|  |
| --- |
| HES-SO |
| Projet C++ : Logic Emulator |
| Rapport |

|  |
| --- |
| Gilles Mottiez  16/06/2018 |

Table des matières

[Introduction 2](#_Toc516727340)

[But et cahier des charges 2](#_Toc516727341)

[Développement 2](#_Toc516727342)

[Chargement des fichiers 2](#_Toc516727343)

[Décodage du fichier Json et génération du modèle logique 2](#_Toc516727344)

[Calcul des sorties selon le type de portes et des entrés 2](#_Toc516727345)

[Affichage graphique 2](#_Toc516727346)

[Modification des entrées et validation des modifications 2](#_Toc516727347)

[Problèmes et solutions 2](#_Toc516727348)

[Résultats 2](#_Toc516727349)

[V0.0 2](#_Toc516727350)

[V1.0 2](#_Toc516727351)

[V2.0 2](#_Toc516727352)

[Conclusion 2](#_Toc516727353)

# Introduction

# But et cahier des charges

Concevoir un software qui permette la simulation d’un système de logique numérique. Le fonctionnement est le suivant :

* Des fichiers JSON sont écrit selon un modèle précis, qui est défini dans l’annexe « JSON\_logicProtocol ».
* Le software permet de charger un de ces fichiers, de générer un modèle de ce fichier en porte logiques, d’en calculer l’état de ses portes et d’afficher une représentation graphique.
* Grâce à une fenêtre textuelle, l’utilisateur peut changer des valeurs sur des entrées sans avoir à modifier tout le fichier et à le recharger complétement

Versions :

* V0.0
  + Charger un fichier Json dans le software
  + Implémentation des portes AND, OR, NOT
  + Création de la représentation graphique
  + Génération de la logique
* V1.0
  + Création de bloc logiques tels que XOR, NAND, NOR, …
  + Possibilité d’éditer le code directement à l’écran pour la modification des signaux d’entrée
* V2.0
  + Bascule D

# Développement

## Chargement des fichiers

## Décodage du fichier Json et génération du modèle logique

## Calcul des sorties selon le type de portes et des entrés

## Affichage graphique

## Modification des entrées et validation des modifications

# Problèmes et solutions

# Résultats

## V0.0

## V1.0

## V2.0

# Conclusion