

L'environnement de travail de Delphi 5

- **A la fin de ce chapitre, vous serez capable de :**
 - Connaître l'interface de travail de Delphi
 - Savoir comment travailler avec les éléments de l'interface
 - Savoir comment créer son premier projet

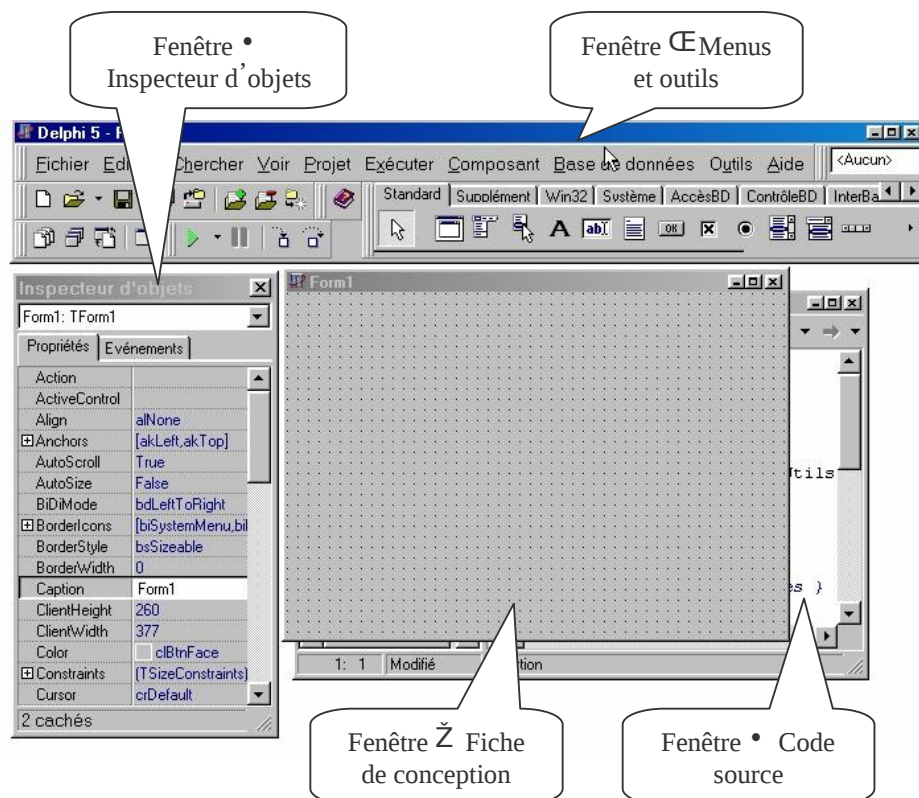
- **Travaux dirigés :**
 - Exercice n° 1** : Insérer un bouton dans une fiche
 - Exercice n° 2** : Créer un événement

Introduction :

Delphi diffère des autres langages classiques tel que Pascal ou Fortran, là où on doit directement écrire des instructions à la suite d'autre jusqu'à la fin; Delphi ressemble plus à un atelier où dispose d'une boîte à outils et d'un ensemble d'objets qui servent à fabriquer une application. On l'a bien dit; sous Delphi, on écrit pas une application mais on la fabrique.

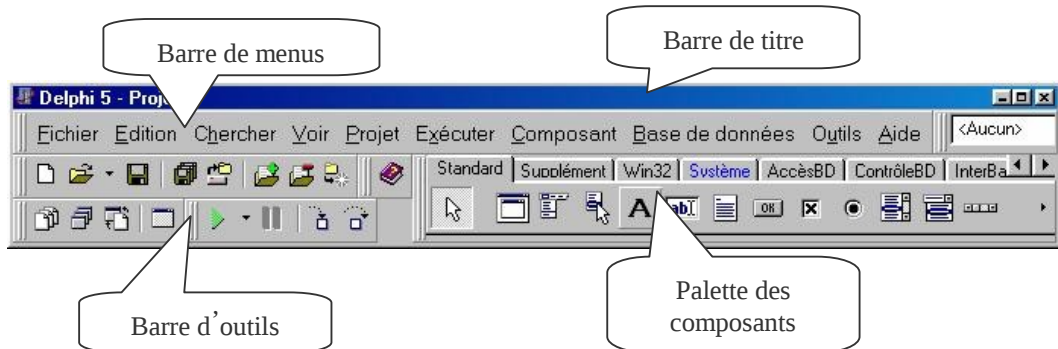
1- Interface de Delphi :

Après l'exécution de Delphi, vous pouvez constater que l'écran se compose de quatre (04) fenêtres :

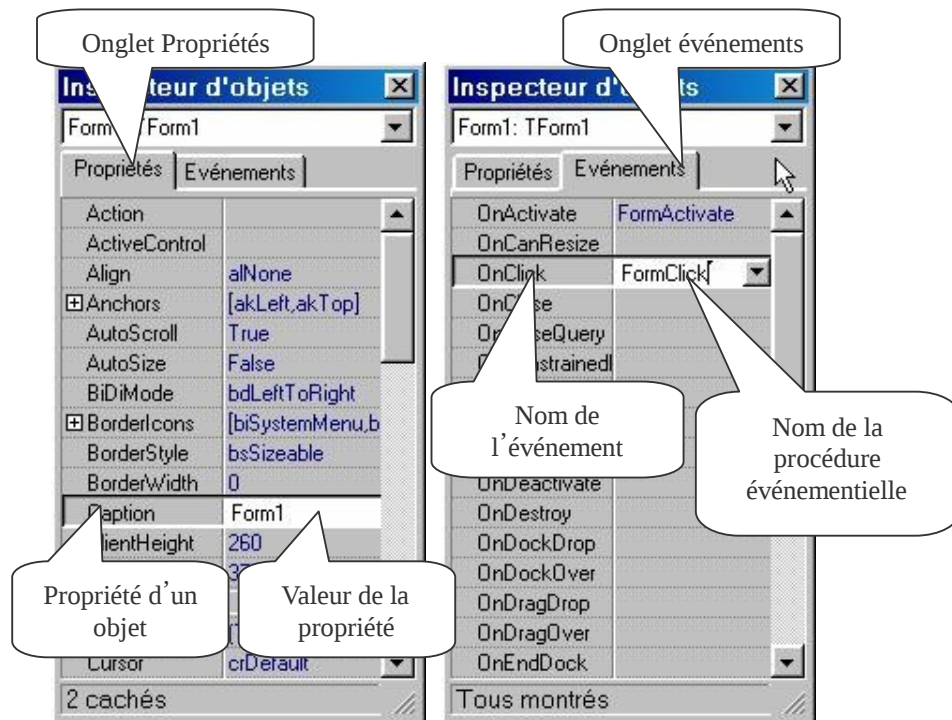


☒ La première fenêtre occupe la partie supérieure de l'écran ; elle contient :

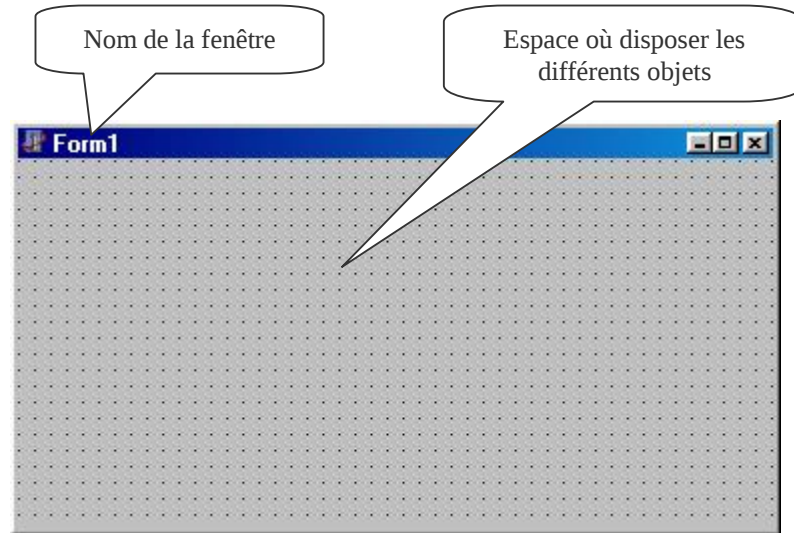
- La barre de titre
- La barre de menus
- La barre d'outils
- La palette des composants



La deuxième fenêtre se trouve par défaut à gauche de l'écran, elle correspond à l'Inspecteur d'objets : elle est composée de deux volets (onglets), l'un correspond aux propriétés d'un objet, l'autre aux événements auxquels l'objet peut répondre.

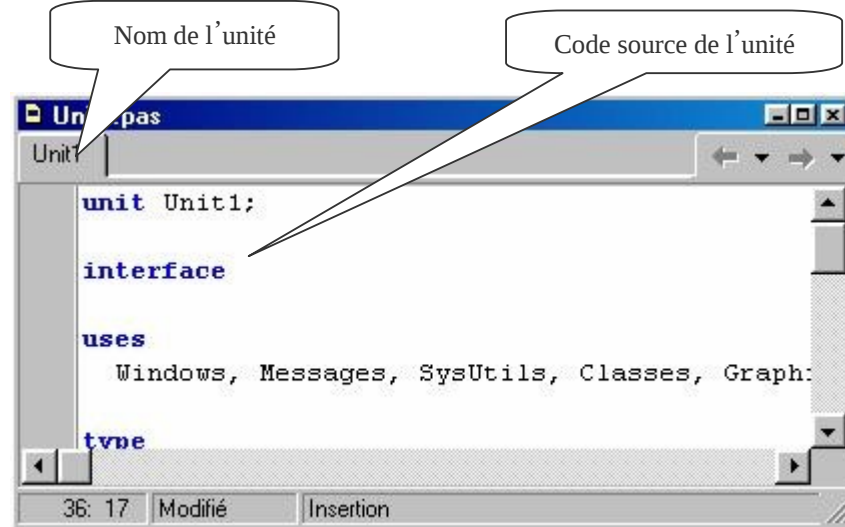


- Æ *La troisième fenêtr*e : elle constitue la fiche principale de la future application, il s'agit d'une fenêtr vide dans laquelle, on placera les divers objets (composants), on y trouve beaucoup d'appellations tel que Forme (Form), Feuille ou Fenêtr, nous utilisons l'appellation Fiche.



- Æ *La quatrième fenêtr*e est caché sous la fenêtr *Fiche*, on l'appelle « *L'Editeur de code* », cette fenêtr contient le code source relatif à la fiche, elle représente une unité de l'application parmi d'autres.

~ **Remarques :**



- Chaque fenêtre Fiche correspond à une fenêtre Unité.
- Une application Delphi peut regrouper plusieurs fiches donc plusieurs unités.

2- Travailler avec l'interface :

Dans ce paragraphe nous allons revenir sur les éléments importants de l'interface, c'est-à-dire ceux que nous utiliserons le plus souvent.

2-1- A quoi servent les menus et la barre d'outils :


Comme vous l'avez remarqué l'interface de Delphi comme toute interface sous Windows possède des menus déroulants. Presque toujours les mêmes (Fichier, Edition, ...).

Nous allons nous intéresser ici à quelques commandes plus urgentes :

Æ Fichier/Nouvelle application :

Lorsque vous exécutez Delphi, il vous crée automatiquement une nouvelle application avec une fiche « Form1 » et son unité « Unit1 » que vous pouvez utiliser pour construire votre application. La commande *Nouvelle application* a le même effet.

Æ Fichier/Nouvelle fiche :

Cette commande vous permet de rajouter d'autres fiches à votre application selon le besoin, la fiche insérée sera superposée à celles déjà existantes, dans la barre d'outils elle est représentée par l'icône .

Æ Fichier/Ouvrir :

Cette commande sert à ouvrir une unité déjà existante, l'ouverture de l'unité (.Pas) provoque l'ouverture automatique de la fiche qui lui correspond.

Æ Fichier/Ouvrir un projet (Ctrl+F11) :

Un *Projet* représente une application entière, l'ouverture d'un projet (.Dpr ou .Dpg) provoque la fermeture du projet en cours, car un seul projet peut être affiché à la fois. Le terme projet signifie « *Application* » ou « *Logiciel* ».

Æ Fichier/Enregistrer (Ctrl+S) :

Cette commande permet d'enregistrer les unités relatives au projet en cours, les noms de fichiers sont suivis de l'extension (.Pas).

Æ **Fichier/Tout enregistrer**  :

Cette commande permet d'enregistrer les unités ainsi que le projet lui-même avec tous les fichiers ressources.

Les unités sont sauvegardés avec l'extension (.Pas)


Le projet est sauvegardé avec l'extension (.Dpr) n'oubliez pas de choisir un répertoire qui doit contenir votre projet.

Æ **Projet/Compiler projet (Ctrl+F9)**  :

Cette commande permet de compiler votre projet et de déceler les erreurs s'il y a lieu.

Æ **Exécuter/Exécuter**  :

Cette commande permet de lancer l'exécution du projet en cours.

Æ **Exécuter pas à pas (F8)**  :

Cette commande permet d'exécuter votre projet pas à pas pour cerner le problème dans votre code source.

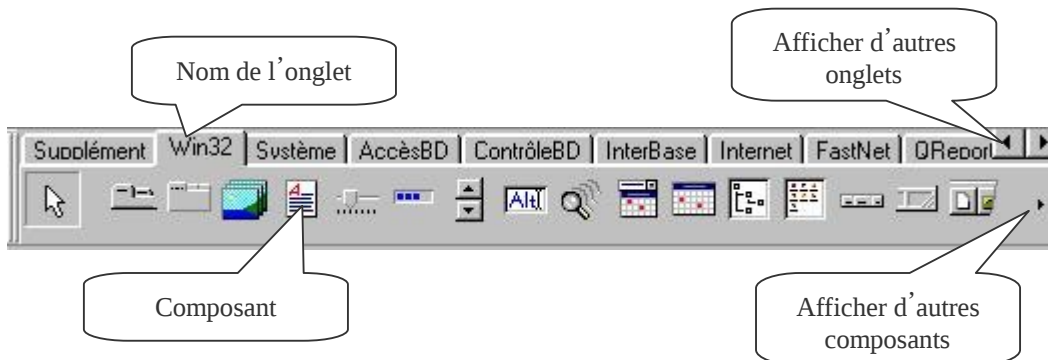
Æ **Exécuter/Réinitialiser le programme (Ctrl+F2)** :

Cette commande permet de rompre l'exécution du programme en cours.

D'autres commandes seront expliquées à fur et à mesure que l'on avance dans les notions.

2-2- A quoi sert la palette des composants :

La *Palette des composants* contient un ensemble de composants répartis par famille, chaque famille figure dans un onglet de la palette.



C'est quoi un composant ?

Pour voir mieux la signification d'un composant, prenons l'exemple d'un atelier de fabrication de meubles, on veut par exemple fabriquer une table, on sait à l'avance quels sont les éléments qui constituent une table :

- Un plateau en bois
- 4 pieds
- Pièces d'assemblage.

Le fabricant utilise directement les éléments ci-dessus pour monter une table. Et si l'on suppose que ces éléments ne sont pas disponibles, le fabriquant serait obligé de les fabriquer, chose qui est loin d'être pratique.

Alors, par simulation, on peut dire qu'un projet est constitué de plusieurs éléments qu'on appelle composants, ces composants peuvent être disponibles au préalable et peuvent servir à la fabrication de plusieurs types d'applications.

Le composant serait donc un programme (petit projet) qui exécute une tâche élémentaire ou une fonction déterminée, ce programme est créé puis inséré dans la palette pour une réutilisation probable. En effet tout le monde peut créer ses composants et les insérer dans la palette des composants.

Certains composants sont proposés par Delphi, ce sont ceux là qui vont servir à la création de nos projets.

2-3- A quoi sert l'inspecteur d'objets :

Comme on l'a déjà dit, l'inspecteur d'objets est composé de deux onglets : **Propriétés** et **Evénements**

a- Les propriétés :

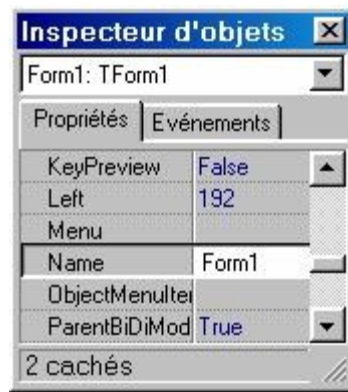
L'onglet **propriétés** affiche une liste de propriétés pour chaque composant inséré sur la fiche, si aucun composant est inséré, l'inspecteur d'objets affiche les propriétés relatives à la fiche.

Parmi les propriétés importantes et qui reviennent souvent :

• **Name**

Propriété qui a pour valeur le nom de l'objet (composant) ou de la fiche, l'inspecteur d'objets propose des noms par défaut qu'il numérote automatiquement, il est recommandé de laisser les valeurs de *Name* telles quelles sont.

Name est une propriété qui est commune à tous les objets



• **Caption**

C'est une propriété qui apparaît lorsque le composant inséré possède une description ;

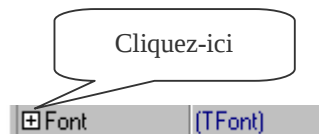
Par exemple :

La fiche possède une description qui s'affiche sur la barre de titre, pour la modifier, il suffit de changer la valeur de la propriété *Caption*.

Par contre le composant Edit ne possède pas de description donc la propriété *Caption* n'a pas lieu d'être.

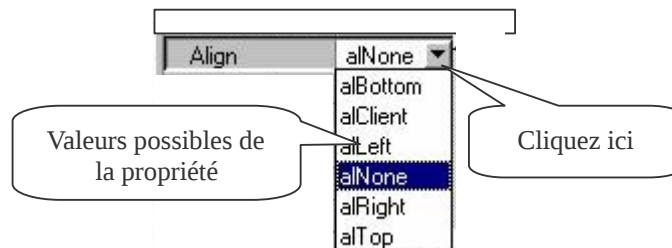
- Æ Certaines propriétés possèdent des sous propriétés, celles-ci sont précédées d'un signe « + », il suffit de cliquer dessus pour voir apparaître les sous propriétés.

Par exemple : La propriété *Font*



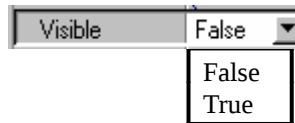
- Æ D'autres propriétés proposent un choix de valeurs parmi lesquelles il faut en sélectionner une.

Par exemple : La propriété *Align*



Æ D'autres propriétés prennent deux valeurs possibles « True » ou « False ».

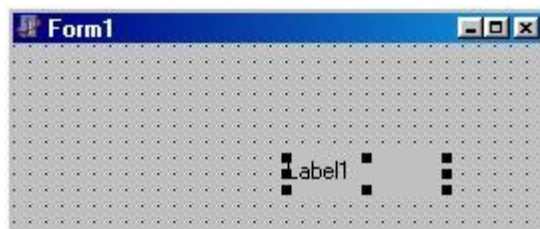
Par exemple : La propriété *Visible*



• **Exemple d'application :**

Pour illustrer l'utilisation des propriétés, essayons d'insérer un composant dans la fiche et attribuons-lui ses différentes propriétés :

F Insérer le composant « Label » de la palette des composants Standard symbolisé par **A**, pour cela cliquez sur le composant et déposez le sur la fiche en cliquant sur l'endroit voulu.



F Dans l'inspecteur d'objets :

- Visualisez la propriété *Name*, par défaut le nom *Label1* est attribué au composant *Label*, vous pouvez le changer ou le garder.
- Sélectionnez la propriété *Caption*, par défaut Delphi lui affecte la description « Label1 », changez cette valeur en mettant à la place la description suivante : 'Bonjour le monde' vous allez apercevoir que le texte change automatiquement sur le composant *Label1*.
- Cliquez sur le signe « + » qui précède la propriété « Font », une liste de propriétés apparaît :
 - i. Mettez la valeur « CLYellow » pour la propriété *Color* (couleur de caractère).
 - ii. Mettez la valeur *Arial* pour la propriété *Name* (nom de police).
 - iii. Mettez la valeur 16 pour la propriété *Size* (taille de caractère).

- Pour modifier la couleur de fond du Label :
Sélectionnez la propriété *Color* qui est placée juste après la propriété *Caption*, et affectez-lui la valeur `CLRed`.

~ **Remarque :**

*On voit bien que les propriétés agissent sur le composant dès la conception en revanche il existe d'autres propriétés qui agissent uniquement lors d'exécution d'application pour voir, essayez de changer la valeur de la propriété *Visible* et affectez-lui la valeur `False` vous remarquerez qu'il y a aucun changement. Cette valeur a pour effet de rendre invisible le composant à l'exécution.*

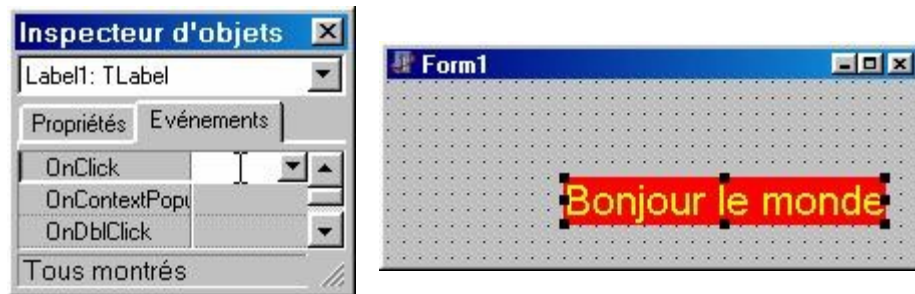
b- Les événements :

Chaque objet possède des événements qui peuvent le gérer, certains événements sont communs à tous les objets, d'autres sont spécifiques.

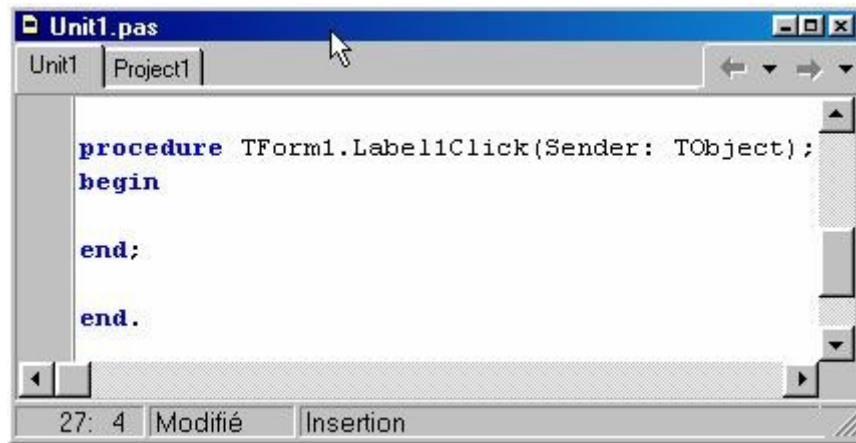
Un événement permet déclencher une procédure.

Pour voir le sens d'un événement, revenons à l'exemple d'application précédent :

- F Cliquez une fois sur le label « Bonjour le monde » pour le focaliser (le sélectionner).
- F Remettez la propriété *Visible* à `True`.
- F Cliquez sur l'onglet événement pour voir apparaître les événements relatifs au composant *Label*.



- F Sélectionnez l'événement *OnClick* et double cliquez sur la zone de saisie ou bien tout simplement double cliquez sur le composant lui-même, le résultat sera le même.
La fenêtre du code apparaît à l'écran et une procédure relative à l'événement est créée automatiquement.



Il ne reste qu'à écrire les instructions relatives à l'action qu'on veut avoir.

L'événement *OnClick* signifie qu'on exécute une action lorsqu'on clique sur le composant *Label1*, pour cela essayons de créer cet événement qui est le suivant :

On souhaite que, lorsqu'on clique sur « Bonjour le monde » on doit avoir le texte 'Bonsoir le monde' qui s'affiche à sa place avec un fond bleu, pour cela rajoutons l'instruction suivante :

```
Procedure TForm1.Label1Click(Sender: TObject);
Begin
Label1.Caption := 'Bonsoir le monde' ;
Label1.Color := Clblue ;
End;
```

Exécutez le programme, puis cliquez sur 'Bonjour le monde' et constatez le résultat.

Conclusion

Jusqu'ici on a pu avoir une idée sur la manière dont Delphi travaille, en utilisant son interface vous allez vous en apercevoir que la création d'applications devient un jeu d'enfants.

Travaux dirigés n°1

L' environnement de travail de Delphi 5

Exercice n° 1 :

Dans cet exercice :

- Utilisation des propriétés des objets.
- Enregistrement d'un projet.

Créer un programme qui affiche une fenêtre avec au milieu un bouton qui possède les caractéristiques suivantes

Description : Ok

Taille : 16

Police : Arial

Ø Mettez les caractéristiques suivantes pour les la fiche

Titre Fenêtre principale

Couleur de fond vert-citron

Hauteur 150

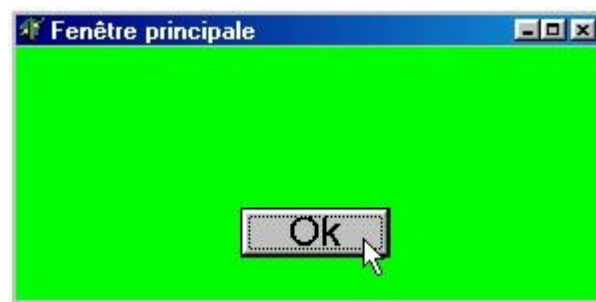
Largeur 300

Position centre de l'écran

Ø Enregistrer le projet sous le nom *premier.dpr*

Ce qu'on doit obtenir :

On doit avoir la fenêtre suivante :



Exercice n° 2 :

Dans cet exercice :

- Ouverture de projet.
- Utilisation des événements.

- Ø Ouvrir le projet *premier.dpr* précédemment créé.
- Ø Ajouter une zone d'affichage ayant comme description 'Bienvenu dans mon premier programme'
- Ø Créer l'événement suivant :
« Lorsque je clique sur le bouton *Ok* le contenu du label devient 'J'ai réussi mon premier programme' »
- Ø Insérer un 2^{ème} bouton qui va servir à quitter le programme.

Ce qu'on doit obtenir

On doit avoir la fenêtre suivante :



Lorsqu'on clique sur le bouton *Ok* la fenêtre devient comme suit :



Solutions TD 1

Exercice n° 1 :

Ce qui a été demandé :

On devait avoir la fenêtre suivante



Sur votre micro :

Suivre les étapes suivantes :

- F Ouvrir Delphi, s'il n'est pas encore ouvert ou choisir dans le menu **Fichier** le sous menu **Nouvelle Application**. Le projet est maintenant constitué d'une seule fenêtre appelée *Form1*.
- F Ouvrir Delphi, s'il n'est pas encore ouvert, ou choisir dans le menu fichier le sous menu **Nouvelle Application**. Le projet est maintenant constitué d'une seule fenêtre appelée *Form1*.
- F Dans la palette des composants choisir l'onglet **Standard** placer un bouton de commande *Button* sur la fenêtre *Form1*. Ce bouton a pour nom *Button1*
- F Dans l'inspecteur d'objet et dans l'onglet **Propriétés** définir la propriété *Caption* de cette fenêtre et lui attribuer la valeur 'Fenêtre principale'.
- F Cliquer sur la propriété *Color* de cette fenêtre et dans la liste déroulante choisir la valeur 'clLime'.
- F Définir la propriété *Height* de cette fenêtre et lui attribuer la valeur '150'.
- F Définir la propriété *Width* de cette fenêtre et lui attribuer la valeur '300'.

- F Dans la palette des composants choisir l'onglet **Standard** placer un bouton de commande *Button* sur la fenêtre *Form1*. Ce bouton a pour nom *Button1*
- F Dans l'inspecteur d'objet et dans l'onglet **Propriétés** définir la valeur *Caption* de ce bouton et lui attribuer la valeur 'Ok'.
- F Cliquez sur le signe + de la propriété *Font* les autres propriétés s'affichent.
- F Définir la propriété *Size* et lui attribuer la valeur '16'
- F Définir la propriété *Name* de la police en choisissant dans la liste la valeur 'Arial'.
- F Enregistrer le projet en sélectionnant **Tout Enregistrer** dans le menu **Fichier**.
- F Une boîte de dialogue s'affiche, Sélectionner, de préférence votre répertoire de travail, donner un nom à l'unité, par exemple *Uform1*, et appuyer sur **Enregistrer** l'unité sera enregistré sous l'extension '*.pas*'. Donner un nom au projet, par exemple *Premier* et appuyer sur **Enregistrer** le projet sera enregistré sous l'extension '*.dpr*'.
- F Enfin Compiler et Exécuter le projet en choisissant **Exécuter** dans le menu **Exécuter** ou appuyer sur **F9**.

Exercice n° 2 :

Ce qui a été demandé :

A la fin de cet exercice on devait obtenir :



Lorsqu'on clique sur le bouton *OK* la fenêtre devait être comme suit :



Sur votre micro :

Pour réaliser cet exercice on suit les étapes suivantes :

- F Ouvrir Delphi, s'il n'est pas encore ouvert, choisir dans le menu Fichier le sous menu Ouvrir un projet... Puis sélectionner le projet *premier.dpr*.
- F Dans la palette des composants choisir l'onglet Standard, placer un 2^{ème} bouton de commande *Button* sur la fenêtre *Form1*. Ce bouton à pour nom *Button2*.
- F Dans l'inspecteur d'objet et dans l'onglet Propriétés définir la valeur *Caption* de ce bouton et lui attribuer la valeur 'Quitter'.
- F Définir ces propriétés de police (*Size, Name*) comme celle du bouton *Ok*.
- F Dans la palette des composants choisir l'onglet Standard, placer une zone d'affichage *Label* sur la fenêtre *Form1*. Cette zone à pour nom *Label1*.

F Définir la propriété *Caption* de ce bouton en lui attribuant la valeur 'Bienvenu dans mon premier programme'.

Maintenant la maquette de l'application est prête. il est donc nécessaire de rajouter le code correspondant à chaque composant.

F Double-cliquer sur le bouton *Ok* pour écrire l'événement correspondant. L'événement consiste à remplacer le texte existant par le texte 'J'ai réussi mon premier programme' lorsqu'on clique sur le bouton *Ok*. Cette procédure événementielle doit être comme suit :

```
Procedure TForm1.Button1Click(Sender: TObject);  
Begin  
  label1.caption:='J'ai réussi mon premier  
  programme';  
End;
```

F revenir à la fiche *Form1* puis double-cliquer sur le bouton *Quitter* et rajouter l'instruction qui va mettre fin à l'application, cette procédure doit être comme suit :

```
Procedure TForm1.Button2Click(Sender: TObject);  
Begin  
  Close;  
End;
```