Parachutes

Le Club de Parachutisme de Moléson va faire un saut dans le vide! Bob, Alice et Max vont embarquer dans un avion, sauter et aterrir en souplesse.

Les buts de cet exercice sont:

- De renforcer vos connaissances en POO: Classes, Instances, Attributs, Méthodes, Constructeur, ...
- De vous faire découvrir et implémenter le principe de base d'un moteur de jeu vidéo
- De vous faire manipuler des listes d'objets (Collections)
- De mettre en oeuvre les bonnes pratiques de la POO
- De vous faire appliquer les bonnes pratiques de Git

Etapes

0. (Ensemble) Prendre connaissance la théorie sur le contenu et le nommage des commits Git. A partir de là, chaque étape doit faire l'objet d'un commit (sauf s'il s'agit de théorie)

1. Créer un projet de type console, nommé "ParaClub", dans votre repository local (de votre fork), puis:

    - Ajouter la classe "Config" dans votre projet, avec le code suivant dedans:

      ```
      static class Config
      {
          public const int SCREEN_HEIGHT = 40;
          public const int SCREEN_WIDTH = 150;
      }
      ```

      Attention à bien mettre le mot-clé `static` !
    - Utiliser les méthodes de la classe Console pour dimensionner la fenêtre. Pour accéder à vos constantes depuis n'importe quel endroit du code, servez-vous de la syntaxe:

      ```
      Config.SCREEN_HEIGHT
      ```

2. Faire une classe `Plane` et l'utiliser pour afficher l'avion au moyen d'un tableau de string. Par exemple:

    ```
    private string[] view =
        {
            @" _                          ",
            @"| \                         ",
            @"|  \       ______           ",
            @"--- \_____/  |_|_\____  |  ",
            @"  \________ --------- __>-} ",
            @"        \_____|_____/   |  "
        };
    ```

3. Faire bouger l'avion de gauche à droite en se basant sur la structure de moteur de jeu:

```
while (true)
{
    // Modifier le modèle (ce qui *est*)
    plane.update();
    ...

    // Modifier ce que l'on *voit*
    Console.Clear();
    plane.draw();

    // Temporiser
    Thread.Sleep(100);
}
```

4. Faire une classe `Para` qui modélise un parachutiste, l'utiliser pour mettre un parachutiste nommé "Bob" dans l'avion (mais sans qu'on ne le voie)

5. Prendre en compte les touches frappées par l'utilisateur pour terminer le programme quand il tape `ESC`. Snippet:

```
    if (Console.KeyAvailable) // L'utilisateur a pressé une touche
    {
        keyPressed = Console.ReadKey(false);
        switch (keyPressed.Key)
        {
            case ConsoleKey.Escape:
                ...
                break;
            ...
        }
    }
```

6. Permettre à l'utilisateur de "pousser" le parachutiste hors de l'avion en pressant la barre d'espace. Avec un parachute fermé, le parachutiste ressemble à ceci:

```
     private string[] withoutParachute =
     {
         @"     ",
         @"     ",
         @"     ",
         @"  o  ",
         @" /▒\ ",
         @" / \ ",
     };
```

7. Faire tomber le parachutiste de trois lignes à chaque cycle du game engine. Le faire s'arrêter sur le sol

8. Faire ouvrir le parachute au parachutiste à la moité de la hauteur de l'écran. Avec le parachute ouvert, il ne descend plus que d'une ligne à chaque cycle. Avec un parachute ouvert, le parachutiste ressemble à ceci:

```
    private string[] withoutParachute =
    {
        @" ___ ",
        @"/|||\",
        @"\   /",
        @" \o/ ",
        @"  ▒  ",
        @" / \ ",
    };
```

9. Etudier et comprendre la classe List<T>

10. Faire un club de parachutistes dans le programme principal avec une liste de parachutistes `List<Para>`. Modifier votre avion pour qu'il aie à son bord une liste de parachutistes (ce ne sont pas obligatoirement ceux du club!) et qu'il en éjecte un à chaque fois que l'utilisateur presse la barre d'espace.