Play Pac-Man on the Palm
ECE145 Senior Design Project
Professor Chou

By: Prima Pamplona
Deepa Sahni
Tommy Valverde
March 16, 2001
# TABLE OF CONTENTS

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Introduction</td>
<td>3</td>
</tr>
<tr>
<td>Background and Related Work</td>
<td>3</td>
</tr>
<tr>
<td>Development Plan</td>
<td>4</td>
</tr>
<tr>
<td>Development Schedule</td>
<td>5</td>
</tr>
<tr>
<td>Project Design</td>
<td>5</td>
</tr>
<tr>
<td>Project Specification</td>
<td>7</td>
</tr>
<tr>
<td>Platform Specific</td>
<td>8</td>
</tr>
<tr>
<td>User Interface</td>
<td>8</td>
</tr>
<tr>
<td>Project Evaluation</td>
<td>8</td>
</tr>
<tr>
<td>Qualitative</td>
<td>8</td>
</tr>
<tr>
<td>Quantitative</td>
<td>9</td>
</tr>
<tr>
<td>Concluding Remarks</td>
<td>9</td>
</tr>
<tr>
<td>Credits</td>
<td>10</td>
</tr>
<tr>
<td>Bibliography</td>
<td>11</td>
</tr>
</tbody>
</table>
**Introduction**

In the 1980s, a popular graphics machine came out. Children, even adults, were engaging in a new era of technology. Board games were out, video games were in! Atari, a game console, is the earliest we can remember. Why bother going to arcades when you can bring the arcade game right to your home? This overwhelming new development of graphics brought games that used to be played on the computer (using 5 1/4" floppy disks) to a Vector Monitor based game cartridge. Over time, companies have improved on the old, bringing better quality with a smaller platform.

Now developers for the PalmOs are somehow programming graphics on it. However, as Eurogamer.com has pointed out, "We know the Palm has its own games, but until now, they have been mainly limited to 2D board / puzzle games." But why have games on the PalmOs when Nintendo’s Gameboy already exists? Because every Palm Powered handheld includes basics that all users need, including an address book, mailbox, and a to-do list, all in a system designed to be so small and lightweight that you can carry it comfortably in your pocket.

We know that Pac-Man is already out there for the Palm, but we wanted to make a game that everyone is familiar with and know how to play. The objective of this project was initially to create a game using the J2ME platform, and then render it into 3-D using the MiniGL library. Unfortunately, we could not even get past the installation of J2ME. So, in the short amount of time that we had left, we knew we had to learn the palm platform in CodeWarrior, which none of us were familiar with, create a game, and then incorporate the MiniGL library.

**Background and Related Work**

My group and I first began our research by logging on to www.palm.com to read about palm pilots, their applications, and features. We were very surprised to see what this handheld device has to offer. They are becoming more and more popular everyday. “The Palm OS® is the standard for handheld computing, a new form of computing focused on helping people manage and access information at any time, in any location.” Palm Pilots allow individuals to manage their personal information, enter data, as well as accessing the Internet.

“Palm Powered handhelds are the systems of choice because they give you the most freedom of choice. Palm Powered handhelds have by far the most users, the most software programs, the most hardware options, and the most manufacturers; and yet are still the simplest handhelds to learn and use. All of this diversity lets you make your handheld the perfect system just for you, and gives corporations the security of knowing that they are investing in the open, dynamic handheld standard.”
Palm Powered handhelds already have vast amounts of multimedia options, including voice recorders, color screens and hundreds of games.

Since our project involves developing a game using MiniGL, we researched what this 3D package was all about. MiniGL is basically derived from OpenGL (Open Graphics Library) which is the computer industry's standard application program interface (application program interface) for defining 2-D and 3-D graphic images. Before OpenGL was created, any company that was trying to develop a graphical application had to rewrite the graphics part for each operating system platform and had to have an understanding of the graphics hardware as well. So this application made life a lot easier for the programmer interested in multi-dimensional graphics.

MiniGL is a library that seeks to achieve OpenGL 1.1 compatibility. The goal of this library is to allow compilation of GL code in a Palm development environment. MiniGL also allows easy porting of 3D OpenGL code to the Palm Computing platform. The purpose of this library is to allow users the world of 3D interaction. Using this application, porting time is cut dramatically.

After researching a little about MiniGL and its features, we started thinking about what kind of game we wanted to develop. There were many existing 3-D games on the Palm. Some games that I found that implement MiniGL are Quake, Quake2, Kingpin, and Half-Life. I also found something called WickedGL, which is an enhanced MiniGL support.

WickedGL features:

1) Improved visual quality

2) High tolerance to heavy rendering loads

3) Extra acceleration for high resolution high quality modes

All three of us like Pac-Man, and there are several existing versions of Pac-Man already on the PalmOS. We never came across a 3-D Pac-Man game, so 3-D Pac-Man it was!

**Development Plan**

The development of our project was done using the Metrowerks CodeWarrior development environment and the PalmOS SDK running on Windows ME. We used the C programming language to code this project. The Constructor of the environment was used to develop the application’s GUI. We obtained a copy of the CodeWarrior application from: http://www.palmos.com/dev/tech/tools/cw/, which also provided us with a tutorial. Our cross-development methodology will include using the PalmOS Emulator. Using the PalmOS platform, we will be doing emulations.
We have worked on the same set of files because none of us are really familiar with this new software. The Memopad resource files, included in the CodeWarrior application, gave us with a general understanding of how the source code should be structured. These files also provided us with a skeleton for our program.

Test cases were used to test the functionality of our program including the movement of the Pac-Man and the flow integration of our introductory modules.

Development schedule:

- Organize a group of the same interest and come up with a topic relating to the Palm. We decided to work on a graphics design game using the MiniGL library to possibly make our game 3-D. (Weeks 1-3)

- Programming the GUI
  1. Provide an application icon for the Launcher. To launch an application, users navigate to the launcher screen and tap on an icon. (Weeks 4-5)
  2. Specify the Application Icon Name and Application Icon using the Project Settings panel in Constructor. (Weeks 4-5)
  3. Provide a base screen that offers buttons for the entry and exit of our application (Week 6)
  4. Draw the maze for the Pac-Man game (Week 7)
  5. Correlate the created buttons to make the bitmap move in all four directions (Week 8)
  6. Program database to keep score of each dot that the Pac-Man eats (Week 9)

Concurrently, each one of us will be doing research on the side. Prima will be researching palm specifications; Tom will be working on the user interface and Deepa will further look into MiniGL, how to use it and its capabilities.

Project Design

Our program consists of several modules.

- PalmOS application - this application will obtain commands through the user interface.

- Buttons – This module will handle the entry, exit, menus, and directions that the user will control including the maze application, which is the actual interface the user sees.
• Pac-Man and Ghosts – A Pac-Man game needs its hero and its bad-guys. These bitmaps are created through the use of the CodeWarrior’s Constructor catalog. Drag and drop, create and name the resource ID.

• Maze - What’s a Pac-Man game without the maze? The maze is created for the Pac-Man to go in all of the conventional directions. Within this module are the buttons for exit, start, and the direction keys.

• Database – This module keeps track of the score each time the Pac-Man eats a dot.

We are very proud of our Options form since we incorporated a great deal of animation. Here is a snapshot of our bitmaps. When the game is run, the user can view Pac-Man being chased by the ghosts. We also have several options: Top Scores, Directions, Level, Credits and Exit Screen, which can be viewed below.
Since we only had one detailed module, the maze, there was no purpose to have a header file. Our maze looks something like this:

![Maze Diagram]

**Project Specification:**

Our program will run on Palm OS 3.5 or above only. This application targets the user to engage in the classic Pac-Man game, where the player-controlled Pac-Man feasts on dots, while ghosts try to feast on him. The first few modules include introduction, menus for the start and exit of the game, directions to the game, and choice of which level to play. Viewer can also click on the Credits tab to see development information. The choice of levels (5) in our application is supposed give the player the option to make Pac-Man move a little faster, the higher the level. The maze module is the actual game interfaces which the user controls Pac-Man. The application itself is the movement of the Pac-Man within the walls of the maze. Each movement - forward, backward, up, and down is controlled by tapping the buttons made on the bottom of our maze. In the Pac-Man’s path are the dots which he eats to collect points. There will be a dynamic database that is stored within the Palm to keep track of which user has the highest score. An Artificial Intelligence implementation for the ghost (Pac-Man’s pursuing nemesis) will be needed, since the user only controls the Pac-Man.

We came across a timing constraint when trying to make Pac-Man move. For example, when the user wants Pac-Man to move right, he/she clicks on the right arrow. Pac-Man
does not move right away. It takes time for the system to read that event from the event queue.

Platform-Specific:

The platform we are assuming is the Palm OS IIIe. The only thing we need for our application is the Palm organizer. Platform configurations, and memory capabilities are still unknown.

User interface

The game is compiled into an executable file: .prc which, when is then imported into our emulator. It is executed as a normal application on the Palm with an animation area that contains the maze, power pellets, one Pac-Man, and one ghost. The actual animation area is 128x160 within the constraints of the PalmOS display window. There are three options on the start screen: Start, Options, and Exit. Start takes the player directly to the maze and Exit takes the user out of Pac-Man and leads to the Palm Applications screen. Options gives the user the choice to view the top scores, to get directions on how to play the game, pick a level (1-5), view the credits, or to exit the options, taking the player back to the start screen. The maze consists of the actual maze, an exit and pause button, and an area where the player can view the top score and the level that they are playing on. There will be four action buttons. Each button directs the movement of the Pac-Man up, down, left and right.

Project Evaluation

Qualitative:

Specification: So far we have been able to get as far as two actual modules. One of the modules include the introduction, credits, levels, and the start and exit of our game. The most difficult problem we had been stuck on for the longest time was trying to get the bitmaps to move using the WinDrawBitmap() function. Once we figured out that, we still needed to implement the user controls, which are the buttons we made ourselves, to move the Pac-Man in all four directions. We still have not gotten as far as making the dots that the Pac-Man eats, and the database, which stores the score.

Design: We drew the maze and the buttons to conform to the limitations of the Palm’s display screen, which is only 160x160. Our maze takes up 128x160 of the display screen. To draw the walls, we created each line with the Bitmap Forms in the Constructor’s catalog and named each of their resource ID’s separately. We conformed to this method in drawing the Pac-Man as well as the ghosts. The introduction screens, which are: introduction, options, levels, and credits are all linked together using the Event
handler written in C. The best method for this was to use IF ELSE and SWITCH statements as seen in our source code.

Development: Since we are using the CodeWarrior environment, and the MemoPad.c tutorials, we have been successful in cross compiling, and MAKE-ing the .prc as an executable file for the Palm. The PalmOS Emulator is used to test our game application. At this point we are successful in going through the introductory forms, getting to the maze, to start the game, and moving the Pac-Man using the buttons we created at the top of the maze, indicating L->left, R->right, U->up, D->down.

Quantitative:
Performance: In terms of performance, we are able to execute successfully, but are not able to determine what types of extra memory copying, or other overhead there is.
Power usage depends on how long the user plays our game. Like any other application on the Palm, the longer the usage, the more likelihood the batteries will drain. Other than that, our game runs like any other application on the Palm.

Concluding Remarks

This project was very exciting, yet challenging. All three of us are really interested in computer gaming, so this project was of great interest to each one of us. We learned that there is quite a bit of planning, designing and coding involved with developing a game. We did, however, come across some obstacles. We had the PacMan code written in Java and it was good working code. Originally, we wanted to program on a Java Virtual Machine, namely J2ME. We had problems installing this on our home PC, so we couldn’t continue with using Java. We asked the professor and other students for some help on this matter, but we got no response. So at the very last minute, we turned to CodeWarrior. Our proposed project entailed creating a 3-D game on the Palm using the miniGL library. Unfortunately, we didn’t make it to the 3-D part. If we had more time and more resources, we definitely would have had 3-D PacMan on the ball. Our game could also be improved by using the FrmRemoveObject() function so we wouldn’t have to draw so many bitmaps. We ended up drawing 63 bitmaps of PacMan! We couldn’t get any help drawing this function. As an alternative, we are hiding and showing bitmaps accordingly. We would also have liked to have a database that keeps a record of the high score, which we were not able to get to due to time.

If we could have done anything differently, we would have planned a little differently. We left the programming to the very last minute. This was mostly because of the fact that Java did not work for us, so we had to learn to program using CodeWarrior. We could have improved our project by also having some pre-knowledge on miniGL. We really didn’t know how to incorporate this library into our program. One of us posted a message on the Noteboard, but unfortunately, no response. On a positive note, we did succeed in creating a successful working game on the Palm, and we are very proud of it!!
Credits

*Project Write Up* – Prima Pamplona (Introduction, Development Plan, Project Design, Credits); Deepa Sahni (Background and Related Work, Project Specification,); Tommy Valverde (Project Specification, Concluding Remarks, Bibliography)

*Programs written in* –

- CodeWarrior: Tommy Valverde
- J2ME (Java): Prima Pamplona, Deepa Sahni
  
  • Although, all of us worked concurrently on the same set of files, credit is given to group members who did the bulk work.

*Bitmap creation* – Prima Pamplona, Deepa Sahni

*Testing* – Prima Pamplona, Deepa Sahni

*Source Codes Used* – MemoPad.rsrc, MemoPad.c; MemoPad_win tutorial found at [http://e3.uci.edu/01w/15278/tutorial/MemoPad_win/](http://e3.uci.edu/01w/15278/tutorial/MemoPad_win/)

*Techniques Learned* – Jack Yuang, Jason Chen, and Barry Chen helped us out by showing us their initial source code to get us started; Hong Yu helped us out by showing us how to implement WinDrawBitmap().
Bibliography

1. http://www.standalone.com/ - company that has already made the pac-man game

2. http://www.opengl.org/developers/about/overview.html/ - website that gives information about OpenGL, that it is an interactive 2D and 3D graphics application programming interface (API)

3. http://www.dsbox.com/minigl.html – very informative website that shows the development and features of miniGL


