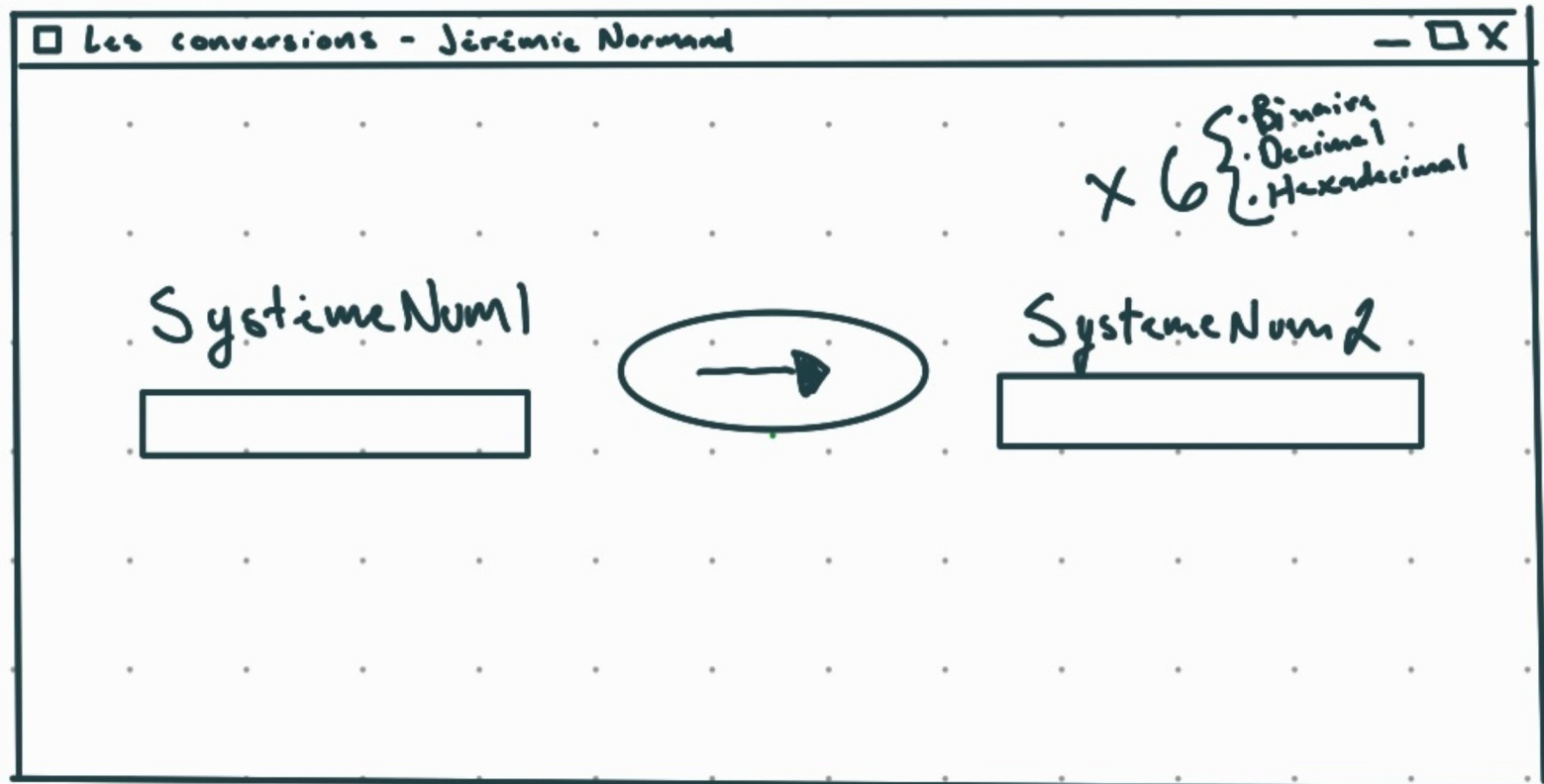


PGS

ÉBAUCHE

Lab-1

## Interface Graphique



# Éléments importants

## BOUTONS :

Binaire  $\rightarrow$  Décimal :

- Valider Si Binaire
- Convertir Binaire En Décimal

Décimal  $\rightarrow$  Binaire :

- Valider Si Décimal
- Convertir Décimal En Binaire

Hexa  $\rightarrow$  Décimal :

- Valider Si Hexadécimal
- Convertir Hexadécimal En Décimal

Decimal  $\rightarrow$  Hexa :

- Valider Si Decimal
- Convertir Decimal En Hexadecimal

Binaire  $\rightarrow$  Hexa :

- Valider Si Binaire
- Convertir Binaire En Hexadecimal

Hexa  $\rightarrow$  Binaire :

- Valider Si Hexadecimal
- Convertir Hexadecimal En Binaire

# BILAN :

Validation (x3) :

- Binaire - Decimal - Hexadecimal

Conversion (x6) :

- BD - DB - HD - DH

- BH - HB

# Algorithmes

## CONVERSIONS:

```
int BintoDec(str binaire)
    int decimal : 0;
    foreach (chr chiffre in binaire)
        if (chiffre == '1')
            decimal += 2^(binaire.length - 1 - chiffre.index);
    return decimal;
```

```
str DecToBin(int decimal)
    str binaire : null;
    while (decimal > 0)
        binaire : "{decimal % 2}{binaire}";
        decimal /= 2;
    return binaire;
```



```
int HexToDec(str hexadecimal)
```

```
int decimal : 0;
```

```
foreach (chr chiffr in hexa)
```

```
if (chiffr >= '0' || chiffr <= '9')
```

```
    decimal += 16(hexa.Length - 1 - chiffr.Index) * int(chiffr);
```

```
elif (chiffr >= 'A' || chiffr <= 'F')
```

```
    decimal += 16(hexa.Length - 1 - chiffr.Index) * (10 + code(chiffr) - code('A'));
```

```
return decimal;
```

```
str DecToHex(int decimal)
```

```
str hexa : null;
```

```
while (decimal > 0)
```

```
    hexa = "{decimal / 16}{hexa}";
```

```
    decimal /= 16;
```

```
return hexa;
```

```
str BinToHex(str binaire)  
return DecToHex(BinToDec(binaire));
```

```
str HexToBin(str hexa)  
return DecToBin(HexToDec(hexa));
```





```
bool ValiderDimensionDec(str decimal)
return decimal.Length < 4 ? true : false;
```

```
bool ValiderDecimal(str decimal)
bool valide : false;
if (int.TryParse(decimal, out valide))
    if (int(decimal) >= 0 && int(decimal) < 256)
        valide : true;
return valide;
```

```
bool ValidDimensionHex(str hexa)  
    return hexa.Length < 3 ? true : false;
```

```
bool ValidHexadecimal(str hexa)
```

```
    bool valide : true;
```

```
    foreach (char chiffre in hexa)
```

```
        if (valide)
```

```
            if (code(chiffre) < code('0') || code(chiffre) > code('9'))
```

```
                if (code(chiffre) < code('A') || code(chiffre) > code('F'))
```

```
                    valide : false;
```

```
    return valide;
```