Layout

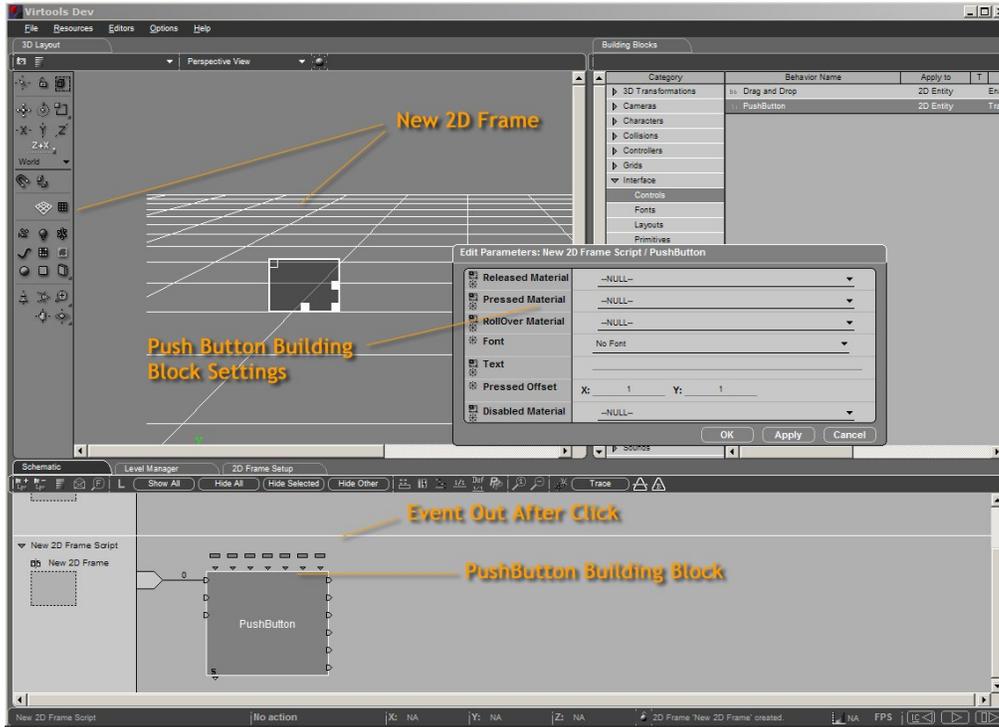
The 3D Layout window shows your workspace. Here you can view and build your project. There are tools for the basic Transformations (Rotate, scale, translate) and selection tools and it's the place to create new object, like 2D frames, 3D frames, materials, textures, Lights, Curves. Note that you cannot build actual 3D objects in Virtools, therefore you need to build them in 3D modeling packages like Maya or 3D Max.

The only objects you can build are 2D frames and 3D frames. These actually represent empty object in either 2D or 3D space, to which you can assign materials and textures. 2D frames are mostly used to create buttons or text on the screen. They are handy for building a user interface. An example of creating a custom button with a 2D frame can be seen in example 1.

3D frames are like other 3D object in your scene but they have no shape. They can be used to define a certain spot in 3D space, or you can assign 3D text to it.

EXAMPLE 1: CREATING A BUTTON WITH A 2D FRAME

1. Click the 2D frame button in the 3D layout to create a new 2D frame.
2. Place the 2D frame at the right spot, using the translation tool.
3. Create new materials for the button, with the 'create new material' button in the 3D layout window.
4. Search for the new materials in the Level Manager, and click right mouse and choose setup.
5. Define the right properties for the material, and choose whether or not you want to use a texture.
6. Drag and drop the PUSHBUTTON BB on to the 2D frame.
7. Finally choose the right materials for the right events.



Level Manager

In the level manager you can see all the objects currently placed in your scene. The Level manager allows you to view the specifics about each object in your scene (from 3D objects to textures to lights to cameras). By right clicking on an object in the level manager you can go to the setup of that object. That's where you can edit object specific parameters like position for 3D objects and diffuse color for materials.

Besides the setup windows you can also assign certain attributes to objects in the level manager, by right clicking on an object and going to the Add Attribute window. Attributes can be of different sorts and are often used by certain building blocks. As an example, you can assign the attribute Floor to an object, which is used by the building block Keep Character On Floor to know what object it should consider as the floor.

As a last remark about the level manager, you can define parameters like active at scene start, active now and reset at scene start there.

These can all be very useful when building an application. They define if certain objects will be in your scene when the application starts, or that you will activate them later.

A Data Resource Set

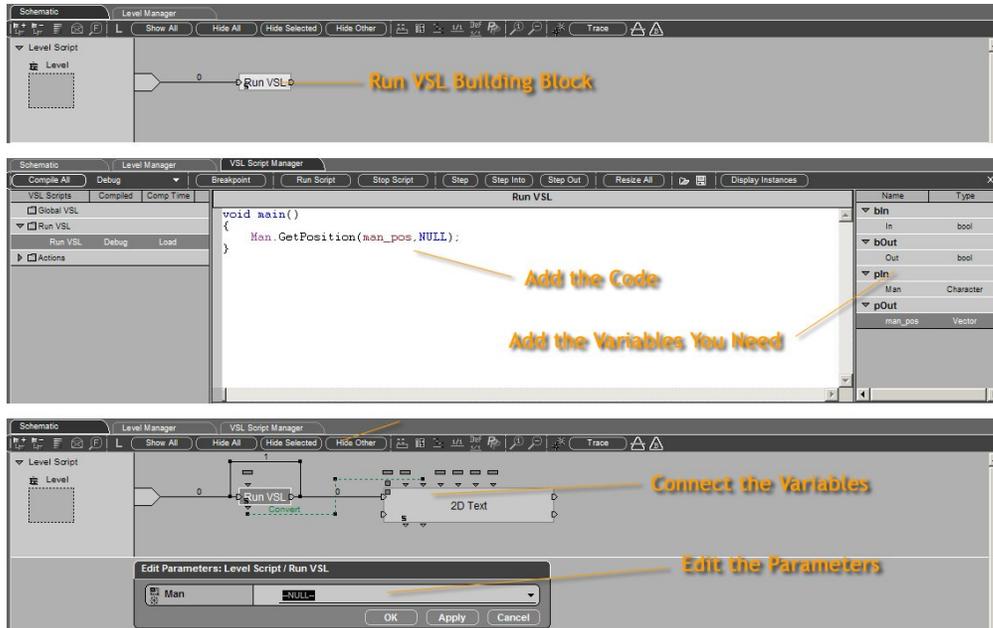
Virtools lets you import objects, textures, animations, etc directly through the Resources → Import File menu. An easier and more structured way of managing and importing your objects is by making a Data Resource Set. This can be done with the option Create a New Resource Set in the Resources menu. This will automatically generate a directory structure at a, by the user defined, place on the computer. This directory structure contains folders like; Characters, Textures, 3D Entities and some more. Now you can put your data in the directory structure and immediately view and import them in Virtools. You can see this Data Resource Set on a sheet next to the Building Blocks.

Run VSL

A good way of learning to use the Building Blocks is building them yourself a couple of times. This can be done through the use of the embedded VSL scripting language and the building block RUN VSL. An empty RUN VSL Building Block only has an input and an output parameter (bIn and bOut, which are Booleans) and no target parameters (pIn and pOut, which can be of any type). Press 'V' on the building block and you can write the script. The syntax of VSL is a lot like C. There are a lot of predefined types and classes you can use. After you programmed the Run VSL Building Block, you can use the pIn's and pOut's by connecting them to the target parameters of other Building Blocks. In example 2 this is demonstrated.

EXAMPLE 2: CREATING A RUN VSL BUILDING BLOCK

1. Put a new Run VSL Building Block on the Schematic and press 'V' on it.
2. On the right side add the pln variable named 'Man', by right clicking and assign it type 'Character'
3. On the right side add the pOut variable named 'man_pos', by right clicking and assign it type 'Vector'
4. Now in the middle add the VSL code, this will put the position of a Character in the vector 'man_pos'
5. Press Compile All and return to the Schematic
6. Now in the Schematic you can see that a pln and a pOut are available
7. Click right on the Run VSL and choose Edit Parameters.
8. Add your character.
9. Loop the Run VSL Building block if you want the result to be update every frame.
10. Now you can use the 'man_pos' pOut for example as input for a 2D Text Building Block



3.2 INITIAL CONDITIONS

Objects

Applied to objects Initial Conditions represent all current parameters, like position and orientation, of the object at the time of pressing the button (Set IC for selected). Using Initial Conditions is not a necessity, but a mere useful option. Virtools has the advantage of being a real-time editable software applications, which means you can adjust parameters when you project is running. One has to be careful about when to press the Set IC's for selected button. As an example, if you have a character walking around in your project, than if you press Set IC's after you've just walked you character to a different position than the previous starting position, the next time

your character will start at the position it was when you pressed Set IC's.

Scripts

Applied to scripts, Initial Conditions represent all the current parameters of the building blocks. So if you Set Initial Conditions when some parameters are still set to NULL (and you change them afterwards) the next time (after return to Initial Conditions) that parameter will be set to NULL again.

You have to think about when to press the Set IC's Button, if you do it can be very handy.

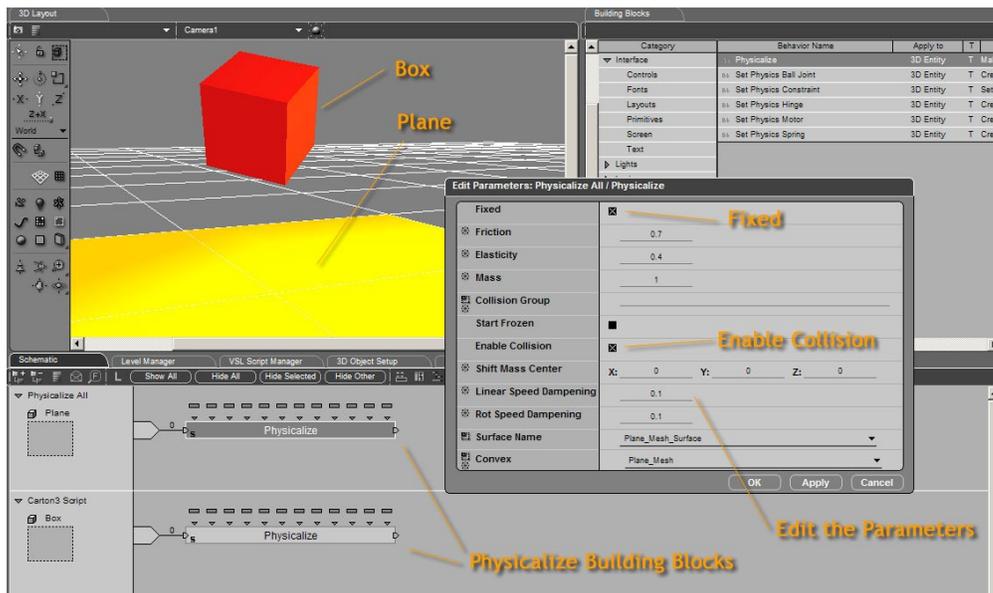
3.3 PHYSICS PACK

The Physics Pack is an additional package to make modeling certain physical aspects of a game easier. Such aspects could be gravity, speed (or the increase in speed) or bouncing. For example, without the physics-pack it is hard to model the increase in speed of the skater in a natural way. By physicalizing certain objects they can interact in a natural way. This is done with the Physicalize Building Block. With other building blocks like Physics Impulse or Physics Constraint you can give a Physics object an impulse in a certain direction, or constrain the direction it can move to.

I used this aspect of Virtools extensively to make the skating look natural (although the corners still remain tough to model). Example 3 shows a way to use the Physics Pack.

EXAMPLE 3: USING THE PHYSICS PACK

1. Import a Plane and a Box to the Virtools Environment
2. Set the positions of the Plane and the Box
3. Note that, now you should use Initial Conditions on the position of the box, to do this click right on the Box and choose Set Initial Conditions.
4. Now drag and PHYSICALIZE Building Block on to both the Plane and the Box (two Building Blocks)
5. Click right on the PHYSICALIZE BBlock of the Plane and choose Edit Parameters, check the fixed attribute and the enable the collision attribute.
6. Click right on the PHYSICALIZE BBlock of the Box and choose Edit Parameters, leave the fixed attribute unchecked and the enable the collision attribute.
7. Click the Restore Initial Conditions button next to the Play button.
8. Click Play.
9. Now you should see the Box falling down on the Plane and Bouncing realistically.
10. Now you can use the 'man_pos' pOut for example as input for a 2D Text Building Block



3.3 PITFALLS

Although I'm quite optimistic about Virtools, there were some problems that I couldn't solve. Because of the interface of the schematic, Virtools appears to process its information in a parallel nature. Especially in more complicated applications, there are lots of different scripts applied to different objects. Because you don't know which scripts are being processed first, problems can arise.

In the last version of Amsterdam City Speedskating, I still have this problem. Sometimes when you want to start skating after you just went through a corner, the controls don't work anymore. I think this is due to the fact that it is not clear in which follow the scripts are being processed.

Virtools has a solution for this in the form of the priority attribute. You can set the priority for scripts and Building Blocks, which should solve this problem. In my case it didn't.

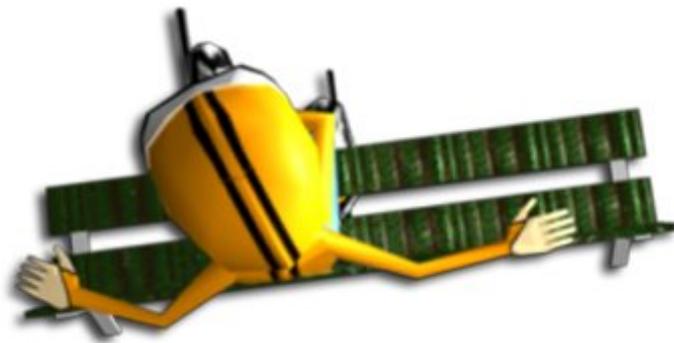
A second problem I have is with the fonts. Every time I put a new version online, Virtools changes the font types I have chosen. Virtools claims that you can solve this problem, by un-checking the dynamic property (In the settings of the Create System Font Building Block). This doesn't seem to work.

4 Discussion

In this report some important (start-off) aspects of using Alias Maya and Virtools to develop games were presented, with a focus on Virtools. Virtools can be an easy to use platform for quickly creating games of prototype games.

Although in the beginning it takes some time to learn, fast gains can be made by doing some simple exercises and using the Help function extensively. Using the Run VSL Building Block helped me a lot and made me understand the concept of Building Blocks.

The Physics Pack makes Virtools even more suitable for fast game programming, since modeling physical concepts can be a tedious job. Although I know my game, Amsterdam City Speedskating, is hardly ready and certainly not functioning as a 'real' game, I think Virtools can be a nice platform for future internet games.



Literature

- [1] Daniel, Developing with Virtools,
<http://www.cs.vu.nl/~eliens/design/@archive/games/virtools-dloran.pdf>
- [2] Laura en Wing Hong, Animation with Maya and Virtools,
<http://www.cs.vu.nl/~eliens/design/@archive/games/virtools-l-wh.pdf>

Appendix A

Shortlist of Maya Tutorials:

<http://www.simplymaya.com/>

<http://www.arch.columbia.edu/gsap/11718/>

http://www.cqtutorial.com/c2/Alias_Maya

<http://www.3dm3.com/>