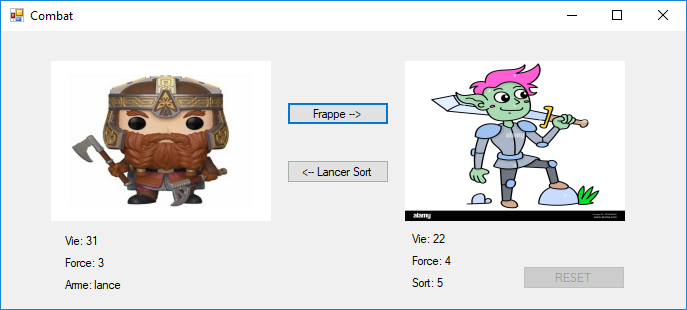
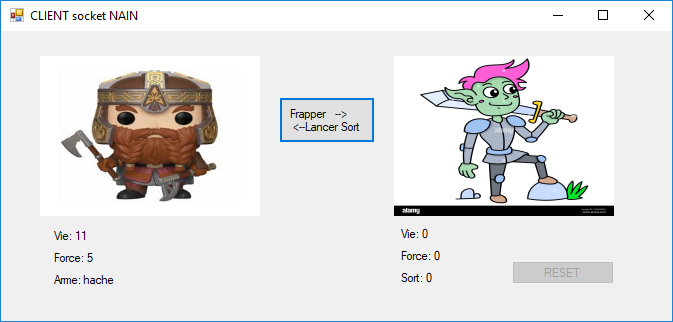
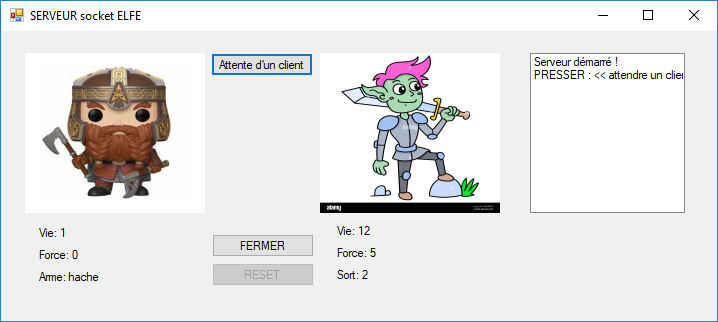
**TP5b Combat Elfe-Nain par socket (10%) 16/11/21**

Nous voulons transformer ce populaire jeu de combat en un jeu de combat à distance. Nous utiliserons les mécanismes de sockets pour le faire.



Voici les interfaces des projets fournis par **Léa**. Le projet **client** (le joueur est le nain) et le projet **serveur** (le joueur est l’elfe).

**Le projet serveur (elfe) : Le projet client (nain) :**



En vous servant de l’exercice à compléter du cours, vous devez compléter les projets zippés fournis par **Léa** et les envoyer pour correction par **GitHub** au plus tard **le 29 novembre** sans pénalité (-.5 par jour de retard, 0 après le 6 décembre).

**Barème de correction :**

**Le client : \_\_\_\_\_\_\_\_ / 3 pts**

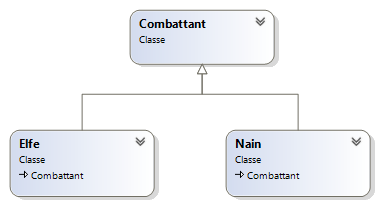
La note peut être diminuée selon les critères suivants : ne plante pas (-50%), respect des normes de programmation (déclarations et noms des variables et méthode, commentaires pertinents, -.25 par non-respect), optimisation (éviter les répétitions de bloc de codes par utilisation de boucle ou méthode, -.25 à -.5 lorsqu’applicable), ménage (enlever le code inutile, -.1 par omission).

**Le serveur : \_\_\_\_\_\_\_\_/ 5 pts**

La note peut être diminuée selon les critères suivants : ne plante pas (-50%), respect des normes de programmation (déclarations et noms des variables et méthode, commentaires pertinents, -.25 par non-respect), optimisation (éviter les répétitions de bloc de codes par utilisation de boucle ou méthode, -.25 à -.5 lorsqu’applicable), ménage (enlever le code inutile, -.1 par omission).

**Remise par GitHub : \_\_\_\_\_\_\_ / 2 pts**

Les classes suivantes sont fournies dans les projets et sont les mêmes dans les 2 projets :



public class Combattant

{

public int Force { get; set; }

public int Vie { get; set; }

public Image Avatar {get; set;}

}

public class Elfe:Combattant

{

public int Sort{ get; set; }

public Elfe(int vie, int force, int sort)

{

Vie = vie;

Force = force;

Sort = sort;

Avatar = Image.FromFile("elfe.jpg");

}

public void LancerSort(Nain nain)

{

nain.Vie = nain.Vie - Sort;

if (Sort > 1)

Sort--;

if(nain.Force > 1)

nain.Force--;

}

}

public class Nain:Combattant

{

public string Arme { get; set; }

string[] tArme = {"hache","pioche","lance" };

public Nain(int vie, int force, int noArme)

{

Vie = vie;

Force = force;

Arme = tArme[noArme];

Avatar = Image.FromFile("nain.jpg");

}

public void Frapper(Elfe elfe)

{

elfe.Vie = elfe.Vie - Force;

}

}