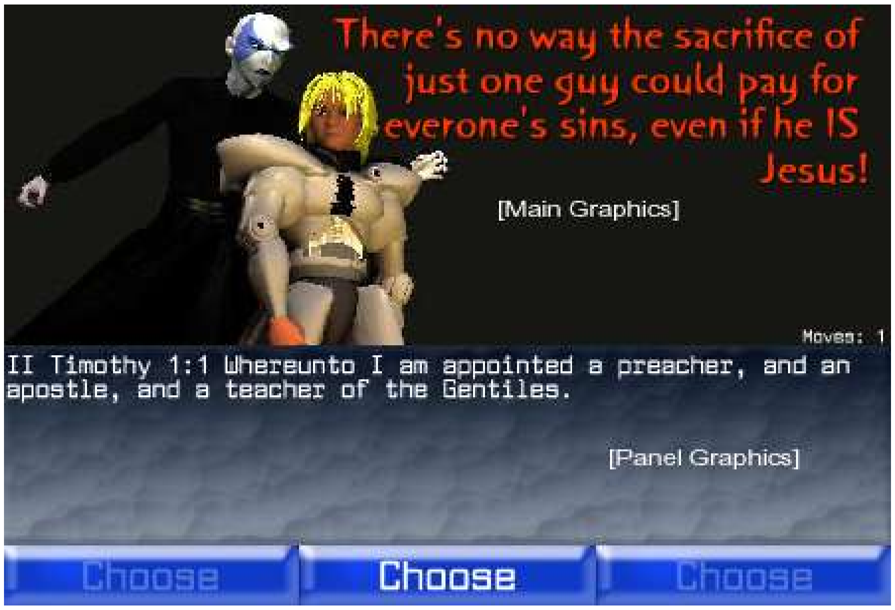
MicroRPG

MicroRPG is a simple script driven engine for making tiny web games that can teach, take the player on a journey, and use Bible verses to defeat enemies. It is very much like those "pick-a-path" adventure books. The script is a separate text file on the server, and (except for the fonts) all of the media is loaded externally. This means that the game can be completely changed without editing the original flash file. The scenes can be image files (.jpg) or Shockwave files (.swf) which can be animated and contain sounds. It can perform combat: choose one of three Bible verses to defeat a lie. You can branch between three choices to allow the player to explore a maze. You can have simple pages with a "next" button to teach, set up the story, or have the player walk down a path. It can have multiple endings, so that there may be degrees of winning. There is no limitation on the number of scenes, so even though the system is simple, it is possible to make a large world for the player to experience. It should be easy to change the scripting on a regular basis (given a library of graphics) so that users will have motivation to keep coming back to the site. Although the program is finished, the current sample makes use of crude placeholder graphics and should not be thought of as a real product.

Graphics

All graphics may be either .jpg or .swf files. The animation rate for swf files should be 30 frames per second.


There's no way the sacrifice of just one guy could pay for everone's sins, even if he IS Jesus!
[Main Graphics]
Moves: 1
[Panel Graphics]
Choose
Choose
Choose

II Timothy 1:1 Whereunto I am appointed a preacher, and an apostle, and a teacher of the Gentiles.

The overall screen is 450 x 300 pixels and is divided into two main sections: the Main Scene area (450 x 170) and the Panel area (450 x 130). The Panel area has a row of three buttons on the bottom, each button 150 x 30. Notice the Move counter in the lower right of the main panel. This indicates the total number of moves the player has made and can act as a score (the lower the better).

The main scene files

450 x 170 and are the view into the world. The critical art should be on the left or bottom, because the “lie” text is right justified. Dark colors work well in this version.

Title file

450 x 170 is seen when the game first starts.

Hit file

450 x 170 runs for about 2 seconds and is shown whenever you successfully hit an enemy.

Panel file

450 x 130. It is placed at the bottom of the screen to run underneath the Main Scene. It has three button areas imbedded in it that are 30 pixels high. This give a text area of 100 pixels for the Bible verses, etc. This panel can be a shockwave file, which is how you could make the background music.

Scripting

The script file is named script.txt and must be in the same directory as the program. It is a plain ASCII text file. The * character is used to separate fields in this scripting system. Be careful with your returns (paragraph marks). They are legal, but will be interpreted as paragraphs with a hard return and a blank line. Be careful with double and single quotes and apostrophe's. If you are working in Word, it tend to convert them to non-ASCII characters. Do not pad text with spaces. Media file names may be relative (in the same directory as the program or in a sub-directory of the current directory), or absolute (as in http://www. etc.). When uploading the file to you web site, make sure that it is uploaded as text, not binary. The first line of the script file must be as follows:

&scrip=TitleFile*HitFile*PanelFile*Introduction

These are the file names of the respective graphics, plus the welcome message. Yes, you do need the &scrip= at the beginning (not that it is scrip, not script).

Example:
&scrip=images/title.swf*images/pow.swf*images/panel.swf*Welcome to The Great Commission!
Cain will tell you lies, and you must choose the correct verse to overcome each lie.

Nodes:
The rest of the file is composed of node blocks, followed by one "*" on a line. Each node block starts with the header line:

Node Header
*label*image*Lie text

"label" is the label for the node and gives a name to the node so that other nodes can link to it. This may be empty.
"image" is the file name of the image for this node, and is required.
"Lie text" is the text to be written over the image and represents the lie spoken by the enemy. It may be blank.

Combat Node
The Node Header, followed by three lines in some combination of the following formats:
**Verse
Or
*ok*Verse

The one with the "ok" is the correct answer. And at least one of the lines should be "ok". The combat image is displayed with the lie (if any) on it, and the player chooses between the three verses. If the player chooses the correct verse, he sees the "hit" scene and continues on to the next node in the file. Otherwise he is take to the "lose" node (one and only one node in your game must have the "lose" label.

Next Node
The Node Header, followed by the following line:

*next*Line of text

The “next” indicates that this is a next node. Next nodes put up the graphics with text below and a next button to continue on to the next node in the file.
“Line of text” is the text to be displayed.

Stop Node
The Node Header, followed by the following line:

*stop*Line of text

The “stop” indicates that this is a stop node. Stop nodes put up the graphics with text below and ends the game. This can be used for winning or losing (and in-between states).
“Line of text” is the text to be displayed.

Choose Node
The Node Header, followed by three lines in the following format:
*link*Line of text

The player chooses from the three text blocks and is taken directly to the node that has the same label as “link”. This is case-sensitive (“Link” is not the same as “linK”). You can loop, skip section, build an entire maze, etc. The decision might be conceptual, rather than direction oriented: “Should I take the blue pill?”

Notes
You need to have at least one “lose” node.
The “lie text” is optional, and you may imbed text into your graphics instead.
Remember to end the file with a single “*” after all of the nodes.
If you use sub-directories, remember to use the forward slash”/” to be compatible with Unix/Linux.
You probably should use spaces and capital letters in your file names, as this can cause confusion when uploading: Windows does not care about capitalization, but Linux does. This can cause your script to work on Windows, but not on Linux.