PALM GAME DESIGN

Pieter Spronck
Universiteit Maastricht IKAT/Infonomics
P.O. Box 616
NL-6200 MD Maastricht, The Netherlands
E-mail: p.spronck@cs.unimaas.nl

KEYWORDS
Gaming, handheld computers, software engineering, design.

ABSTRACT

Though the Palm is mainly a business tool, many games have been developed for it and more are published daily. This article starts by examining the status of game development on the Palm today. To give an indication on what designing guidelines a game developer for the Palm should take into account, several "rules of thumb" are presented. A description is given of a successful Palm game, "Space Trader", and areas where this game fails are indicated. The article concludes by looking forward into the near future of Palm game development.

INTRODUCTION

Since 1998 with the release of the Palm III, handheld computers have become all the rage (Williams 1999). Starting out as replacements for electronic agendas, newly developed applications turned them into calculators, web browsers, notebooks, email-managers, translators, e-books, databases, barcode-scanners and even remote controls. Of course, games weren't left behind. The number of downloadable games for the Palm runs in the thousands, most of them shareware or freeware. Many of these games are not really worth your while, but there are gems to find for those who keep their eyes open.

This paper examines today's status of Palm game design, gives a few rules of thumb for the technical designing of handheld games, illustrates this by the story of a successful Palm game, and will venture a look forward into the near future of handheld games. It will be limited to discussing games for the Palm OS, which currently has the biggest market share, but most of the statements here are just as applicable to competitors of this OS.

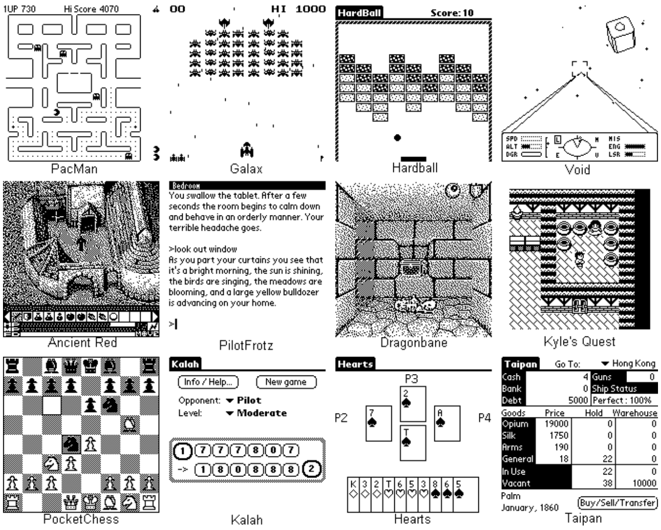
PALM GAMES TODAY

When viewing the landscape of Palm games as it is today, we see games of many types and qualities. There are arcade games like "PacMan", "Galax" and "Hardball"; action games like "Void" and "Ancient Red"; text adventures playable with "PilotFrotz"; role playing games like "Dragonbane" and "Kyle's Quest"; board games like "PocketChess" and "Kalah"; card games like "Hearts"; and strategy games like "Taipan" (see figure 1).

Most games are fairly simple, finding their inspiration in games from the early '80s, but recently, especially since the advent of colour Palms, more complex games have entered the Palm landscape. The evolution of games on the Palm goes fast, and developers try to catch up with even the latest PC games. For instance, the first role playing game for the Palm was a port of the ancient, text-based game "Rogue". This was followed soon by "Kyle's Quest" that was inspired by the first "Final Fantasy" games and "Dragonbane" that was inspired by "The Bard's Tale". Now there is even "Ancient Red", a Palm game reminiscent of the popular "Diablo".

A game like "Ancient Red" looks very beautiful, but does not set the standard for Palm game development. There are several reasons for that. Firstly, though the game does support greyscales, the graphics are too intricate to play it on anything less than an 8bit colour screen. Secondly, even on a colour screen, the interface with its excruciatingly small fonts and pixel-hunting stylus interaction is unclear and difficult to use. Thirdly, the game's memory usage is enormous. Fourthly, it has a steep price. Though the game seems to be popular with some of the people who bought a Palm purely for pleasure, it has only a limited audience.

"Ancient Red" seems to be an exception to the rule that most Palm games are designed and built by a sole programmer. The Palm doesn't allow much in the way of graphics or sounds – the capabilities of the Palm can be compared with those of the first XTs or of a Commodore 64 without a sound chip. Therefore you don't have to be an artist to build a game. Furthermore, for the design of a new game you can be inspired by the multitude of games that have populated the gaming realm since the first release of "Pong". All of this makes the current Palm gaming world an ideal place for an aspiring game programmer to leave his mark. A good Palm game can be produced by a single programmer in a matter of months. There is a large audience awaiting such efforts, and it is even possible to start a small one-person business in Palm games.



Bedroom
You swallow the tablet. After a few seconds the room begins to calm down and behave in an orderly manner. Your terrible headache goes.

>look out window
As you part your curtains you see that it's a bright morning, the sun is shining, the birds are singing, the meadows are blooming, and a large yellow bulldozer is advancing on your home.

>

Cash	4	Guns	0
Bank	0	Ship Status	
Debt	5000	Perfect : 100%	

Goods	Price	Hold	Warehouse
Opium	19000	0	0
Silk	1750	0	0
Arms	190	0	0
General	18	22	0
In Use		22	0
Vacant		38	10000

Palm
January, 1860

Figure 1: Several Palm Games

RULES OF THUMB

Handheld computers are not, like PCs, general-purpose machines. Palms are not about putting a PC in your pocket, they are meant for executing quick, simple tasks, anytime, anywhere. Adding ten numbers on a Palm is acceptable; filling out a large spreadsheet to calculate the total costs of sending your kid through college is not. One of the reasons the Palm is far more successful than its derivatives is that the Palm hardware, as well as the Palm OS, is geared towards such quick tasks (Rhodes and McKeehan 1999). However, this philosophy puts serious limitations on the design of Palm games. This paragraph presents several rules of thumb that take these restrictions into account. Some of these rules are derived from the "Palm OS Programmer's Companion" (3Com Corporation 1999), but here they are especially geared towards game design. Note that these rules only give pointers on designing a well-functioning Palm game, not necessarily a fun one.

1. The Screen Resolution Drives the Design

A Palm screen is limited in size to only 160x160 pixels on a 6x6 cm area, with at most 256 colours. The game designer has to make do with that. Even worse, in order not to restrict the audience too much, the game should work with monochrome screens, since even greyscale support has only recently been added to the Palm OS. The screen resolution is a serious limitation in the functionalities a game can offer. The best way to handle this is not to design a game and then see how it fits on the screen, but to take the screen restrictions into account at every moment during the design.

Of all Palm applications, especially games often make mistakes in screen design. There are games that create their own, small fonts that are unreadable. There are games that use tiny buttons that are easy to miss with the stylus. There are games that use rich colours and use a dithered, unusable version of that to support monochrome screens.

Because of the physical restrictions imposed by the screen, the functionality of each screen should be limited. Scrolling using scrollbars should be avoided. There should be no more than around four buttons on the screen, and they should be big enough to be easily tapped with the stylus. If both colour and monochrome screens are supported, the monochrome version should be designed separately from the colour version.

2. Games Are Small and Use Little Dynamic Memory

Memory is a valuable resource for the Palm. Not only is it used to run applications; it is also used to store them and their databases when they are not in use. There is no hard drive. The earliest Palms only have half-a-megabyte of memory or even less. Most of the Palms today are limited to two megabytes. New Palms will commonly have at least eight megabytes, but it'll take a few years before all the older ones are replaced.

There are two problems with the memory limitation. The first is that the user is severely restricted in the number and size of the installed applications. Games are usually not a priority and will be quickly deleted when there's a lack of memory – especially the larger games. When deliberating whether or not paying a shareware fee for a game, the user's decision will not only be guided by the monetary costs, but also by the memory costs. The quality of a large game must be very high to get a user to pay for it.

The second problem is that an application cannot dynamically assign much memory. The size of the dynamic heap of a Palm is restricted, in the worst case to 32K. On a PC this kind of problem is solved by "swapping out" memory. On a Palm this can be emulated by storing allocated records in a database. However, one should realize that this database is built in the generic Palm memory, and therefore limited by the size of the memory the user left unused after installing applications, which usually isn't that much.

It should be noted that especially colour images are serious memory hogs. One full-screen colour image will take around 25K of memory. While this doesn't seem that much, one should realize that many applications as a whole are even smaller than that. One way around this is to make graphics an optional feature of a game and to store them in a separate database, which users can install if they have memory to spare.

3. Game Control Is by Parsimonious Use of the Stylus

Most PC games use a variety of input devices. Normally you can use a keyboard and a mouse, sometimes a joystick is required. None of these is delivered with the Palm. The standard Palm has three ways of inputting data: by tapping the screen, by writing on the graffiti pad, and by pushing one of the shortcut buttons. Of these, screen tapping is the main method to control a game. This is completely different from controlling PC games, and the design of a game should take this into account.

To answer the question why the graffiti pad should not be used: the pad is used for inputting texts, but is not as easy to use as a keyboard. The writing of text should, especially in games, fulfil only a minor role. Pick-lists are often a good alternative.

To answer the question why a game should refrain from reprogramming the buttons (except for the up/down button, of which the use is application-dependent): the buttons are used to switch to the major tasks the user performs with the Palm, and as such reprogramming them is annoying especially to the business user. Many games, mainly arcade games, break this rule. Of course, it may be hard to create an arcade game without reprogramming the buttons. In practice, however, there are many games which reprogram the buttons while a better way of controlling the game would be using the stylus. In case one finds the buttons are indeed the only viable input device for the implementation of a certain function, one should at least make sure they get released at each and every opportunity.

Even though tapping is the way to control a game, the number of taps necessary to execute that control should be as small as possible. Main functions should require no more than one tap. "Double-tapping" (equivalent to double-clicking with a mouse) is theoretically possible but awkward and therefore totally out of the question.

4. Exiting a Game Takes No More Than a Second

Currently, the normal use for a Palm is not "playing games". It is used as a business device, a portable extension of a PC, a small tool to support tasks done "in the field". The average Palm user will have a few games loaded to entertain him while standing in line or to kill some time while travelling, but these are not the reason why he or she bought the Palm.

Palm users can't be as patient as PC users, since they are usually "on the move". Even a few seconds delay in activating their desired application is too much. They ask their Palm for information and want it now, not in a moment. Therefore, games should allow the user to instantly switch to another application. This means they should be quick to shut down.

Some games break this rule by taking a considerable amount of time to save their state. Other games even do worse and prohibit the user from leaving the game at any given moment, either because that means the user will instantly "lose" the game, or simply because it refuses to give the control back to the OS at that time. These are serious problems.

5. A Game Session Can Be as Short as One Minute

Palm users tend to fire up a game when they have a few moments to spare. They are waiting in line for a cashier or for the bus to arrive. Often they don't know how much time they will have, they just want to play a few turns and be able to put the game away at any given moment. A game should allow them to do that.

This means that the game's state should be obvious at a glance. Even if a game could take an hour to play from start to finish, any possible state in which the game can be loaded should be clear to the user, even if he played the previous session a week before.

It also means that a game can be saved in any possible state. Some games, when exiting, don't save the actual game state but a previous game state. That's not much of a problem if it means the user loses a few seconds of playtime. However, if one game-turn takes a couple of minutes and the state is only saved at the end of a turn, the user should at least play a whole turn to make any progress, so he can't play a one-minute session.

6. Sound Is Optional

When a user plays a game on a PC, he can, if he wants (and the neighbours agree), have sound blasting from enormous speakers. Palm games, however, are usually played in public. Computer game sounds, those simplistic Palm sounds in particular, are very annoying to bystanders, and if they are a required feature of a game, the game cannot really be played at times when the user would like to. Besides, sounds can be turned off for the Palm as a whole, and many users only have "alarm" sounds turned on.

7. Games Don't Need a Manual

When a user starts a new game, he doesn't want to study a manual; he wants to spend five minutes on playing a game. A paper manual, of course, is out of the question anyway, since he won't be carrying it with him. But also on-line manuals are not an acceptable method of training. The Palm screen is too small to read text efficiently, and besides, reading long manual texts only costs time which could better be spent on game-playing.

The solution is to make the game interface easily understandable. In principle, the screen should tell the user everything he needs to know. For board games, which may have complex rules, this is not always possible, but even then at least the interaction with the game should flow naturally. A short instruction text, three screens long at the most, is acceptable if it's easy to access and it only needs to be read once.

The designer can count on the user tapping things that look like buttons or selection lists. So, a good way of creating an easily playable interface is to create buttons for the actions a user can perform and label them in a clarifying manner. This may be problematic because the number of buttons can't be too large, and labels should be short for lack of room. That's exactly where a designer's job comes in.

8. Games Are Thoroughly Tested

This rule is a wide open door, of course. However, it should be particularly noted for Palm games, since Palm users commonly have little patience with buggy software. Games are not required software, there are many free or pretty cheap alternatives available, and storage capacity is valuable. The game designer must expect that a game that annoys a user, even only once, will be deleted. Only serious testing, not only for bugs but also for playability, may avoid this from happening.

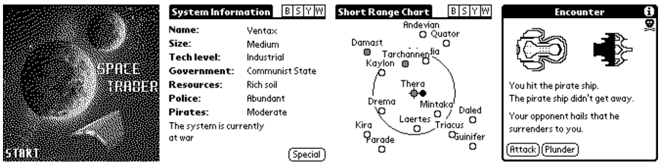
SPACE TRADER

One way to create a game that adheres to the rules of the previous paragraph is to keep it simple. That is actually what has been done for most of the Palm games that exist today. However, that does not mean that complex games are impossible to create for the Palm, though it is more difficult to keep a game playable. Especially the screen limitation needs serious consideration for complex games, since they usually offer the player many possibilities which must be presented somewhere on the screen. This paragraph presents experiences with the design of a complex Palm game.

The History of Space Trader

Halfway the year 2000 there were no complex strategy/trading games for the Palm. The few trading games that existed had very simple mechanisms and depended mainly on luck. As an experiment I decided to build a space trading game, with complex trading rules, geared towards the strategic player. As inspiration I used the famous '80s game "Elite", removing the 3D flight mode and increasing the trading aspects considerably. I implemented the game using CodeWarrior release 6. After a thorough beta testing phase I released it September 2000 as freeware under the name "Space Trader". The first release was quickly followed by a few new releases in response to player comments, adding some graphics and making interface enhancements (see figure 2).

The game has been received surprisingly well, getting almost unanimous good reviews from players, websites and magazines, and receiving several nominations as "the best Palm game" in some category. The number of players, nearly one year after the initial release, I estimate between 100,000 and one million. The current version of the game allows the player to trade ten different kinds of goods in a galaxy comprised of 120 solar systems, each with their own size, technological development level, political system, special resources and special events; to fly ten different kinds of ships equipped with a selection of different kinds of weapons, shields and gadgets; to become a trader, bounty hunter, pirate or smuggler; and to go onto a several quests of varying difficulty.



Name:	Ventax
Size:	Medium
Tech level:	Industrial
Government:	Communist State
Resources:	Rich soil
Police:	Abundant
Pirates:	Moderate

The system is currently at war

You hit the pirate ship.
The pirate ship didn't get away.

Your opponent hails that he surrenders to you.

Figure 2: Several "Space Trader" Screens

Space Trader Design

The gameplay of "Space Trader" consists of two parts: "trading" and "travelling". When trading, the player is docked at a space station, where he can switch to many different functions, like buying and selling of cargo, buying equipment, visiting a bank or choosing a new solar system to fly to. While travelling, the game is in a modal state, and the player has to negotiate his way past several encounters with police, pirates and other traders he meets underway. The player can interrupt the game at any moment and restart it at that same point later. If the interruption happens in a fight, the encounter screen tells him everything he needs to know. If it happens while docked, all information about current cargo, current ship, current location and current quests is accessible.

The game is designed as a continuous journey towards some final goal, which takes a couple of hours of playtime to reach. The main activity of the player is seeking good trading opportunities, which is a strategic task, because it is only slightly randomized (to simulate small local market fluctuations) and the best deals are made when the player succeeds in exploiting the characteristics of neighbouring solar systems. All the factors which play a role in the determination of the prices of the different trade items are explained and justified in a separate manual, which isn't required reading but which might make the game more fun to play.

Besides hoarding money to fulfil the final goal, the player must use his earnings to upgrade his ship and equipment, to manage to survive encounters with the pirates, which become stronger and more numerous while the game progresses. Combat itself is simple and does not rely on reflexes, but mainly on making prudent decisions on when to fight, when to flee, and when to surrender (most players abhor surrendering, but especially on the harder difficulty levels it's sometimes the best choice).

This design carries in it the risk of repetitiveness. A reasonably successful attempt has been made to resolve this by having the following features:

- There are so many factors which influence the trading balance, that the trading experience differs notably from solar system to solar system.
- There are quests which the player gets offered occasionally, many with special rewards.
- The player can choose his own playing style, and can switch between styles if he likes.
- The player can attempt to influence the success rate of his own playing style by choosing the right equipment and mercenaries.
- The enemy AI is simple but takes into account the player's style, strength and success rate.
- Dying is possible in the game, but always avoidable for a smart player.

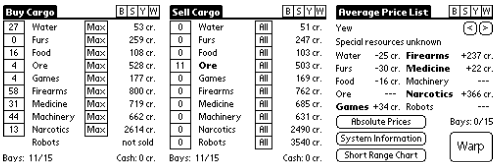
For the user interface it was decided to strive for clarity by using standard fonts and buttons, by spreading functions over as many different screens as needed, and by avoiding the use of abbreviations.

Space Trader and the Rules of Thumb

After presenting rules for the successful design of Palm games, it's interesting to examine how well my own exploits confirm to them. "Space Trader" mainly sins against three of them in the following ways:

- The game is not small. The first release was 150K in size, which is acceptable but strains the limits. Later releases increased this to 315K for the colour version. This is mainly the result of adding graphics. The graphics have been compiled in the game itself and are not stored in a separate database, so the user is forced to install them. A black-and-white version has been made available that is "only" 230K in size.
- The game gets controlled mainly with the stylus, but the stylus use is not parsimonious. "Space Trader" simply contains too many screens. A better design would have integrated more functions in fewer screens. In figure 3, the three screens in the game that are used for the trading of cargo are shown. After the first release, a player suggested allowing the user to buy cargo immediately from the average price list. For the next version, I added the ability to buy cargo by tapping on one of the cargo types listed on the average price list. This would popup a screen showing the details of this cargo item and asking the user if he wanted to buy it, and if so, how many canisters (with the possibility to buy "as many as possible" by tapping one button). This greatly increased the user-friendliness of the trading function and decreased the number of necessary screen-switches considerably. With a bit of thinking, it would be possible to also integrate the selling function, producing one "Cargo Trading" screen. It should be noted that many reviews have mentioned the ease of use of "Space Trader" and the clarity of the screens. Integrating more functions in one screen could mean sacrificing some of that clarity, but it would probably be worth it.
- The game is more in need of a manual than I would like. There is a huge in-game manual in the form of help screens and several menu-items that explain certain aspects of the game. I don't believe these are all necessary, and in fact I have discovered that many players don't read the help texts at all. However, a player must at least be instructed on how to buy and sell cargo, how to equip his ship, and how to travel to other solar systems. The need for a manual for these aspects would

be greatly reduced if the following changes were made: the addition of a "nerve centre" screen from which the basic functionalities would be accessed and the reduction of the number of screens as already mentioned above.



Buy Cargo

27	Water	Max	53 cr.
0	Furs	Max	259 cr.
16	Food	Max	108 cr.
4	Ore	Max	528 cr.
4	Games	Max	177 cr.
58	Firearms	Max	800 cr.
31	Medicine	Max	719 cr.
44	Machinery	Max	662 cr.
13	Narcotics	Max	2614 cr.
	Robots		not sold

Bays: 11/15 Cash: 0 cr.

Sell Cargo

0	Water	All	51 cr.
0	Furs	All	247 cr.
0	Food	All	103 cr.
11	**Ore**	All	503 cr.
0	Games	All	169 cr.
0	Firearms	All	762 cr.
0	Medicine	All	685 cr.
0	Machinery	All	631 cr.
0	Narcotics	All	2490 cr.
0	Robots	All	3540 cr.

Bays: 11/15 Cash: 0 cr.

Average Price List

Yew

Special resources unknown

Water	-25 cr.	**Firearms**	+237 cr.
Furs	-30 cr.	**Medicine**	+22 cr.
Food	-16 cr.	Machinery	---
Ore	---	**Narcotics**	+366 cr.
Games	+34 cr.	Robots	---

Bays: 0/15

Figure 3: "Space Trader" Cargo Trading Screens

Lessons Learned

The following statements I would offer as "lessons learned" from the production of "Space Trader":

- **Graphics *are* important**. The first version of "Space Trader" only contained a simple graphical "Start" and "Victory" screen. I got many requests to add more graphics, and did so by adding ship pictures and enhancing the existing illustrations. I took care that the graphics were not just "eye-candy", but fulfilled a useful role in the game. After that, requests for more graphics were seldom ventured.
- **Graphics are not *that* important**. The graphics in "Space Trader" are very simple, but I get virtually no complaints on that. Players seem to focus on the strategic aspects of the game. And graphics do have a bad side: adding them doubled the memory footprint of the game.
- **The fewer screens, the better**. When docked at a space station, the player of "Space Trader" has access to ten main screens and about the same number of sub-screens. This requires far too many screen-switches. A redesign of the game would probably make do with about four main screens and a few sub-screens, without losing functionality.
- **Don't use the menu except for housekeeping functions**. "Space Trader" makes the mistake of making some of the basic functionalities only accessible through the menu. This means some players completely forget about them. The menu should be used only for housekeeping.
- **Keep interest going by having developments and a few surprises**. The most appreciated aspects of "Space Trader" are the quests, which form small stories in the game, and the few surprises the game has in store for the player. A better version of "Space Trader" would have far more and different quests, and would offer multiple solutions for them.
- **Keep supporting low-level Palms**. A surprisingly large number of players write to me how glad they are that there are still developers that write games for older Palms, so it seems worthwhile to keep supporting them. This may, however, prove to be impossible for games that require more advanced functionalities, like elaborate graphics.
- **Incorporate a savegame feature**. Intentionally, I left out the savegame feature for "Space Trader", because it would remove all tension from the combat sequences. However, there are three good reasons to have it: Firstly, if a player moves on to another Palm or if his Palm suffers a serious crash, he will have lost his current game; secondly, it allows players to play multiple parallel games on one Palm; thirdly, if a player reports a bug, having access to a savegame is an important help in solving it. When adding savegames, a good solution for the combat issue could be penalizing the player for restoring a saved game, for instance by disallowing ranking in the highscore table.

THE FUTURE OF PALM GAMING

Palms have dropped in price and have become popular not only as a tool for business users, but also as a "cool gadget" for everyone else, especially youngsters. These people are far more interested in games than in business applications. The Palm Corporation seems to find this enough reason to forego their philosophy of focussing on simplicity and starts to push the Palm more and more as a games machine. This means it is very likely that in the near future we will see the release of Palms specifically for gamers, with extra graphic capabilities (perhaps a 320x320 screen with 16bit colours), extra sound capabilities and an earphone jack, a lot more memory or memory cards on which software can be preloaded, easy Internet access for multi-player games, and perhaps a built-in joystick. In fact, hardware which supports some of these features is already available. The consequences for the Palm OS are minimal, it's mainly the hardware that will be changed. Palms without these extras will, as long as the game-specific Palms are clumsier than regular Palms, still be available for business purposes. Just like what happened with PCs, the gamers will use the most expensive and advanced machines.

The consequence is, however, that game programmers must decide whether they will build a game which runs on all Palms, or whether they will produce a game for a "GamePalm". The first will restrict the game's features, the second the audience (though it will, of course, mean that the designer can relax on the rules presented earlier in this paper). At first, a single programmer is probably better off concentrating on the basic Palm, because it is likely that professional programming teams will quickly start working on producing games for the enhanced Palm. Not long after, I expect there will be no room anymore in the Palm game development world for a one-person business. Therefore, the statement at the start of this article that the Palm gives an ideal opportunity for a sole game programmer to make a name for him- or herself, has an expiration date. The time to grab that opportunity is now. It will probably have passed in a few short years.

REFERENCES

3Com Corporation. 1999. *Palm OS Programmer's Companion.* Part of the Palm OS Software Development Kit.

Rhodes N. and J. McKeehan. 1999. *Palm Programming: The Developer's Guide*. O'Reilly & Associates.

Williams J. 1999. *Games to Go: The PalmPilot Series*. GamaSutra. Available at: http://www.gamasutra.com.