

## Postmortem: **Games Kitchen's *Wireless Pets***

By David McQueen

When The Games Kitchen began *Wireless Pets*, WAP (Wireless Access Protocol) was as much in its infancy as the company and the publisher was.

Let me give you some background. The first WAP enabled mobile phones had just reached the market, and network operators were pushing them to customers as 'the Internet on your phone', which it isn't, or at least not in any useful sense. It would be much more accurate to say that it was AN Internet on your phone, as only websites in WML (Wireless Mark-up Language; a variant of HTML based on XML) could be viewed. There were virtually no websites and no applications, games or otherwise. However, WAP was not yet seen as the donkey it is now commonly viewed to be, and everyone was jolly excited.

The Games Kitchen had been going for almost a year before we started the contract for *Wireless Pets*; however, we were only two people (business partner David Thomson and myself). Digital Bridges (DB) had been started just up the road from us (we are both based in Scotland), and we had worked together on a few simple projects, such as Roulette and Fruit Machine, back when DB were also only two people.

*Wireless Pets* was an ambitious title. We'd never done anything that big or complex, we had to expand the company for this title, and DB had to expand their platform to handle it. Heck, we weren't even sure if WAP (or, more accurately, the usually poor implementation of WAP found on the new devices) could handle it. But we did it anyway, and it became the biggest WAP game in Europe, with over 15 million minutes of airtime on 18 operators. The SMS version looks set to repeat that success. This is the story of both WAP and SMS versions.

## What Went Right

**1. Simple Interface.** WAP is an browser based online only environment; nothing resides or runs on the device. Playing a WAP game means selecting choices by selecting a link, which communicates back to the server, and the server generates a new page for you to view based on what you've just done, and updates your game state. Everything runs on the server, and thousands of people from multiple countries can be accessing at once. In this respect, it is a lot like massively multiplayer games, except that there's no client application other than the browser.

The medium really forces the design to be menu-driven; indeed, there is virtually no other way for the user to have any meaningful choice. The issues with this are:

- The only menus possible are lists of links (as found on many HTML pages), and links can't be images in WML
- Typical screen size is 70x40 pixels, or four lines of text

So what most users will be faced with are lists of text links, and indeed only a very small window onto the list (four options visible at a time at most). There's not even anything like drop-down menus which are typical in the Windows interface. Therefore, a significant issue is designing a simple user interface in which the user won't get lost, but which still offers a good number of choices to the user.

*Wireless Pets* achieved this fairly successfully; the reason we know this is that not one of the over 350,000 players has complained about the interface. If it doesn't get in the way, it must be OK. Also, there are over 60 ways to interact with the pet, giving the user plenty of flexibility.

The simple interface hides a relatively deep game. Like games such as *Creatures*, *Wireless Pets* is surprisingly complex, although most of this complexity is hidden and the game seems simple to the user. The pets have many hidden statistics and variables, as do the food, drink and play objects, and each object has a different effect on each of the 6 pets. Having this end of the game running on the server meant that it could be as complex as we wanted it to be without worrying about device limitations, so the choices the user makes can have interesting and complex effects. For example, pets may get any one of a number of illnesses if their health variable gets low; and incidentally, there's no way for the user to directly affect the health variable.

SMS is an even more limited technology. Text messages are sent to and from the user and server; just ASCII text, nothing else, and with a limit of 160 characters. Try designing a game for that! Thankfully, we weren't trying to replicate the WAP gaming experience for SMS - the WAP version has more than 60 different ways to interact with the pet, and it is impossible to let the user know what 60 options they have in only 160 characters of instructions.

Therefore, the interface for SMS game is very limited, as it has to be. It is much more like the Tamagotchi games than the WAP version is, having only 3 real options (feed, play and clean). Again, though, this did not mean that the pets were as simple as that, although they were less complex than those found in the WAP version.

**2. Personality.** It may seem odd to talk about personality in media which can have 70x40 black and white graphics (WAP) or 160 characters of ASCII text (SMS), but this is what we recognised we had to do. The user must feel some sort of emotional attachment to the pet, otherwise it's entirely meaningless.

Before we did any game art, all of the characters were designed on paper then as full colour vector graphics. This was extremely useful for marketing support materials. All of the pets were designed so that they would reduce down well to 40 pixels in height.

Since the game graphics are so small, we knew we had to do more to convey personality, and we did this by having the pet write a diary each night. It's worth pointing out at this point that the game is 'always on'; the user links up (WAP) or sends a message (SMS) to interact, but if they don't see their pet for a few days it will get hungry, bored and very lonely. At midnight on the server, the game does a daily update, which includes changing some of the pet's variables and also writing a new, three sentence diary. The sentences are selected from a database based on the pet's variables and the priority of individual pets (for example, the dog is more worried about the amount of exercise it's getting than the ghost is). Diary entries are not as simple as 'I feel hungry'; they convey personality. Sample diary entries for feeling hungry are:

- "I blame Dave for making me hungry" (the cat always blames the owner for anything bad)
- "Me first mate was here today but there still be problem with them thar rations" (the turtle is a salty old sea dog)
- "ROAR!" (the dinosaur is perpetually angry)

The SMS game was a good bit trickier to try and get an emotional attachment due to the limitations. We again used diary entries, though these had to be short due to the 160 character limit, so only one entry was 'written' by the pet. Although we couldn't use graphics, we did use smileys to give a visual representation of how the pet was feeling. A typical update message would look like this:

-Spot (Dog)-  
This is too messy even for a dirty dog like me!  
  
Happy? :-)  
Fed? :-D  
Clean? >-(  
Healthy? :-|  
  
Respond with Feed, Play, Clean, Quit

Which at 143 characters (including spaces and hard-carriage returns) is at the maximum we could use, since although the technical limitation is 160 characters, most languages are 10-20 percent lengthier than English.

**3. Innovation.** As I said at the beginning, we didn't know quite what UNITY or indeed WAP itself could manage. We made quite a number of technical innovations which would be taken for granted on other platforms, even relatively simple ones such as Game Boy.

- *Wireless Pets* was the first published WAP game to feature animation, and we had to figure out how to do that by ourselves
- It was the first published game to use UNITY's multiplayer code to do pet shows
- Other features that were added to UNITY (or DB's servers) included code for timed events, a revised database structure and image caching control

Many of these features are standard now in UNITY and on other WAP platforms, but at the time were ground-breaking. All of these innovations, although all except animation are transparent to the user, combined to make an envelope pushing game.

**4. Mass Appeal.** One of the features of WAP and particularly SMS is the broad market for the devices. Every mobile phone in Europe sold in the last five years or so has had SMS capability, and inter-operator SMS isn't a problem in Europe as has been in the US. Over 24 billion SMS messages are sent each month, despite its lack of popularity in the US. Almost every phone now sold in Western Europe has WAP capabilities, although usage isn't nearly as high as it is for SMS.

Basically, these are mass market media, and in a way that the games industry never gets close to. Around 60 percent of the population (in the UK, and this probably hold true for most of Western Europe) has devices that can play these games, and the games are marketed by network operators, who really do market the games to everyone.

This means that games for WAP and SMS will be marketed at many non-gamers. They are unlikely to be enthused by many typical gaming genres, such as sci-fi and fantasy. The games will also be played by many non-gamers, who are unused to gaming conventions, even really

simple ones ("Why didn't the game start from where I left off? I have to press 'save'? Why? Who decided that?"). Another factor worth considering is that most gamers will have dedicated games machines, be they console or PC, and so will likely be uninterested in a gaming platform featuring a 70x40 black and white screen.

Undoubtedly, one of the reasons *Wireless Pets* has become so popular is its appeal to non-gamers. The subject matter is based firmly in the real world, the objectives are obvious and not goal oriented, everyone knows what a dog is and that it was to be fed and played with, and the pets look cute as buttons. People are very familiar with the concept of animals with human features and personalities, and of course *Tamagotchi* and *Pokemon* took the world by storm, appealing to a market that games hadn't reached before.

The simplicity of the user interface also helped significantly in this respect. There are no hard controls to learn. Despite the number of variables and complexity of interactions, most of this was hidden from the user, or presented to them in a way anyone could understand (for example, through the diary entries). The game also features a level of resource management; users have a weekly allowance to spend on food, toys and medicine, but again this is a concept not foreign to anyone.

**5. Platform.** Despite its flaws and limitations, particularly at the start of the project, UNITY is a good platform for WAP and SMS. It took away a number of tasks which would otherwise have been very difficult and time consuming to write, such as user identification, login, localisation and data management. Without UNITY to handle this, the game would easily have taken twice as long to write.

---

## What Went Wrong

**1. Text.** The amount of text I had to write for this game almost killed me. There are 500 unique diary entries for each of the 6 pets in the WAP version, and 350 unique diary entries for all six pets (featuring four new pets) in the SMS game. It was insane; 50 different ways of saying "I'm hungry" for ten different animals, each in a way that conveys a bit of personality.

One solution to this would be dynamic text generation. However, when the game is translated into many languages, this is very problematic. English is fine, but other languages have new and unusual forms of sentence construction to torment you with, not to mention gender specific nouns, verbs and adjectives (not necessarily the same gender in every language). The programmers had lots to do; as well as producing the game, I was doing the art and text, and there's virtually no art. To get it out the door ASAP, I took it upon myself to complete what I did not realise would be such a monumental task.

There are over 40,000 words of text between the text games. I wrote my first novel without realising it. I'm never doing that again. Localization cost a fortune too; thankfully the publisher shouldered that burden.

**2. Testing.** Testing was something that really should have gone better. At the time, we had no way to load test the WAP version of the game. So, when the game was launched on the first operator, they load tested the game on Friday, simulating 10,000 users accessing it at once. It fell over, and what's more, the game was due to launch on Monday. Monday couldn't be pushed back as the operator had already done loads of marketing to users. Let me tell you, that was one busy weekend, but we got it done.

Even the tiniest flaw adds up to big problems when there are so many users accessing the game. Now, DB have load testing tools and also better guidelines on garbage collection and preventing memory leaks. In fact, these were ready for the SMS release of the game, and we had no problems with the launch of that.

**3. Limited Scope.** One of my personal disappointments about the project is the lack of any real multi-player mode. Since everyone is playing on the same server (or more accurately, three servers), the system is set up nicely for some sort of user-user interaction. However, we had a limited timescale and budget, and we were pushing the platform as it was, so we shelved the concepts we had.

The only part of the game that is in any way multi-player is the pet shows. They are very simple though; users choose to enter their pet in the shows, and at midnight on Sunday all the entered pets are compared using a weighted total of all their stats. The winner, and those within a certain number of points of the winner, get extra in-game money as prizes. Although it is multi-player, it doesn't feature any user-user interaction.

It's a shame really as, since the devices have to be connected to a network in order to play the game, it seems obvious to use this (perhaps the only) advantage the device has over other gaming devices to create a better user experience. Time and budget won out, but on the plus side, keeping the scope limited did mean that we delivered on time.

**4. Publisher friction.** Now for everyone's favourite bit of the postmortems; the bitchy developer's revenge. Actually, I'm sorry to disappoint those of you looking for a slagging match, but DB is not at all bad to work with, it's just that some conflicts of interest arose. The issues have now been resolved, we work with DB regularly, and we have a constructive and friendly working relationship.

One feature of the contract is that The Games Kitchen would retain the *Wireless Pets* IP. To begin with, DB used the characters, logos and other materials on a number of occasions without our permission or approval. This was the cause of constant consternation, and some heated emails were exchanged. Now DB has an approval loop system in place and know where they stand when using external IP. Our disagreement with DB at the time meant that they had their system in place before they worked with IP belonging to companies who go around packing fully loaded lawyers and who aren't afraid to use them.

Another issue that arose requires a little more explanation of WAP, so bear with me. When WAP was released, many operators were unprepared for it and had not implemented any sort of billing system. Since implementing such a system would mean that they'd only have to pay out to content suppliers, they were in no rush to do it. DB were understandably keen to get into as many operators as possible, and so gave many of them the game for free on the understanding that they would pay when their billing system was implemented. Of course, this made implementing the billing system even less attractive for operators, and we missed out on quite a few months of royalties because of this.

We felt that DB should have stood firmer, though we do understand their point of view, and certainly it achieved their aim of being on many operators. We are happy to report that all operators (or at least all those Wireless Pets is on) now have their WAP billing mechanism sorted, so this problem has gone away, and in the long term DBs tactic has achieved the same end result. Although we were unhappy at the time, we look back without any anger. This has never been an issue for SMS; since text messaging has been popular for some time, billing has been in place for quite a while.

All of the friction arose as both The Games Kitchen and DB were young companies at the time. They did not yet have their systems in place for working with IP, and we wanted the royalties too quickly. Both companies are better for the experience and a lot wiser on developer-publisher relations, a fact evidenced by our good relationship today.

**5. Ongoing support requirements.** One of the features of operators implementing their billing systems was that we sometimes needed to change the code so that it would register billing points and so on. As new devices came out, the code would sometimes need changed. Annoyingly, future releases of UNITY were not always backwardly compatible, so on occasion we needed to change the code as the platform changed. The data was also designed to be flexible, so we could add new pets and objects in over time, which we did, including a promotional pet for the Big Brother game show.

All of this meant that the support required was greater than we had originally anticipated, and sometimes ate into other projects that we had running for DB. Generally, DB was quite accepting of the need to compromise on support and new development, though there was some understandable friction from time to time. Knowing what we know now, we should have allocated more time for support of *Wireless Pets* to prevent any future project slippage.

## Conclusion

*Wireless Pets* was a landmark title for us. It was The Games Kitchen's first major project, and it was an incredible success, immediately making a name for us in this emerging market. For DB, it proved their business model worked and that there is a market for mobile games.

*Wireless Pets* is good software delivered on time, and that has certainly given us the confidence to tackle bigger projects. This is essential for us as a company; mobile devices are getting more and more powerful and complex, and better and better technologies are emerging. We learnt a lot which has stood us in good stead in our future projects.

**Game Data**

***Wireless Pets***  
[The Games Kitchen](#)

**Publisher:** Digital Bridges

**Team Size:** 3

**Estimated Budget:** \$40,000

**Length of Development:** 7 months (total for both versions)

**Release Date:** March 2001 (WAP), March 2002 (SMS)

**Platforms:** WAP, SMS

**Development Hardware:** Linux based PCs for coding, Windows PCs for design, art and management

**Development Software:** Corel Draw, jCVS, NetBeans, Nokia MobileInternet Toolkit, PaintShop Pro, PyWeb Deckit, UNITY,

**Project Size:** 17,000 lines of code, 40,000 words of in-game text and 290 graphics (total for both versions)

[Return to the full version of this article](#)

Copyright © 2016 UBM Tech, All rights reserved