GameBuilder, an Outdoor AR Creation Software

by

Chuan Zhang

S.B., Massachusetts Institute of Technology (2008)

Submitted to the Department of Electrical Engineering and Computer Science
in partial fulfillment of the requirements for the degree of

Masters of Engineering in Electrical Engineering and Computer Science

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

February 2010

Author ....
Department of Electrical Engineering and Computer Science
January, 19, 2010

Certified by ....
Eric Klopfer
Associate Professor
Thesis Supervisor

Accepted by ....
Dr. Christopher J. Terman
Chairman, Department Committee on Graduate Theses

GameBuilder, an Outdoor AR Creation Software

by

Chuan Zhang

Submitted to the Department of Electrical Engineering and Computer Science
on January, 19, 2010, in partial fulfillment of the
requirements for the degree of
Masters of Engineering in Electrical Engineering and Computer Science

Abstract

In this thesis, the author designed and implemented GameBuilder, a desktop application targeted towards 11-14 year olds for creating Outdoor AR games. The software attempts to reduce its learning curve by limiting the feature set, using a guided user interface, adding WYSIWIG-like display, and suggesting to the user a model which matches the software model. The author then conducted two studies to test the efficacy of the software. The first study was a free observation of children using the software during a workshop. The second study was a controlled comparative study where two populations designed games using either the GameBuilder or existing software. Based on observations and analysis of designed games, the author conludes that GameBuilder does offer advantages over existing software. Future improvements to the software are suggested.

Thesis Supervisor: Eric Klopfer
Title: Associate Professor

Acknowledgments

The author would like to acknowledge Professor Eric Klopfer for his intellectual support and guidance which made the author's thesis a success. The author would like to thank Project Managers Josh Sheldon and Judy Perry for their day to day encouragement and honest feedback. The author especially appreciates their expertise and aid in conducting research studies with subjects. Finally, the author would like to thank Lisa Stump for the programming guidance during implementation and Nyls Braak for continuing to develop GameBuilder into a usable product.

Contents

1 Introduction 15

1.1 Outdoor Augmented Reality Background 15

1.2 Motivation for Having Children Design Games 16

1.3 Motivation for GameBuilder 18

2 Design of GameBuilder 21

2.1 GameBuilder Design Principles 21

2.2 GameBuilder Implementation 22

2.3 GameBuilder Overview 23

2.3.1 Game Menu 23

2.3.2 Command Menu 25

2.3.3 Map View 33

2.3.4 List View 34

2.3.5 Selected Object 34

3 Cambridge Science Festival GameBuilder Pilot Study 37

3.1 Study Procedure 37

3.2 Observations 39

3.3 Game Analysis 39

3.4 Survey Results 40

4 Comparative Usability Study: GameBuilder vs. Editor 43

4.1 Study Design 43

4.2 Study Procedure 44

4.2.1 Subject Recruitment 44

4.2.2 Overall Study Procedure 44

4.2.3 Introduction 45

4.2.4 Walkthrough 45

4.2.5 Game Design 47

4.2.6 Survey Period 48

4.3 Data Analysis Method 49

4.3.1 Which software is easier to learn, GameBuilder, Editor, or inconclusive? 49

4.3.2 If GameBuilder is easier to learn, is it because of the user interface changes implemented in Chapter 2? Is it due to something else? 50

4.3.3 What are the most common mistakes subjects make using either GameBuilder or Editor? 50

4.3.4 What additional features do subjects want when designing games? 51

4.4 Results 51

4.4.1 Which software is easier to learn, GameBuilder, Editor, or inconclusive? 51

4.4.2 If GameBuilder is easier to learn, is it because of the user interface changes implemented in Chapter 2? Is it due to something else? 54

4.4.3 What are the most common mistakes subjects make using either GameBuilder or Editor? 57

4.4.4 What additional features do subjects want when designing games? 58

4.5 Discussion 59

5 Conclusion **61**

A Evaluation Rubric **67**

B Brainstorming Worksheet 73

C Survey 75

List of Figures

1-1 HP iPAQ 16

1-2 Editor Main Window 18

2-1 GameBuilder Main Window 24

2-2 Editor Menu 24

2-3 Editor Map Tool 25

2-4 GameBuilder New Game Window 25

2-5 GameBuilder New Object Wizard Step 1 26

2-6 GameBuilder New Object Wizard Step 2 27

2-7 GameBuilder New Object Wizard Step 3 27

2-8 Editor Properties Panel 28

2-9 Editor Properties Window 29

2-10 Editor Adding InfoPage 30

2-11 GameBuilder Edit Rules 31

2-12 Editor Triggers and Anti-Triggers 31

2-13 GameBuilder Edit Introduction Window 32

2-14 Editor Edit Introduction Window 33

2-15 Map View in GameBuilder (left) and Editor (right) 33

2-16 GameBuilder List View and Selected Object 34

2-17 Editor Overview Panel 35

3-1 Example CSF Game 40

4-1 Total Completion Time 52

4-2 Total Distress 53

4-3 Total Difficulty Rating 53

4-4 Step 9 Completion Time 55

4-5 Step 15 Completion Time 55

4-6 Step 14 Completion Time 56

List of Tables

4.1 Types of Data 49

Chapter 1

Introduction

1.1 Outdoor Augmented Reality Background

The Scheller Teacher Education Program (STEP) defines Augmented Reality Games as games which use the handheld computer to provide a layer of data that "augments users' experience of reality by connecting data on the handheld (e.g., multimedia), to physical space" [2]. Outdoor Augmented Reality games are a subset of AR Games using GPS technology that focus on using location: the sights, landmarks and scenes as the base real world information. The game can then overlay any imaginable virtual information which fits the purpose of the game. The earliest of these games, Environmental Detectives, allowed players to search for the source of a toxic spill on MIT's campus. Players could travel to different locations and conduct a variety of environmental tests. Players could then use the gathered evidence to form conclusions about the source of the spill.

In TimeLab, another game, players were given the role of time travelers from a future devastated by global warming [3]. They could change the past only in small ways, such as passage of legislation in Cambridge, MA. However, these small changes would have a butterfly effect which could alter the course of the future. Players walked around a "future" MIT campus and learned about what possible legislation could have the greatest positive impact on the future. Player visited the Charles River and imagined constant flooding. Players visited MIT's nuclear reactor to learn

Figure 1-1: HP iPAQ

about how nuclear power could affect future global warming.

There are a few basic concepts in Outdoor AR games. First, when the player starts the game, he or she sees an Introduction. Game objects are virtual bits of information that have an associated location which the player can visit. Game objects can be classified for game purposes into items or non player characters (NPC's). When a player visits a game object by standing within a predetermined radius of the object's associated physical location, the player sees the object's description. Then, the player sees one or more pages of information connected to the game object (InfoPages) which can be text or video. Finally, triggers and anti-triggers allow objects to appear or disappear after the player visits another object.

Until recently, all Outdoor AR games have been played on Windows Mobile PDA's with GPS capability (See Figure 1-1). However, future versions of the software will support the Android and iPhone platforms.

1.2 Motivation for Having Children Design Games

Prior to the beginning of this research, nearly all Outdoor AR games had been designed by researchers or teachers. A normal game development process might have been: researchers picked an interesting topic or theme such as climate change or learning about evolution while at a zoo. Researchers might also have picked a specific critical thinking skill such as evaluating noisy data in the context of predicting weather. Researchers then carefully designed an Outdoor AR game which attempts to best convey the information or teach the skills to the children. STEP developed an interest in utilizing the Outdoor AR platform in a different way. Instead of using the platform as a tool to teach a specific lesson plan, the platform could be used as a content creation tool. Children (11-14) would design their own games and then play their own/peers' games. The goal of the games could be to simply entertain, tell a story, or develop a skill in the player.

Designing games which accomplish any of the above purposes can be very difficult. Even though STEP researchers and collaborating teachers are experienced educational professionals, all currently released Outdoor AR games have gone through months of design, revision, and testing in order to ensure the best experience for the players.

Despite the difficulty, children can gain many important skills through game design which may include developing creativity, constructing complex objects out of basic components, picking relevant information to support an idea.

First, game design allows an opportunity for the child to be creative, similar to drawing a picture or writing a short story. Teachers or facilitators can lead brainstorming activities to encourage children to design more interesting games. While current Outdoor AR activities such as TimeLab or Environmental Detectives also provide opportunities for the child to be creative, game design differs by focusing exclusively on content creation. Because creativity is especially important in a knowledge based economy, activities which promote creativity have advantages over other activities.

Second, game design develops a child's ability to construct complex objects out of basic components. For example, a child could first brainstorm a game which describes a student's adventure after he or she got lost in the woods. However, after brainstorming, the child must then construct the story out of the available elements in the Outdoor AR platform. For example, the child must make a "bear" NPC and maybe a "honey" item. If he wants to convey that after the player sees "honey", the "bear" appears, he must create a trigger from the "honey" to the "bear". This skill has many analogs in the adult world. For example a computer scientist who wants to create a new algorithm must understand how to construct the algorithm

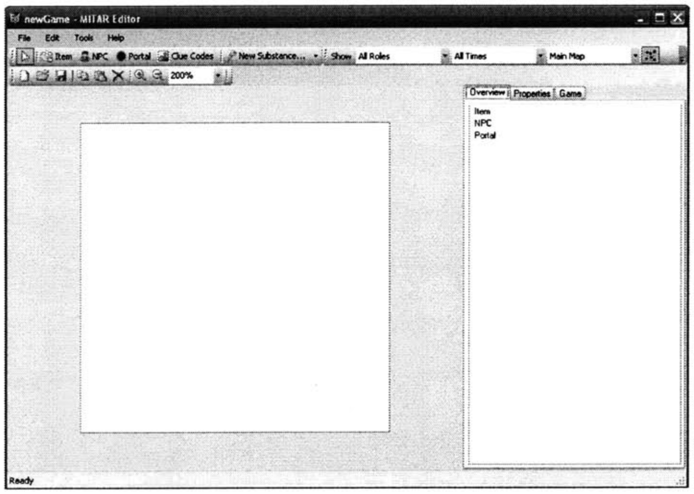


Figure 1-2: Editor Main Window

from preexisting code modules.

Finally, Outdoor AR games give the child an opportunity to pick relevant information to support an idea. Outdoor AR games focus on the interaction between the physical environment and the player. STEP believes children who design games will typically be constrained to a certain location (their school or a local park). While this area might contain many locations for the player to visit, in order to develop a great game, the child must determine the most suitable locations (based on scenery, smell, or sounds) which will best achieve his goals for the game. In many scientific fields, there is often a lot of information most of which is irrelevant. In order to evaluate theories, scientists must be able to identify which facts are relevant.

Given these benefits of game design, STEP moved forward to create curricula focused on having children (11-14) design their own Outdoor AR games.

1.3 Motivation for GameBuilder

Prior to the author's work, all Outdoor AR games were designed using the Editor, a desktop application for Windows (Figure 1-2). The Editor offered all the features of the Outdoor AR platform and successfully served as the primary game development environment for Outdoor AR.

For the purpose of the game design workshops, most children would attend a one-time workshop rather than attend multiple times. If Outdoor AR were to be a consistent component of their curriculum, it might make sense to invest a steep upfront cost in learning full-featured software. However, this makes less sense if most children will design games only once. Given a limited time frame, STEP researchers needed to ensure that more time was spent on the core activity of designing games and less time was spent on learning the tools necessary to support the core activity. STEP researchers decided that Editor would not be suitable for children's game design activities due to the software's complexity and its user interface, both of which increased its learning curve. First, Editor allows for a rich feature set used in early Outdoor AR games. An example of one of these features is a "spill", developed for Environmental Detectives. A "spill" is a geographic region that has associated data, but that has no onscreen visual representation that a player can see. However, players can conduct virtual tests in a spill region to get pieces of the data associated with the spill.

Second, Editor's design follows a standard user interface similar to many productivity software's used in office environments. Adults, because of prior experience to such software, might have a much easier time learning the software compared to children.

STEP decided to develop a new game design software called GameBuilder targeted specifically towards children for the purpose of supporting these workshops. The author commenced design and development of GameBuilder in June 2008.

Chapter 2

Design of GameBuilder

2.1 GameBuilder Design Principles

Nielsen lists 5 major criteria to judge software usability: learnability, efficiency, memorability, errors, and satisfaction [4]. Because GameBuilder will mostly be used infrequently and with novice users, we focused primarily on increasing learnability at the possible expense of the other criteria. GameBuilder's implementation followed 4 primary design principles.

1. *Feature Reduction.* The author eliminated many advanced features used by researchers in previous Outdoor AR games, such as clue codes, spills, portals, roles and chapters. In order to effectively use these advanced features, designers needed to have played multiple Outdoor AR games. Given a novice audience and limited time of workshops, most designers would not be able to understand these features. On the first design iteration, GameBuilder only supported objects and their descriptions and an Introduction. However, support for triggers and anti-triggers (called Rules) were subsequently added (see 2.2). An argument could be made that the Editor could have been used and the children could have been told to ignore the advanced features. While this option is possible, it would require more screen real estate, increase menu and toolbar complexity, and allow for more possibilities of errors.

2. *Guided User Interface.* The Editor was designed primarily for flexibility and ease of use for expert STEP Researchers. In general there are multiple ways to accomplish each task and the tasks can be accomplished in any order. The GameBuilder presents a guided user interface by, using step by step wizards. Dryer suggests that wizards are aptly suited for tasks that are either difficult or infrequently performed [1]. Most users of GameBuilder will have had no prior experience of GameBuilder. Furthermore, STEP expects most users of GameBuilder will use it infrequently. Thus the use of wizards is justified based on Dryer's criteria.

3. *WYSIWIG - like capability.* Almost all document preparation programs are WYSIWIG, in which the user sees at all times a representation similar to the final output. This display is advantageous because it reduces the number of design/testing cycles needed compared with non-WYSIWIG (e.g. LaTeX). GameBuilder will display a skinned output which represents what the player will see on his or her handheld screen when visiting an object.

4. *User Model.* According to Spolsky, the central tenet behind user interface design is that the user model (how the user thinks about the software) corresponds to how the program will behave [5]. An example is that the user thinks about windows on a desktop like pieces of paper on a desk. Clicking on a window in the background should bring it to the foreground, just like if a user grabbed a paper and put it on top. For GameBuilder, we attempt to use common well-known icons which immediately suggest the corresponding function. For example, we use pencil icons for editing and a notepad to suggest writing in a particular area.

2.2 GameBuilder Implementation

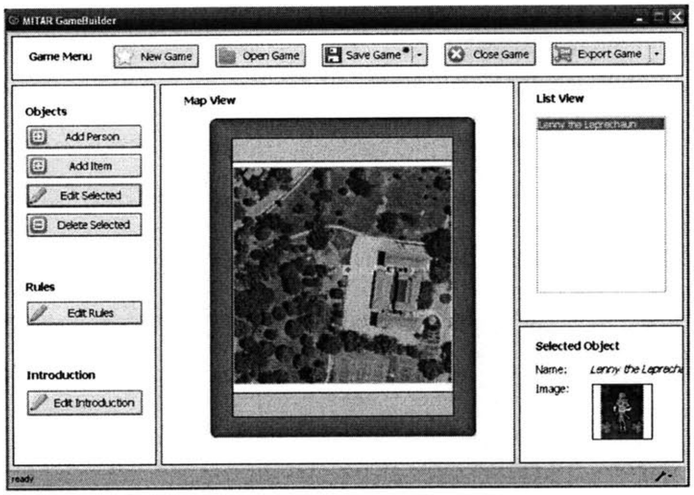
The alpha version of GameBuilder was created over a 6 month period from June 2008 to December 2008. GameBuilder was implemented in C# for Windows operating systems. A round of user interface mock ups were created using Microsoft Paint and then critiqued and revised. After the basic layout had been finalized, the author commenced programming. Originally, GameBuilder was not designed to support triggers and anti-triggers (Rules). During the late summer of 2008, a pre-alpha version of GameBuilder was used at a National Science Foundation funded Local Investigations of Natural Science (LIONS) at the Missouri Botanical Gardens. Students in the target age group developed games set at two local locations. After observing the session and games designed, researchers decided that allowing only games with objects and their descriptions would be too basic [3]. To faciliate more complex thinking, researchers requested that triggers and anti-triggers be included in future versions of GameBuilder.

2.3 GameBuilder Overview

In the next section, we present a comprehensive overview of the GameBuilder software as of September 2009. Additionally, we also contrast aspects of GameBuilder's user interface with corresponding aspects of Editor's user interface. Many features were added or modified by Nyls Braak who took over development from the author in January 2009. Figure 2-1 shows the main window of the GameBuilder after a game has been loaded. The main window is divided into 5 main sections: Game Menu, Command Menu, Map View, List View, and the Selected Object View.

2.3.1 Game Menu

The Game Menu contains 5 buttons, each with an icon and text. The Save Game and Export Game buttons have drop down menus to allow for non-default behavior such as "Save As" and "Export To". Compared to the Editor (Figure 2-2), GameBuilder required fewer menu options and larger buttons were able to be used. The expandable drop down menus of Save Game and Export Game allow additional options without forgoing a one click action. The icons for the Game Menu were chosen to follow Design Principle 4. The Open Game folder icon and Save Game disk icon are standard



Name: Lenny the Leprecha

Figure 2-1: GameBuilder Main Window

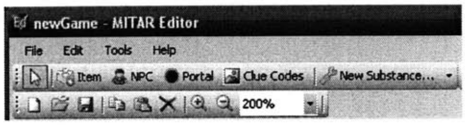


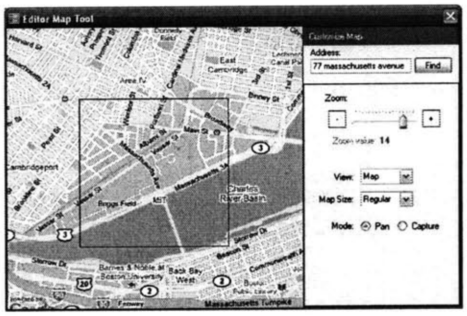
Figure 2-2: Editor Menu

computer icons while a star and a red x are used for New Game and Close Game respectively. For Export Game, a shopping cart is used to suggest checking out or taking a product (the designed game) and leaving.

In the Editor, Open, Save, and Close Game are options in the File menu item as well as in a shortcut toolbar. The Export Game feature is located under the Tools menu item.

The process of creating a new game differs significantly between Editor and GameBuilder. Because the Editor was designed for researchers, flexibility was an important consideration. The Editor contains a Google Maps tool which allows the user to 1.) take a picture of any place on earth and 2.) store the latitude and longitude coordinates (Figure 2-3).

Because, most child game design would occur in a classroom setting and the game location would be predetermined (near the workshop location), it was unnecessary to have the flexibility of the Editor's Map Tool. Instead, GameBuilder requires that a



Address: 77 massachusetts avenue

Zoom value: 14

View: Map

Map Size: Regular

Mode: ● Pan ○ Capture

Figure 2-3: Editor Map Tool

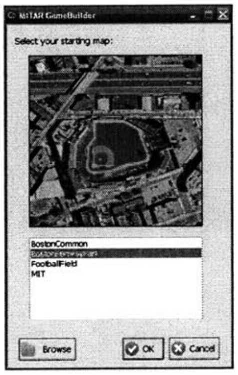


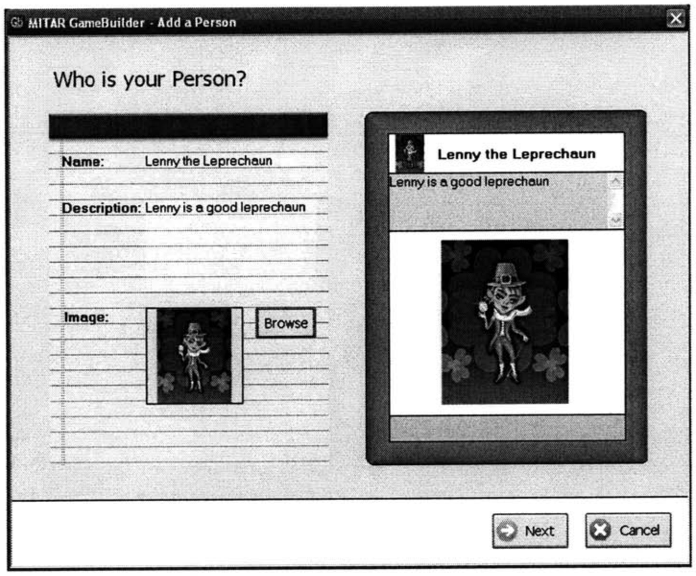
Figure 2-4: GameBuilder New Game Window

researcher or teacher first create a "template" game using the Editor to specify a map and coordinates. Figure 2-4 shows the GameBuilder's New Game window. The user has the choice of all templates which are found in the path or he or she can browse for another template. A large preview of the map of the template is also displayed.

2.3.2 Command Menu

The command menu of the GameBuilder contains buttons to Add/Edit/Delete objects, create triggers/anti-triggers (called Rules), and edit the Game Introduction. To follow Design Principle 2, addition and subtraction symbols are used as icons for adding and deleting objects respectively. A pencil icon is used to represent editing Objects, the Introduction, and Rules.

When a user clicks on either the Add Item or Add Person button in the Command Menu, a three step wizard appears which guides the user in creating their object


MITAR GameBuilder - Add a Person

Who is your Person?

Name: Lenny the Leprechaun

Description: Lenny is a good leprechaun

Image:

Lenny the Leprechaun

Lenny is a good leprechaun

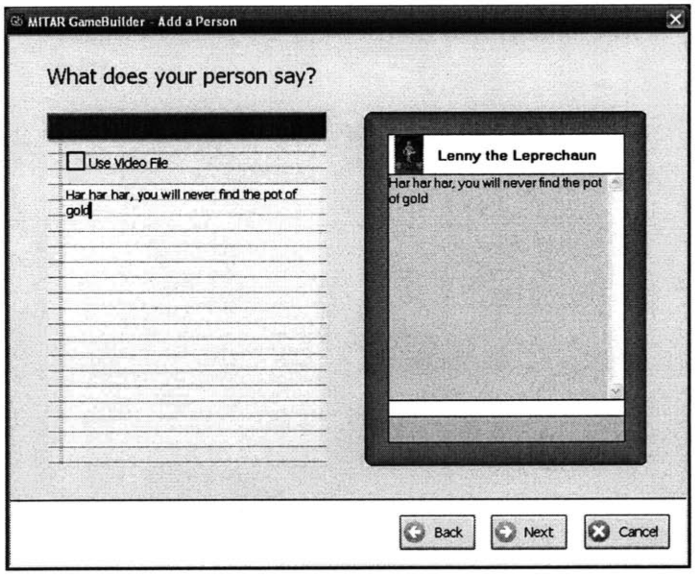
Figure 2-5: GameBuilder New Object Wizard Step 1

(Design Principle 2). Figure 2-5 shows the first step of the wizard, which allows users to name, describe, and choose a picture for their object.

The window is divided into two roughly equal halves. The right half represents how the object will appear to the player using a handheld. Whenever an edit is made to the boxes on the left, a corresponding change is immediately seen on the right. This WYSIWIG-like feature (Design Principle 3) helps users see the consequences of their actions. The notepad skin on the left half is used to immediately suggest editing the text (Design Principle 4). Clicking next brings the user to the Step 2 of the wizard (Figure 2-6).

Step 2 corresponds to adding an InfoPage in the Editor. While the Editor allows the user the choice of adding any number of InfoPages, GameBuilder only allows the user to include one InfoPage in an effort to streamline the design process. Furthermore, the word "InfoPage" is never used. Instead, the wizards simply asks "What will the person say". Clicking next brings the user to the Step 3 (Figure 2-7).

Step 3 asks the user to choose the color and shape of the icon and then place it onto the map. Within this step, GameBuilder further guides the user by presenting numbered steps (Design Principle 2). The player may select 1 of 6 different icons to



MITAR GameBuilder - Add a Person

What does your person say?

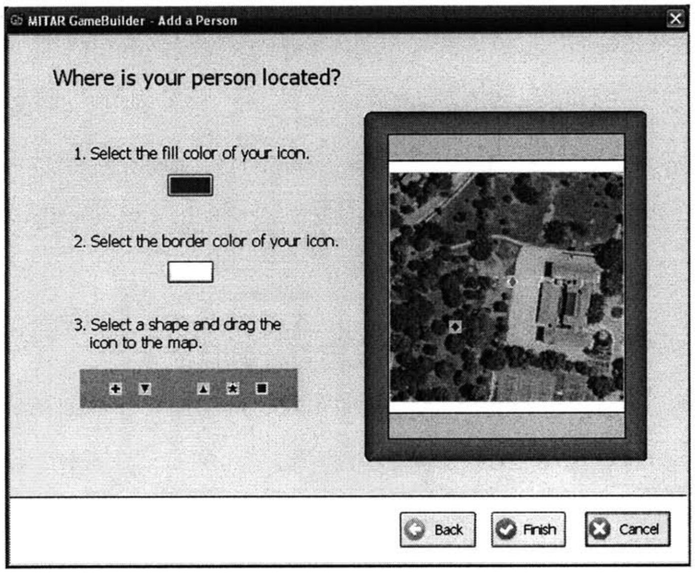
☐ Use Video File

Har har har, you will never find the pot of gold

Lenny the Leprechaun

Har har har, you will never find the pot of gold

Figure 2-6: GameBuilder New Object Wizard Step 2



MITAR GameBuilder - Add a Person

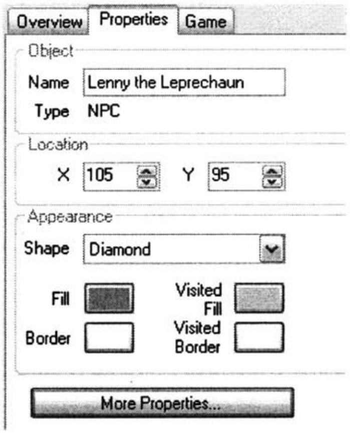
Where is your person located?

1. Select the fill color of your icon.

2. Select the border color of your icon.

3. Select a shape and drag the icon to the map.

Figure 2-7: GameBuilder New Object Wizard Step 3



Object
Name: Lenny the Leprechaun
Type: NPC

Location
X: 105
Y: 95

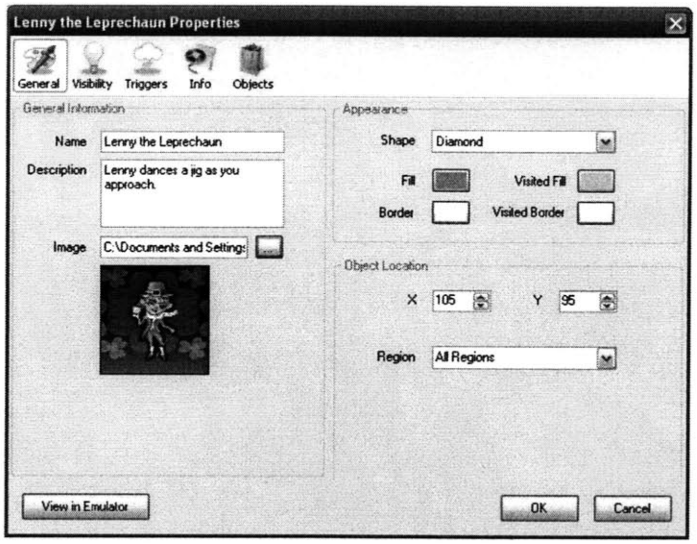
Appearance
Shape: Diamond
Fill
Visited Fill
Border
Visited Border

Figure 2-8: Editor Properties Panel

represent an object. The icon selection and icon placement are combined into one action. The user simply selects the icon he or she wants from the gray bin and drags it to the desired location. If he changes his mind, he simply drags another one onto the map and the icon originally chosen will snap back to the gray bin. In the Editor, a text menu is used to select the icon type. This is redundant since the actual word descriptions are irrelevant.

The Editor allows much more flexibility and choice when adding objects. First, the user must click on either an item or NPC icon (Figure 2-2). Then, he or she places the icon in the desired location on the Map. After this, the user is free to modify any property of the object. If the item has been selected, a Properties Panel to the right of the map will appear as shown in Figure 2-8. This panel allows quick modification of the name, icon color, icon type, and border color. The location of an item can be changed simply by dragging and dropping it to a new location on the map. If a GameBuilder user wishes to modify only the description of an object, he or she must still click through all 3 steps of the wizard. This is one disadvantage compared to the Editor.

If the user double clicks on the icon or clicks on the More Properties button in the Properties Panel, a Properties Window will appear as shown in Figure 2-9. The Properties Window allows editing of almost all aspects of the object except for the InfoPage.



General Information

Name: Lenny the Leprechaun

Description: Lenny dances a jig as you approach.

Image: C:\Documents and Setting:

Appearance

Shape: Diamond

Fill

Visited Fill

Border

Visited Border

Object Location

X: 105 Y: 95

Region: All Regions

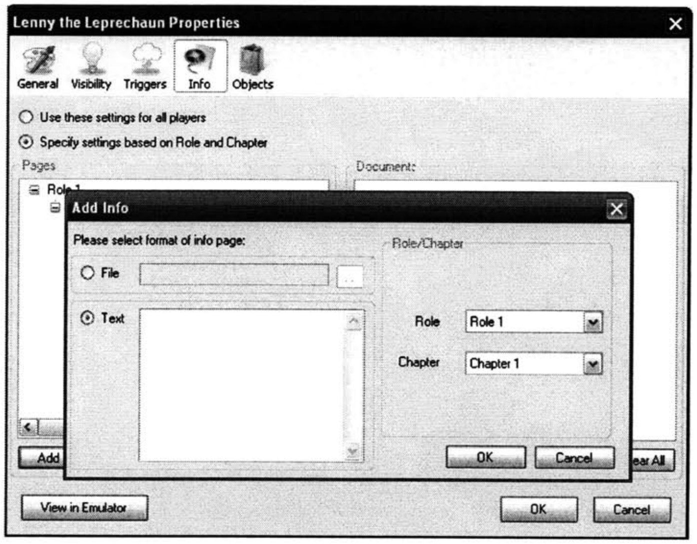
Figure 2-9: Editor Properties Window

Adding an InfoPage in the Editor is especially challenging in the Editor. It requires the user to click on the Info tab of the Properties Window and then click on the Add button to create a new page. This process is shown in Figure 2-10. At this point, 2 modal windows are open.

Adding an Object represents the biggest user interface difference between Editor and GameBuilder. While Editor treats the act of adding an object as simply placing a blank object onto the Map, GameBuilder requires the user to flesh out a description before creating the object. GameBuilder utilizes a strict linear path to guide the user. The Editor is flexible and allows the user to modify settings in any order. Because researchers believe that almost all objects created by users of the target age group can be represented with just one InfoPage, this was streamlined into the Object creation process.

Editing an Object in the GameBuilder uses the same wizard as Adding an Object, the only difference being that the fields are pre-populated. Thus, the same advantages and disadvantages are present when compared with using the Editor.

Deleting an Object is nearly the same in the GameBuilder and Editor. There is a Delete Item button in the Command menu and an "X" button in the Editor toolbar. Also, selecting an object in the map or list view and pressing the delete key also works



Lenny the Leprechaun Properties

○ Use these settings for all players

● Specify settings based on Role and Chapter

Add Info

Please select format of info page:

○ File

● Text

Role: Role 1

Chapter: Chapter 1

Figure 2-10: Editor Adding InfoPage

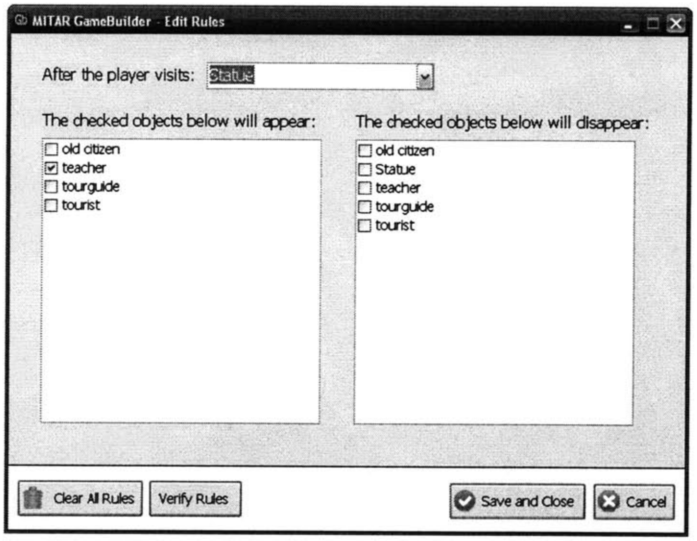
in both Editor and GameBuilder.

In GameBuilder, triggers and anti-triggers are referred to as Rules instead. We chose this name because we thought it be simpler to understand. The Rules window (Figure 2-11) can be accessed by clicking on the Edit Rules button in the Command Menu.

The Edit Rules window is designed so that a sentence can be read. For example, "After the player visits Statue, teacher will appear and old citizen will disappear". To change the object whose visit by the player causes the appearances and disappearances, the user must click on the dropdown menu. In order to change which objects will appear or disappear, the user clicks the checkboxes. The Clear All Rules will eliminate all triggering and anti-triggering for all objects. The Verify Rules will check that all objects will be visible at some point.

For the Editor, the interface for creating triggers and anti-triggers is accessed through a tab of the More Properties Window. The window is shown in Figure 2-12.

Overall, the interface is similar to that of the GameBuilder. However, in GameBuilder, rules are treated as a property of the entire game accessed through a top level menu button. In the Editor, the triggers and anti-triggers are treated as properties



After the player visits: Statue

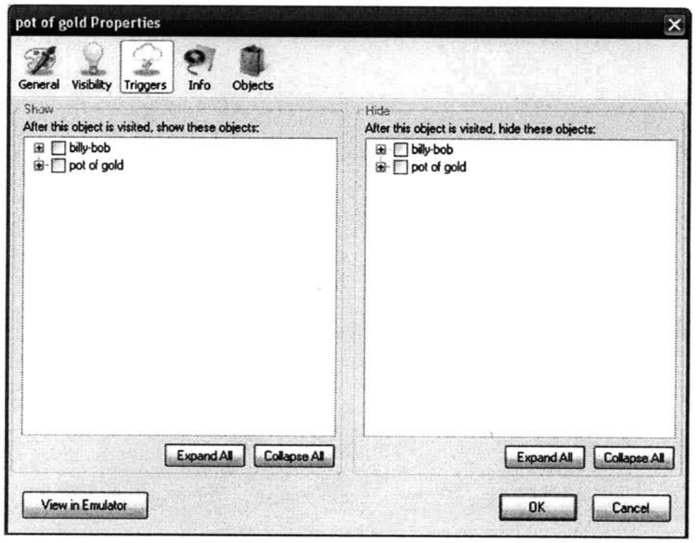
The checked objects below will appear:

- □ old citizen
- ☑ teacher
- □ tourguide
- □ tourist

The checked objects below will disappear:

- □ old citizen
- □ Statue
- □ teacher
- □ tourguide
- □ tourist

Figure 2-11: GameBuilder Edit Rules



Show

After this object is visited, show these objects:

- ⊞ □ billy-bob
- ⊞ □ pot of gold

Hide

After this object is visited, hide these objects:

- ⊞ □ billy-bob
- ⊞ □ pot of gold

Figure 2-12: Editor Triggers and Anti-Triggers

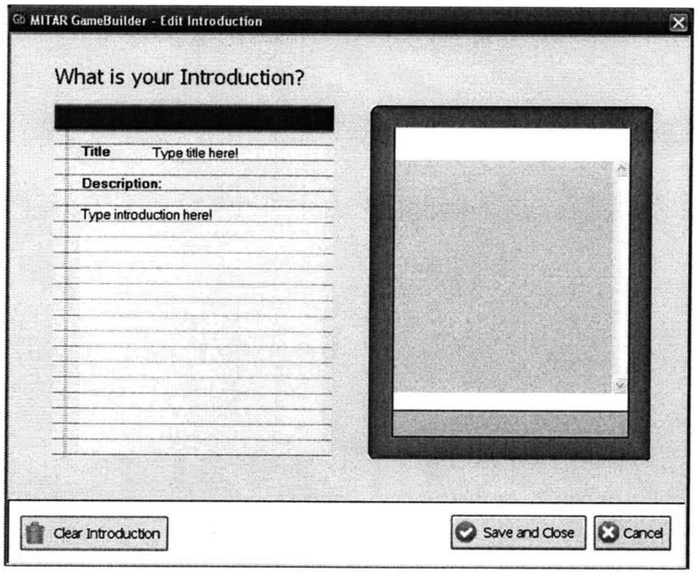
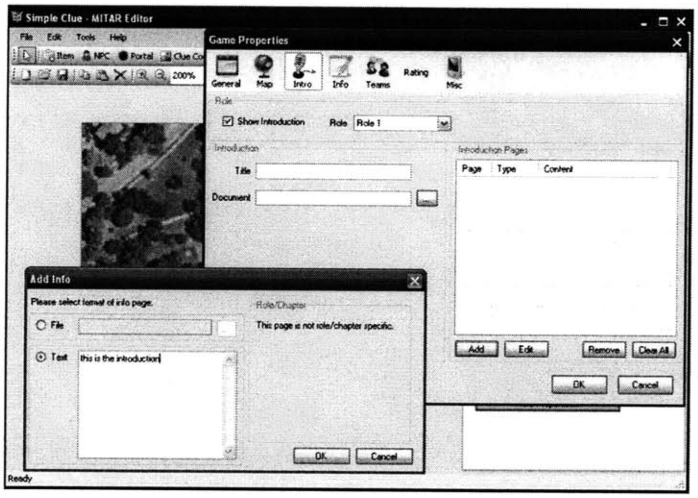


Figure 2-13: GameBuilder Edit Introduction Window

of the object itself. One advantage of the GameBuilder approach is that if the user first creates all the characters and then wants to go through and make the rules all at once, it is easy to change from character to character with the drop down menu. This requires only one action to switch between characters. In the Editor, this requires closing the existing trigger window, opening the properties window of another item and clicking the triggers tab (3 actions).

The last button of the Command Menu is the Edit Introduction button. The introduction consists of a game title as well as a page that the player sees when he or she first starts the game. In GameBuilder, the user simply clicks on Edit Introduction and a window shown in Figure 2-13 appears. The window is similar to the Add Object Wizard where the user edits the left half and the right half displays the output.

Figure 2-14 shows how a user would change the title and introduction of the Editor. The user would first click on the Game tab in the main window. They would then click on More Properties to bring up the Game Properties window. Then they would click on the Intro tab. There is the option to add a title and to add multiple introduction pages. Compared to the GameBuilder, the Editor's interface has more flexibility but is also more complex.



☑ Show Introduction Role: Role 1

Please select format of info page.

○ File

⊙ Text: this is the introduction

This page is not role/chapter specific.

Figure 2-14: Editor Edit Introduction Window

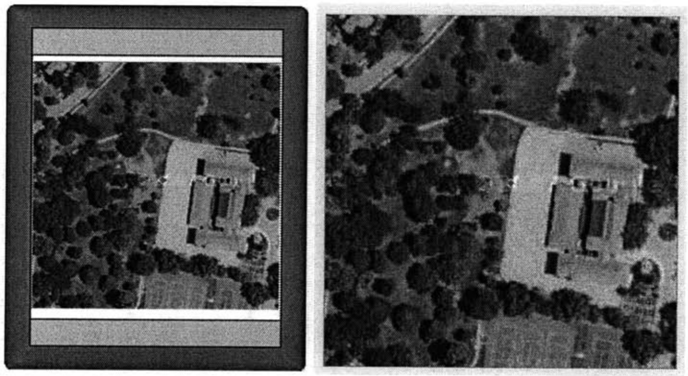
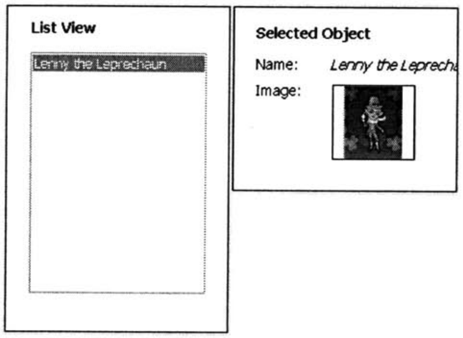


Figure 2-15: Map View in GameBuilder (left) and Editor (right)

2.3.3 Map View

Figure 2-15 shows both the GameBuilder's Map View and the Editor's Map. Both are very similar and display the location of the game and all current objects. The GameBuilder's Map View has a skin around it to simulate the handheld device. The user is able to move items by dragging and dropping their icons in the Map. Clicking on an object will highlight it. Double clicking on an object's icon in the Editor will open up the Object Properties Window. The same action in GameBuilder will launch the Edit Object Wizard.



List View

Lenny the Leprechaun

Selected Object

Name: *Lenny the Leprech*

Image:

Figure 2-16: GameBuilder List View and Selected Object

2.3.4 List View

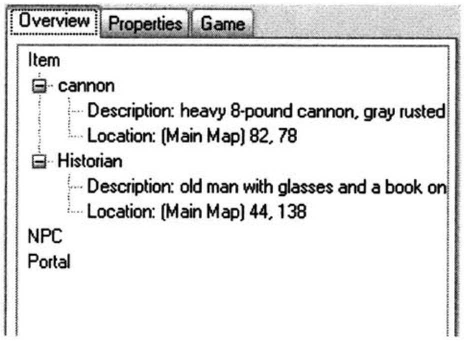
The GameBuilder also features a List View which serves as an alternative way to see the items/NPC's present in the game (Figure 2-16). The List View is synced to the Map. When the user selects an object's icon in the Map View, the corresponding entry in the List View will be highlighted and vice versa. Similarly, users can launch the Edit Object Wizard by double clicking on the object's name In the List View.

2.3.5 Selected Object

The Selected Object box, shown in Figure 2-16, is a small box that displays the name of the currently selected object. The only additional information the Selected Object box provides is a preview of the picture.

The Editor combines the functionalities of the List View and the Selected Object box into the Overview Panel shown in Figure 2-17.

The Overview panel separates objects into Items and NPC's and lists their names descriptions and locations. If the user double clicks on a particular object, the Object Properties Window will launch. However, unlike the GameBuilder, if the user selects an item on the map, the Overview panel will disappear and the Properties panel for that object will appear.



Item
- cannon
  - Description: heavy 8-pound cannon, gray rusted
  - Location: (Main Map) 82, 78
- Historian
  - Description: old man with glasses and a book on
  - Location: (Main Map) 44, 138

NPC

Portal

Figure 2-17: Editor Overview Panel

Chapter 3

Cambridge Science Festival GameBuilder Pilot Study

Researchers performed a pilot workshop of the GameBuilder software as part of the Cambridge Science Festival (CSF), on April 25, 2009. Researchers observed children using GameBuilder to design games and then playing the games.

3.1 Study Procedure

The pilot workshop was advertised as one event of the CSF. The CSF website contained an online form which allowed parents to enroll their child in the pilot workshop. Potential subjects were required to be between 11 and 14 (the target audience for GameBuilder software).

The pilot workshop took place on MIT's campus from 9 AM to 2 PM. A portion of the workshop took place in an electronic classroom and the rest of the workshop was conducted outside in MIT's Killian Court. A total of 10 subjects participated in the workshop. The workshop was divided into the following phases:

1. Introduction to Outdoor AR Games (inside)

2. Playing an Outdoor AR Game (outside)

3. Introduction to GameBuilder Software (inside)

4. Brainstorming Ideas for Good Games (Inside)

5. Designing the Game (inside)

6. Playing the Designed Game Outside (outside)

First, researchers gave a basic overview of Outdoor AR to the subjects. Researchers explained how GPS worked, what items and NPC's were and went over a specific game example. Next, the subjects went outside, were divided into pairs, and played a simple game about a leprechaun for 40 minutes. This reinforced what they had just learned about Outdoor AR in the classroom.

After subjects played the example game, they returned to the classroom. Researchers divided the subjects into 2 groups of 2 and 2 groups of 3. Each group received a Windows XP laptop with the latest release of GameBuilder at that time. Researchers used a projector to demonstrate how to perform basic tasks on the GameBuilder such as creating items and NPC's, writing descriptions, adding pictures, and using triggers. Subjects followed along on their laptops.

After the GameBuilder tutorial, researchers led a brainstorming session about potential game ideas. This helped subjects start thinking about what kinds of games would be interesting and fun to design and play. After the brainstorming, groups were given approximately an hour to discuss and design their games. Researchers walked from group to group answering questions from the subjects. While researchers had provided a collection of images for groups to use for their items and NPC's, almost all groups requested guidance on downloading pictures of their choosing from the Internet.

During a short lunch break, researchers copied every group's games onto a number of handhelds. Researchers then led subjects back outside to Killian Court and each group was given time to play their own game. Two groups finished playing their own game and played the games of other students. This period lasted for about 1 hour.

Finally, each subject filled out a survey about their experiences (Appendix C). This survey consisted of 5 quantitative questions about the difficulty of using GameBuilder, 4 quantitative questions about satisfaction with the designed game and 3 open response questions.

3.2 Observations

Because this was a pilot study, researchers did not have an observation checklist during either the design or playing game period. Therefore, the majority of collected data comes from the analysis of the designed games and survey results.

In general, there was a large maturity discrepancy between the subjects. One group had an older child who dominated the design, typing and using the mouse during design, and holding the handheld while playing. Because enrollment was voluntary, the subjects seemed to be from families of high socio-economic standing whose parents wanted them to learn.

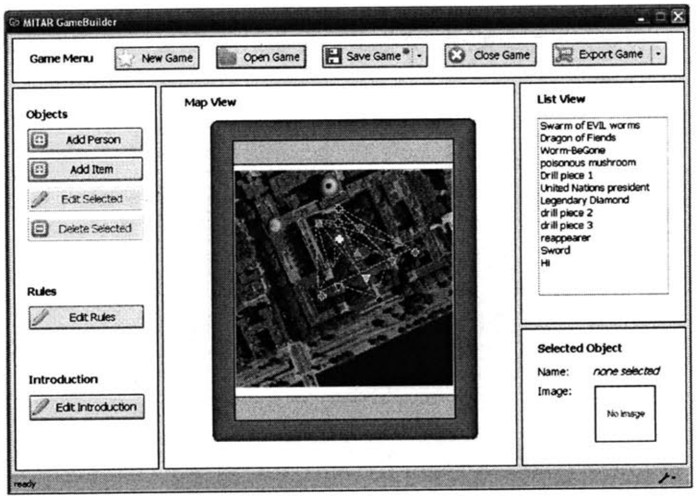
3.3 Game Analysis

Two of the four games designed are described below.

One very complex game featured a non-linear storyline. The player's task was to find the three drill bit pieces. Once these had been found, the player would win. It was possible for the player to meet dragons and poisons which could kill the player and make him lose. This game showed evidence of careful design and thinking through the effects of triggers and anti-triggers. Figure 3-1 shows this game loaded into GameBuilder.

Another group's game was about Halloween. The player traveled to different houses to trick or treat. However, there was no way to win. The player would always encounter an infected house where they would have their candy eaten by zombies.

Overall, all of the games were pretty well designed. They all had interesting storylines, rich items/NPC's and heavy use of triggers and anti-triggers. Most of them did not incorporate the MIT setting into the story. Instead, they simply pretended that they were somewhere else.



Swarm of EVIL worms
Dragon of Fiends
Worm-BeGone
poisonous mushroom
Drill piece 1
United Nations president
Legendary Diamond
drill piece 2
drill piece 3
reappearer
Sword
Hi

Name: *none selected*
Image:

Figure 3-1: Example CSF Game

3.4 Survey Results

The 3 open response questions yielded interesting feedback, some expected and some unexpected. The first question asked about a software feature that was difficult to use. Several subjects reported that the process of downloading pictures from the Internet was cumbersome. Researcher observations also supported this claim. In particular, subjects had a limited understanding of file/folder structure. After subjects had downloaded images, they would have a difficult time finding the download location using GameBuilder. This could be significantly improved in future versions with an image finder tool integrated into GameBuilder. Subjects could conduct a search and then immediately use the picture without worrying about the file path.

Another general problem was classification and organization of the image library. Researchers had chosen about 30 clipart pictures for subject use and put them in one folder. Image files were organized alphabetically rather than by concept and were not searchable by tags. This could be why all groups wanted to download their own files, because it was simply too difficult to search for a matching picture. Also, if the groups downloaded the image to the same directory as the pre-populated images, they often would not remember the image filename and would have a hard time finding the image they just downloaded. Again, an integrated image finder tool could eliminate these hassles.

For the second question, more than half of subjects commented that adding items and NPC's was very easy. One subject wrote: "When I started making a person, I didn't understand how to make a person. Then I got the hang of it and it became very easy." This fits with the purpose of GameBuilder. The wizard was designed specifically to allow for a smooth path to creating the items.

The third question asked for a functionality subjects wished they could do. One subject wrote: "I wish you could fix a mistake quickly, while you're outside. Then my glitch never would of[sic] happened“. This subject's group had created a relatively complex game with many triggers and anti-triggers but realized they had made a mistake once outside. This raises the issue of the natural disconnect between where the game is designed (usually in a classroom) and where it is played (outside). Researchers mitigate the effects of this disconnect by iteratively designing and going outside to test. Furthermore, adults are more capable than children of imagining what will happen and foreseeing consequences of certain actions.

There are two general approaches to remove this disconnect, roughly described as "designing outside" or "playing inside"'. To "design outside", Tiffany Wang developed the Remote Editor, an editor which worked on the handheld. Unfortunately, this solution is constrained by the lack of a keyboard and a mouse for the PDA. An evolution of this concept might be to utilize a laptop enabled with GPS. STEP has current solutions and future solutions to enable "playing inside". The GameBuilder and Editor both can interface to an Emulator. Robert Falconi is working on a FlowView tab for the Editor which allows users to skip through the game. Finally, a more immersive approach might be to synchronize the editor with Google Street View to virtually debug the game in the classroom.

There were many other feature requests, some implemented but not available for GameBuilder and some which not implemented for AR. One subject wanted a "hotspot" where people could trigger an event on visiting. Currently, this feature cannot be implemented using the GameBuilder and can be simulated as visiting an item/NPC with a large trigger radius with the Editor. Another subject wanted something to happen whenever the player left a certain area. These requests expose one weakness of the Outdoor AR paradigm in that it is focused on meeting NPC's and finding items. It is not designed to handle events such as "a lightning bolt comes out of the sky", which then cause other things to happen in the game. Another subject wanted "more complicated rules w boolean operators, if visited Person A and Person B then show Object A." Other requests included an inventory and having hidden items that you bump into.

Chapter 4

Comparative Usability Study: GameBuilder vs. Editor

4.1 Study Design

Researchers performed a comparative software usability study of GameBuilder and Editor. The study was designed to answer the following questions:

- Which software is easier to learn, GameBuilder, Editor, or inconclusive?

- If GameBuilder is easier to learn, is it because of the user interface changes implemented in Chapter 2? Is it due to something else?

In addition, researchers recorded general observations which could help answer:

- What are the most common mistakes subjects make using either GameBuilder or Editor?

- What additional features do subjects want when designing games?

4.2 Study Procedure

4.2.1 Subject Recruitment

Researchers contacted potential subjects through email advertisements. Potential subjects were required to be between 11 and 14 (the target audience for GameBuilder software) and use computers more than 3 hours per week. Researchers gave a $10 gift certificate as compensation.

All experiments were performed between September 20, 2009 and November 18, 2009. A total of 14 subjects completed the experiments. One subject started the experiment but did not finish due to time constraints. 13 studies were conducted on MIT campus in the TEP laboratory. One subject was recruited from a local middle school and the researcher conducted the experiment on site.

Experiments were conducted on a Windows XP environment with the latest release of GameBuilder and Editor as of September 20, 2009. For features requiring Internet access, the Firefox browser was used. For the one on site study conducted at a middle school, Internet access was not available and Step 13 of the walkthrough was skipped. Also, the experiment was conducted using a laptop rather than a desktop.

4.2.2 Overall Study Procedure

The researcher assigned each subject to test either the GameBuilder or Editor software. The researcher attempted to follow consistent procedures for all 14 experiments. Inconsistencies and their possible effect on data analysis and conclusions are documented.

Experiments consisted of four phases, the introduction, walkthrough, game design, and survey. The total experiment time ranged from 45 minutes to roughly 2 hours. Variance in the walkthrough completion time accounted for almost all of the variance in total experiment time.

4.2.3 Introduction

During the introduction, the researcher informed the subject the purpose of the study was to test game design software developed at the lab targeted for children ages 11 to 14. The researcher then briefly demonstrated an Augmented Reality game running on a handheld in manual (non-GPS) mode.

The researcher outlined the parts of the experiment for the subject (walkthrough, game design, and survey) and explained that the experiment's purpose was to test the software rather than to test the subject.

The subject was informed that if he or she got stuck, he should pretend the researcher was not present and attempt to fix the problem alone. However, if the subject was seriously lost, the researcher would step in and assist him. Unfortunately, the researcher did not inform the first two subjects of this. This could affect results by showing that subjects asked for help more when not told to first try to fix problem alone.

Finally the researcher asked the subject to talk out loud during the walkthrough and the game design period. The researcher demonstrated by describing his intermediate actions in visiting google.com ("I am now clicking on Firefox", "I am now typing in the address bar w, w, w, dot, g, .". The purpose of this request was to give the researcher insight into the subject's thought process.

4.2.4 Walkthrough

During the walkthrough period, the subject was given a walkthrough packet describing how to use either the GameBuilder or Editor. The subject worked through the packet step by step.

Two walkthroughs, one for GameBuilder and one for Editor, were developed for the study. Both versions covered the same features in the same order in the context of having the player find a leprechaun's Pot of Gold. The first four steps ask the subject to provide content describing the leprechaun and Pot of Gold. The next 15 steps each cover features including:

- adding, moving an item
- changing description, InfoPages, and pictures
- downloading images from the Internet to use for items
- causing items to appear and disappear (triggers, anti-triggers)

The specific instructions to implement the features differed between the two walkthroughs because the software user interface differed.

The GameBuilder walkthrough was created first and then a corresponding Editor walkthrough was created. Thus the ordering of feature in the walkthrough was optimized for GameBuilder. Due to differences in the user interfaces, a feature ordering optimized for GameBuilder would not necessarily be optimal for Editor. However, maintaining the same feature ordering is an important variable to control. This could have an effect on our results by reducing the satisfaction and mastery associated with using Editor. For future studies, a better approach would be to analyze both pieces of software and compromise on a feature ordering.

While the subject completed the walkthrough, the researcher made observations about behavior and progress. The researcher recorded completion time (the start and end time of each step), distress indicators whether the subject 1.) went down the wrong path, 2.) had a long period of inactivity, 3.) asked for help, and any interesting behaviors which occurred. An example might be that the student clicked "Save" when the instructions asked him to use "Save As" or if the student performed the wrong action and moved on.

If the subject asked a question, the researcher would first respond by asking them to try a few things or to reread the walkthrough. If the researcher believed the student was stuck either by continuing down the wrong path or having a long period without progress, the researcher would give hints until the subject was back on track. Unfortunately, no consistent criterion for intervention was developed. This adds noise to our completion time data but should not have a directional skew towards either software.

Throughout the walkthrough, if the subject stopped talking out loud, the researcher would encourage him or her by saying "What are you doing now?", or "Can you describe what you are doing?".

4.2.5 Game Design

After the walkthrough period, the researcher informed the subject that the game design period would begin. Each subject was asked to use what he or she had learned in the walkthrough to design his own game.

First, the subject was provided with a brainstorming worksheet (Appendix B). The worksheet asked the subject to select four possible locations (Fenway Park, Gillette Stadium, Boston Common, and MIT) for the game and then write a rough outline of the game in a blank space given. Factors the subject was told to consider were:

- What do you want the player to learn when playing your game?
- What characters or items do you want in the game?
- What is the goal of the game?
- Will items/characters appear or disappear?

For the first two experiments, Gillette Stadium and Boston Common were not options and instead Central Park, New York City and Lake Michigan Lake Shore, Chicago were provided. However, because researchers decided Boston area subjects would not be familiar with these locations, they were replaced. Furthermore, Gillette Stadium and Fenway Park may be more appealing to males, while Boston Common and MIT are gender neutral. For future experiments, we should improve the balance of locations by making all gender neutral or also including locations that may have more appeal to female subjects.

The researcher gave the subject up to 5 minutes to work on the brainstorming worksheet. The same worksheet was used for both GameBuilder and Editor experiments. Some subjects wrote down one or two sentences about their game while others filled the entire box.

After brainstorming, the subject was given 25 minutes to build their own game. Before beginning, the researcher reminded the subject to reference the walkthrough in case he or she forgot how to use certain features. After 20 minutes, the researcher informed the subject 5 minutes were left. At 25 minutes, the researcher stopped the subject. Some subjects finished designing their game before the full 25 minutes passed. During game design, the researcher responded to help requests in the same way as before. The researcher would first direct the subject to the walkthrough or to "try to figure it out yourself". If the subject was stuck for a significant time, the researcher would intervene.

Some subjects also asked about available features not explored in the walkthrough or features not implemented in the software. The researcher would respond either "Yes, the feature is available but please try not to use it", or "No, the feature is not available".

Finally, subjects sometimes asked questions about the game content, such as "Should I add Lenny the Leprechuan to my game?" The researcher responded that the game content was entirely up to the subject.

During the game design period, the researcher recorded observations related to the following questions:

- Did the subject look back in the walkthrough for help? What did he need help with?
- Did the subject start down the wrong path? What did he actually want to do?
- Did the subject get confused and ask for help? Which feature was it?
- What general features did the subject spend the most time in? What general features did the subject spend the least time in?

4.2.6 Survey Period

After the game design period, the subject was given a survey designed to evaluate their experience (Appendix C). Two sets of quantitative questions and three open

Table 4.1: Types of Data

Data Type	*Phase*	*Quant/Qual*
Completion time data for each step	Walkthrough	Quant
Distress indicators (Asking for help, going down wrong path, or long pause with no action) for each step	Walkthrough	Qual
Observations each step	Game Design	Qual
Game design observations	Game Design	Quant
Game metrics (Number of characters created, images used, triggers/anti-triggers)	Game Design	Qual
Satisfaction Responses	Survey	Quant
Open Response Questions	Survey	Qual

response questions comprised the survey.

4.3 Data Analysis Method

A variety of data was collected as described in the procedure below and is shown in Table 4.1.

4.3.1 Which software is easier to learn, GameBuilder, Editor, or inconclusive?

If the average total walkthrough completion time for GameBuilder was significantly less than that for the Editor, this might suggest that GameBuilder was easier to learn. An alternate unlikely explanation would be that the walkthrough for GameBuilder was written more clearly.

Exploring completion time differences for specific features between the populations would also be interesting. For example, if the average completion time for the trigger feature differed significantly between the populations, this could suggest that one software's user interface is better adapted for that particular feature.

The number of total times the subject indicated distress could also indicate that one piece of software was easier to learn than the other. Similarly, the number of distress indicators for particular features can also be explored.

Qualitative observations made by the researchers could also suggest differences between the software. For example if during the game design phase, subjects using Editor repeatedly referred back to the walkthrough for help with one feature, this would be useful.

Finally, survey data could suggest differences between each software's ease of use. Two open response questions address what features subjects thought were the easiest and most difficult for each piece of software.

4.3.2 If GameBuilder is easier to learn, is it because of the user interface changes implemented in Chapter 2? Is it due to something else?

If results suggest that GameBuilder is easier to learn, we must evaluate whether this advantage is caused directly by the design principles researchers chose when designing GameBuilder. This can be evaluated by looking at completion times and distress rates for particular tasks. If the user interface for performing the task was changed specifically due to our design principles, then we did find support that our design principles increased learnability. Researchers can also observe what particular mistakes are made in the Editor which are not made in the GameBuilder.

4.3.3 What are the most common mistakes subjects make using either GameBuilder or Editor?

During the walkthrough, a high number of distress indicators over the subject population would indicate a particularly challenging feature. Qualitative observations could explain why these features were difficult such as what questions subjects asked or in what wrong path subjects pursued.

During the game design, as opposed to during the walkthrough, the subject is free to choose what features to use or not use. Thus, observing subjects either avoiding, having difficulty with (looking back through the walkthrough, asking for help), or incorrectly using certain features would indicate those were challenging.

Finally, the survey has an open response question asking which feature was the most difficult. These responses could also provide insight.

4.3.4 What additional features do subjects want when designing games?

The walkthrough presented the same set of features to subjects, independent of whether they used the Editor or GameBuilder. However, these features comprise only a small subset of the Editor's features. For the most part, subjects were discouraged during game design from using features outside this set.

A survey open response question asks the subject what additional features would be desired. These features would need to be added to GameBuilder but could possibly already be implemented in Editor.

In addition, observations of questions subjects asked during the walkthrough or game design could also suggest desired additional features.

4.4 Results

Due to the small sample sizes, 6 for the GameBuilder group and 8 for the Editor group, it will be difficult to conduct statistical tests of sufficient power. However, distributions of quantitative data are shown to suggest possible relationships.

4.4.1 Which software is easier to learn, GameBuilder, Editor, or inconclusive?

To determine which software was easier to learn, we looked at completion time, distress indicators, survey data.

Figure 4-1 shows the distribution of completion times for the GameBuilder and Editor population. For the final completion time metric, we only included steps 5-12 and 14-19. Steps 1 through 4 asked the subject to provide content and did not involve

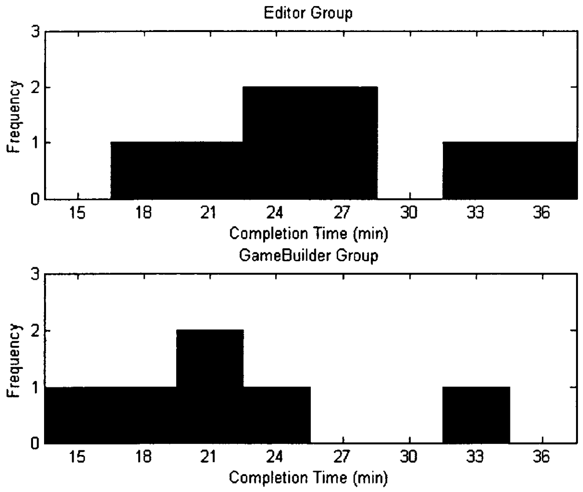


Figure 4-1: Total Completion Time

the software design. Step 13 involved using Firefox to download pictures. One subject did not perform this step and it is not relevant to the software design.

From Figure 4-1, a moderate difference can be seen between the completion time distributions of the Editor and GameBuilder populations. If we treat the 33 minute point as an outlier, this difference becomes very strong. This lends support to our hypothesis that GameBuilder is easier to learn because subjects take a shorter time to work through the walkthrough.

Figure 4-2 shows the distribution of the total number of times subjects were in distress during the walkthrough. As can be seen, for the Editor group, 3 subjects were in distress more than 10 times during the walkthrough while none were for the GameBuilder group. Furthermore, for the GameBuilder group, one subject was in distress only 1 time during the walkthrough. This suggests that overall subjects using the GameBuilder experienced less distress during the walkthrough.

Next, we looked at the first five survey questions regarding difficulty. The first question asked about the overall difficulty of the software while the other questions asked about the difficulty of performing specific tasks. The lower the score, the easier the feature was to perform. We built a metric by adding the responses from all 5 questions together. Figure 4-3 shows the histogram for both populations.

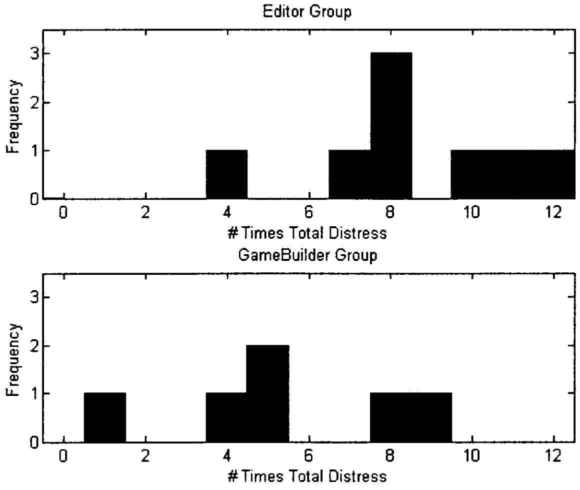


Figure 4-2: Total Distress

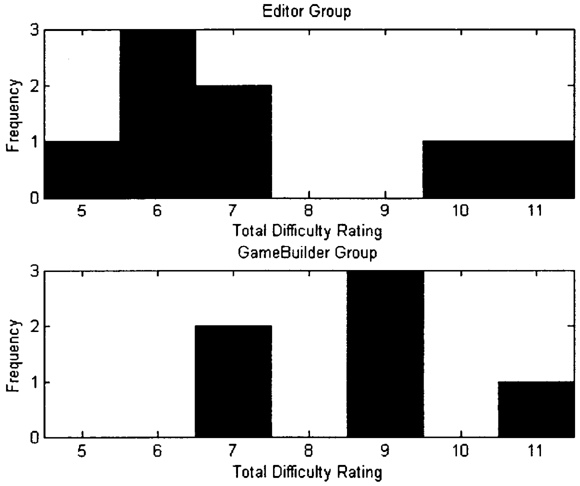


Figure 4-3: Total Difficulty Rating

There seems to be no clear difference between either population group. However, several factors could explain this. Subjects who participated in the experiment are most likely high achievers who excel relative to their peers. Regardless of a task's difficulty, subjects might write down that it was easy. Also, it is much easier to use a relative scale and compare the difficulty of performing two tasks rather than assign an absolute difficulty to a task. Because each subject only used and evaluated one software, there is no basis for comparison. It makes sense that there is no information to be gained from the difficulty survey.

During the game design period, multiple subjects in the Editor group referred back to the walkthrough in order to add triggers and anti-triggers as well as to add InfoPages. We did not see any such references with the GameBuilder group.

4.4.2 If GameBuilder is easier to learn, is it because of the user interface changes implemented in Chapter 2? Is it due to something else?

We looked at differences between the completion times for independent steps. The mean and median of each step's completion time for the GameBuilder and Editor populations were compared. The median was chosen because it is less sensitive to outliers. Three steps which showed a significant difference were Step 9, Step 15, and Step 14. Figure 4-4 and 4-5 show the histograms for Step 9 and Step 15 respectively.

As can be seen, the average completion time for Step 9 is much higher for the Editor group than for the GameBuilder group. If we remove an outlier in the GameBuilder group at 6 minutes, this difference becomes even clearer.

At first glance, there seems to be only a moderate difference in Step 15 completion time between the two populations. However, the same subject who took 6 minutes in Step 9 also took 5 minutes in Step 15. If we exclude this subject, the histograms strongly suggest subjects took longer on average in Step 15 while using the Editor.

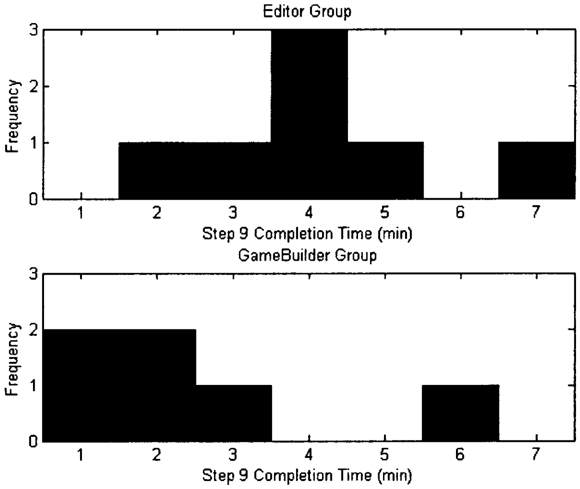


Figure 4-4: Step 9 Completion Time

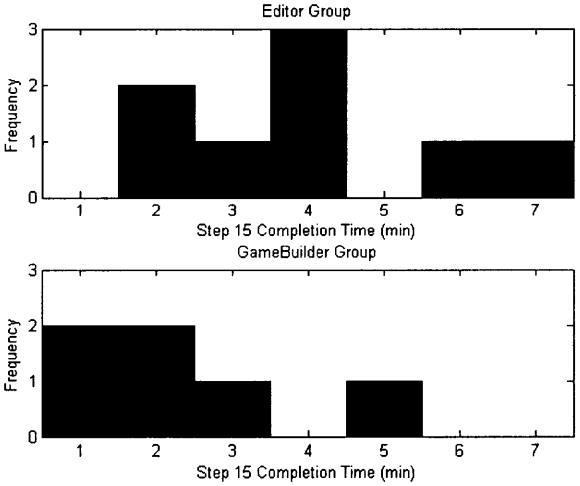


Figure 4-5: Step 15 Completion Time

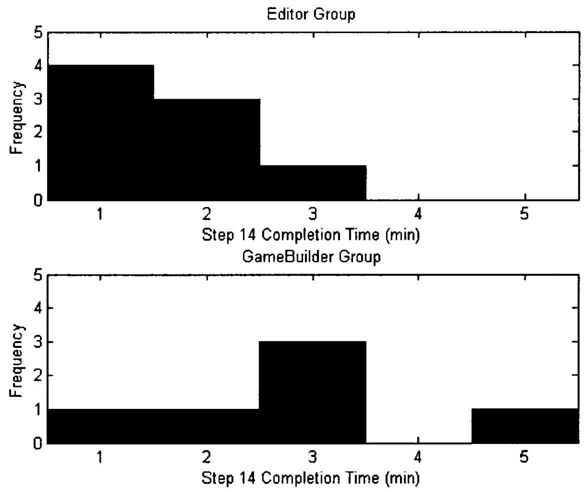


Figure 4-6: Step 14 Completion Time

Step 9 and 15 are particularly interesting because they correspond to adding an InfoPage and changing an InfoPage respectively. In the Editor, this feature requires several clicks to use, with three windows open simultaneously. The GameBuilder was designed specifically to integrate this into the Item/NPC wizard. Thus, it is encouraging to observe a decrease in completion time corresponding to the user interface change.

We observed many problems during the Editor walkthrough with Step 9 consistent with our previous analysis (Section I). Many subjects became confused because there were two sets of windows (the Pages and Documents windows), each with their own Add buttons. Secondly, when the subject tried to add an InfoPage, it would default to File first rather than Text. The large number of actions required to add an InfoPage offered many opportunities for users to get lost or confused.

Step 14, the step associated with creating an item, adding a name, description, and image, showed that the median completion time for the GameBuilder population was significantly higher than that of the Editor population, which does not support our hypothesis. Figure 4-6 shows the histogram.

This relationship does not seem particularly strong if the outlier at 5 minutes is removed. Furthermore, there is no hypothesis why this feature could be more easily performed with Editor than with GameBuilder. Thus, we consider it an accidental relationship.

4.4.3 What are the most common mistakes subjects make using either GameBuilder or Editor?

Using triggers and anti-triggers incorrectly was a common mistake for both populations. 2 out of the 8 Editor subjects and 2 out of 6 GameBuilder subjects used triggers/anti-triggers incorrectly. 1 GameBuilder subject did not attempt to use triggers or anti-triggers at all. The user interface for adding triggers and anti-triggers does not differ significantly between the Editor and GameBuilder. An example of a typical mistake could be: A user has three characters A, B, and C. After the player visits A, B will appear and A will disappear; after the player visits B, C will appear and B will disappear. The user might accidentally put a trigger from A to C or an anti-trigger from A to C.

An emulator has been developed for AR Games which allows users to walk through the game while designing. This emulator was not introduced during the experiment and might have helped subjects understand the temporal consequences of adding triggers and anti-triggers. In general, it is difficult for subjects to accurately create a 3-D story (time and location) when only given a 2-D surface (location). Nyls Braak also added another tool, a button which when clicked tells if all the items can be visited. This avoids the problem where there is a cycle of triggers which means nobody shows up originally. This tool was available but also not introduced. In addition to the general purpose emulator and the visibility checker, perhaps another tool designed solely to help subjects walk through the triggers and anti-triggers could be integrated into both the GameBuilder and Editor.

One of the most common mistakes which Editor users made was double clicking on the map when attempting to double click on an icon. This would bring up a Game Properties window which looked similar to the More Properties window which edited an object. Because the More Properties Window closely resembled the Game

Properties window, the user would often not realize he had made a mistake and become confused. Because there are other ways to access Game Properties, this research suggests that the double click access of Game Properties should be removed.

4.4.4 What additional features do subjects want when designing games?

In one particular instance, a subject wanted an NPC to say different things to the player depending on what the player had visited/achieved. While this feature is not supported, he was able to clumsily replicate it by creating otherwise identical NPC's which appeared and disappeared based on triggers. Supporting this natively would be a useful future feature.

The last open response of the survey asked subjects to suggest something they wish the software could do. The responses varied widely. For the GameBuilder population, most of the responses asked for features not currently implemented and outside the current AR Game framework. These included, "carrying items and using them", "designing background pictures for InfoPages and Description", and having "NPC's and items move around". One subject also asked for a spell checker.

The Editor group mostly suggested features that were not introduced in the walkthrough but are currently within the AR Game framework. Unfortunately, these features are not supported by the GameBuilder. These included:

1. "have different maps and travel between them" (portals and regions implement this)

2. "say different things at different times to different people" (roles and chapters)

3. "cannot do something until after you have been somewhere" (clue codes)

4. "animated NPC's" (video)

This feedback suggests that 11-14 year olds desire to create advanced interactive games. This implies that limited features enforced by GameBuilder can be detrimental. An alternative might be to slowly allow features to be introduced, or added onto the GameBuilder as users become more and more advanced.

4.5 Discussion

The experiment provided support for our hypothesis that the GameBuilder would be easier to learn compared to the Editor for the target age group. However, due to the low number of samples, additional research should be performed.

Specifically, we found strong support that the GameBuilder's streamlined InfoPage interface allowed subjects to learn the software more quickly than using the Editor's corresponding interface. This was a specific design choice and it is encouraging to see results which support this decision.

The conclusions drawn from our current experiment only apply when users are taught the software in a structured manner. While in many cases, teachers first give students a tutorial, it is also possible teachers might provide minimal support or users will learn outside a formal workshop setting. It is common for adults that have bought a piece of software or a device to not read the instruction manual. Nonetheless, products are designed such that the learning how to use the most common features is intuitive and obvious. Future research should compare how easily subjects can learn each software in an unstructured environment.

We hypothesize that in these cases, GameBuilder will show a marked advantage over the Editor. Due to the reduced feature set, there are fewer options users can choose from and fewer mistakes to make. In a structured teaching environment, users are told which options to choose. In an unstructured environment, users make informed guesses based on prior software experience or user interface guidance. The reduced feature set could help prevent costly errors.

Originally, we had planned to compare the quality of the games created during the free design period. We hypothesized that the better the game designed with a piece of software was, the better the software was. We did not find any significant difference between the games designed by the Editor group and the GameBuilder group using our complexity metrics or observations. Upon further reflection, this result makes sense. First of all, the brainstorming worksheet ensured that the game was designed before the subject started using the software. The subject would then be mostly constrained by the design and simply implement the design he wrote. Thus, the software had little impact on the complexity of the game.

Secondly, completing the walkthrough necessarily meant that the subject had a basic understanding of how to use all features in the walkthrough. While it may be true that the Editor is harder to learn than GameBuilder, subjects on average also spent more time with the Editor and should have ultimately achieved the same level of mastery. Thus, we did not and should not have expected to see any game quality differences between the two populations during the game design period.

Overall, it is important to note the huge variety in the complexity of the brainstormed games. Some subjects designed simple games with two characters while others had complex storylines with up to 16 triggers and anti-triggers. We did not find any significant correlations between the complexity of the game and the mastery of the software. In order to stimulate users to create exciting and interesting games, the first and most important task must be to teach them to be creative and good story-tellers. The technical know-how of using the software should have a secondary emphasis, a means to an end rather than the end itself.

Our experiments have shown that the GameBuilder is easier to learn in a structured manner than the Editor. We have not yet tested the relative ease of learning in an unstructured manner but hypothesize the GameBuilder will do even better. Unfortunately, we found from survey questions that users of the target age group can be advanced thinkers who would like to use complex features. The GameBuilder pares down their options, a high cost for the improved ease of learning. In future versions of GameBuilder, we should try to maintain ease of learning while still making advanced features available. This is discussed more in the Conclusion.

Chapter 5

Conclusion

Over the course of his thesis, the author designed and implemented GameBuilder, conducted a pilot study during the Cambridge Science Festival, and conducted a comparative usability study between GameBuilder and Editor.

GameBuilder was designed following 4 design principles in order to increase learnability as much as possible. In a time constrained environment, this would increase the time spent designing games and lessen the time spent learning how to build games.

During the pilot study during the Cambridge Science Festival, user feedback and researcher observations suggest that the streamlined add object interface worked well. Users for the most part designed complex and richly described games. Many of the users desired more complex features.

During the comparative user study, we found evidence that the time for performing a structured walkthrough was shorter overall and specifically for adding InfoPages with GameBuilder. This seems to suggest that GameBuilder is able to be learned more quickly in a structured environment. We hypothesized that in a free session, users would be able to learn GameBuilder even more quickly than Editor.

Originally, we planned to use game quality during free period to measure the learnability of the software. We expected that users who used a more learnable software would develop better games. We found that there was no correlation between game quality metrics and software in the comparative user study due to two possible reasons. First, users spent longer times and possibly received more guidance for the Editor walkthrough than the GameBuilder walkthrough. It is possible that after completing the walkthrough, both populations were equally capable of designing games even though it took one population longer to learn. Secondly, learnability plays only a supporting role in game quality, with creativity playing a dominant role. We found that the games developed at CSF were across the board richer and better designed than those designed during the comparative user study. We attribute the CSF advantage to a prior brainstorming session led by researchers, group interaction to generate more ideas, a longer time to work on the game (1 hour vs. 25 minutes), and experience playing a game before attempting to design one. Researchers should use the methods listed above to encourage children to become more creative during workshops.

In both CSF and comparative user study, students had a difficult time using triggers and anti-triggers correctly. As explained in Chapter 3, causes for this include not debugging and not teaching children to use available visualization tools. Researchers should teach children how to use these tools to help avoid mistakes when designing. Future innovations to remove the design/play spatial disconnect should also be integrated into GameBuilder.

Few of the developed games used location as a core element of the game. One participant designed a game about baseball at Fenway Park while another met Tom Brady at Gillette Stadium. However, only one of these games used a specific location within the map in a manner which made sense (the player had to pass through the ticket booth at Fenway Park). All games designed for the CSF were played at MIT. However, designers completely ignored the MIT setting and overlaid their own fantastical setting (zombie house, dragon's lair). If STEP's specific goal is to focus on creating games which utilize aspects of the location (sights, sounds, smells) to further the game goal, researchers must focus the children more on the location. A good exercise might be to have a brainstorming session outside where the group walks around and point out interesting locations which might fit a story.

Finally, one critical tradeoff of increasing learnability of GameBuilder was reducing the supported feature set. During design, STEP did anticipate that a game with only objects and descriptions would be too simplistic and added triggers and anti-triggers. However, user comments suggest that even more advanced functionalities were desired, some of which are not even supported by the full Editor. A user interface which is simple at first glance, but expandable beneath the surface would be able to accomplish learnability without reducing feature set. One exciting development is the integration of StarLogo language into Outdoor AR. One can imagine advanced users writing their own code fragments to implement all kinds of new functionalities.

In conclusion, the GameBuilder is currently in use and is serving its purpose of providing a quickly learnable game design tool. Extensive user testing conducted by the author and fellow researchers revealed many insights into AR software and game design. Based on this research, the STEP team is hard at work improving the Outdoor AR platform for the future.

Bibliography

[1] Dryer, C.. "Wizards, guides, and beyond: rational and empirical methods for selecting optimal intelligent user interface agents" Proceedings of the 2nd international conference on Intelligent user interfaces 1997

[2] Klopfer, E. and Squire, K.. "Environmental Detectives - The Development of an Augmented Reality Platform for Environmental Simulations." Educational Technology Research and Development 2005

[3] Klopfer, E. and Sheldon, J., "Augmented Your Own Reality Student Authoring of Science-Based Augmented Reality Games" 2009 (in press) New Directions in Youth Development: Theory, Practice, and Research

[4] Nielsen, J.. "Usability 101: Introduction to Usability." 23 August 2003. useit.com. Retrieved 10 January 2009. http://www.useit.com/alertbox/20030825.html.

[5] Spolsky, J.. User Interface Design for Programmers APRESS, 2001

Appendix A

Evaluation Rubric

MITAR Software Study

Date: ________________
Subject Name: ____________________
Software Being Tested (Circle One): GameBuilder / Editor

Pre-Sit Down:

1. Has parent signed Parent consent form? __________
2. Has subject signed Subject assent form? __________
3. Is software ready? __________
4. Is clipboard ready with new rubric? __________

Introduction:

1. Introduction – Hi, my name is Chuan Zhang, I work at the MIT Teacher Education Lab.
2. Purpose – We are here to test a piece of software. The software is used for creating MITAR games. AR stands for Augmented Reality. Augmented Reality builds a world on top of the physical world. Our games are played with handheld devices like the one shown here (SHOW). When you move in real life, your icon in the game will move. For right now, I am controlling it with my hands but you can imagine if we were outside.... (SHOW VISIT).
3. Outline – The study is going to take about 1 hour. We are going to first do a Walkthrough of the software to get you acquainted. Then you will have about 20 minutes to create your own game.
4. Questions and Concerns – Remember, we are here to test the software today, not to test you. If you have trouble following the instructions, its really important that you tell us. This can help us improve it for other kids. To help us do this, please help us by speaking out loud what you are thinking. For example say, "I am clicking on the start menu, I can't find minesweeper".
5. Stopping / Bathroom Breaks – Finally, you can stop anytime during the study if you don't feel comfortable in any way. Any information you give after it leaves the room will not be associated with your name. If you need to go to the bathroom, it is right around the corner just let me know and I can take you. OK, are you ready to begin??

Walkthrough:

TASK #	Go Down Wrong Path	Have a Long Pause with No Action (> 10 seconds)	Ask for Help	Time Taken in Step (Rounded to nearest minute)	Comments
1					
2					
3					
4					
5					
6					
7					
8					
9					

10					
11					
12					
13					
14					
15					
16					
17					
18					

In Between Walkthrough and Free Period:

Thanks and good job on the walkthrough. So now, do you think you'll be able to create your own game? Before you start, I'd like to give you a few minutes to think about your game. For this game, you can choose 4 different locations (New York City Central Park, Chicago Lakeside, Fenway Park, and MIT). Think about what characters and items will be in your game. Think also about what you want the player to learn when visiting the characters and items. Do you want any of them to appear or disappear? It might help if you wrote some down.

Let me know when you are ready and we can start. Also, like we said before, please help me by speaking out loud what you are thinking.

(Show them how to load their desired template in the software of choices).

Free Period Observations:

Did the subject look back in the walkthrough for help? What did he need help with?

Did the subject start down the wrong path? What did he actually want to do?

Did the subject get confused and ask for help? Which action was it?

What general actions did the subject spend the most time in? What general actions did the subject spend the least time in?

End Step:

1. Administer the Survey
2. Thank them and their parents for their participation.
3. Make a copy of the assent and consent form and give it to the parents
4. Ask them if they would like a copy of the study: YES or NO
5. Give them the $10 Gift Certificate
6. Backup their files and prepare a fresh Game File

Appendix B

Brainstorming Worksheet

Creating Your Own AR Game:

There are four locations you can use for your game:

- Gillete Stadium, Foxborough, MA
- Fenway Park, Boston
- Boston Common
- MIT

Take some time and think about the questions below.

- What do you want the player to learn when playing your game?
- What characters or items do you want in the game?
- What is the goal of the game?
- Will items/characters appear or disappear?

Write down some ideas in the space provided. When you are ready to build the game, let me know.

Appendix C

Survey

MITAR Software Usability Study Survey

First Name ____________________

PART A: For the next set of questions, please circle the number that is closest to how you feel and think about each statement.

5 = Very Difficult
4 = Somewhat Difficult
3 = Average
2 = Somewhat Easy
1 = Very Easy

1. How difficult was the software to use?	1	2	3	4	5
2. How difficult was it to add an item/person?	1	2	3	4	5
3. How difficult was it to change an item/person's picture?	1	2	3	4	5
4. How difficult was it to change an item/person's icon?	1	2	3	4	5
5. How difficult was it to make things appear and disappear?	1	2	3	4	5

PART B: the next set of statements, please circle the number that is closest to how much you agree with each statement.

1 = Agree Strongly
2 = Somewhat Agree
3 = Neutral
4 = Somewhat Disagree
5 = Disagree Strongly

1. The software was enjoyable to use.	1	2	3	4	5
2. You are proud of the game you designed.	1	2	3	4	5
3. Your best friend would enjoy playing the game you designed.	1	2	3	4	5
4. You would like to spend another half hour further designing your game.	1	2	3	4	5

PART C: For the next set of questions, please think carefully and write a sentence or two.

1. What was something you tried that was difficult to do with the software?

MITAR Software Usability Study Survey

2. What was something you tried that was easy to do with the software?

3. What is something you wish the software could do that it cannot do?

Thank you for your time!