

**Designing Flash Applications For the Pocket PC
Focus on Game Design**

Submitted By: Galya Caspi

Date: March 24, 2003

Course: Research Project

Table Of Contents:

Introduction	pg.1
What Exactly Is the Pocket PC?	pg.2
The Pocket PC and Game Design	pg.2-3
The Pocket PC Interface	pg.3
Pocket Internet Explorer (PIE)	pg.3
Landscape Versus Portrait	pg.3-4
Graphic Optimization	pg.4-5
Button Design and User Input	pg.5-6
Conclusion	pg.6-7
Image 1a: Prototype Welcome Screen	pg.8
Image 1b: Prototype Welcome Screen	pg.9
Image 2: Prototype Game Screen	pg.10
References	

Introduction

Pocket PC's are becoming increasingly popular as the evolution of technology continues, and Flash has been recognized as the application that delivers the best results for these devices. In response to this apparent trend, I decided to dedicate my research towards developing and designing flash applications, specifically games, for the Pocket PC. The game I decided to implement was Frogger. I will discuss the creation of the game systematically in relation to the various sections of this paper. This will help to explain why the game was designed the way it was as well as further illustrate important points brought up in this paper about game creation for the Pocket PC.

What Exactly Is the Pocket PC

Before designing anything, the target platform must be defined. The newest operating system for the PPC is PPC 2002. Any device that runs Pocket PC 2002 has very consistent specifications such as: minimum 32 MB of RAM, 32 MB ROM, and an Intel 206 Mhz StrongARM SA-110 processor. Some of the best Pocket PC's have a maximum of 64 MB of RAM and can display 6, 356 colours. Unfortunately, device specific Flash players are necessary for the Pocket PC because some will vary. In addition, the pocket PC is capable of running a lot of the same software that is familiar to normal PC's, such as PIE (pocket Internet Explorer), Outlook Express, and Pocket Word. More detail will be presented about how to compensate for the slower speed and limited memory on the Pocket PC.

The Pocket PC and Game Design

The nice thing about designing games for the PPC is that once the target device has been established processor speed and memory is consistent among all devices thus guaranteeing a consistent playback. Even though the PPC is capable of playing sounds and performing computations, when it comes to designing flash games, simultaneous flash functions such as tweening, actionscripting and music must be minimized for the application to run smoothly. Memory limitations of the PPC also play a key role in flash game design. Unless the user has a CompactFlash card, the RAM on the PPC also stores all additional applications and files desired by the user. This is another reason that restricting concurrent actions and events is necessary for proper game design for the PPC. When developing the game Frogger, all animation was done via actionScript (not by motion tweening), and no sound was added to the application. Animation was also kept to a minimum, with only very simple movements. This was to avoid putting too much stress on the Pocket PC's processor as well as to preserve memory.

Another integral part to game design in addition to knowing the target platform, is defining your target. In general, the people who own Pocket PC's are executives over the age of 25 who most likely prefer simple games that are not complex and help pass time in between meetings or other events. This was another reason why I decided to create Frogger. It is a simple, quick game that everyone can enjoy. Now that the limitations of the PPC and target audience have been

defined, we can discuss some other important issues regarding designing applications for the Pocket PC. The next few sections will discuss usable screen area on the Pocket PC and how to maximize this space.

The Pocket PC Interface

The Pocket PC generally has a command bar at the bottom of the screen and a navigation bar at the top, and the remaining area is used to view applications and content. Most Pocket PC's have an absolute pixel dimension of 240 x 320, however taking scrollbars into consideration and the PIE address bar, which most users leave on leaves a mere 240x240 area to work with. Even 240 x 240 may invoke scrollbars, so often it recommended to design slightly smaller than that. The dimensions used for Frogger were: 240X240.

Pocket Internet Explorer (PIE)

Since the application being developed will more than likely be viewed via the Internet, it is important to know the interface structure of PIE, which is quite similar to the general interface of the PPC. All the familiar navigation tools found in IE such as back, forward, home, refresh etc. can be found within the command bar at the bottom of the screen. Since space is limited, users are advised to turn the address Bar off so as to have more usable area. Frogger was designed assuming that most users do not turn the address Bar off.

Landscape Versus Portrait

Flash content on the Pocket PC can be viewed portrait style or landscape style. Usually, the portrait orientation is chosen because it is more straightforward since the input buttons are oriented in the same direction. The input buttons are still

accessible if the application was designed in a landscape view, although the PPC menus would unfortunately still be vertical. Software such as Jimmy Software's JS Landscape allows developers to turn the entire PPC interface sideways. Landscape view offers a space that is usually more appealing and easier to design for. When designing for a landscape orientation, the design must be created vertically in the flash authoring environment.

Graphics Optimization

The newest Pocket PC's offer 16-bit colour displays, which nearly provides photo quality. However, older models still use 12-bit colour displays, which may result in banding - poor colour transitions within gradients. Adding a little bit of noise will increase the file size, but ultimately create a much nicer looking image.

Creating graphics within flash requires the use of a 12-bit palette. Generally, it is better to use bitmaps as opposed to vector graphics. The graphics should be imported into Flash at the size needed and not scaled down within the authoring environment since this will not reduce file size.

Here are some other tips for optimization:

- Running the flash application at medium quality as opposed to high quality greatly reduces the strain on the PPC processor and provides very similar results to high quality.
- Minimize the use of gradients, and do not use a motion tween with a gradient (*Frogger did not use any gradients, all graphics were solid colours*).

- Movie clips or graphic symbols should be used for repeating graphics – this limits file size. (*Frogger makes great use of repetitive movie clips. The same cars, logs and turtles are constantly being re-cycled. Please refer to Image 2*).
- Minimize the use of motion tweens in the application. (*No tweening was used in Frogger, animation was ActionScript driven*).
- Flatten grouped objects; yet keep a separate reference file with grouped objects in tact in case any changes are necessary.
- Minimize the amount of vector points found in the graphic by using the pencil tool over the brush tool. (*Frogger used bitmap images, not vector graphics*).

Button Design and User Input

Input for the PPC is quite different than for the regular PC. The stylus and five-way button are the devices used for user input. The stylus is similar to a mouse, yet there are only three states: the down state, the up state and the hit state. Rollover buttons are not used for the PPC. Buttons have to be designed at a reasonable size so that their function is easily determined. In addition, buttons should not be placed too close together in case the user slips and taps the wrong button. The design of the application should be forgiving for mistakes such as those. The five-way button acts as the arrow keys, and the middle key behaves the same as pressing enter. Designs that require concurrent key presses cannot be implemented because the 5-way button has no in-between states. Since most users are unfamiliar with this control it would be worthwhile to explain it's

functionality and how it relates to the application, especially if the 5-way button is an integral part of the program. The relevant Actionscript used to detect key presses on the PPC are: Key.UP(38), Key.RIGHT(39), Key.DOWN(40), Key.LEFT(37) and Key.ENTER(13) for up, right, down, left and enter actions respectively. All these actions, except the enter key, will fire repeatedly while the key is depressed. When creating Frogger, the above code was used to move Frogger up, down, left and right. In addition, large, legible buttons with 'simple to determine' functionality were created. No rollovers were used, however different down state were implemented to indicate that something is happening. Another important element in the design of Frogger is the welcome screen, which requests the user to tap on a button with the stylus in order to play. This helps bring the window with the application into focus on the Pocket PC before the game begins (Image 1a, 1b).

Conclusion

Although the Pocket PC is a handy device, it presents many challenges in terms of application development. Creative solutions are always being presented, and this paper discusses many of them. When designing games, for the Pocket PC or any other device, make sure that it is user friendly. Understanding the target device and the users are two of the most important areas relating to game design on the Pocket PC. Designing an application to fit in the usable screen area of the Pocket PC without the appearance of invasive scrollbars is most likely one of the largest challenges for designers. Another important issue is user input and the drawbacks associated with the 5-way control button and the stylus. When

designing any application for the Pocket PC it is important to keep these problem areas in mind early on in the project in order to properly decide how to compensate for these limitations.

References:

Lentz, Warren, Bill Turner, Ian Chia, *Flash the Future*. (San Francisco ; No Starch Press, Inc., 2002)

<http://www.microsoft.com/mobile/pocketpc/learnmore/software>,

<http://www.macromedia.com/software/flashplayer/pocketpc>,

<http://www.zdnet.com.au/reviews/coolgear/pda/story/0,2000023548,20266209-4,00.htm>