# Projet de "Langages à Objets Avancés

Rapport

Mohamed Lezhari KHORISSI

2021/2022

# Introduction :

le projet a pour objectif de simuler un circuit combinatoire avec un langage orienté objet: le C++).

Les classes sont représentées par un diagramme de classe.

# Présentation du programme :

Le programme consiste d'une classe mère "Gate" abstraite, dont tous les portes du circuit sont des classes héritantes de cette dernière.

La classe a la possibilité d'ajouter des portes au circuit (schéma).

## Les portes logiques:

Les portes logiques ont un attribut de type vecteur de "Gate" puisque on peut avoir des portes avec 2 arguments et une porte avec un seul (Not), donc ce choix est plutôt un choix d'optimisation.

```
OrGate::OrGate( Gate* const g1,  Gate* const g2) : GateLogique{{g1,g2}} {Calcul();}
```

La Fonction "Calcul" permet d'un côté calculé le résultat logique et avoir la formule et d'autre côté: actualiser ces valeurs à chaque changement apporté à la porte. Elle est redéfinie dans chaque classe puisque il y'a une différence de logique et de formule.

```
void NxorGate::Calcul(){
    std::string s = this->FonctionLogique("nxor");
    this->setFormule(s);
    this->setVal(!(this->getPortes().at(0)->getVal()^this->getPortes().at(1)->getVal()));
}
```

# Les entrées/sorties:

une classe "InputOutputGate" représente les portes d'entrée et de sortie, la valeur logique de l'input est donné lors de la création de l'objet et peut être contrôlé après.

```
InputGate* a = new InputGate('a',0);
```

Les outputs ont une fonction de calcul exactement comme les portes logiques.

# Circuit :

Un circuit se compose de 3 vecteurs (inputs , potes logiques et outputs) et un schéma pour la visualisation, le schéma est construit par rapport au nombre de portes d'entrées : On remplit un vecteur 2D de strings vides puis on construit pas par pas le schéma final.

La fonction "AfficherCircuit" permet le pas par pas, puisque à chaque fois le schéma est réinstallé et la valeur logique est recalculée.

La fonction "Save" permet de sauvegarder le circuit en un fichier texte.

# Schéma :

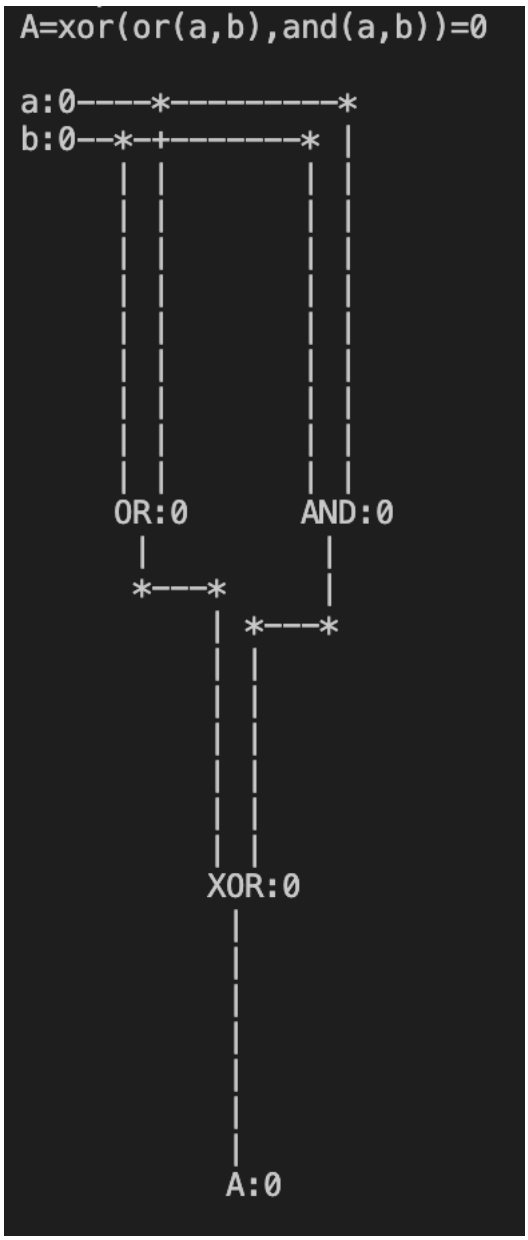
Le schéma est très interactif avec la classe circuit puisque c'est de là où on peut construire l'affichage (dessiner des tirets, gérer les intersections ...), et  du coup les fonctions de schéma sont utiliser par la fonction "AfficherCircuit".

Aide: https://www.daniweb.com/programming/software-development/code/216430/add-a-little-graphics-to-your-console

Les fonctions "Ajout" permets d'actualiser le vecteur 2D pour avoir de l'espace pour les différentes portes.

La fonction "ShowcaseLigne" ajoute une ligne vertical ou horizontale au schéma (gère aussi les intersections).

"GateOut" et "GateLogic" permet de trouver l'emplacement des portes et compléter le schéma.




La grande hauteur des portes et des fils permet d'éviter un affichage encombré et un risque d'une erreur "out of bounds"

# Conclusion:

Le simulateur de circuit combinatoire fonctionne bien, on peut visualiser le mode par à pas.

Néanmoins on remarque des fonctionnalités manquantes, qui n'ayant pas été établies , notamment synthétiser le circuit à partir de son expression textuelle et le relecture du circuit depuis un ficher.