# Jeu du Simon

## Projet Embarqué

## Réalisé par :

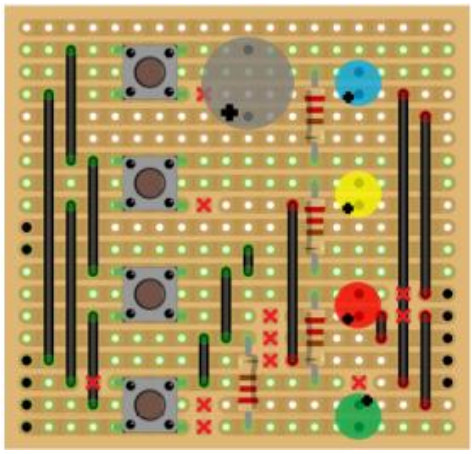
*João CARVALHO SANTOS*

# Table des matières

# Le schéma

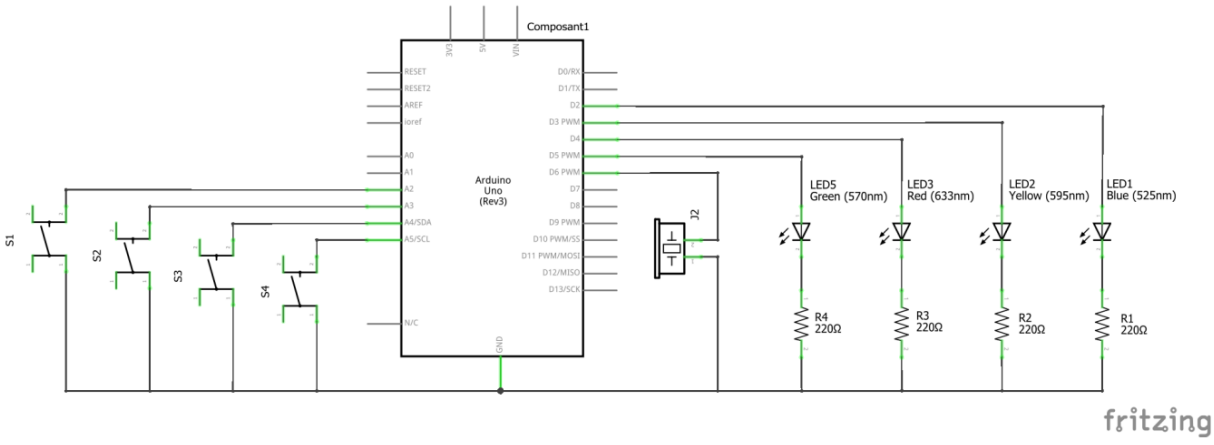
## Schéma physique

Pour le Jeu du Simon 4 boutons colorés sont nécessaires, dans ce cas 4 boutons et quatre LED's ont été placés en parallèle pour les remplacer. Des résistances de 220Ω sont reliés à la cathode de chaque LED à l'aide de ponts en métal. Un buzzer est aussi présent, ce dernier s'occupe des sons. Des coupures de piste (représentées par des croix) ont aussi été faites pour éviter des courts-circuits. Le tout a été placé /soudé sur un veroboard, et le veroboard sur un Arduino. Voilà le schéma physique présenté avec l'image ci-dessous :



## Schéma logique

Chaque LED est reliée à une sortie logique/digitale (ports 2 à 5) ainsi que le buzzer (port 6). Les boutons sont reliés à des entrées logiques/analogiques (ports A2 à A5). Chaque composant est relié à la terre (GND). L'image ci-dessous représente le schéma logique :



# Stratégie utilisée

## Diagramme de flux

Pour commencer ce projet j'ai fait des diagrammes de flux à l'aide de draw.io. Ces diagrammes m'ont aidé à construire « la base » du code, à savoir quel chemin parcourir pour arriver à un certain endroit, ainsi que à mieux construire mes boucles. La dernière version de mes diagrammes des flux se trouvent en annexe.

## Établir les entrées/sorties

Pour débuter le code, j'ai inclus la librairie « Bounce2.h ». Cette librairie m'aidera à contrôler quand est-ce que l'état logique du bouton est passé de 0 à 1 et vice-versa (différent de tester s'il est à 0 ou à 1). J'ai initialisé chaque bouton, LED et le buzzer. J'ai remplacé chaque port par une constante, de cette forme, au lieu de devoir à chaque fois utilisé « A2 » par exemple, je pourrais juste utiliser le mot « buton1 ». Pour finir j'ai écrit quelques lignes de code de teste, qui m'ont permis de facilement tester le fonctionnement de chaque composant, je les ai supprimés après avoir fini tous mes testes.

## Coder suivant mon diagramme

Le moment de commencer à coder le jeu est arrivé ! J'ai commencé calmement à « traduire » mon diagramme, en mettant chaque bout de code dans fonction et en commentant le principal, de cette forme je ne me perdrais jamais. En faisant ça j'ai pu trouver des erreurs et bugs, qui venait déjà de mon diagramme de flux. J'ai donc décidé de mieux découvrir les erreurs et de comment faire juste, et après chaque solution, je mettais à jour chaque différence entre mon code et mon diagramme de flux.

### Tester

Je testais le jeu à la fin de chaque fonction, de cette manière, je ne me perdais pas et savais où chercher au cas où il y avait un bug. Quand j'ai fini le code, j'ai fait plusieurs tests, mais j'ai aussi demandé a d'autres personnes de le tester. La faite d'avoir différents points de vue pour tester un code m'a beaucoup aidé.

### Vérifier le code une dernière fois

Quand j'ai fini de coder, et de bien tester le jeu, j'ai pris mon temps de bien regarder chaque ligne de code. J'ai mis à jour mon en-tête, j'ai fini de commenter et j'ai cherché des fautes d'indentation, d'aération et d'optimisation.

## Conclusion

### Partie physique

Le circuit du jeu est réussi ! Toutes les demandes du cahier des charges ont été réussies, tous les composants fonctionnent bien seuls, ainsi qu'entre eux, cependant, il n'y est pas parfait. Les ponts ne sont pas très beaux visiblement et certains composants ne sont pas très droits, de la soudure en trop peut aussi être trouvé sur les pâtes de certains composants. J'ai trouvé la partie physique du jeu assez dure, mais faisable avec du calme et patiente, ça m'a beaucoup plu.

### Partie logique

Le code du jeu est réussi ! Toutes les demandes du cahier des charges de cette partie ont tous aussi été réussis, le jeu fonctionne bien, le code est commenté, aucun bug a été trouvé, cependant, il n'y est pas parfait. Le code pourrait faire moins de lignes, et les extra demandes (2$^{ème}$ mode de jeu et accélération de la séquence tous les 3 codes) n'ont pas été réussites. J'ai trouvé la partie logique très dure, même avec de l'aide et beaucoup de recherches, car c'est la première fois que je code un jeu physique, mais pour finir, ça m'a quand même plu.

## Documentation

La documentation du jeu est réussie ! Toutes les questions peuvent être répondu à partir de ce document. Une mise en page simple en thème bleu a été mis en place, ainsi qu'une page de garde, une table de matières, des en-têtes et des pieds de page, ainsi que des images. J'ai trouvé la documentation assez facile et plaisante.