

La Programmation
sur
Pocket PC

Benedetti Benoit Riitano Stéphanie

Juin 2004

Avant-propos

Le logiciel envahit notre vie quotidienne: On le retrouve dans nos cartes à puces, nos organisateurs, nos téléphones, nos voitures, nos domiciles. . . , et l'évolution des différents matériels n'est pas pour ralentir ce phénomène, bien au contraire. La diversité de l'offre et des terminaux aux capacités restreintes rendent difficile la mise en place d'un environnement de développement: Comment fournir un langage et des outils permettant de concevoir et réaliser des applications adaptables le plus facilement possible à l'ensemble de ces équipements, qu'ils soient mobiles ou embarqués, avec leurs caractéristiques propres?

Cette problématique est d'autant plus importante que la résoudre , et donc "unifier" ces différents appareils, permettrait à celui qui y parviendrait de prendre un avantage dans le domaine de la production.

programmation sur Pocket PC: Les organisateurs de poches ([Personnal Digital Assistant ou PDA](#)) ont connu un engouement prononcé ces dernières années de part leur ergonomie, la richesse des applications installées et installables ainsi que part leurs capacités multimédia qui en font des terminaux parfaits pour le déploiement d'applications graphiques.

Nous étudierons plus particulièrement le développement sur Pocket PC (organiseurs basé sur le système d'exploitation Windows Mobile 2003, lui-même basé sur Windows CE .NET 4.2, ces différentes notions étant détaillées dans le chapitre suivant) car son architecture nous permet d'aborder les notions évoquées précédemment:

- Quel modèle Microsoft a-t-il désiré mettre en place en réponse à la diversité des périphériques et quels outils met-il de fait à la disposition des développeurs pour Pocket PC?
- Quelles autres alternatives de développement pour Pocket PC existe-t-il?

Notre étude se décomposera donc ainsi:

- Généralités sur le Pocket PC:
 - Windows CE;
 - Caractéristiques d'un Pocket PC;
- Les solutions de développement:
 - Différents outils et langages disponibles pour Pocket PC
 - Approfondissements des outils officiels Windows
 - J2ME: une plateforme de développement à ambition multi-périphériques mobiles applicable aux Pocket PC
- Etude de cas:
 - Utilisation de l'environnement Visual Studio .NET 2003.
 - Utilisation de l'environnement SuperWaba.

Vous trouverez en annexe différentes références sur les Pocket PC qui vous permettront d'approfondir les différentes notions, environnements et outils abordés dans ce document.

Table des matières

Avant-propos	i
1 Windows CE .NET 4.2 et Pocket PC	1
2 Solutions de développement pour Pocket PC	7
3 Etude de cas	19
4 Conclusion de notre étude	25
A Références	27
Remerciements	31

Chapitre 1

Windows CE .NET 4.2 et Pocket PC

1.1 Windows CE .NET

En réponse à la grande diversité de terminaux mobiles et embarqués, Microsoft propose une approche orientée système pour ces appareils:

Windows CE est un système d'exploitation supportant plusieurs familles de processeurs (ARM, MIPS, SH...), qui disposent d'une faible empreinte mémoire et destinés à être embarqués sur des matériels très variés: périphériques mobiles, clients légers Windows, téléphones, passerelles, guichets automatiques, etc... A cette fin, il est constitué de modules que le constructeur peut assembler en fonction des caractéristiques de son matériel. Cette assemblage s'appelle une *configuration*.

Le système d'exploitation Windows CE intègre un sous-ensemble des API Win32 communes aux autres systèmes Windows 32 bits (9x, Me, NT4, 2000, XP). Bien qu'un certain nombre d'API supplémentaires permettant d'accéder aux spécificités matérielles des PDA telles que les écrans tactiles ont été rajoutées, le modèle de programmation reste commun: de nombreuses fonctions sont donc identiques d'un système à un autre.

En 2002, Windows CE a été rebaptisé **Windows CE .NET** (mais continue à être désigné plus simplement par Windows CE) car il supporte effectivement le Framework .NET, qui a connu un engouement ces derniers temps, et permet l'exécution de programmes écrits en Visual Basic .NET, C# ou encore J#. Nous reviendrons plus en détails sur ces différentes notions ultérieurement.

Il existe plusieurs configurations prédéfinies de Windows CE, correspondant à des profils matériels standards, définis en partenariat avec les constructeurs, dont les Pocket PC et les Smartphone font partie. Nous allons revenir sur la configuration Pocket PC dans le bref historique qui suit.

1.1.1 Historique

Révéle au grand public en 1996, Windows CE a permis à Microsoft de prendre pied dans le monde des PDA, dominé à l'époque par le Palm Pilot et ses dérivés, tournant tous sous le système Palm OS.

Dans ce contexte concurrentiel, la première version de Windows CE (v1.0) a connu un démarrage difficile. En effet, cette première mouture contenait de très nombreux défauts, parmi lesquels une lenteur générale particulièrement flagrante.

Néanmoins, une fois les premiers défauts de jeunesse corrigés, la nouvelle version (v2.0) a tout de même réussi à convaincre plusieurs constructeurs et les nouvelles machines sont apparues fin 1998, déclinées en deux gammes:

- Les *Handled Pc* (ou H/PC) équipés d'un écran en mode paysage (pouvant être tactile) et d'un clavier;

- Les *Palm-Sized PC* (ou PS/PC), équipés uniquement d'un écran tactile en mode portrait.

A partir de l'année 2000, une nouvelle version (v3.0) nommée *Pocket PC* a commencé à remplacer les anciennes machines de type PS/PC, et a réellement amorcé l'entrée de Microsoft sur le marché des PDA. iPaq notamment, la machine de Compaq, a connu un vif succès avec le million d'unités vendues atteint à la fin de la première année de lancement.

Fin 2001 est apparu le *Pocket PC 2002*, présentant une évolution quant au système d'exploitation. Ont suivi de près de nouvelles machines puissantes intégrant cette mise à jour.

En 2003 est sortie une version rebaptisée Windows Mobile 2003 pour Pocket PC (appelée *Pocket PC 2003*) basée sur Windows CE .NET 4.2: Elle est très orientée connectivité et intègre le support du Wifi et Bluetooth, technologies de réseau sans fil qui équipent de plus en plus les appareils mobiles.

Comme les noms des plateformes, des numéros de versions et les modèles des PDA équipés de Windows CE peuvent prêter à confusion, voici un tableau récapitulatif:

<i>Nom du modèle</i>	<i>Nom de la plateforme</i>	<i>Version de Windows CE</i>
H/PC 2.0	HPC	2.0
H/PC 3.0	HPC Pro	2.11
Palm-size PC 1.0	Palm PC	2.1
Palm-size PC 2.0	Palm PC	2.1
Pocket PC	Palm PC	3.0
Pocket PC 2002	Palm PC	3.01
Pocket PC 2003	Palm PC	4.2

TAB. 1.1 – *Différentes versions de Windows CE*

1.2 Caractéristiques du Pocket PC

Comme décrit précédemment, un Pocket PC est un organiseur de poche intégrant une configuration du système d'exploitation Windows CE, les modèles récents bénéficiant de la version Windows Mobile 2003.

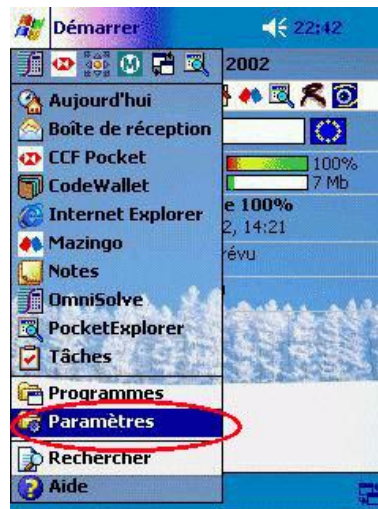


FIG. 1.1 – L'écran aujourd'hui et menu démarrer du Pocket PC

Un développeur doit tenir compte de certaines contraintes durant la phase d'élaboration de ces programmes:

L'écran: petitesse, résolution maximale faible, nombre variable de couleurs. . . ;

Le microprocesseur: 400MHz est un maximum actuellement et les différents types suivant les machines impliquent aux applications écrites dans des langages compilés d'être recompilés pour chaque microprocesseur cible;

Les mémoires: ROM et RAM présentent un grand maximum de 128 Mo.

Mais aussi de certaines caractéristiques propres à ces appareils:

- Ecran tactile, utilisation du stilet;
- Manipulation d'une main par contrôles de navigation;
- Wifi, Bluetooth, GPRS, infrarouge.

De plus, une application doit respecter un certain nombre de règles pour s'intégrer correctement au Pocket PC. Ces règles sont énoncées dans le cahier des charges "Designed for Windows for Pocket PC", dont nous vous présentons les grandes lignes.

Le Pocket PC définit un modèle particulier pour le cycle de vie des applications qu'il héberge: elles peuvent être lancées mais pas arrêtées manuellement. Elles restent en mémoire, et l'utilisateur passe de l'une à l'autre comme s'il les lançait, par le menu **Démarrer** ou par le dossier **Programmes**. En cas de manque de mémoire, le système ferme automatiquement l'application la moins récemment utilisée.

Ce mécanisme offre à l'utilisateur une sensation de fluidité et de disponibilité, lui évitant d'avoir à utiliser une liste de tâches. Mais il impose que les applications ne soient instanciées qu'une fois et qu'elles sauvegardent leur état. Leur lancement ou leur réactivation doivent avoir le même effet. De plus, l'application doit pouvoir être arrêtée par le système correctement, sans affichage de boîte de dialogue ou traitement consommateur de ressources.

Le bouton **Smart Minimize** (*Réduction Intelligente*), en forme de **X** à droite dans la barre de navigation, existe dans toutes applications, sauf dans l'écran **Aujourd'hui**, pour les réduire facilement et afficher au final cet écran. Une application reste en mémoire et peut être réactivée à tout moment.

La barre de navigation doit afficher en permanence le titre de l'application, et ne doit pas être exploitée à des fins spécifiques, pour afficher des boutons, par exemple.

La barre de commande, située en bas de l'écran, cumule les fonctions de barre de menu et d'outils. Les éléments de menu étant affichés à gauche, et les boutons à droite. Il ne doit pas y avoir de menu **Fichiers**, les menus ne doivent pas comporter d'accélérateurs, les raccourcis éventuels ne doivent pas être affichés, les opérations et boutons standards doivent respecter un ordre d'affichage. . .

Dans un souci de cohérence ergonomique et d'économie de mémoire, les ressources fournies par le système (boîtes de dialogues, alarmes. . .) doivent être utilisées et non reproduites.

Le bouton **SIP** (*Soft Input Panel*) affiche les outils de saisie (clavier ou reconnaissance d'écriture). Les applications doivent prendre en compte l'espace que peut consommer le SIP (80 pixels), soit en plaçant toutes les zones de saisies en dehors de cette zone, soit en proposant un ascenseur. Les onglets et autres éléments de l'interface doivent être retaillés de manière à être pleinement fonctionnels lorsque le SIP est affiché. Les boîtes de dialogues doivent s'afficher dans la zone restant libre lors de l'affichage du SIP.

API de référence du Pocket PC de la librairie MSDN:

- ActiveSync;
- Bluetooth API;
- CE Messaging (CEMAPI);
- Configuration Service Providers;
- Connection Manager;
- Control API;
- Control Panel Extension DLL Definitions;
- Device Management API;
- Game API (GAPI);
- HTML Control;
- Inkx Control;
- Input Panel;
- MIDI;
- Object Exchange (OBEX);
- Pocket Outlook Object Model (POOM) API;
- Projects Control;
- Remote API (RAPI);
- RichInk Control;
- Telephony;
- Today Screen;
- User Interface;
- Voice Recorder Control.

1.3 Conclusion

Un bref aperçu de ces règles et caractéristiques met en évidence la configuration mise en place par Microsoft pour les Pocket PC (et tout autre matériel tournant sur Windows CE), configuration qui a un impact sur les différentes solutions de développement existantes: Que ce soit celles mises à disposition par Microsoft et s'intégrant, de fait, parfaitement à de telles machines, ou encore

d'autres proposées par des sociétés tiers. Nous vous proposons un rapide tour d'horizon de ces différentes solutions dans le chapitre suivant.

Chapitre 2

Solutions de développement pour Pocket PC

On peut distinguer 2 sortes d'outils de développement: Ceux qui permettent de développer directement sur le Pocket PC (rares et peu pratiques pour la saisie de code) et ceux qui nécessitent une phase de développement sur le PC de bureau pour ensuite être déployés sur l'appareil. Nous allons présenter différentes solutions dominantes du moment appartenant à ces deux catégories, la deuxième étant la plus représentée pour le moment.

2.1 Les outils officiels Microsoft

Les tous premiers kits de développement pour Windows CE ont été fournis par Microsoft sous forme de packages à intégrer à Visual C++ et Visual Basic. Ces premiers SDK gratuits nécessitaient néanmoins de posséder une licence de Visual Studio. Dans l'optique d'inciter des développeurs à écrire des applications pour Windows CE, ces SDK ont été remplacés par une version autonome de Visual Studio, baptisé *eMbedded Visual Tools 3.0*, gratuite. Ont suivi, avec l'explosion de la plateforme .NET, des outils adaptés à cette plateforme au travers de *eMbedded Visual C++ 4.0* et d'extensions apportées à Visual Studio .NET.

2.1.1 eMbedded Visual Tools 3.0

Cet outil permet de réaliser des applications en Visual C++ (*eMbedded Visual C++*) ou en Visual Basic (*eMbedded Visual Basic*). Il est composé de SDK additionnels spécifiques à chaque plateforme, dont le Pocket PC, et d'un environnement de développement intégré ressemblant très fortement à Visual Studio 6.0 (pour la partie C++) et à Visual Basic (pour la partie VB) comme en témoignent ces captures d'écrans:

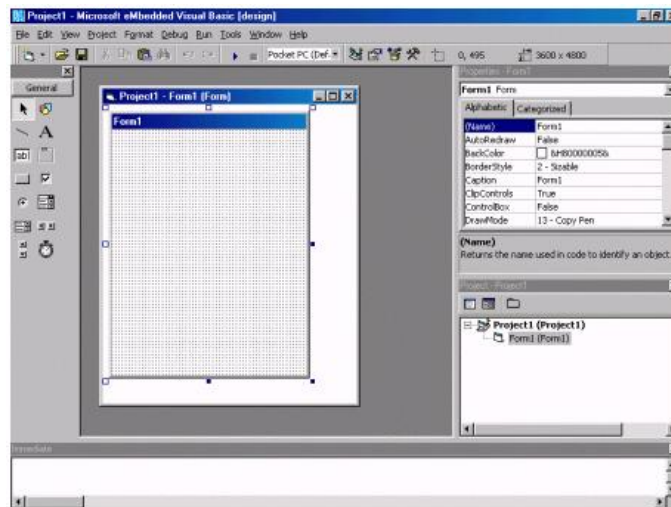


FIG. 2.1 – l'environnement eMbedded Visual Basic

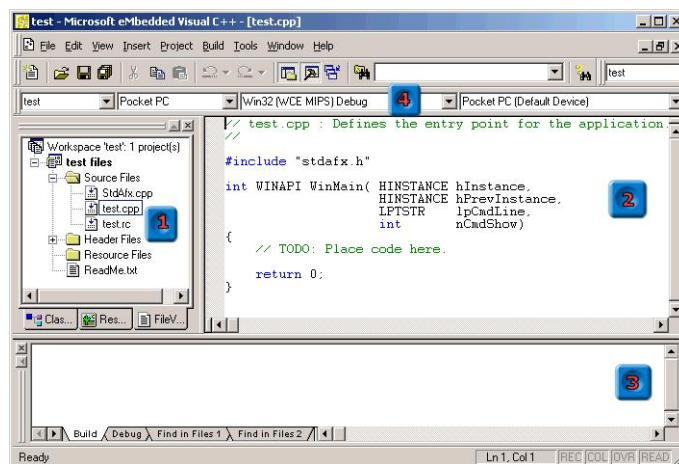


FIG. 2.2 – l'environnement eMbedded Visual C++

eMbedded Visual Basic

En raison de plusieurs facteurs, dont l'intérêt croissant que porte la communauté des développeurs pour Visual Basic .NET, cette technologie tend à être laissée de côté: Utilisez Visual Basic .NET pour créer de nouvelles applications Pocket PC et cette version pour la maintenance d'anciennes applications.

eMbedded Visual C++

Utilisez cet outil pour:

- des applications natives;
- des pilotes;
- des jeux nécessitant des performances accrues;
- des applications fortes consommatrices de ressources

Les fonctionnalités de cet environnement présente des similarités avec leurs homologues pour Windows, plus quelques spécificités:

- Un espace de travail qui regroupe les différents projets;
- un éditeur de textes;
- un éditeur de ressources;
- une aide en ligne;
- des émulateurs pour tester directement les applications;
- des outils distants: pour visualiser les objets créés, pour éditer la base de registre. . .
- un *Platform Manager* permettant de configurer et gérer les communications entre les outils distants, le débogger et les machines cibles.

On s'aperçoit donc que tout développeur avec des connaissances en C++ et de l'API Win32 voire de Visual C++ pourra écrire rapidement des applications avec un tel outil. Malheureusement, ces outils ne sont pas compatibles avec la nouvelle mouture de Windows CE et font place à de nouveaux outils plus orientés .NET.

2.2 Développement .NET

L'architecture .NET distingue deux types de développement:

Les développements natifs: Permettent de réaliser des applications exécutables directement sur la machine cible;

Les développements .NET: Fournissent du code binaire intermédiaire exécuté par une machine virtuelle.

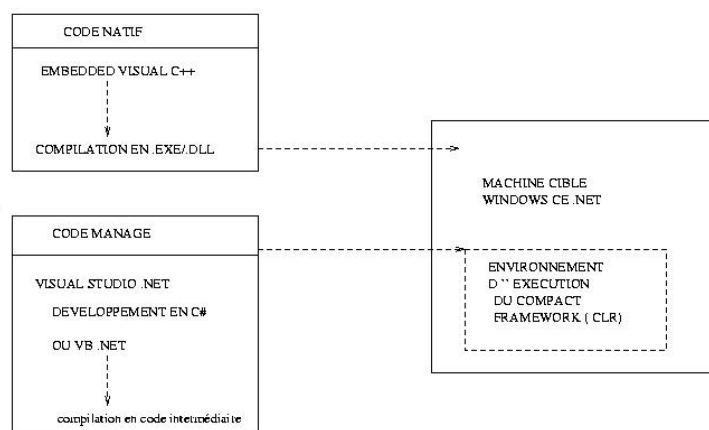


FIG. 2.3 – Les modes de développement .NET

Le Framework .NET est l'association d'une infrastructure logicielle, le CLR (*Common Language Runtime*) et d'une bibliothèque de classe, la *Base Class Library*. Le Framework permet l'exécution de programmes écrits dans un langage .NET: Visual Basic .NET, C#, J# et C++.

2.2.1 eMbedded Visual C++ 4.0

La partie eMbedded Visual Basic a donc complètement disparu, quant à eMbedded Visual C++ 4.0 vous pourrez développer, pour les derniers modèles de Pocket PC équipés de Windows Mobile 2003, du code natif pour cette nouvelle version.

L'environnement de travail et les fonctionnalités restent similaires à son prédécesseur, avec quelques améliorations:

- Intégration des mécanismes d'exceptions C++;
- Amélioration du debugger;
- Amélioration des émulateurs;
- etc...

La prise en main d'un tel outil ne s'en verra que facilitée.

2.2.2 Visual Studio 2003

Le Compact Framework

Le Compact Framework est une adaptation du Framework .NET au Pocket PC et Smartphone. Il en constitue en fait une version allégée, et démontre ainsi la capacité multiplate-forme de .NET.

Les fonctionnalités du Framework ont été reprises dans leur ensemble en écartant les classes et méthodes non indispensables, pour limiter au maximum l'empreinte mémoire. Par exemple, dans le Framework, une méthode est souvent déclignée avec plusieurs surcharges. Dans le Compact Framework, une seule version de la méthode est généralement conservée (celle qui attend la totalité des arguments possibles).

Extensions de Visual Studio 2003

Visual Studio 2003 fusionne le Framework et le Compact Framework au travers du *Smart Device Programmability* (SDP), et permet de développer des applications pour Pocket PC 2002 en C# et Visual Basic .NET en utilisant les mêmes outils et techniques que pour des ordinateurs de bureau. Le développement d'applications managées s'en trouvent donc optimisé. Vous devrez toutefois installer le SDK Pocket PC 2003, sorti après Visual Studio 2003, pour développer des applications pour Pocket PC 2003. La base de données Microsoft dédiée au Pocket PC est SQL CE, présentée comme une adaptation de SQL Server 2000. Le développement d'applications Windows avec .NET repose sur les *Windows Forms*, qui constituent une partie des classes du Framework, et ont été portées sur le Compact Framework pour le Pocket PC et le Smartphone. Ces classes comportent tous les éléments pour réaliser des applications de type GUI (*Graphical user Interface*). Les applications web (*Web Forms*) pour ASP.NET ont été adaptées au Compact Framework pour construire des applications mobiles supportées par une large gamme de terminaux mobiles.

Au vue de ces différents outils, on s'aperçoit que Microsoft a fait un réel effort pour permettre aux développeurs de travailler sur des terminaux mobiles. Mais, comme à son habitude, cette volonté s'inscrit dans une politique de fidélisation pour promouvoir et inonder la communauté de ses produits. Nous allons maintenant nous intéresser aux autres alternatives de développement disponibles pour Pocket PC.

2.3 La programmation Java sur Pocket PC

La plate-forme de développement Java ayant suscité un fort engouement ces dernières années, il n'est pas étonnant de trouver plusieurs implémentations de machines virtuelles Java pour les systèmes Windows CE. Nous allons d'abord revenir sur la politique de Sun pour l'embarqué connue sous le nom de J2ME (*Java 2 Micro Edition*) qui nous permettra de mieux appréhender les différentes solutions pour Pocket PC.

2.3.1 La politique multi-périphériques de Java

J2ME a pour but de fournir un langage reconnu, Java, et des outils permettant de concevoir et réaliser des applications adaptable le plus facilement possible sur l'ensemble des équipements

mobiles ou embarqués. J2ME n'est pas une nouvelle technologie spécifiquement internet comme pouvait l'être les applets ou J2EE, mais il s'agit d'une réponse au besoin de concevoir des applications de plus en plus puissantes, sûres et rapidement pour des systèmes de plus en plus complexes et présents.

L'une des grandes particularités de J2ME est qu'il cible aussi bien les terminaux mobiles, comme les téléphones ou les PDA, que les systèmes embarqués, tels les systèmes informatiques se trouvant dans les voitures ou à notre domicile. Pour cela, des catégories de terminaux et de systèmes ont été établies, auxquelles correspond ce que l'on appelle une *configuration*, ou couche logique d'abstraction. A l'intérieur de ces catégories, un *profil* est adapté à chaque type de terminal pour correspondre au plus petit dénominateur commun des fonctionnalités de ces terminaux. De la sorte, une application tournant sur un téléphone J2ME peut également tourner sur un Palm Pilot sans adaptation, en s'appuyant sur un profil commun, le MIDP.

On en retire que les briques de base de J2ME sont la configuration, le profil et les packages optionnels:

- Une configuration est une machine virtuelle et un ensemble minimal de classes de base et d'API. Elle spécifie un environnement d'exécution généralisé pour les terminaux embarqués et agit comme plate-forme Java sur le terminal;
- Un profil est une spécification des API Java définie par l'industrie et utilisée par fabricants et développeurs à destination des différents types de terminaux spécifiques;
- Un package optionnel est, comme son nom l'indique, un package qui peut ne pas être implémenté sur un terminal particulier.

Les différences entre les API J2ME et J2SE résident essentiellement dans les classes, le tableau suivant dressant la liste des classes de l'API J2ME:

Classe	Description
java.util.TimeZone	Gestion de la zone horaire
java.util.Date	Gestion des dates
java.util.calendar	Gestion des calendriers
java.lang.thread	Gestion des threads
java.lang.System	Gestion des systèmes

TAB. 2.1 – Les classes de l'API J2ME

J2ME comprend aujourd'hui deux configurations: CLDC et CDC.

La configuration CLDC

Le rôle de CLDC (*Connected Limited Device Configuration*) est de définir une plate-forme Java standard adaptée aux terminaux légers dotés de peu de ressources et fonctionnant en mode connecté: téléphones, anciens PDA... La configuration CLDC ne définit que les bases communes à l'ensemble des terminaux: entrées-sorties, réseau, sécurité, internationalisation. En ce qui concerne les fonctionnalités de plus haut niveau, c'est au profil spécifique du terminal de les prendre en charge: gestion du cycle de vie de l'application, interface utilisateur, gestion des événements. Les classes héritées de J2SE par CLDC proviennent des packages java.lang, java.io et java.util. un certain nombre ont été exclues notamment java.lang.applet, java.lang.awt, java.lang.rmi, java.lang.security et java.lang.sql ainsi que les classes Swing et Corba. Elles ont été écartées, entre autre, pour des limitations de la machine virtuelle ou parce-que trop consommatrices de mémoire.

CLDC n'hérite que de quelques classes de J2SE dans le domaine de la connectivité réseau. De nouvelles classes ont été définies et regroupées sous le nom de GFC (generic Connection Framework) dans le package javax.microedition.io. La configuration CLDC est la base du profil MIDP (*Mobile Information Device Profile*). Un MIDlet est une application développée pour MIDP de façon analogue à une applet pour le web: elle possède un cycle de vie et à pour conteneur un téléphone mobile par exemple. Compte tenu du nombre limité de classes disponibles dans les API MIDP, le dévelop-

pement de MIDlets avec J2ME est plus simple que le développement d'application traditionnelles en J2SE.

La configuration CDC

La configuration CDC (*Connected Device Configuration*) fournit une machine virtuelle Java 2 complète, appelée CVM (*Convergence Virtual Machine*), ainsi qu'une bibliothèque de classes minimale et les API permettant de porter Java sur les terminaux embarqués de type CDC. Le CDC est un superensemble de CDLC. A ce titre, un profil compatible CLDC est aussi compatible CDC. L'API CDC comprend un sous-ensemble minimal de l'API de J2SE permettant de supporter une machine virtuelle Java, comprenant les packages suivant :

Classe	Description
java.lang	Système de la machine virtuelle
java.util	Utilitaires
java.net	Datagramme UDP et entrées-sorties de fichiers réseau
java.text	support de l'internationalisation (message d'erreur, etc. . .)
java.security	Support de la sécurité et du cryptage

TAB. 2.2 – Packages de J2SE repris dans la configuration CDC

Il existe plusieurs profils CDC. Le premier est le Foundation Profile, sorti en même temps que CDC. Il le complète en ajoutant de nombreuses bibliothèques absentes de J2SE. Comme son nom l'indique, ce profil est censé servir de base à d'autres profils. L'API PersonalJava a été redéfinie comme étant le Personal Profile et repose sur le Foundation Profile.

Le Foundation Profile est un ensemble d'API Java qui forment avec CDC un environnement d'exécution J2ME complet destiné aux terminaux embarqués. Les implémentations de ce profil utilisent un sous-ensemble de la plate-forme J2SE, le package java.awt ayant disparu entre autre.

Classe	Description
java.lang	Support du langage Java
java.util	Utilitaires
java.net	Ajoute les connections socket et HTTP TCP/IP
java.io	Support des entrées-sorties du langage Java
java.text	support de l'internationalisation (message d'erreur, etc. . .)
java.security	Ajoute la signature du code et les certificats

TAB. 2.3 – Packages de J2SE repris dans le Foundation Profile

il est à noter que l'API du Foundation Profile est fondée sur celles de J2SE 1.3.

Le PJAE (*PersonalJava Application environment*) est un environnement d'exécution d'application Java destiné à tourner en environnement réseau sur des terminaux cosommateurs personnels tels les PDA. L'API du PJAE repose sur l'API du JDK 1.1.6 comprenant essentiellement les packages des applets, de l'interface utilisateur, des Beans, des entrées-sorties, des mathématiques, de la gestion du réseau, de l'accès à une base de données SQL et de RMI avec quelques modifications (java.awt a de nombreuses classes et méthodes modifiées afin de fournir les fonctionnalités correspondant aux capacités du système graphique et de fenêtrage du terminal). Le PJAE a été modifié et rebaptisé dernièrement Personal Basis Profile et un autre profil, surper-ensemble de celui-ci, a été élaboré: le Personal Profile bénéficiant de nouveaux apports:

- Gestion des composants lourds et légers de AWT;
- Gestion des entiers et déimaux longs;
- Gestion des beans;
- Gestion des Xlets. . .

2.3.2 Conclusion

Sun a essayé de fournir plusieurs configurations et profils pour sa plate-forme J2ME, par contre il semble qu'il fasse peu d'efforts au niveau de l'implémentation des configurations adaptables au Pocket PC et vous serez obligé de passer par des sociétés tiers pour avoir une machine virtuelle java compatible J2ME voire des outils de développements.

2.3.3 Les machines virtuelles Java

IBM Websphere Micro Environment

De son côté, IBM propose sa propre solution compatible J2ME avec le *Websphere Micro Environment* (la machine virtuelle J9) et l'outil de développement *Websphere Studio Device Developer*. Cet outil présente l'énorme avantage de proposer des machines virtuelles pour de nombreux ordinateurs équipés d'OS différents notamment pour les Pocket PC équipés de processeurs ARM.

<http://www-306.ibm.com/software/info1/websphere/index.jsp>

Jbed

La société Esmertec propose des machines virtuelles compatibles aux configurations J2ME (Jbed CLDC et Jbed CDC). Il est à noter que l'implémentation de la partie AWT a été faite pour tirer au maximum parti des spécificités de l'interface graphique de Windows CE. D'autre part, le produit intègre un support complet de JNI (*Java Native Interface*) qui permet aux développeurs d'utiliser toutes les fonctions système à partir des classes Java. De plus, cette machine virtuelle semble disposer d'une très forte optimisation des performances (c'est avec l'ancienne version de cette JVM appelée Jeode et anciennement vendue par Insignia que nous avons pu tester des applications java car cette JVM est présente sur le cd d'installation du matériel qui nous a été confié. Esmertec a depuis racheté Insignia).

<http://www.esmertec.com/>

NSIcom CrEme

NSIcom édite aussi une machine virtuelle fondée sur PersonalJava. Cet outil offre un certain nombre de fonctionnalités supplémentaires, telles qu'un module d'optimisation fondé sur la technologie JIT, un sous-système de gestion mémoire paginée, une version améliorée du toolkit graphique *Truffle AWT*. De plus, CrEme intègre les packages java JNI, RMI, Swing, SQL, Math, Zip/Jar et java.security.

<http://www.nsi.com/products/creme.asp>

Waba et SuperWaba

La société Wabasoft propose une machine virtuelle sous licence GPL, déclinée en deux versions. La première, Waba, est une machine virtuelle minimale développée pour des systèmes vraiment légers. Entre autres, SuperWaba, distribuée sous licence GPL, apporte de nombreuses améliorations à ce noyau initial, tant au niveau des performances qu'au niveau du support des interfaces utilisateurs spécifiques à chaque OS cibles dont Windows CE. Site officiel SuperWaba:

<http://www.superwaba.com.br/fr/default.asp>

Site français:

<http://www.superwaba-france.com>

Penbase

Avec son outil *Penbase Designer*, Penbase propose un environnement de développement d'applications mobiles de bases de données pour Pocket PC. Il intègre un moteur de base de données relationnelle, *Penbase Engine*, particulièrement compact (environ 90Ko), une machine virtuelle

Java, *Penbase Virtual Machine*, et le langage *PenJ* implémentant un sous-ensemble du langage Java et de l'API Swing. <http://www.penbase.com/product/pusdesigner.html>

Conclusion

2.3.4 Les autres outils et langages disponibles

Syllem MPS

La société Syllem, anciennement Palmware, propose le *Multi-Platform System*, MPS, qui permet de réaliser des applications qui s'exécuteront sur une machine virtuelle propriétaire, comparables aux JVM. Cet environnement est disponible pour de nombreuses plate-formes, y compris les Pocket PC. Le développement se fait à partir d'un IDE, le *Visual Application Generator 2*, qui supporte nombre de langages de programmation connus.

http://www.syllem.com/Products/Product_Overview.asp?Id=2

PenRight Mobile Builder

Syware Visual CE

Syware propose un outil de création de formulaires et de bases de données pour le système de base de données intégré du Pocket PC. La synchronisation des données via le PC de bureau peut être effectuée avec n'importe quelle source ODBC.

http://www.syware.com/prodlib/win_ce/vce/vce.htm

NSBasic-CE

NSBasic est un outil qui permet de développer directement sur le PDA. Comme son nom l'indique, il utilise le langage de programmation BASIC. Un environnement de développement pour le PC de bureau est tout de même présent, l'édition de programmes sérieux restant difficile sur les PDA. Le noyau de NSBasic repose sur le moteur Microsoft VBScript, intégré dans Windows CE, enrichi d'un certain nombre d'extensions. À noter que ce produit s'est vu décerné le prix du meilleur outil de développement pour mobile 2003 par Microsoft.

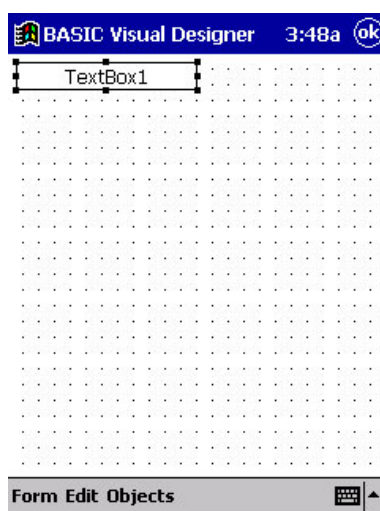


FIG. 2.4 – une capture d'écran du logiciel sur le Pocket PC

PocketC

La société *Orb Works* propose un compilateur C pouvant tourner sur Pocket PC. Il est à noter que le compilateur n'est pas conforme ANSI. Le développement peut être réalisé directement sur le PDA mais l'outil prévoit un IDE pour le développement sur le PC de bureau nommé *PocketC Desktop Edition*.



FIG. 2.5 – une capture d'écran du logiciel sur le Pocket PC

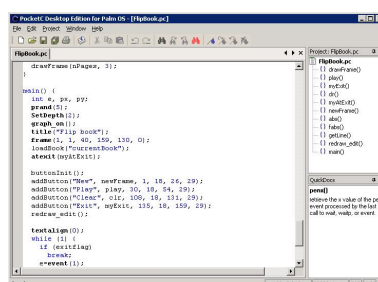


FIG. 2.6 – une capture d'écran de l'IDE

<http://www.orbworks.com/pcpalm/index.html>

Python

Des portages de Python ont été effectués pour Pocket PC. Il suffit d'installer un fichier bin sur le Pocket PC, puis le programme préalablement compilé sur le PC de bureau.

http://www.python.org/download/download_windows.html

Pocket Scheme

Pocket Scheme propose une implémentation du langage Scheme pour Windows CE. Le package est accompagné d'un éditeur de texte permettant la création de scripts directement sur le PDA. Malheureusement, la dernière version ne supporte pas les Pocket PC 2003, mais des améliorations pour cette plate-forme sont en cours.

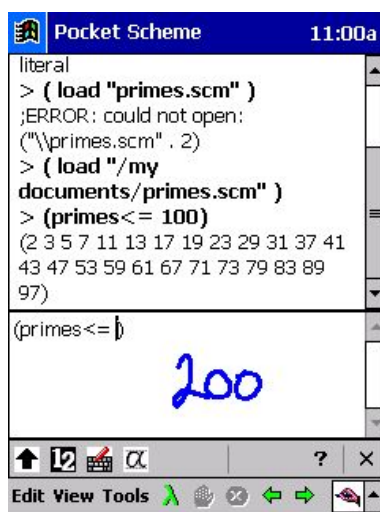


FIG. 2.7 – L'éditeur de texte de Pocket Scheme

<http://www.mazama.net/scheme/pscheme.htm>

JSoftware

Le langage J est un langage de programmation spécialisé pour la manipulation et l'analyse de données mathématiques complexes. Il hérite des fonctionnalités du langage APL et a été porté pour les Pocket PC équipés de processeurs ARM et MIPS.

<http://www.jsoftware.com/index.htm>

Dialect

Dialect est un langage de programmation interprété pour Windows et Windows CE. Les sources sont distribués sous une licence open-source de type Apache. Le code peut être précompilé pour optimiser ses performances.

<http://sourceforge.net/projects/dialect/>

REBOL

Le langage REBOL est un outil de programmation très axé sur les fonctionnalités Internet puisqu'il intègre de manière native de très nombreux protocoles de communication TCP/IP. Le package est disponible pour des dizaines de plate-formes dont Windows CE.

<http://www.rebol.com/>

Squeak

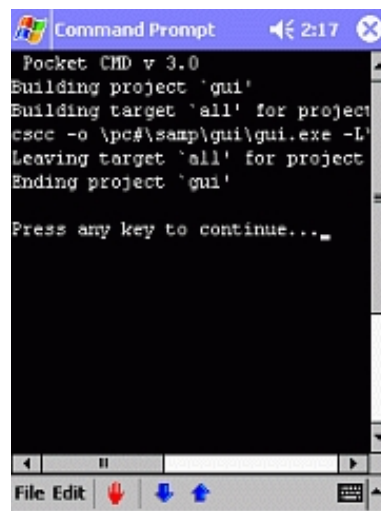
Le langage squeak est une implémentation open-source du langage de programmation Smalltalk-80, dont la partie environnement d'exécution est disponible pour de très nombreuses plate-formes dont Windows CE.

<http://www.squeak.org/about/index.html>

Pocket C#

Pocket C# est le portage pour Pocket PC du compilateur C# de DotGNU. IL propose un compilateur ainsi que les bibliothèques standards du Compact Framework .NET de Microsoft. Le développement a lieu directement sur le PDA.

<http://mifki.ru/pcsharp/index.html>



```
Command Prompt 2:17
Pocket CMD v 3.0
Building project 'gui'
Building target 'all' for project
csc -o \pc#\samp\gui\gui.exe -L'
Leaving target 'all' for project
Ending project 'gui'

Press any key to continue...
```

FIG. 2.8 – *Pocket C# sur le PDA*

Chapitre 3

Etude de cas

3.1 Un programme C# avec Visual Studio

Nous allons voir ici comment écrire un formulaire simple dans la tradition des HelloWorld pour prendre connaissance avec le développement d'applications pour Pocket PC avec Visual Studio 2003.

3.1.1 Création du projet

Lancez Visual Studio 2003 et choisissez un projet Visual C# et une application Smart Device en spécifiant un nom et un répertoire de projet.

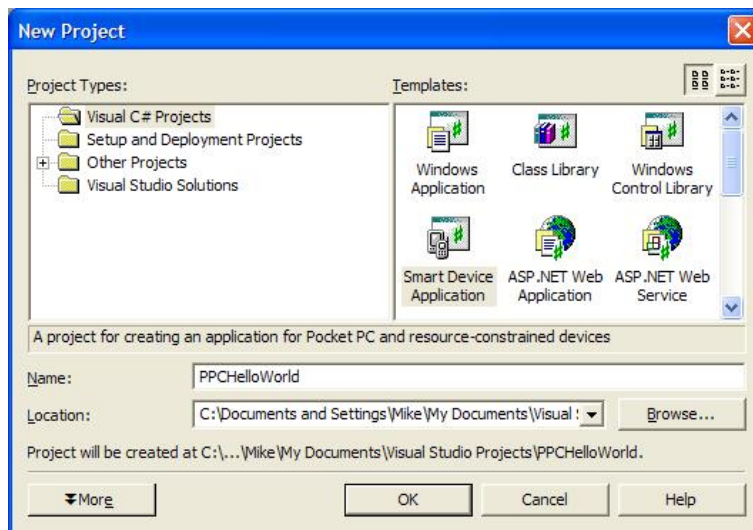


FIG. 3.1 – Fenêtre de configuration du projet

Dans le panneau suivant, choisissez Pocket PC comme plate-forme cible et application Windows comme type d'application.

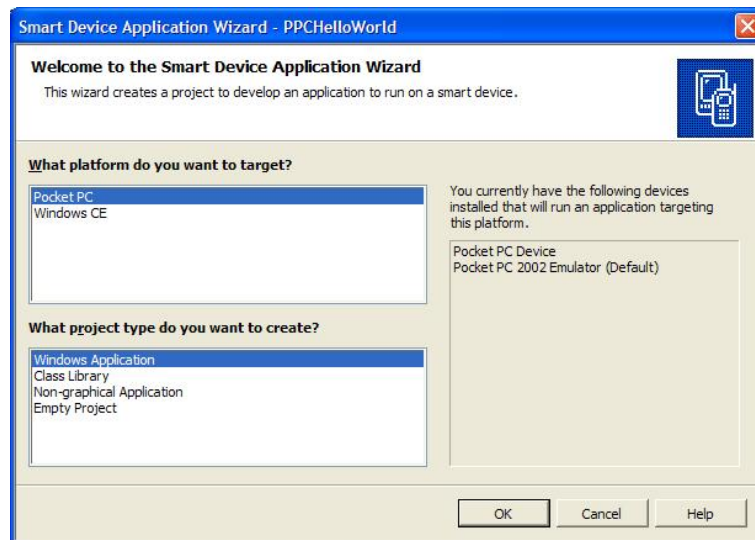


FIG. 3.2 – Fenêtre de configuration du projet

3.1.2 Le formulaire

Par défaut, un projet est créé, avec un formulaire de base. Dans la fenêtre Form1, cliquez en dessous sur l'objet menuItem1 et dans le formulaire, sur "Type Here":Ecrivez Hello World!. Vous venez de créer un objet de la barre menu du formulaire, reste à lier un événement à cet objet.

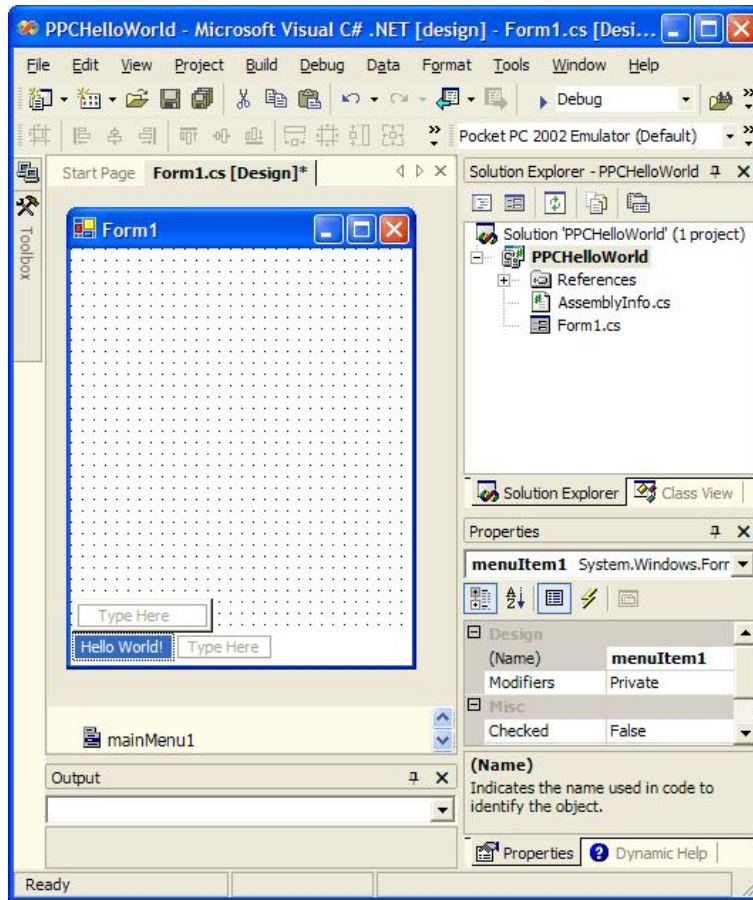


FIG. 3.3 – Fenêtre de configuration du projet

Double-cliquez sur l'objet Hello World pour créer un écouteur pour ce menu lorsqu'on le clique avec le stylet et écrivez : "MessageBox.Show("Hello World!");".

3.1.3 Le Déploiement

Dans la barre d'outils, sélectionnez la cible pour le déploiement : "Pocket PC 2003", puis construisez l'application en cliquant sur "build .cab" (bouton en forme d'armoire jaune dans la barre d'outils). Dans le répertoire de projet, récupérez le fichier .cab propre à votre processeur, copiez le sur le Pocket PC et procédez à l'installation en double-cliquant dessus.

Cliquez sur le programme installé : voilà vous venez de créer votre premier formulaire en C# qui affiche une boîte de dialogue lorsque vous cliquez sur le menu pour votre Pocket PC.

3.2 Un premier programme SuperWaba

Etant donné le peu d'efforts fournis par Sun dans l'implémentation d'une machine virtuelle répondant à la configuration CDC, nous avons choisi de vous montrer un exemple de compilation d'un programme compris dans les démos de SuperWaba, cette JVM faisant figure de référence avec une optimisation de la partie graphique associée à des outils performants, le tout gratuitement.

3.2.1 Installation de l'environnement

SDK

Installer le JDK de Sun (version 1.3 minimum).
 Téléchargez le SDK depuis la page de téléchargement du site officiel SuperWaba (nommée partie 1: <http://www.superwaba.com.br/en/downloads.asp>). Ce fichier zip doit juste être décompressé (disons à la racine de votre ordinateur de bureau).
 Téléchargez à la même adresse, la partie 2: c'est un fichier .zip qui contient les fichiers à installer sur votre Pocket PC: reliez votre PC et votre Pocket PC et cliquez sur l'installateur de ce fichier .zip (ou copiez le fichier .cab relatif à votre processeur sur votre Pocket PC et double-cliquez dessus pour l'installation automatique).

Installation de l'environnement de travail

Téléchargez l'IDE MobileCreator sur le site <http://www.tauschke.com/>. Cet outil vous facilitera le développement d'application (aide en ligne avec l'API SuperWaba, génération automatique d'exécutables...).

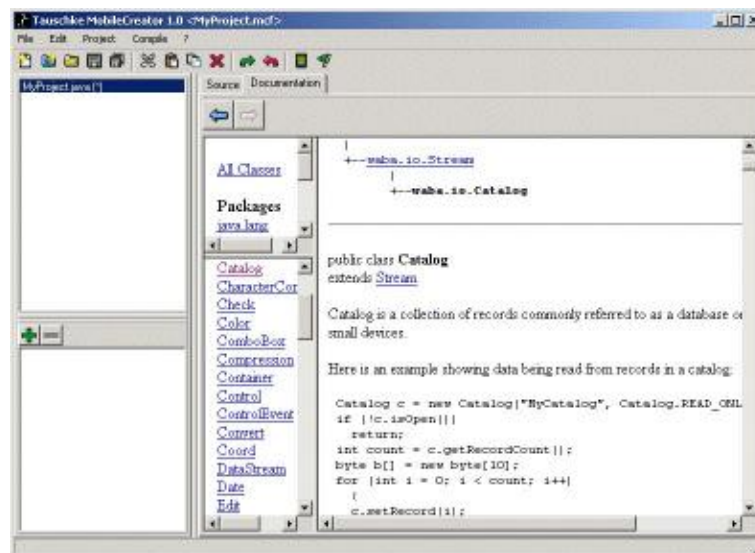


FIG. 3.4 – L'API de SuperWaba intégrée à Mobile Creator

Vous devrez configurer les chemins relatifs au JDK de Sun et au SDK de SuperWaba.

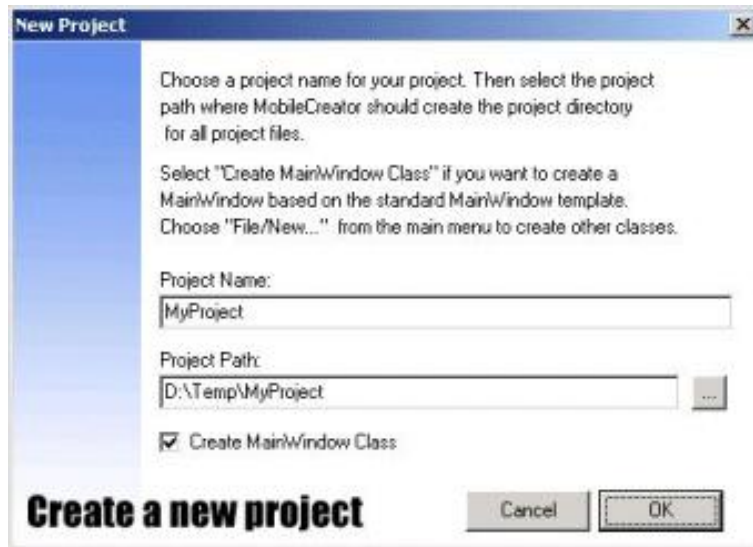


FIG. 3.5 – Fenêtre de configuration des chemins des SDK

Configurez ensuite les programmes Warp et Exegen: Ouvrez l'onglet, et cochez dans l'onglet Warp "start Warp" et dans l'onglet Exegen "start exegen" et "create .cab files for CE "; De cette manière, un fichier .cab à déployer sur le PDA se trouvera dans le répertoire du projet.

3.2.2 Création du projet

Lancez Mobile Creator et créez un nouveau projet (menu Project->New): Donnez lui le nom Ping (le nom de notre application et du main), donnez lui le chemin où vous voulez sauvegarder ce projet et décochez l'option *Create MainWindow Class* car notre projet possède déjà une classe Main.

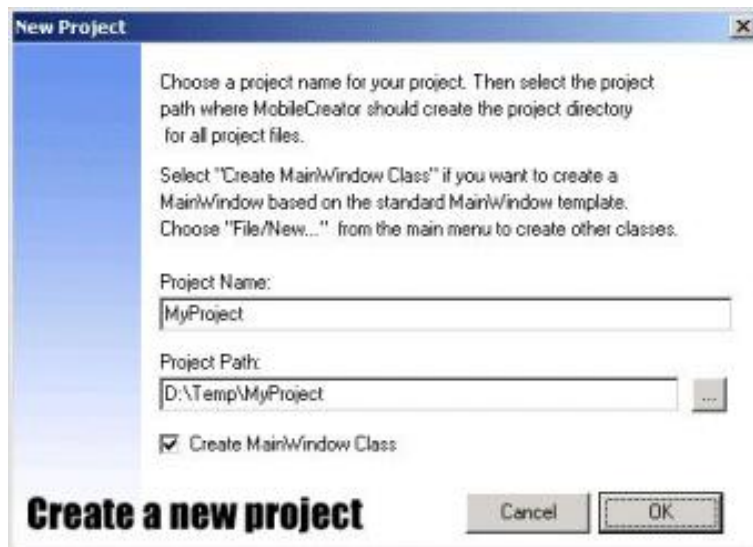


FIG. 3.6 – Fenêtre de configuration du projet

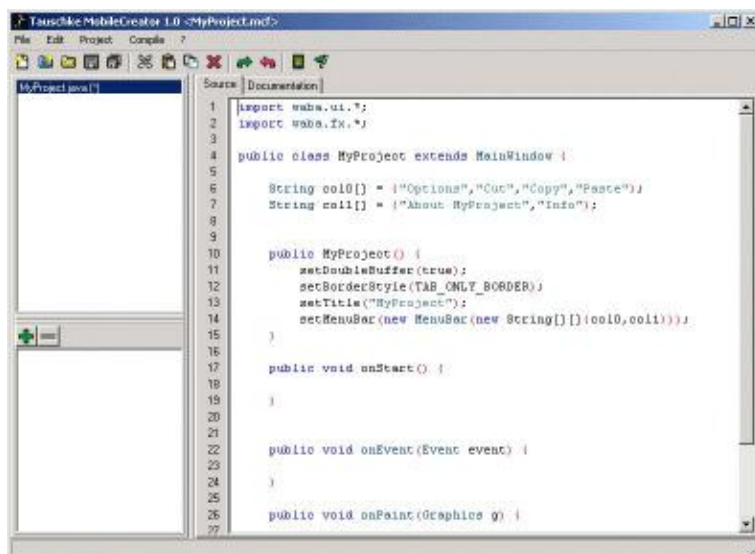


FIG. 3.7 – Fenêtre Principale de Mobile Creator

Importez les classes du projet par la commande (File->Open):Elles se trouvent dans le repertoire demo->games->ping du SDK SuperWaba (Vous pouvez importer toutes les classes d'un coup).

En bas à gauche se situe une fenêtre dans laquelle vous pouvez placer toutes ressources nécessaires au programme par le bouton "+" vert (images,son...): incorporez toutes les ressources du répertoire Ping dans cette fenêtre.

3.2.3 Déploiement

Ne reste plus qu'à compiler les classes et construire les fichiers .cab:cliquez sur le bouton "build" de la barre d'outil (le drapeau vert).

Vous devriez avoir à la racine du repertoire de votre projet des fichiers .cab correspondant à plusieurs machines et processeurs:

Récuperez le fichier Ping_PPC_ARM et copiez le sur le Pocket PC et double-cliquez dessus;les fichiers .pdb et .prc nécessaires à la JVM sont installés. Le programme Ping devrait être disponible dans le repertoire Program Files.

Voilà,vous avez découvert succinctement l'environnement, la compilation, la construction et le déploiement d'une application SuperWaba.

Chapitre 4

Conclusion de notre étude

Il ressort de notre étude que la programmation sur PDA , et donc sur Pocket PC, reste en pleine expansion. De plus, l'évolution de ces outils de développement conditionnera les futurs systèmes embarqués et mobiles de demain et vice-versa, tendant à élargir encore plus la gamme de ces équipements. On notera malheureusement que le monde des utilisateurs de Pocket PC reste dominé par des professionnels et que les personnes désireuses de fournir des outils de développement pour ces appareils sont des sociétés faisant payer des licences. Restent certaines solutions gratuites: les eMbedded Visual tools , mais permettant à Microsoft de populariser sa plateforme Windows CE et de fidéliser une communauté de développeurs, ou encore les offres proposées par Sun , mais celles-ci paraissent disparates et les nombreuses spécifications produites par Sun rendent difficile la compréhension des évolutions de la plate-forme Java pour appareils mobiles. Heureusement, la communauté des possesseurs de Pocket PC s'agrandit de jour en jour et nous pouvons que nous satisfaire de l'émergence de groupes de développeurs passionnés indépendants qui proposent des outils gratuits tel le projet SuperWaba.

Annexe A

Références

A.1 Portails et Communautés

A.1.1 Sites généralistes

Pocket PC paradise:<http://www.pocketpcparadise.com/>
L'univers du Pocket PC:<http://www.universpocketpc.com>
Pocket PC France:<http://www.ppcfrance.com/>
Windows CE .NET:<http://www.microsoft.com/windows/embedded/default.mspx>
Windows Pocket PC:<http://www.microsoft.com/windowsmobile/products/pocketpc/default.mspx>
J2ME:<http://java.sun.com/j2me/index.jsp>
Forum CDC:<http://forum.java.sun.com/wireless/forum.jsp?forum=80>
Forum CLDC:<http://forum.java.sun.com/wireless/forum.jsp?forum=76>

A.1.2 Sites sur le développement

Microsoft Developer Network (MSDN):<msdn.microsoft.com>
GotDotNet:www.gotdotnet.com
dotnet-fr:<http://dotnet-fr.org>
dotnetguru:<http://www.dotnetguru.org>
CodePPC:<http://www.codeppc.com>

A.1.3 Outils de développement

eMbedded Visual Tools 3.0, eMbedded Visual C++ 4.0, SDK Pocket PC 2003 pour VS .NET 2003, image française de l'émulateur...: <http://www.microsoft.com/windowsmobile/resources/downloads/developer/default.mspx>
Wireless toolkit (pour le développement de MIDlets): <http://java.sun.com/products/j2mewtoolkit/index.html>

Bibliographie

- [1] M. RUTKOWSKI. *Programmation Pocket PC*. Éd. CampusPress, Paris, 2002.
- [2] O. DEWIT. *.NET pour la Mobilité*. Éd. Eska, Paris, 2002.
- [3] B. DELB. *J2ME Applications Java pour terminaux mobiles*. Éd. Eyrolles, Paris, 2002.

Remerciements

Nous tenons à remercier Mr Buffa et le Département Informatique pour nous avoir confié les PDA et les différents logiciels qui ont été utilisés lors de l'élaboration de ce projet.