TD

Une cellule mutante dans une fenêtre graphique

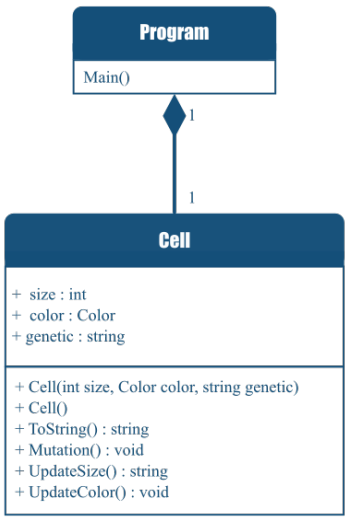
Octobre 2022

Présentation :

Cet exercice va nous amener à manipuler une nouvelle classe permettant de définir un objet Cellule. Cet objet va être amené à changer de taille et/ou encore de couleur. Nous allons donc dans un premier temps implémenter la classe attendue en nous appuyant une fois encore sur un diagramme de classes puis, dans un deuxième temps, nous allons implémenter une interface graphique pour observer les mutations dans une nouvelle fenêtre.

Implémentation de la classe Cellule :

Vous êtes libre de proposer l'implémentation de votre choix en accord avec le diagramme de classes suivant. Les détails de la méthode mutation sont données ci-dessous.



La méthode ***mutation*** correspond à la méthode qui va influencer, en fonction de différentes probabilité, l'information génétique de notre cellule. Cette nouvelle information génétique sera ensuite traitée par d'autres méthodes et pourra conduire à des modifications de l'aspect de notre cellule.
La méthode ***mutation*** parcourt un à un tous les caractères de l'attribut *genetic* de la cellule. En fonction des probabilités suivantes, les changements donnés sont susceptibles d'apparaître dans le code génétique de la cellule.

⋆ Il y a 15% de chance de voir apparaître une mutation du "A" en "T".
⋆ Il y a 7% de chance de voir apparaître une mutation du "T" en "AA".
⋆ Il y a 21% de chance de voir apparaître une mutation du "C" en "G".
⋆ Il y a 4% de chance de voir apparaître une mutation du "G" en "CG".

Entre chaque caractère lu dans le code génétique de la cellule, il y a de plus une probabilité de 5% de voir apparaître entre ces deux caractères un des quatre caractères "A", "C", "T", "G" choisit aléatoirement.
Une fois l'ensemble du code génétique parcourut, et l'ensemble des mutations appliquées, la cellule doit se mettre à jour et opérer des changements si besoin :

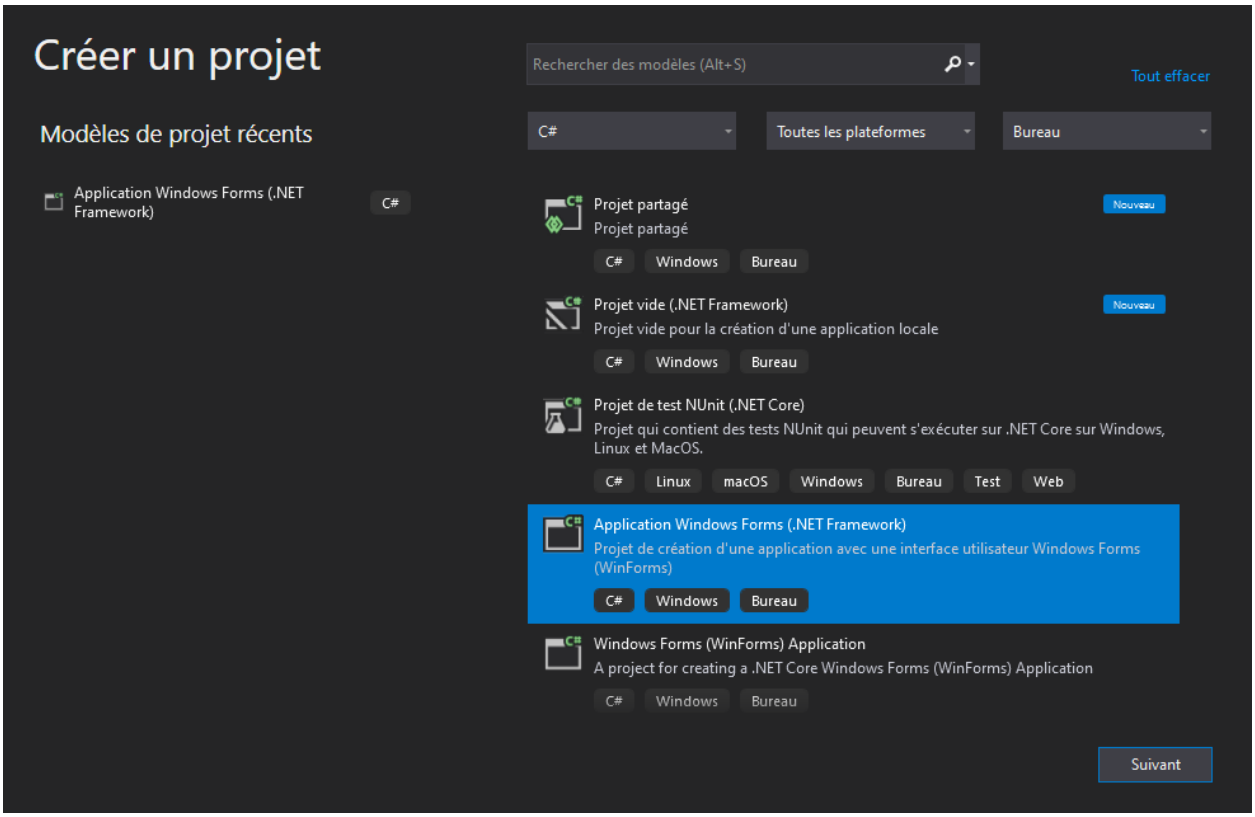
⋆ La **couleur** de la cellule est déterminée par le motif le plus fréquent parmi les motifs suivants :
- **TGT** - Noir (couleur par défaut de la cellule)
- **ATT** - Bleu
- **CTC** - Jaune
- **ACT** - Violet
- **GTC** - Orange
- **GAA** - Vert

Si aucun de ces patterns n'est détecté dans le code génétique de la cellule, alors sa couleur devient le noir.
Si plusieurs patterns sont détectés avec la même occurrence, c'est à vous de proposer la méthode de votre choix pour le choix de la couleur.
Ressource vers des couleurs prédéfinies en C#

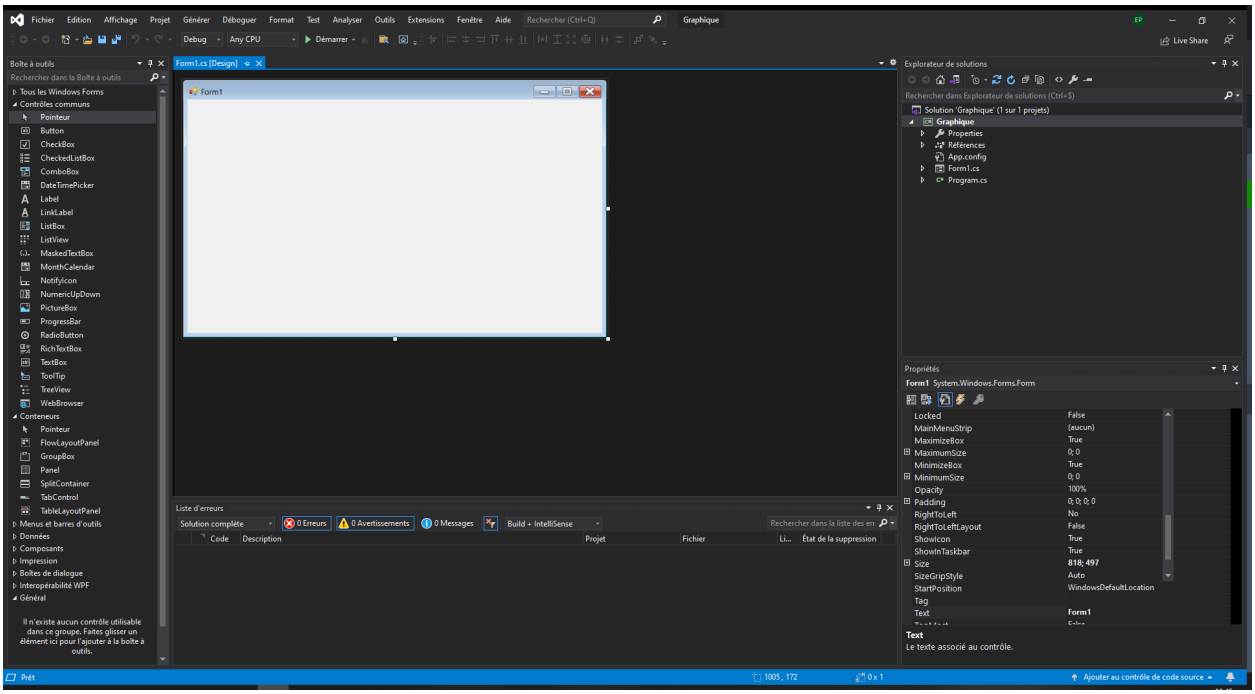
⋆ La cellule est représentée par un cercle donc la taille correspond au diamètre. Par défaut, au début de la simulation, la cellule a un diamètre de 10. La **taille** de la cellule est impactée par la longueur de son code génétique et par le nombre d'occurrence de la lettre "T" dans son code génétique. Elle se calcul comme suit :
- 10 +
- Taille du code génétique divisée par 5 +
- Le minimum entre le nombre d'occurrence de la lettre "T" et la taille précédente de la cellule

Implémentation graphique :

1. Ouvrir Visual Studio
2. Créer un nouveau projet en vous assurant de choisir comme modèle "Application Windows Forms (.NET Framework)

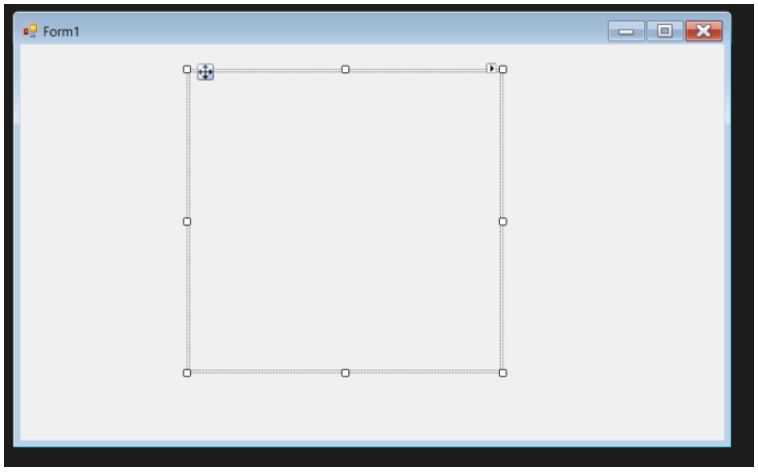


Vous arrivez sur un écran comme celui-ci. Si certain onglet ne s'affiche pas par défaut chez vous, vous pouvez aller les chercher dans le menu affichage (Boîte à outils, Explorateur de solutions).

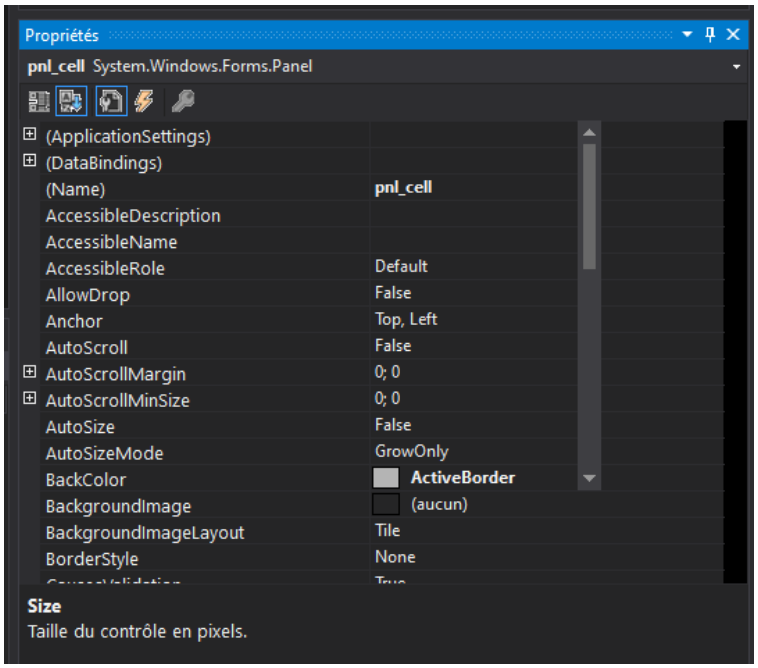


Nous allons travailler pour commencer dans la fenêtre du centre nommé Form1.cs [Design].

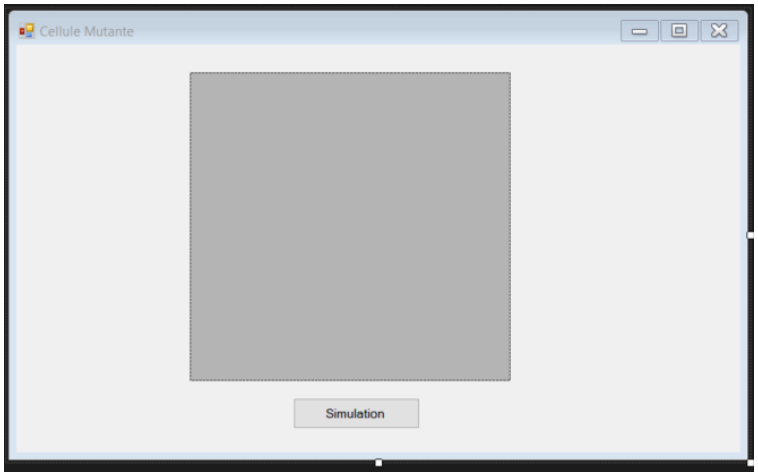
3. Cliquer dans la **boîte à outils**, dans la section **Conteneurs**, sur la ligne **Panel** dessiner un carré dans la fenêtre centrale.



4. Dans l'onglet Propriétés en bas à droite de l'écran, lorsque le Panel est sélectionné, vous accédez à un ensemble de propriétés caractérisant l'objet. Renommer le panel en **pnl_cell** puis attribuer-lui une couleur de fond.

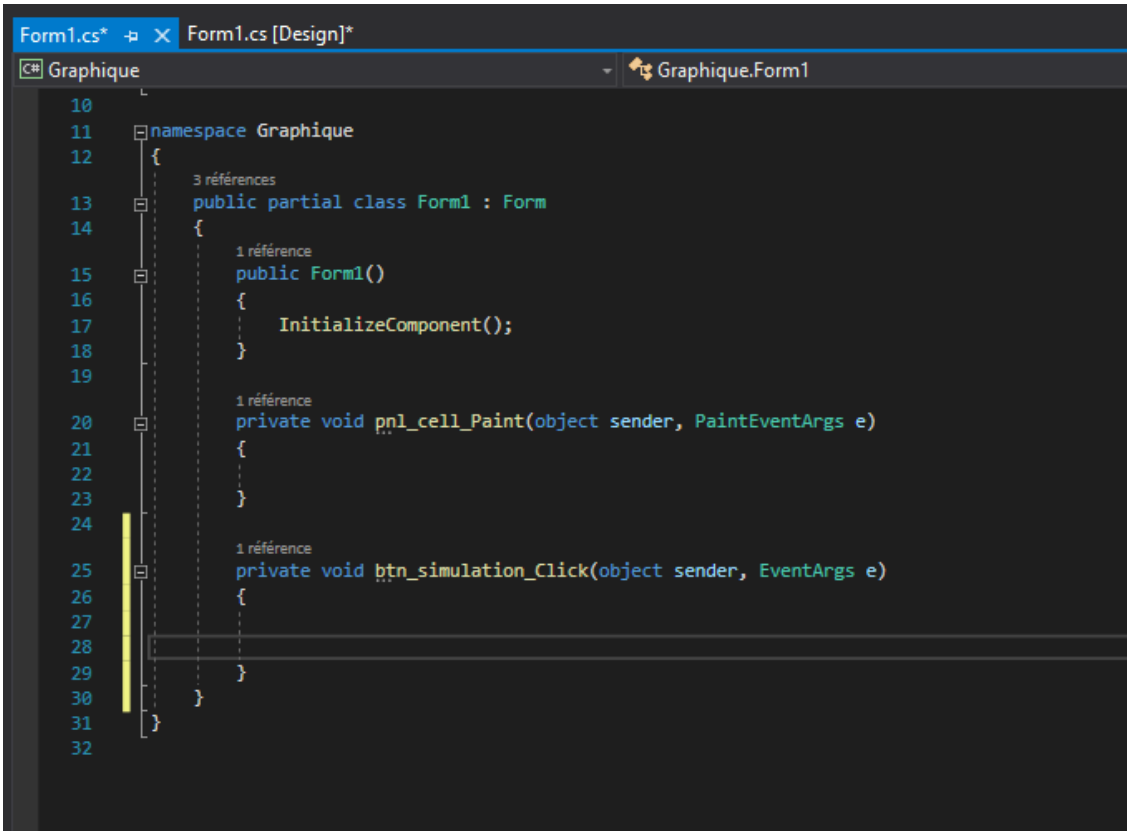


5. Ajouter de la même façon un bouton sur la fenêtre. Renommer-le **btn_simulation** et modifier le texte par défaut en **Simulation**. Modifier également le titre de la fenêtre en **Cellule Mutante**.



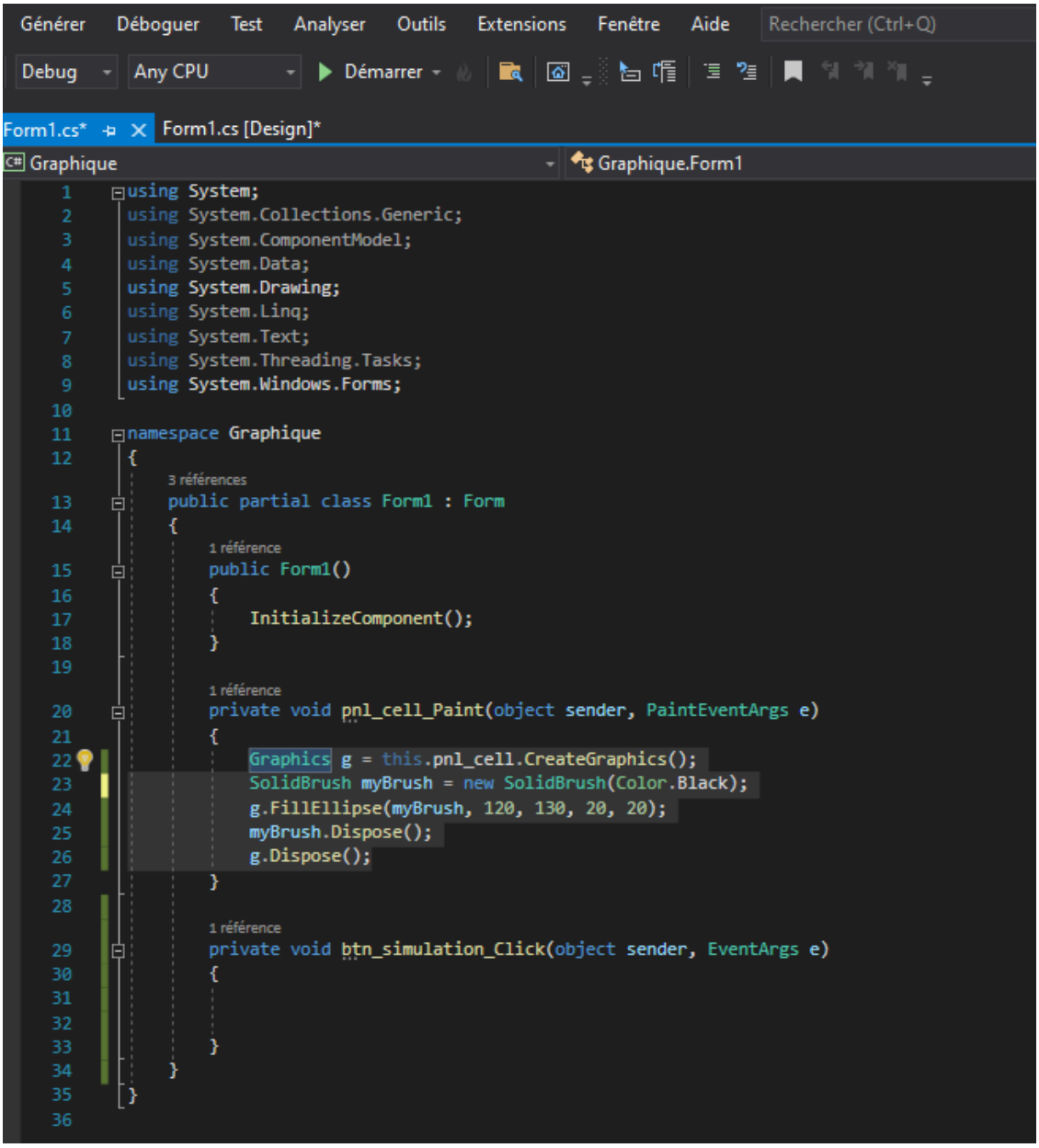
Nous avons maintenant mis en place dans les grandes lignes notre affichage graphique. Vous serez bien sûr vivement encouragés à l'améliorer et à le personnaliser. Pour le moment, concentrons-nous sur les bases et allons voir les lignes de codes qui se cachent derrière cet affichage.

6. Double-cliquer sur le panel dans la fenêtre graphique (ce qui vous conduit sur un nouveau document nommé Form1.cs).
7. Revernir sur Form1.cs [Design] et faire de même avec un double-clic sur le bouton.

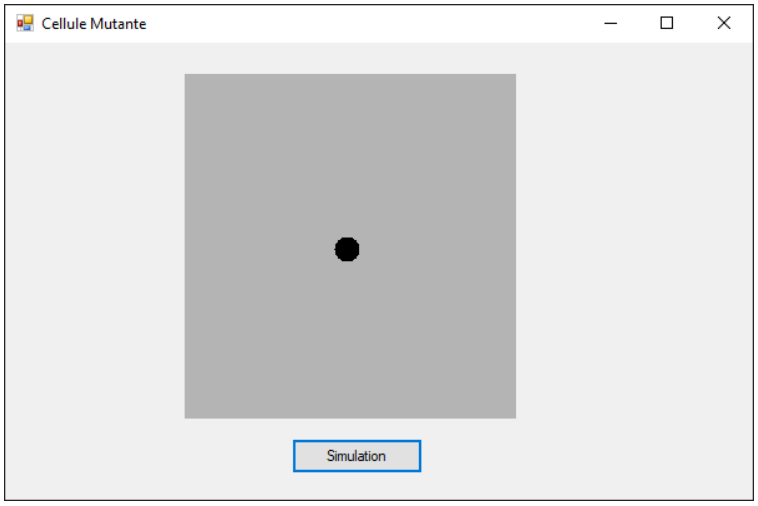


8. Saisir les instructions suivantes dans la méthode **pnl_cell_Paint()** :

```
Graphics g = this.pnl_cell.CreateGraphics();
SolidBrush myBrush = new SolidBrush(Color.Black);
g.FillEllipse(myBrush, 120, 130, 20, 20);
myBrush.Dispose();
g.Dispose();
```



9. Cliquer sur le **bouton Démarrer précédé de la flèche verte** dans le menu de haut de l'écran.



A suivre !