

Logiciel Pricer

BTS Informatique de Gestion
Option Développeur d'applications
Année 2002-2003

Réalisé par Florence CHAU

SOMMAIRE

Remerciement	page 1
---------------------------	---------------

1. L'entreprise

1.1 Statut juridique	page 2
1.2 Activité	page 2
1.3 Personnel	page 3
1.4 Equipement informatique	page 4

2. Mon rôle dans l'entreprise

2.1 Tâches effectuées	page 5
2.2 Origine du projet	page 6

3. L'analyse du projet

3.1 Etude de l'existant	page 7
3.2 Objectif	page 9
3.3 Contrainte	page 10
3.4 Etude des solutions possibles	page 10

4. La mise en place du projet

4.1 Apprentissage des langages concernés	page 11
4.2 Développement du projet	page 12
4.3 Tests	page 19
4.4 Mise en ligne	page 19

Conclusion	page 20
-------------------------	----------------

REMERCIEMENT

Je remercie tous ceux qui ont contribué à la réalisation de ce projet : le personnel de la société Egide Développement et les enseignants du CEFP pour leur encadrement et leurs encouragements.

Un grand merci à Jean-Marcel Dalbarade, associé de Egide Développement de sa disponibilité et de la confiance qu'il m'a accordée en me confiant le développement du logiciel Pricer.

J'associe aussi naturellement mon tuteur Philippe Mauriras à la réussite de ce projet, qui malgré sa surcharge de travail a pris le temps de m'épauler quand j'ai rencontré des problèmes.

J'aimerais également remercier mes collègues Fabrice Teixeira, Christophe Chabanois et Emmanuel Guinard pour leurs bons conseils, leur bonne humeur et leur aide pendant ces deux années.

Enfin j'adresse mes remerciements aux administrateurs et membres des sites traitant des langages ASP.NET et C# tels que www.asp-magazine.com, www.dotnet-fr.org, www.charpfr.com, www.asp.net pour n'en citer que quelques uns.

1. L'entreprise



Créée en 1985, **Egide Développement** est une Société de Service et d'Ingénierie (SSII). Elle accompagne les entreprises et les organisations dans le développement de leurs systèmes d'information et l'intégration des technologies, de la conception à la mise en service des applications.

1.1. Statut juridique

<i>Dénomination :</i>	Egide Développement
<i>Siège social :</i>	23, rue des Mathurins
<i>Forme juridique :</i>	Société Anonyme
<i>Capital social :</i>	38 200 €
<i>Activité principale :</i>	Prestation de services informatiques
<i>Chiffre d'affaires 2002 :</i>	258 000 €

1.2. Activité

➤ Développement de logiciels

Les informaticiens d'Egide sont spécialisés dans les langages de développement Visual Basic, C, C++ et Java. Ils utilisent les systèmes de gestion de base de données SQL Server et Oracle. Nous travaillons plus particulièrement dans les domaines de l'Education, de la Finance et de la gestion.

Au service de l'**Education**, les logiciels d'EAO¹ pour la formation en finance dans l'enseignement supérieur et Egimmo pour la comptabilité et la gestion des établissements de l'Education Nationale (3000 utilisateurs) ont été développés.

Pour la **finance**, Egide a créé des logiciels standards d'aide à la décision comme le progiciel Matsevery pour le Crédit Agricole.

Egide facilite également la **gestion** commerciale de l'entreprise. Nous pouvons assurer la gestion des prospects, des devis et des commandes... Nous nous chargeons par exemple de la gestion des films des studios de cinéma Phoenix Billancourt.

➤ **Intégration de systèmes**

En **réseau**, Egide conçoit et assure la maintenance des systèmes de micro-ordinateurs connectés sur un ou plusieurs sites. Il a notamment mis en place un réseau 100 postes réparti sur trois sites pour la Banque de Gestion Privée.

En matière de **technologies Web**, nous hébergeons et développons notamment les sites de Voyageurs Du Monde, élu meilleur site des Voyageurs français et de Bookstorming, librairie en ligne de publications d'art contemporain.

➤ **Autres**

Certains informaticiens sont envoyés en mission dans d'autres entreprises et y travaillent alors pour le compte d'Egide pour des périodes plus ou moins longues.

Dans le domaine du multimédia, Egide offre ses services pour tous les stades de réalisation d'un cd rom, de l'écriture de la charte graphique au pressage du cd.

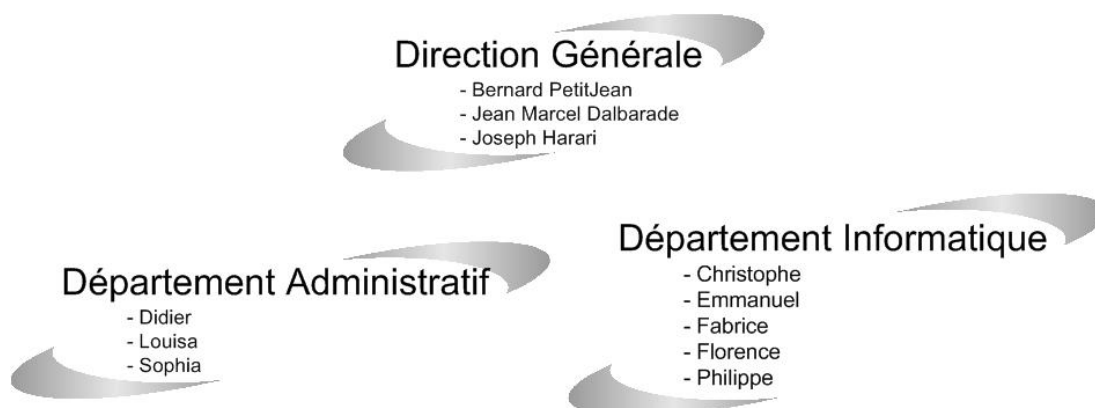
Enfin, des formations à Windows sont également proposés aux utilisateurs.

1.3. Personnel

Pour effectuer tout ce travail, la société est composée de 4 associés professeurs d'université, 4 développeurs, 2 secrétaires et une informaticienne en alternance. Nous avons tous de bonnes connaissances informatiques, chacun devant être à même d'assurer l'assistance téléphonique.

¹ EAO : Enseignement Assisté par Ordinateur

Composition de l'entreprise



1.4. Equipement informatique

Pendant la première année de ma formation j'ai travaillé sous le poste DB dans le développement Web. La mémoire vive était toujours saturée, le disque dur d'un giga octet également, je devais régulièrement redémarrer la machine, désinstaller les programmes que je n'utilisais pas pour libérer de l'espace et les réinstaller plus tard. A partir du moment où j'ai travaillé avec un serveur Apache sur la machine pour développer en PHP¹ : une nouvelle machine s'imposait, je passais en effet plus de temps à attendre qu'à travailler. Aussi on m'a attribué le poste XP de Christophe laissé vacant le temps de sa régie.

Liste des ordinateurs à Egide

Nom Ordinateur (*)	Système d'exploitation	Configuration
Arthur **	Windows 2000 Serveur	P3 800 MHz 320 MO de ram
Bajor **	Windows NT4	Pentium Pro 200 MHz 48 MO de ram
DB (Didier)	Windows 98 SE	Pentium Pro 48 MO de ram
Dell2000 (Emmanuel)	Windows 2000 Serveur	P3 800 MHz 256 MO de ram
Eagle (Fabrice)	Windows 98 SE	P3 800 MHz 256 MO de ram

¹ PHP : langage de programmation Internet qui s'exécute du côté du serveur et non du navigateur client, contrairement au HTML par exemple.

ED (Jean-Marcel)	Windows 98	Pentium Pro 96 MO de ram
JMD (Louisa)	Windows 98 SE	P2 266 MHz 64 MO de ram
P4**	Windows XP Pro	P4 2.4 GHz 512 MO de ram
Primo**	Windows NT4	P2 400 MHz 256 MO de ram
Sirius (Philippe)	Windows NT4	P2 400 MHz 256 MO de ram
Sophia (Sophia)	Windows 98 SE	P2 192 MO de ram
XP (Florence)	Windows XP Pro	P3 1GHz 128 MO de ram

* Le nom de l'utilisateur du poste au moment du projet est mentionné entre parenthèses

** Les postes non attribués peuvent faire office de serveur, servir pour les tests, être prêtés aux stagiaires, etc.

L'ensemble des postes est relié en réseau, ils ont tous accès à Internet par le biais d'un routeur.

2. Mon rôle dans l'entreprise

J'ai intégré Egide Développement en septembre 2001, en dépit du peu de connaissances que j'avais en informatique. En effet, l'entreprise avait à l'époque besoin de gens pour effectuer des tâches qui n'exigeaient pas de hautes qualifications. Par ailleurs, elle cherchait, à terme, à embaucher du personnel. Je pouvais ainsi m'intégrer dans la société au fur et à mesure, en assimilant sa culture et ses méthodes.

2.1. Tâches effectuées

Le niveau de difficulté du travail à faire augmentait progressivement.

Mes tâches du début consistaient principalement à **mettre à jour des sites Internet** de voyagistes tels que Voyageurs du Monde, Comptoirs des Déserts, du Maroc, etc. J'ai pu ainsi me familiariser avec le langage HTML et ses feuilles de style.

Plus tard, j'ai de nouveau été amenée à travailler avec ces clients pour effectuer des modifications diverses sur leur site Internet comme changer leur page d'accueil, mettre en cadre les pages, etc.

Je me suis ensuite occupée quelque temps des dossiers Gérance et Copropriété. Il s'agit d'un ensemble de programmes de gestion des propriétaires d'un immeuble. J'ai été chargée du **débogage** et de l'**ajout de quelques fonctionnalités** en Visual Basic. Pendant toute la durée du projet, j'étais la principale responsable, répondant aux appels du client et me déplaçant chez lui en cas de problèmes.

Par la suite, j'ai travaillé presque uniquement en collaboration avec Jean-Marcel Dalbarade. Associé de l'entreprise, il est aussi maître de conférence en finance à l'Université de Dauphine. La plupart de mes travaux ont alors toujours gardé un rapport avec la finance.

On m'a confié la **réalisation d'un tutorial** dans les langages HTML et Javascript du jeu Virtual Trader. Ce dernier simule un marché boursier où le joueur achète et vend des devises (dollar, euro, yen).

J'ai adapté les cours et les QCM tirés du livre¹ de Jean-Marcel à la plateforme Ganesha¹ qui permet aux élèves de s'auto évaluer en marquant les chapitres lus et les notes obtenues dans les QCM.

Enfin j'ai mis en place le **site Formation Finance** afin de réunir le tutorial de Virtual Trader, les cours de finance disponibles et les tarifs.

2.2. Origine du projet

Egide avait dans ses archives un ancien projet de "pricer" programmé en Visual Basic. La société qui l'a développé a fait faillite avant d'en terminer la réalisation. Elle en a alors fourni le code source au client Mondher Bellalah.

On m'a proposé de terminer le développement du logiciel pour l'intégrer dans le site de Formation Finance.

¹ Jean-Marcel Dalbarade, *Introduction au marché des changes*, 1995

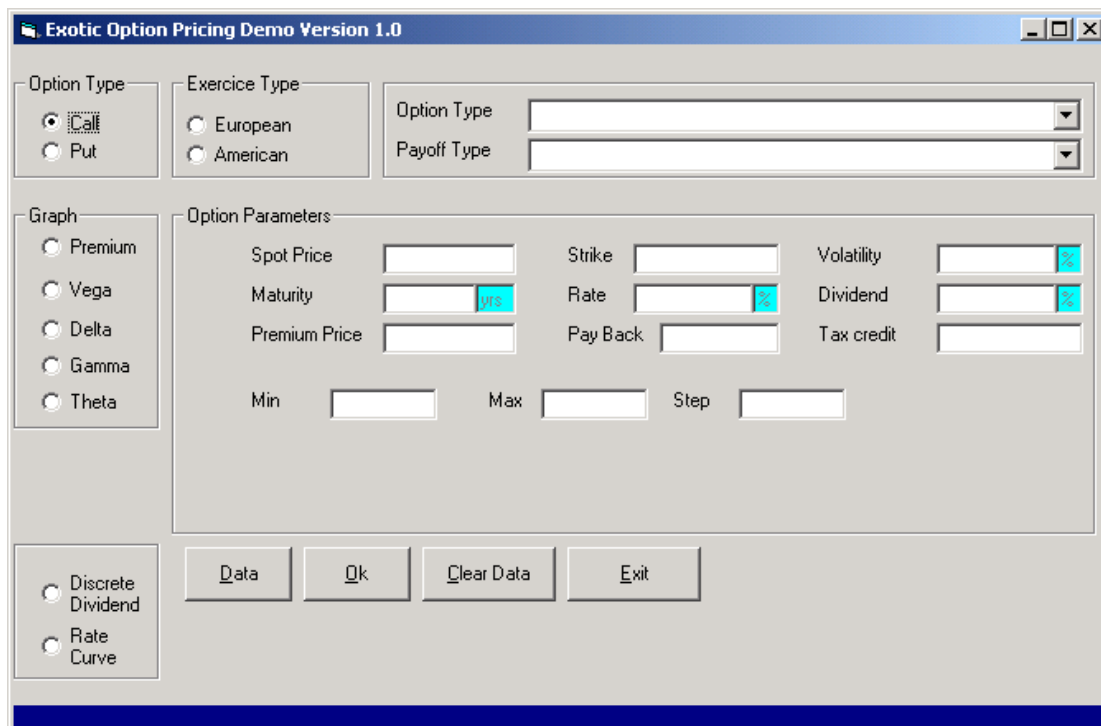
¹ Plate-forme gratuite de téléformation. Plus d'informations sur le site Internet : www.anemalab.org

3. L'analyse du projet

3.1. Etude de l'existant

❖ Un "pricer" en Visual Basic 6

Nous partons d'une ébauche de programme déjà existante en Visual Basic. Elle prévoit de nombreuses fonctionnalités, mais beaucoup ne sont pas opérationnelles.



❖ Un peu de finance

➤ Qu'est ce qu'un "pricer" ?

Le terme anglais de "pricer" est devenu courant dans le domaine financier : il permet de calculer le prix d'une option. Ce prix est aussi appelé "premium".

Une option est un *droit* d'achat ou de vente d'un actif à une date donnée.

L'actif en question, appelé sous-jacent, peut être un bien immobilier, une action, de l'or, des devises...

➤ Types d'option

Avant d'en calculer le prix, il faut apporter quelques précisions sur le type d'option qu'on souhaite acheter ou vendre.

Le call est un droit d'*achat* de l'actif, le put un droit de *vente*.

Une option est américaine lorsque la transaction s'effectuera à une date précise tandis que l'option est dite européenne lorsque l'achat ou la vente s'effectuera *jusqu'à* une date donnée.

Ces généralités s'appliquent à l'option Vanilla, qui est la plus classique et la plus basique. Il existe cependant d'autres types d'options, avec des variantes. Pour une option Moyenne par exemple, le prix garanti à l'échéance est la moyenne des cours de la période plutôt qu'un montant fixé à l'achat de l'option.

Le prix d'une option donnée peut se calculer selon différents types de modèles.

➤ Calcul de l'option

Plusieurs facteurs entrent en ligne de compte dans le calcul d'option : le spot, le strike, la volatilité, la maturité, le taux d'intérêt et le dividende.

Selon le type d'option calculé, d'autres variables pourront être nécessaires au calcul.

Spot : prix actuel du sous-jacent.

Strike (ou prix d'exercice) : prix auquel l'acheteur de l'option achètera ou vendra l'actif à la date donnée. Son montant est supérieur ou égal au spot.

Volatilité : amplitude de variation du cours de l'actif. Elle mesure en quelque sorte l'incertitude sur l'avenir du cours de l'actif : plus la volatilité augmente et plus on a de chances de voir le prix subir une forte hausse ou une forte baisse. On obtient ce taux à partir d'un historique.

Maturité (ou date d'échéance) : date fixée au delà de laquelle l'option n'a plus de valeur. C'est aussi la date limite à laquelle l'acheteur de l'option peut exercer son droit d'achat ou de vente.

Taux : taux d'intérêt du moment.

Dividende : revenu potentiel du détenteur de l'actif.

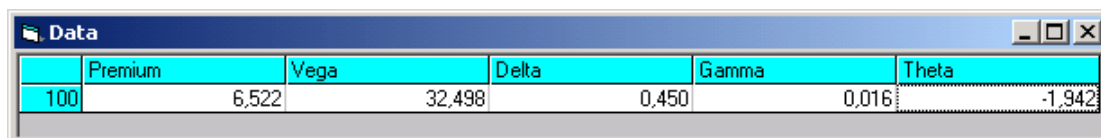
➤ Les grecques

Outre le prix de l'option, nous voulons aussi connaître l'importance de l'influence que chaque facteur peut avoir ce dernier. En effet, tout changement d'une des variables a une influence directe sur le prix. Nous mesurons ces risques avec des grecques comme Véga, Delta, Gamma et Thêta.

	mesure la sensibilité du prix de l'option par rapport à	
Véga	→	Volatilité
Delta	→	Spot
Thêta	→	Maturité

Le gamma correspond à la vitesse de réaction du risque Delta à une variation du sous-jacent.

Dans la pratique



	Premium	Vega	Delta	Gamma	Theta
100	6,522	32,498	0,450	0,016	-1,942

- Le prix théorique de l'option (ou *premium*) est ici de 6,522 €.
- Le Véga de 32.498 prévoit que le prix de l'option augmentera de $32.498 \times 1\%$ si la volatilité augmente de 1%.
- Le Delta de 0.450 signifie que le prix de l'option augmentera de 0.45 € si le prix de l'actif augmente de 1€.
- Le Gamma de 0.016 annonce une hausse de 0.016 de Delta si le premium change. Il constitue une sorte de valeur de secours par rapport au prix, delta étant très sensible.
- Le Thêta de -1.942 signale une hausse de 1.942 € sur le prix de l'option si une année s'écoule.

3.2. Objectif

Nous voulons créer une version de démonstration accessible en ligne sur Internet.

Elle permettra de calculer les options européennes Vanilla, Barrière et Moyenne.

Nous laisserons transparaître dans cette version les étendues de la version finale, qui proposera un affichage graphique et qui prendra en charge les options américaines Barone-Adesi & Whaley et Binomial Model .

Le code source utilisé pour le calcul des formules était déjà disponible dans l'ancienne version du programme. Cependant, beaucoup d'erreurs de programmation s'étaient glissées dans le code source et les résultats étaient parfois tout à fait improbables.

Tout le reste est à refaire, dans la mesure où j'utiliserai un autre langage de développement du pricer où l'approche sera probablement différente de celle en Visual Basic.

3.3. Contrainte

J'avais environ trois mois pour apprendre le langage et développer le pricer.

3.4. Etude des solutions possibles

Pour qu'il puisse être utilisé sur Internet, nous pouvions soit utiliser le langage Java, soit un langage .NET.

Seulement, nous voulions tester le nouveau produit phare de Microsoft qu'était Visual Studio.NET. Il est vrai que le fait que cet environnement de développement puisse à terme intégrer une vingtaine de langages tous compatibles entre eux représente un intérêt non négligeable, entre autres avantages. Au moment, Visual Studio.NET 2002 ne supportait que C++.NET, VB.NET et C#.NET.

C# et VB.NET sont les deux langages les plus accessibles pour un débutant.

Au premier abord il peut paraître évident de choisir Visual Basic.NET pour un programme initialement écrit en Visual Basic.

Le fait est qu'il fallait de toute manière réécrire une très grande partie du code pour intégrer l'orienté objet. Les approches en VB et en VB.NET sont tout à fait différentes. Le développement en VB.NET n'est donc pas tellement plus avantageuse qu'en C# sur ce point.

Les langages .NET étant relativement récents, personne encore à Egide ne s'est familiarisé avec. Il est donc important de pouvoir trouver de l'aide sur Internet, or la documentation C# est beaucoup plus abondante que celle de VB.NET.

Cela s'explique par le fait rôle phare qu'a voulu donné Microsoft au C# dans la plate-forme .NET. En effet, il doit allier la puissance de C et C++, la portabilité de Java tout en rendant la programmation plus simple, plus souple. Il est même gratuit.... Bref, C# c'est une nouvelle ère qui commence ?

Sur un plan plus "personnel", je me suis auto formée en Visual Basic au tout début du BTS, c'est à dire à un moment où je connaissais à peine le HTML, et où en cours, il n'avait encore abordé ni algorithmes, ni aucune notion de programmation. Mon apprentissage de ce langage a donc été assez abstrait, contrairement au C abordé de manière structurée par la suite à l'école.

De plus le langage principal de développement à Egide est le C++, duquel C# s'inspire. Les informaticiens de la société pouvaient donc m'apporter éventuellement leur aide.

Enfin, l'écriture de Pricer en C# nous permettait à Egide de s'approprier plus complètement le projet, et de repartir sur un code propre.

➤ **Concertation avec les autres**

Jean-Marcel favorisait plutôt un développement en Visual Basic.NET, il pensait ainsi la tâche plus aisée.

Néanmoins, d'une part Philippe, Emmanuel et Christophe pensaient eux que le C# avait plus d'avenir et serait plus intéressant à connaître, d'autre part la société ne pouvait pas laisser la révolution C# se faire sans elle...

Il a ainsi été décidé que Pricer serait développé en C#.

4. La mise en place du projet

4.1. Apprentissage des langages concernés

Avant de se lancer dans le développement à proprement dit du projet, il fallait commencer par une phase d'apprentissage du langage C# qui a pris environ un mois.

Toutefois, les informations que contiennent les livres sont souvent assez abstraites : mon support d'apprentissage¹ ne comprenait pas d'exercices, je comprenais le cours mais je n'étais pas dans une logique de programmation. Je ne savais pas si en développant j'aurais le réflexe de me dire que telle et telle possibilité existait, que telle méthode était plus appropriée. L'absence de pratique ralentissait mon apprentissage du langage. Pour cette raison j'ai bien souvent préféré directement développer en C#, quitte à rechercher les informations nécessaires au fur et à mesure.

¹ Bradley L. Jones, *C# En 21 jours*, 2001

Je pensais qu'il fallait d'abord coder le programme en C# pour Windows et qu'après quelques modifications mineures, ce même programme serait directement utilisable sur Internet. L'idée qu'un navigateur puisse supporter un langage non consacré à Internet était assez intrigante. Pour en avoir le cœur net, et mesurer la faisabilité du projet, Jean-Marcel m'a demandé de faire une première ébauche de programme en C# qui fonctionnerait sur Internet.

En réalité il fallait dans un premier temps d'abord définir l'interface Web, en HTML et ASP.NET pour supporter le C#, et après seulement le lier à un fichier tiers qui ne contiendrait que la partie fonctionnelle.

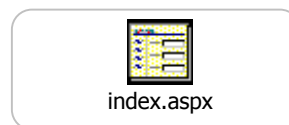
J'ai écrit une fonction d'addition pour le test. Elle nous a permis de voir le fonctionnement de .NET et de rassurer Jean-Marcel.

Dans les faits j'ai parcouru rapidement quelques pages de l'ouvrage traitant d'ASP.NET sans vraiment m'y attarder. Mon apprentissage a été plus orienté sur le C#.

4.2. Développement du projet

❖ Vue d'ensemble

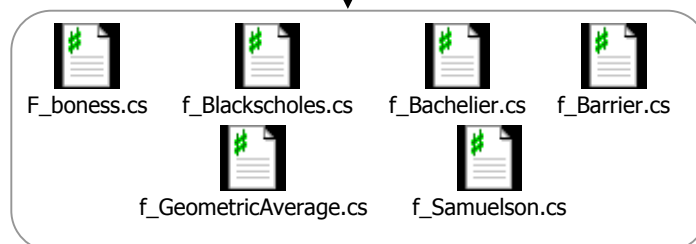
Interface



Point d'entrée



Formules de pricing

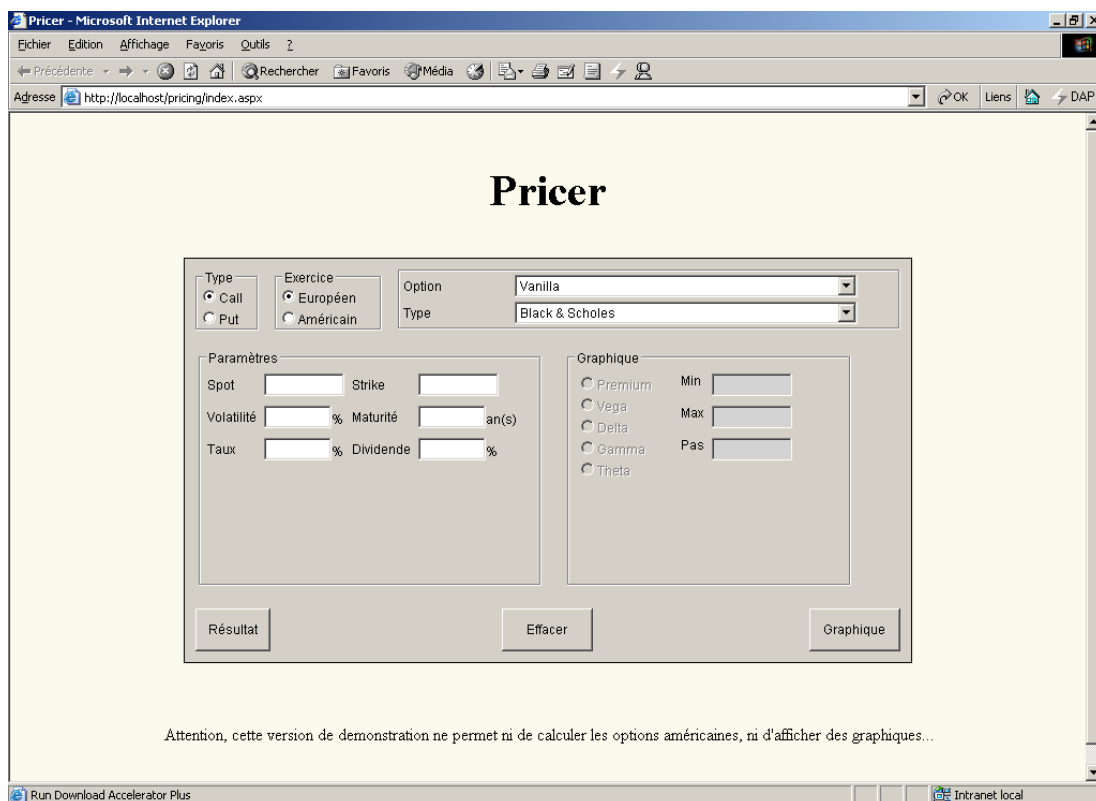


L'ensemble des éléments graphiques de base sont contenus dans le fichier *index.aspx*.

Le point d'entrée est dans le fichier *index.aspx.cs*. Ce dernier contiendra l'ensemble des événements associés aux contrôles de la page Internet, l'initialisation des variables, et d'autres fonctions. Les formules seront appelées à partir de ce fichier.

❖ Interface du Pricer

Nous avons choisi d'utiliser une interface type Windows. Elle permet à l'utilisateur de spécifier le type d'option dont il souhaite calculer le prix et de renseigner les champs nécessaires au calcul.



➤ Interdépendance des contrôles

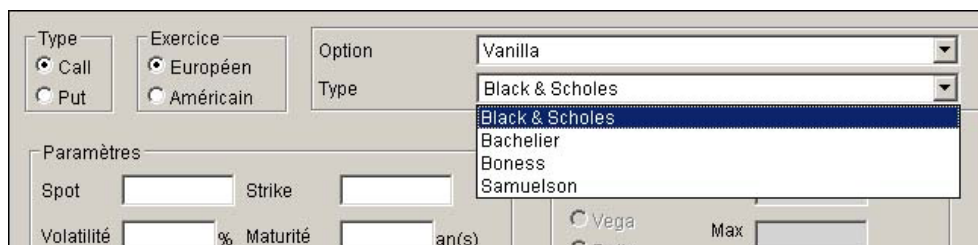
Une fois les contrôles statiques posés, il faut tenir compte de l'interdépendance des contrôles dynamiques pour pouvoir les ajouter en fonction des besoins. Nous n'avons par exemple pas besoin de renseigner une barrière de prix pour calculer une option Vanilla.

Etat des dépendances entre les contrôles

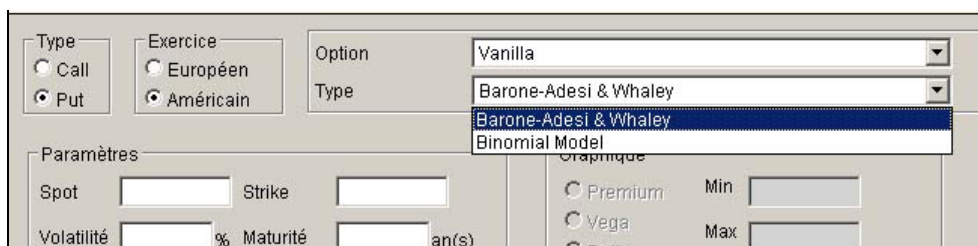
Exercice	Option correspondante	Type correspondant	Ajouter le contrôle
Européen	Vanilla	Black & Scholes	Taux de retour Rhô Alpha Barrière
		Bachelier	
		Boness	
		Samuelson	
	Barrière simple	Up & Out	
		Up & In	
		Down & Out	
		Down & In	
	Moyenne Prix	Moyenne géométrique	Date Montant
Américaine	Vanilla	Barone-Adesi & Whaley	
		Binomial Model	

Les modèles disponibles pour l'option Vanilla selon que l'exercice soit européen ou américain ne sont donc pas les mêmes :

Option Vanilla dans un exercice européen



Option Vanilla dans un exercice américain



➤ Affichage des résultats

Les résultats s'afficheront en dessous des paramètres à renseigner, une fois que l'internaute aura cliqué sur le bouton résultat. Le reste du temps, le tableau est invisible.

	Premium	Vega	Delta	Gamma	Theta
100	3,6083	36,0523	0,4826	0,0245	-1,4401

❖ Formules du pricer

➤ Implémentation des formules par classe

Dans le programme originel en Visual Basic, les formules avaient été décomposées en plusieurs sous fonctions réutilisables. Aussi j'ai décidé de rassembler ces sous fonctions de base et de les introduire dans une nouvelle classe AllFormules dont dériveront l'ensemble des formules. Ainsi, ces fonctions existeront par défaut pour tous les modèles de calcul.

Au lieu d'avoir l'une des deux structures suivantes :

```
//En précisant l'emplacement de la
fonction à chaque fois qu'elle sera
appelée

public class Blackscholes {
    protected static double act(...) {
        return Math.Exp(-R * T);
    }
    (...)

    public double pricing(...) {
        (...)
        A = act(T, R);
        (...)
    }
}

public class Boness {
    public double pricing(...) {
        (...)
        A = Blackscholes.act (...);
        (...)
    }
    (...)
}

public class Barrier {
    public double pricing(...) {
        (...)
        A = Blackscholes.act (...);
        (...)
    }
}
```

```
//En recréant la fonction redondante
dans chaque classe qui l'utilise

public class Blackscholes{
    protected static double act(...) {
        return Math.Exp(-R * T);
    }
    (...)
}

public class Boness {
    protected static double act(...) {
        return Math.Exp(-R * T);
    }
    (...)
}

public class Barrier {
    protected static double act(...) {
        return Math.Exp(-R * T);
    }
    (...)
}
```

Nous aurons :

```
//Classe contenant les fonctions basiques
public class AllFormules
{
    protected static double act(...) {
        return Math.Exp(-R * T);
    }
    (...)
}

//Les classes Boness et Barrier hérite des fonctions de la classe AllFormules
public class Boness : AllFormules{
    public double pricing(...){
        (...)
        A = act(T, alpha);
        (...)
    }
}

public class Barrier : AllFormules {
    public double pricing(...){
        (...)
        A = act(T, alpha);
        (...)
    }
}
}
```

➤ Première formule

La programmation d'ensemble terminée, il me restait à coder individuellement chaque formule. C'est ici que je me suis servie du code existant. J'ai commencé par réécrire la formule Black & Scholes de l'option Vanilla en exercice européen.

➤ Gestion des erreurs

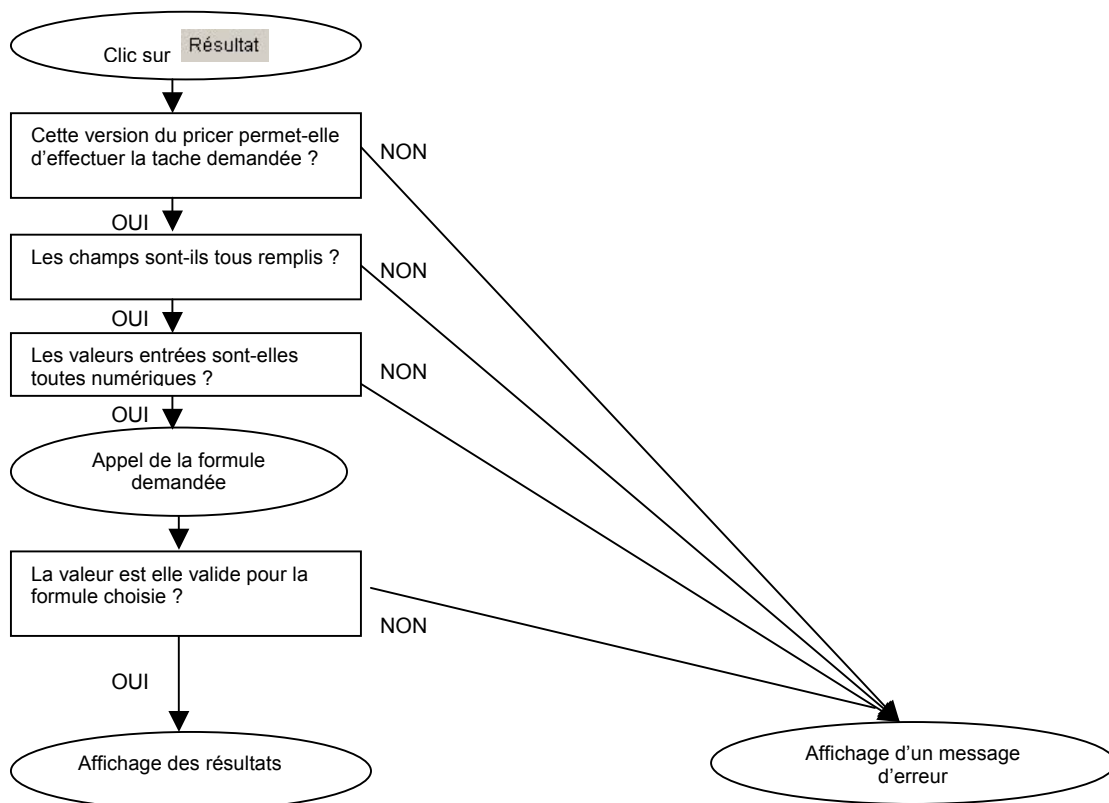
Quand l'utilisateur veut utiliser une des fonctions inactives pour la version de démonstration, un message d'erreur s'affiche :



Avant d'envoyer les valeurs à la fonction de calcul choisie, le programme vérifie un à un que les champs remplis contiennent bien des valeurs numériques :

Si la valeur du champ n'est pas autorisée pour la formule choisie (une valeur nulle pour le spot par exemple), la fonction ShowError se lance.

▪ Contrôle des saisies de l'utilisateur



❖ Problèmes rencontrés

A chaque fois qu'un problème apparaissait il fallait le situer et déterminer si la cause était liée à ASP.NET ou C#.

Comme nous l'avons vu précédemment les contrôles sont dynamiques et les formules utilisées varient. Or à la validation de la page, c'était toujours la même formule Black & Scholes qui était employée pour le calcul. En effet, chaque validation de la page réinitialisait toutes les variables, y compris celle du choix de la formule à leur valeur par défaut.

J'ai aussi l'habitude d'utiliser assez abondamment les boîtes de dialogues pour contrôler la saisie de l'utilisateur et gérer les erreurs. J'ai eu beau chercher dans l'aide, impossible de trouver un équivalent en ASP.NET de cet objet de base. J'ai finalement trouvé une alternative en passant par le langage Javascript.

La documentation officielle m'a été utile pour les problèmes de syntaxe dans le code, mais beaucoup moins pour les besoins de réalisation concrète. Je me suis alors tournée sur les forums de programmation en ligne, tels que www.asp.net ou www.csharp-corner.com qui m'ont apporté beaucoup d'éléments de réponses.

Pour le problème de rafraîchissement par exemple, c'est dans un forum que j'ai appris qu'il fallait utiliser l'instruction ViewState pour stocker les variables temporaires nécessaires entre le déchargement de la page et son rechargement.

Les problèmes étaient parfois aussi dues à des erreurs de frappe ou de syntaxe. Le recours aux points d'arrêts, qui stoppent l'exécution d'un programme, facilitait l'isolement des erreurs et leur corrections. Nous pouvons ainsi vérifier une par une les valeurs des variables et situer le problème. J'ai résolu tous les bogues rencontrés par ce moyen.

❖ Devdays

Chaque année, Microsoft fait dans les Devdays une présentation de ses dernières technologies. Dans la conférence de 2003, les possibilités offertes par .NET ont semblé ne pas avoir de limites, avec les webservices à l'honneur. Ceux-ci permettent de réutiliser des services Internet tels quels. On peut ainsi les revendre, se les échanger, sans que chacun ait à réinventer le code à chaque fois.

J'ai mis les formules de Vanilla sous forme de webservices, même si le nombre de personnes potentiellement intéressées est très faible. Les formules de calcul d'option étaient tout à fait appropriées pour l'utilisation de cette technologie.

Malheureusement j'avais déjà assez avancé dans mon programme et, le temps pressant je ne pouvais pas me permettre de ralentir l'avancement du développement pour expérimenter de nouveaux concepts...

4.3. Tests

Le programme compilait et affichait des valeurs. Mais ces résultats étaient-ils pertinents ? Il fallait le vérifier.

J'ai commencé à tester les formules une à une, en vérifiant la pertinence des Véga, Delta, Gamma et Thêta par rapport au premium obtenu. Les résultats affichés pour la formule Up and Out étaient visiblement erronés. Grâce aux points d'arrêt j'ai pu "pister" l'erreur et remonter jusqu'à sa source. Elle était en fait déjà présente dans la toute première formule, que la cinquième et les suivantes utilisaient. Les résultats n'avaient été correctes avec mes valeurs de tests que par hasard...

Je n'ai malheureusement pas eu l'occasion de terminer l'intégralité des tests minutieux, par manque de temps. Mi-février, Jean Marcel m'informait que j'avais à m'occuper en priorité d'un nouveau projet pour le voyageur Voyageurs du Monde, du fait de l'indisponibilité du spécialiste Internet de la société.

4.4. Mise en ligne

Le système d'exploitation du serveur devait être au moins supérieur à Windows 2000 pour supporter l'environnement .NET.

Nous préférons que le site de formation finance et le pricer soient hébergés sur le même poste. Or, d'une part le poste Bajor hébergeant le site n'était pas assez puissant pour supporter ASP.NET en plus, d'autre part nous avions jusqu'ici utilisé un serveur Apache pour le PHP tandis que seul un serveur IIS était compatible à ASP.NET. Nous avons donc renoncé à l'idée.

Le poste Arthur hébergeait déjà les sites de Bookstorming et Voyageurs du monde, mais sous Windows NT. Aussi nous l'avons fait migrer vers Windows 2000 pour le rendre compatible à notre application.

Les abonnés au site Formation Finance peuvent désormais voir des exemples concrets de calcul d'option et même s'exercer seuls !

CONCLUSION

Avant tout, le développement du logiciel Pricer m'aura permis d'apprendre les langages C# et ASP.NET.

Dans la mesure où la logique de programmation était différente de ce que j'avais connu, il fallait garder un certain recul, conserver une vue d'ensemble, repérer les besoins : il était hors de question de commencer à programmer et de rajouter des éléments au fur et à mesure, sans une certaine logique. En effet le programme était amené à évoluer, aussi il fallait faciliter le travail de la personne qui s'en chargerait.

Je me suis rendue compte que sans Internet, le logiciel aurait pris beaucoup plus de temps à se mettre en place. A chaque problème, je pouvais être assurée qu'en cherchant bien, je trouverais une solution sur des sites spécialisés, dans des forums ou des news groups. Sans cela, je n'aurais pas pu être aussi autonome dans l'élaboration de toutes mes actions professionnelles.

Bien souvent aussi, le problème tenait à une lettre, à un chiffre. Les longs moments passés à déboguer m'ont poussée à être plus patiente et m'incitait à plus de rigueur pour la suite.

Mon principal regret finalement serait peut être de ne pas avoir pu exploiter pleinement certaines possibilités de .NET, tels que les webservices et autres fonctions C#, de ne pas pouvoir complètement optimiser le code. Ce manque de temps m'a également empêché de finir les tests. Mais j'espère bien avoir l'occasion d'y revenir à l'avenir...