More from less: thinking outside the box

Xander Wilcke
1720007

Abstract

This essay will discuss how we used out-of-the-box thinking and creativity in order to implement a labyrinth-like environment by solely making use of the basic XIMPEL platform. To this end, we will walk through both the conceptual and detailed design and the final implementation.

We concluded that it is indeed possible to create impressive displays of Interactive Multimedia by solely using the basic functionality of XIMPEL. However, it will require extensive work and coping with the limitations. We therefore recommend using the XIMPEL platform only for simple applications that require basic interaction and are formed around a narrow storygraph.

1 Introduction

With every passing day, it will become more unlikely to know someone that is not connected to a digital network. These networks, which being connected to was once merely a luxury, have crawled their way through the layers of society by the ever growing widespread availability of consumer electronics, e.g. mobile phones and laptops. Due to this availability, information is literally in the hand of the individual, wherever he or she might be.

With the abundance of sources, ranging from local to global, the individual is no longer limited to the conventional media, e.g. television, radio or newspaper. In order to survive the ongoing technological evolution, the media is required to be innovative in their representation of the information. Those that do not adapt, will fall behind and will ultimately cease to exist.

A recent instalment of innovative media is the Interactive Media(IM), which tries to combine the best of both media and gaming. With IM, the viewer is the participant of the story and he or she may decide how the story will unfold. This is achieved by presenting the participant numerous questions or dilemmas during the course of the story. Depending on the amount of decision points and the branching of the storygraph, the story can result in a similar of totally different way.

One of the difficulties with implementing new technologies into multimedia is the accessibility of it. The workings of these technologies, and more specifically

new media, are often unknown to the majority of the people. Therefore, it is paramount not to limit the user' creativity by forcing him or her into extensive programming. In order the appeal to this target audience, a pair of students have created the XIMPEL platform as their grand finale.

While the basic possibilities of the XIMPEL platform might seem limited at first, it can be still be used to create some impressive displays of IM. To this end, we will discuss how we use out-of-the-box thinking to create a labyrinth with an intuitive interface.

2 Conceptual Design

While many projects seem to start by aiming low at first, we decided to do the opposite by aiming at the stars. By having set such a high goal for ourselves, we were motivated to study the possibilities of the XIMPEL platform in much depth. The conceptual design, however, was on a strict time constraint, thus limiting our understanding of XIMPEL and its possibilities at that time.

2.1 Initial Concept

The initial concept was to create some sort of labyrinth in which the participant was able to freely go where he or she wanted. The character would then walk through the corridors until a decision point or an obstacle was encountered. At those locations, the participant was supposed to interact with the application, by either textual input or by clicking on an object with a pointing device, e.g. a mouse.

The labyrinth was to contain a clear starting and ending location, albeit not connected in a linear fashion. In addition, numerous rooms would be scattered around in which the participant would need to answer a question. Only when each of these rooms was found and every question was answered correctly, would the participant be able to reach the ending.

While walking through the labyrinth, the participant would encounter five different types of decision points. These are as following:

- A three way junction which offers the choice of going left, going right or to continue.
- A two way left-right junction which offers the choice of going left or going right.
- A two way left junction which offers the choice of going left or to continue
- A two way right junction which offers the choice of going going right or to continue.
- A dead end, e.g. a room with a single entrance, which offers the choice of turning around.

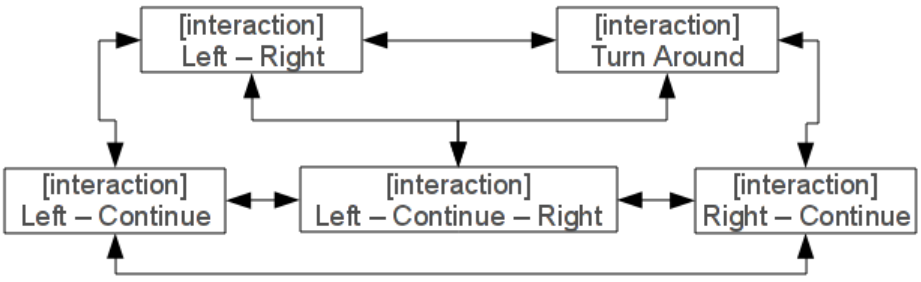

Figure 1: Interaction graph showing the interactions needed for navigation

In addition to those various types of decision points, we incorporated one way junctions, i.e. a left or right turn, and straight paths into the design. As the optional choices at these points would be useless, i.e. having only a single option to select, we omitted them. Focussing solely on the navigational interaction, we would therefore need five different interaction screens, as shown in Figure 1.

In order to keep track of the junction at which the participant would be any given moment, we looked at the usage of variables. Each junction would get an own variable where the value would represent from which way he or she came. That way, we could have used if-statements or the internal leadsto-tag to play the correct clip.

Theme

Although a labyrinth on itself might be interesting enough, we were compelled to construct it around a specific theme. This theme, however, would need to be applicable to our initial concept of a labyrinth, as we wanted to convey a coherent feeling to the participant. In addition, we strived to lift our concept above a simple game or button smash application.

As the mental image of a labyrinth often already has a mythical or fairytale-like connotation, we started by exploring those options. The former was ultimately chosen for the simple reason that we liked the stories we found on our literary study and the possibilities we saw on how to incorporate them.

For our general theme we had chosen for the Norse mythology, one that is not often used. A background story was found after reading through several sagas from that time period. The saga of Heimskringla was selected, which story can be summarised as following:

> The 10th-century Norwegian King Haakon I, surnamed “the Good”, died in a glorious battle while fighting with great honor. After the battle had ended, his trusted friends carried his body northward to Saeheim in North Hordaland. The King was buried there in a large burial mound in full armor and his finest clothing, and put on the way to Valhalla.

> As the king was a Christian, nobody knew if he would be allowed entrance to Valhalla. It would all depend on the bravery and strength he would show as he traveled to the ancient gates at Asgard.

The hardship the King has to bear whilst he proved himself worthy to the Norse Gods was an untold tale that we deemed both interesting and usable for our design.

It was then decided that our labyrinth would represent the path to Valhalla that the King would venture through. If this representation would be the actual physical path, a mental path or a divine path would be let up to the participant to decide for themselves. That way, we were hoping on adding a additional layer of mysticism.

Other aspects of the Norse mythology were incorporated to create a coherent design. The Valkyrie, the angels of death if you will, would have left clues, obstacles and challenges along the way in order to help the King reach Valhalla. These would be centered around the theme as well.

Obstacles and Score

For the obstacles, we ultimately decided to focus solely on questions concerning the mythology. In each of the question rooms, we would as a single question which answer would have been entered by an input field. An instance of such a question would be to ask "Name the two black feathered companions of the lord of Asgard", with which we mean to give the names of the two ravens of Odin. While we tried hard to make our questions Google-friendly, the variations in spelling and correct matching proved to be to challenging. We therefore settled for multiple-choice questions with an undecided penalty for selecting the wrong answer.

Each right answer would now add to a score that itself represented a binary number, e.g. 10011. This would be achieved by setting a starting score of a high integer, starting with one and a tail of zeros, and adding a power of ten with every correctly answered question. At the final door, this score would be evaluated in runtime, allowing the participant to enter if the score contained solely ones, i.e. all answers were correct.

3 Implementation

3.1 Media

In order to create an environment which was both consistent and relevant to our theme, we decided to make use of machinima. As we had to work with a time constraint, we opted of using an existing game engine. We ultimately chose for the game *Dungeon Keeper* as it allows the user to create a custom map and could convey the ecstatics of a labyrinth. In addition, it offered various objects and decorations that were fitting to the theme.

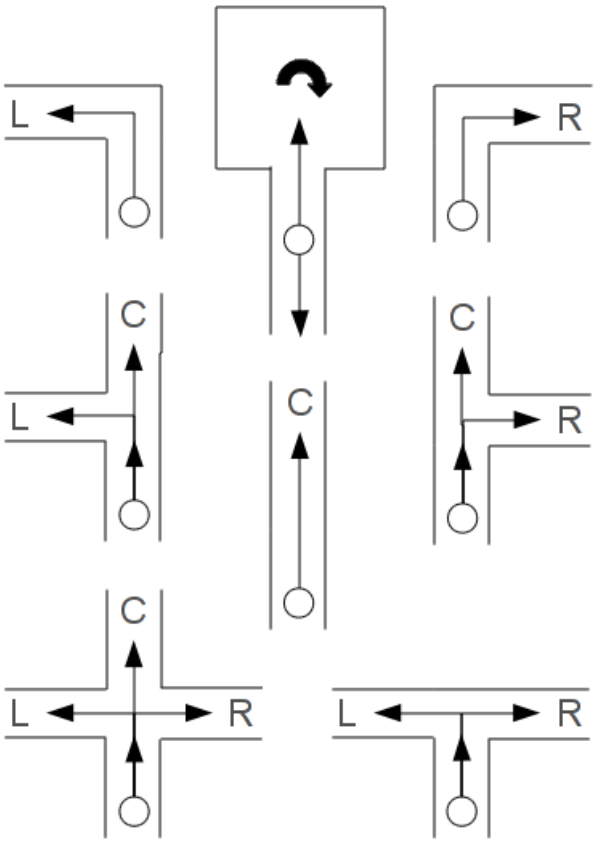

Figure 2: Sketches of the needed animations for navigation

As a good planning is half the work, we started out by sketching simple storyboards, as seen in Figure 2. Excluding both the prologue and epilogue, that would account for eighteen clips. In addition, looping clips were needed at each decision point as we wanted to immerse the participant into the world we created for him or her. The usage of static images would break the immersion. We therefore needed an additional five clips which would need to blend over without it being too distracting.

By making use of a map editor, we created a simple map in which we could simulate all the animations needed. The first prototype used an empty corridor which looked like a cavern of some sort. After a series of test runs, however, it became obvious that this lack of environmental decoration made it difficult to perceive movement whilst moving. To this end, we added torches along the walls. The distance between each torch was the same in order to reuse clips and to give a feeling of consistency.

An additional issue was the velocity at which the participant would walk through the labyrinth. We finally settled on a velocity which was about a quarter slower then the default one, as the latter might have negative effects on the learnability of the layout of the labyrinth.

To record our animations, we used a desktop recording tool; record-my-desktop, an open-source desktop session recorder for GNU/Linux and FreeBSD. As output, the recorder gives Ogg using Theora for video and Vorbis for audio. Post-processing was subsequently done by the usage of Avidemux, a multi-format, cross-platform video editor designed for simple cutting, filtering and encoding tasks. With this editor, the clips were cut to reflect the proper animation and cropped to have a consistent size. In addition, the background sound was filtered as an overlap from one clip to another resulted in distracting differences in that music. In the end, all clips were saved in the Flash Video format, i.e. FLV, by the use of the FFmpeg encoder.

3.2 XIMPEL

3.2.1 Connections

With the clips being ready, it was time to start by testing our design in the XIMPEL framework. First up was a proof of concept in order to test our clips, their transition and to get familiar with the XIMPEL XML code. It displayed a continuing path with an alternating turn going both left and right. The XML representation for this simple animation was relatively straightforward, as can be seen in Listing 1. No interaction was implemented at this stage.

Listing 1: XML snippet of Proof-of-Concept

```
<subject id=''left_turn''>
  <description>Going left</description>
    <media>
      <video file=''one_way_left_turn''
       leadsto=''right_turn''/>
    </media>
</subject>
<subject id=''right_turn''>
  <description>Going right<description>
    <media>
      <video file=''one_way_right_turn''
       leadsto=''left_turn''/>
    </media>
</subject>
```

It was learned that the use of custom made variables could only be made possible by modifying the application code of XIMPEL itself. The same applied to the usage of a field for textual input and the evaluation of the score during runtime. Due to our goal to solely make use of the standard configuration possibilities, we had to reevaluate our design.

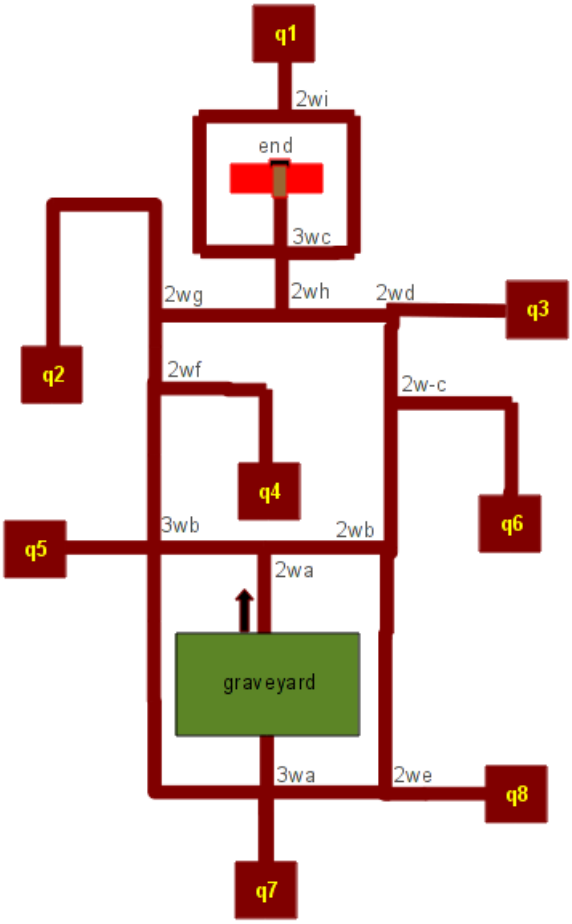

Figure 3: Final design of the labyrinth

Design Alterations

While the original concept of an open labyrinth was still possible, we had to downscale the size and branching points to keep it doable. We ultimately settled on a design that allowed room for eight question and the use of all forms of decision points. Furthermore, we put great effort in designing the layout to be quickly learnable by creating an unique way to each room, e.g. after two right turns or after a left turn. Finally, we incorporated a thematic starting and ending location, respectively a graveyard and a bridge over lava, as represented in Figure 3.

In the new design, each junction is labeled in order to refer to in the XML playlist. The number in the label stands for the number of options that are

available to choose from while the last letter allows us to discriminate between junction with an equal number, e.g. *3wb* is the second three-way junction.

As the usage of variables with a score evaluation at runtime was out the question, we had to connect our clips the had way; by writing the whole layout by hand. To this end, we had to connect every junction and room to the one reachable from that location. For instance, in the case of junction *3wa* we needed the following sixteen connections:

- *3wa* approached from the *graveyard*
- *3wa* left turn whilst approached from the *graveyard*
- *3wa* continue whilst approached from the *graveyard*
- *3wa* right turn whilst approached from the *graveyard*
- *3wa* approached from *Q7*
- *3wa* left turn whilst approached from *Q7*
- *3wa* continue whilst approached from *Q7*
- *3wa* right turn whilst approached from *Q7*
- *3wa* approached from *3wb*
- *3wa* left turn whilst approached from *3wb*
- *3wa* continue whilst approached from *3wb*
- *3wa* right turn whilst approached from *3wb*
- *3wa* approached from *2we*
- *3wa* left turn whilst approached from *2we*
- *3wa* continue whilst approached from *2we*
- *3wa* right turn whilst approached from *2we*

In addition to these connections, we needed left and right turns and straight corridors to create a transitions between locations.

3.2.2 Interaction

Although the basic overlay design of XIMPEL is fairly limited, we discovered that we could create additional layers of different sizes that link to the same object. Therefore, we could display an overlay that is shaped as an arrow, thus adding to the immersion of actually going in that direction. This was a great improvement on the default cubic design.

The overlay design were used in the loop clips that play whenever a decision point is reached. Hovering over an overlay would then change its colour and display a description. Clicking on them would start the playback of that action clip, after which a transition clip is displayed. Finally, the participant would reach a new decision point.

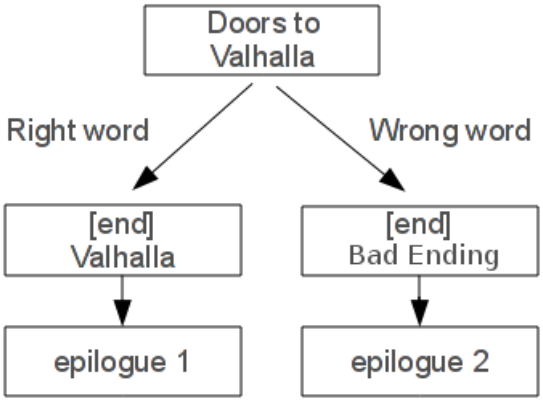

Figure 4: Final interaction

Design Alterations

As noted earlier, the evaluation of the score at runtime was not an option anymore, as well as using a textual input field. The former was a real issue since answering a question correct could now not be observed at the final obstacle. Therefore, we reevaluated the clue and challenge design in order to circumvent these limitations.

In order to prevent the participant to end the application before he or she had the opportunity to actually walk around and do some thinking, we still needed an end challenge. Even the creation of a multiple-choice question would still allow for random answer spamming. To overcome this issue, we decided on using a series of overlays to represent the alphabet, each overlay representing a different letter. As this would constitute a considerable amount of work, we opted on merely applying this technique on the final challenge.

The end challenge would consist of the door to Valhalla. As soon as the participant reached this decision point, an alphabetic interface would display. If the proper key is selected, it would lead to the next frame, which consist of the same loop clip and alphabetic interface. This would continue until the correct answer is spelled out. If, however, an incorrect letter is selected, the participant will fail and is presented with a clip featuring the bad ending, as seen in Figure 4.

To ensure a correct spelling of the final word, we altered the questions into clues, thus eliminating the need to search for the answer on internet. Each clue would represent a letter of the final answer. As we had eight question rooms in our design, the answer would consist out of the same number of letters. To ascend above a simple search game, we encrypted the letters by a simple alphabetic to numeric algorithm, i.e. $1 = a, 2 = b$, etcetera. The clue "First comes 3" would then say that the first letter is the third letter of the alphabet; c.

Finalizing Implementation

To represent all the various connection between the clips, we used a from-to notation. In combination with the design of the layout, this ensured an unique label for each section. The approach of junction *3wa* from junction *2we*, for instance, would be labeled as *3wa_from_2we*. The loop at the decision point would then be labeled *3wa_loop_from_2we*.

The loop function of a junction displays the different choices that can be made. In the case of a three-way junction, these options are going left, going right and continue, as seen in Listing 2. To connect to the proper segment, the direction that was selected is appended to the junction label.

Each decision that the participant makes, is awarded with an increment in his or her score. This score is then later used to evaluate the participant his or her gameplay. The score itself is returned as the number of decisions made. In addition, certain ranges of the score are awarded a title, e.g. scoring less than fifty points would be considered cheating, thus being label equally, and getting a high score might lead to the conclusion that the participant is lost.

Listing 2: XML snippet of decision point at a three-way junction

```
<subject id=''3wa_loop_from_2we''>
  <description>Make a choice at junction 3wa</description>
  <media>
    <video file=''three_way_junction_loop''
     repeat=''true''>
      <branchquestion>Go left, go right or continue?</branchquestion>
      <overlays>
         <overlay>
            <overlaycel x=''50'' y=''275'' width=''200'' length=''80''
             text=''Go Left'' scorevalue=''+1'' leadsto=''3wa_from_2we_left''/>

            <overlaycel x=''300'' y=''275'' width=''200'' length=''80''
             text=''Continue'' scorevalue=''+1'' leadsto=''3wa_from_2we_continue''/>

            <overlaycel x=''550'' y=''275'' width=''200'' length=''80''
             text=''Go Right'' scorevalue=''+1'' leadsto=''3wa_from_2we_right''/>
         <overlay>
      <overlays>
    </video>
  </media>
</subject>
```

At the final stage, we added two different clips and epilogues, for both a good and a bad ending, and a prologue that tells the background story and sets the mood.

Shortcomings

During the implementation phase, we encountered several shortcomings and limitations in the XIMPEL platform that forced us to reevaluated our design. We note that these issues might be easily solvable by modifying the application source. It was our aim, however, to edit solely the XML files that were included with the basic package.

The first limitations were encountered with the usage of overlays. At it happens, there seems to be a limit of twenty-five overlays on a given subject. This limit seems unnecessary and might possibly be just an array length, but it prevented us to represent a western alphabet, which needs one more. The creation of custom designs in overlays by combining smaller overlay blocks was considerately hampered by this limit as well.

Another issue with the overlays seems, ironically, to originate at the user-friendliness of the implementation. The overlays that lead to the same object share their colour changing properties. When, for instance, making all wrong answers lead to the same object, all those answers would light up is only one of them is selected. Creating a unique object for each answer seemed to be the only solution.

A more practical limitation is the lack of a sound manager in the platform. When creating a project comprised out of a lot of short clips, it is more favorable to be able to add sound or music to it without having to rely on the clips to feature them. A music tag, both for objects as for the whole, would therefore be advisable. Furthermore, a volume manager and a mute button are paramount, as poor sound quality and unbalanced sound seem to be abundant.

Finally, more often required options should be made available in the basic package, i.e. by editing the XML files. Relying on *actionscript* knowledge should not be necessary to use the score during runtime or to use a textual input field.

4 Conclusion

We set of to design a labyrinth that incorporated free and unlimited navigation by using the basic XIMPEL platform. To this end, we started by designing graphs for both the interaction and the story. Furthermore, as theme for the application was conceived and the challenges belonging to it.

After creating a proof-of-concept, we discovered the possibilities and shortcomings of the XIMPEL platform. Therefore, we reevaluated the design and modified it considerably in both size and interaction. Finally, a working model was build.

After finalizing project, we can come to the conclusion that it is indeed possible to create an advanced environment in the XIMPEL platform. This is, however, extensive work that requires many man-hours, close to forty different clips and almost five-thousand lines of XML code. An application of this magnitude could much better, faster and more efficient be build in other frameworks. It is noted, however, that the XIMPEL platform does not claim to be more than a player of interactive media.

The XIMPEL platform is a simple tool to create interactive media and simple games. For that purpose, it is probably sufficient. Project that need more than that, or have a wide storygraph, should better look at more task-specific

software.