Elecsim est une plateforme développée en C# destinée à des chercheurs qui désirent simuler un réseau électrique à l’échelle désirée.

Ce dernier possède une plateforme facile à modifier.

Il est possible d’ajouter des sources productrices d’énergie, créer de nouveaux types de producteurs énergétiques, ajouter autant de consommateurs que l’on désire, créer de nouveaux types de consommateurs. Un système de météo est présent qui permet de simuler 3 paramètres :la force du vent, l’ensoleillement ainsi que la température. Un système de marché permet de gérer les prix unitaire pour plusieurs types d’énergies ainsi que le prix auquel l’électricité est achetée depuis l’étrangers.

Production

Lors du développement de elecsim, nous avons décidé de créer trois types de producteurs d’énergie sans pour autant limiter ce nombre à l’avenir (nucléaire, éolien, achat à l’étrangers).

Pour créer une centrale d’un type connu de elecsim, il faut ajouter du code dans la classe Update :

il faut commencer par déclarer le nom de la centrale (x1) au-dessus du constructeur *public update() :*

Centrale x1 ;

Il faut également ajouter une variable pour stocker la valeur de production :

double Xo1

Ensuite, il faut créer une instance au sein du constructeur *public Update()*:

x1 = new Eolien(2000, 1, "x1", market, bruxelles);

x1 est le nom utilisé dans le code lors d’appel de méthodes ou bien quand on doit récupérer ses attributs Eolien est le type de centrale crée (il existe pour le moment Nucléaire, Eolien, et Achat) au sein des parenthèses, il faut mettre dans le bon ordre les paramètres ce la centrale. On commence par la production maximale théorique exprimée en W, ensuite la quantité de CO2 produite par W, market est le nom de la plateforme qui gère les prix de production et d’achat dans notre programme (objet de type Market) et dans le cas de l’éolien, nous avons également en paramètre la plateforme de météo (bruxelles) pour pourvoir récupérer la force du vent et ainsi produire de l’énergie en fonction de ce dernier.

À présent, nous pouvons assigner la centrale à une ligne électrique. Si vous souhaitez raccorder la centrale à une ligne existante, il suffit d’ajouter le bloc suivant dans la partie « ASIGNATION DES LIGNES & GESTION ERREURS SURCHARGES COTE PRODUCTION »

Dans le cas de l’éolien nous pouvons choisir de désactiver sa production en pleine simulation avec cette condition que nous détaillerons plus tard.

if (status\_x1 == true){

Il faut commencer par placer dans la variable la valeur de production en appelant la méthode Get\_prod() de l’objet x1 :

double Xo1 = x1.Get\_prod();

Nous assignons la ligne l1 avec la méthode ligne\_in() et nous mettons en paramètres la production de la centrale :

Xo1 = l1.Ligne\_in(Xo1);

Il est également nécessaire d’ajouter cette production à la production totale se trouvant dans la variables prod\_tot :

prod\_tot += Eo1;

Cette partie gère l’affichage de message quand il y a une surcharge au niveau de la ligne véhiculant la production de x1 en précisant l’endroit et le moment ou la surcharge s’est passée :

if (Xo1 == 0)

{

erreurs += DateTime.Now.ToString();

erreurs += errors.Line\_Overload(l1);

erreurs += "\n";

}

Par défaut, il existe déjà une centrale éolienne donc dans les ligne suivante nous devons incrémenter les valeurs de Co2 émis, du cout de production ainsi que la production de l’ensemble des centrales éoliennes:

prix\_eolien += x1.Get\_prix();

co2\_eolien += x1.Get\_co2();

prod\_eolien += Xo1;

}

Dans cette condition nous affichons un message lorsque la centrale est désactivée durant la simulation :

else

{

erreurs += DateTime.Now.ToString();

erreurs += errors.Deactivate(x1);

erreurs += "\n";

}

Si x1 est un type de centrale déjà existant, il n’y a plus de manpulation s à faire.

Cependant, s’il s’agit d’un nouveau type de centrale, il faudra créer quelques variables en plus avant de passer dans la classe view\_graphe.

Il faut créer une variables associées au cout de production, au CO2 ainsi qu’à la production au-dessus du constructeur *public Update()* :

public double prix\_fusion;

public double co2\_fusion;

public double prod\_fusion;

Il faut également ajouter la production de ce type de centrale dans la variable totale à la fin de la classe :

total += prod\_fusion ;

À présent nous pouvons passer à la classe view\_graphe.

Dans la partie AFFICHAGE, nous devons créer un nouveau type de centrale et y afficher ses infos nous utilisons un richTextBox rtbMessage pour faire cela :

rtbMessage.AppendText("Fusion: "

+ up.prix\_fusion.ToString() + " € " //prix\_x1

+ up.co2\_fusion.ToString() + " g " //co2\_x1

+ up.prod\_fusion.ToString() + " W " //prod-x1

+ "\n");

Pour gérer l’état d’une centrale, il suffit de créer un checkbox

if (cbEolien.Checked)

{

up.status = true;

}

else

{

up.status = false;

}

Nouveau type de centrale