

PROGRAMMATION OBJET 1 - TRAVAIL 2

(15%)

En équipe de 2 ou 3.

SOMMAIRE

Vous êtes responsable de créer une application simple pour jeu de Battleship dans une version très simplifiée. Vous devez utiliser Visual Studio pour créer un projet Application Windows Forms. Le code doit être produit en français et respecter les normes de programmation C#. Durant la réalisation du projet, vous devez utiliser Github. **Prenez le temps de lire l'ensemble du travail avant de débiter.**

À la fin du travail, vous avez des exemples de code pouvant vous aider.

Un vidéo est fourni en annexe pour vous montrer le résultat final.

MISE EN SITUATION

Battleship est un vieux de bataille navale de bateau. L'objectif est de faire feu sur des cases dans l'espoir d'atteindre les navires adverses dont on ignore la position. Dans le vrai jeu, chaque joueur est responsable de placer ses navires de différentes tailles sur son plateau. Dans notre version, nous placerons des navires qui occuperont seulement 1 tuile de manière aléatoire. Ceci dans le but de simplifier votre projet.

[https://fr.wikipedia.org/wiki/Bataille_navale_\(jeu\)#/media/Fichier:KBattleship_Screenshot.png](https://fr.wikipedia.org/wiki/Bataille_navale_(jeu)#/media/Fichier:KBattleship_Screenshot.png)

TRAVAIL

UTILISATION DE GITHUB

La première étape est de créer le projet et de le mettre dans un dépôt sur Github. Ceci est obligatoire. La date limite pour la création du dépôt est le 22 mars.

À la fin de chaque cours où du temps vous est alloué pour travailler sur le travail vous devez faire minimalement 1 commit avec un message significatif. Il est toutefois recommandé d'en faire plus. **N'oubliez pas de pousser avant de fermer l'ordinateur.**

CRÉATION DES FENÊTRES

La fenêtre suggérée est relativement simple :

2 étiquettes « label »

2 « PictureBox »

AJOUT DES FONCTIONNALITÉS

INITIALISATION DU JEU

- Générer une grille de jeu pour le joueur.
 - Placer de manière aléatoire 3 bateaux
 - 2 Bateaux ne peuvent pas être sur la même case
- Générer une grille pour l'ordinateur
 - Placer de manière aléatoire 3 bateaux
 - 2 Bateaux ne peuvent pas être sur la même case
- L'image « Vos tirs » doit montrer une grille de case
 - Une case bleue représente de l'eau
 - Comme on doit ignorer où sont les bateaux adverses, même les endroits où sont les bateaux adverses sont bleus.
- L'image « Les tirs adverses » doit montrer une grille de case
 - Une case bleue représente de l'eau
 - Une case grise représente un bateau. Vos bateaux sont donc visibles.
- Les cases des images doivent être séparées par des lignes.

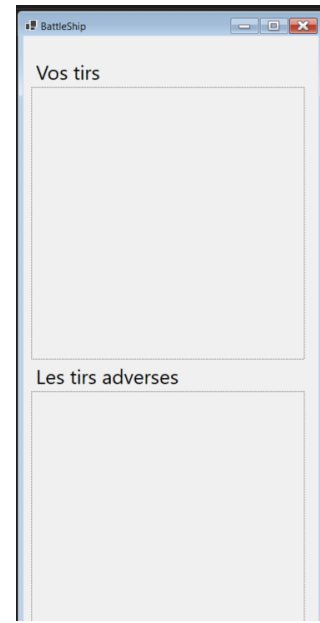


Figure 1 fenêtre du jeu

DÉROULEMENT DU JEU (DANS L'ORDRE)

TOUR DU JOUEUR

- Un clic sur l'image vos tirs permet de faire feu sur la case cliquée.
 - Bien qu'inutile, il est possible de faire feu plusieurs fois sur la même case.
- Un cercle rouge s'ajoute dans la case qui vient d'être tirée sur l'image « Vos tirs »
 - Le cercle persiste jusqu'à la fin de la partie
 - Le cercle doit être centré
- Si un bateau était présent, la case devient grise pour indiquer au joueur qu'un bateau était à cet endroit.
 - Le cercle rouge est tout de même présent.

TOUR DE L'ORDINATEUR

- L'ordinateur sélectionne une case au hasard et fait feu sur celle-ci.
 - Bien qu'inutile, l'ordinateur peut parfois faire feu plusieurs fois sur la même case.
- Un cercle rouge s'ajoute dans la case qui vient d'être tirée sur l'image « Les tirs adverses »
 - Le cercle persiste jusqu'à la fin de la partie
 - Le cercle doit être centré

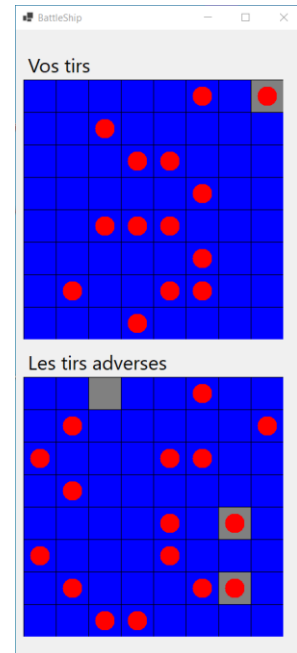


Figure 2 Partie en cours

À LA FIN DE CHAQUE TOUR

- Valider si le joueur ou l'ordinateur a gagné.
- Afficher un message approprié.
- Réinitialiser le jeu pour qu'une nouvelle partie puisse être jouée.
 - Les indicateurs de tirs sont retirés
 - Les bateaux sont replacés aléatoirement sur la grille du joueur et de l'ordinateur.

TESTS

- Que se passe-t-il si on clique plusieurs fois sur la même case?
- L'ordinateur peut-il faire feu sur l'ensemble des cases?
- Les bateaux sont-ils placés aléatoirement?
- Peut-il y avoir plusieurs bateaux sur la même case? (Ça ne devrait pas être le cas)
- Les bateaux placés aléatoirement peuvent-ils se retrouver partout sur la grille?
- Est-il possible de gagner?
- Est-il possible pour l'ordinateur de gagner?
- La seconde partie est-elle fonctionnelle jusqu'à la victoire?
- Est-il possible de faire un clic sur « Les tirs adverses »? (Cela ne devrait pas fonctionner)
- Que se passe-t-il si on clique sur les extrémités des « picturebox »?
- Lorsqu'un de vos tirs atteint un bateau, celui-ci est-il dévoilé (case en gris)?
- Voyez-vous les tirs de l'ordinateur?

CRITÈRES D'ÉVALUATION ET MODALITÉ DE REMISE

CRITÈRES D'ÉVALUATION 44 POINTS

- **Utilisation de Github**
 - Le projet est sur Github et il fut créé avant le ou avant le 22 mars (2 points)
 - Il y a minimalement 1 commit à chaque cours où du temps fut alloué pour avancer ce travail. (3 points)
 - Chaque commit possède un message significatif et informatif
- **Affichage et interface**
 - Les contrôles sont bien nommés (1 pt)
 - Les noms commencent par le type du contrôle suivi de sa fonction (1 pt)
- **Fonctionnalités réalisées en utilisant les notions de programmation orientée objet. Ces critères incluent la validation ainsi que le rafraîchissement des contrôles si nécessaire.**
 - Les grilles s'affichent correctement (10 points)
 - Les bateaux du joueur sont visibles.
 - Les tirs de l'ordinateur sont visibles.
 - Les bateaux de l'ordinateur ne sont pas visibles sauf si un tir a été fait sur celui-ci.
 - Les tirs du joueur sont visibles.
 - Des lignes sont présentes entre les cases.
 - Les bateaux sont placés aléatoirement (3 points)
 - Un clic permet de tirer sur une case (5 points)
 - L'ordinateur réplique avec son propre tir (4 points).
 - Un gagnant est annoncé par une boîte de message (3 points).
- **Qualité du code**
 - Les variables sont bien nommées. (1 point)
 - Il n'y a pas de répétition de code. (2 points)
 - Il y a le minimum possible de variable globale. (2 points)
 - L'indentation respecte les normes. (1 point)
 - Les attributs sont encapsulés. (2 points)
 - Les noms des méthodes contiennent un verbe et commencent par une majuscule. (1 point)
 - Les méthodes ont la bonne visibilité. (1 point)
 - Respect des normes C# (2 points)
 - Pas de caractère accentué sauf dans les commentaires ou les messages
 - Les fonctions commencent par une lettre majuscule à l'exception des événements créés automatiquement par Visual studio.
 - La variable commence par une lettre minuscule.

PÉNALITÉS

- Un projet qui ne compile pas aura une pénalité minimale de **20%** et pourrait aller jusqu'à **40%** selon la gravité des erreurs. Cette pénalité s'applique si l'erreur survient lors de l'initialisation de la forme principale.

- **Un projet qui n'utilise pas les notions de programmation objet n'obtiendra aucun point pour les fonctionnalités.**
- Le code n'est pas en français.
- Un projet qui contient du plagiat recevra la note 0. L'équipe plagiée aura également la note 0. N'envoyez pas votre projet à d'autres équipes. Ne montrez pas votre projet à d'autres équipes.
- Les pénalités pour le français s'appliquent comme prévu par les règles départementales.

MODALITÉ DE REMISE

Remise sur LÉA. Vous êtes responsable de télécharger votre remise et d'en vérifier le fonctionnement.

Création de Github avant le ou avant le 22 mars.

Remise avant le 27 mars 23h59

ANNEXE EXEMPLE CODE

Beaucoup de documentation est disponible en ligne pour vous aider.

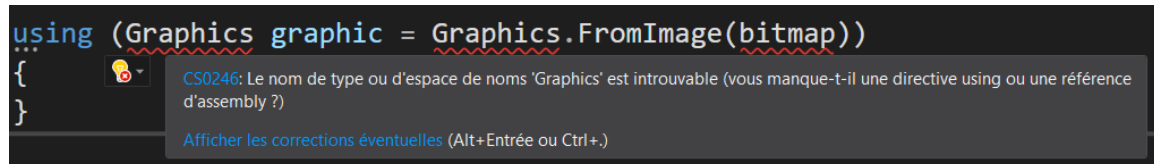
OBJET GRAPHICS

L'objet Graphics contient un grand nombre de méthodes qui vont permettre de dessiner des formes, lignes et images.

Pour l'utiliser, vous devrez l'importer.

```
using System.Drawing;
```

Si vous ne l'importez pas, vous aurez cette erreur :



<https://docs.microsoft.com/en-us/dotnet/api/system.drawing.graphics?view=dotnet-plat-ext-6.0#methods>

COMMENT INITIALISER UN BITMAP EN MÉMOIRE

```
int tailleX = 100;
int tailleY = 100;
Bitmap exempleBitmap = new Bitmap(tailleX, tailleY);
```

CRÉER UN RECTANGLE

Le rectangle est un objet pour déterminer où dessiner.

```
Rectangle positionEtTailleDessin = new Rectangle(positionDepartX,
                                                    positionDepartY,
                                                    largeurX,
                                                    hauteurY);
```

COMMENT DESSINER SUR UN BITMAP

L'objet graphics est très lourd. Using permet d'instancier l'objet et de le libérer lorsqu'il n'est plus requis. (À la fermeture de son accolade)

```
using (Graphics graphic = Graphics.FromImage(exempleBitmap))
{
    //Opération de dessin
}
```

```
//Dessine la bordure d'un cercle ou ellipse
graphic.DrawEllipse(pen, positionEtTailleDessin);

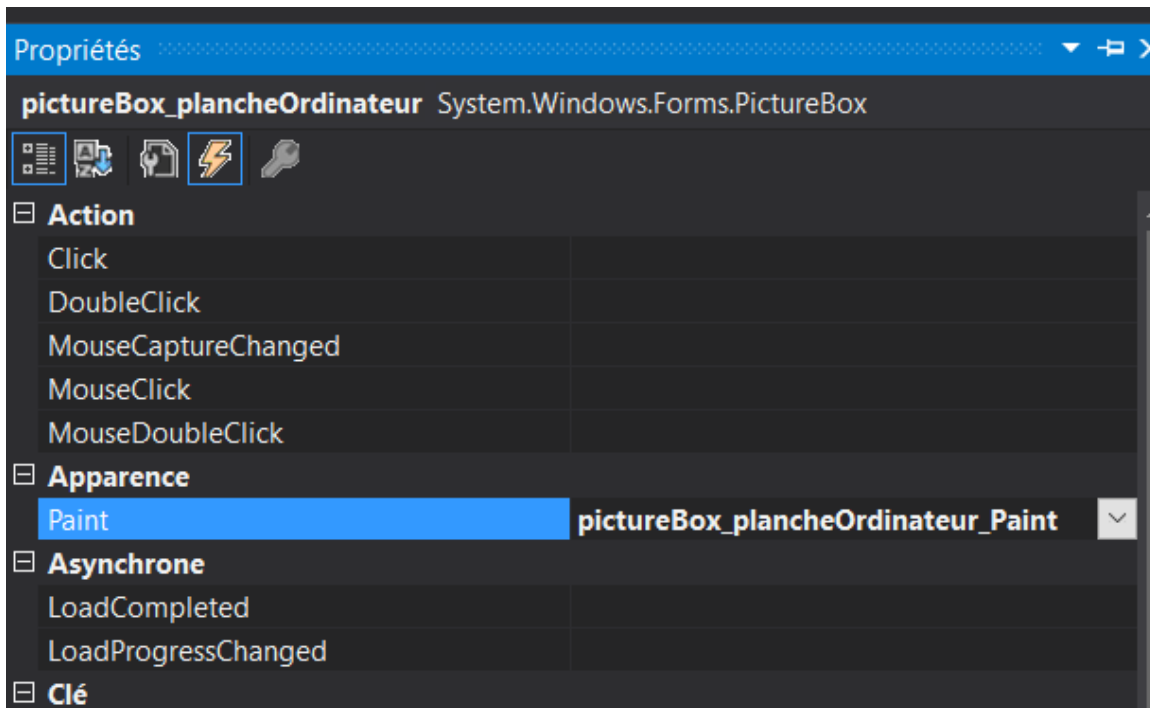
//Dessine un cercle plein ou une ellipse pleine en rouge
graphic.FillEllipse(Brushes.Red, positionEtTailleDessin);

//Trace une ligne du point 1 vers le point 2 de 0,0 à 100,100
graphic.DrawLine(pen, new Point(0,0), new Point(100,100));

//Dessine une image à une position
graphic.DrawImage(unAutreBitmap, new Point(0, 0));
```

PICTUREBOX

Un pictureBox possède un événement « Paint » qui se déclenche lorsque celui-ci se dessine.



DESSINER DANS UN PICTUREBOX

Dans l'événement « Paint » d'un PictureBox, un des paramètres est du type PaintEventArgs. Cet objet contient l'objet Graphics. Il est possible de l'utiliser pour dessiner dans le PictureBox.

```
private void pictureBox_exemple_Paint(object sender, PaintEventArgs e)
{
    e.Graphics.FillRectangle(Brushes.Blue, new Rectangle(0, 0, 100, 100));
}
```

FORCER UN PICTUREBOX À SE DESSINER

La méthode Invalidate() va forcer un PictureBox à se redessiner et donc l'événement Paint sera appelé si elle définit.

```
pictureBox_exemple.Invalidate();
```