

Concordia University, Computation Arts

ASSIGNMENT THEN

CART 360, Fall 2018

Mélina Lopez-Racine

GitHub repository:

https://github.com/mlopezracine/CART360/tree/master/ASSIGNMENTS/ASSIGNMENT_THEN

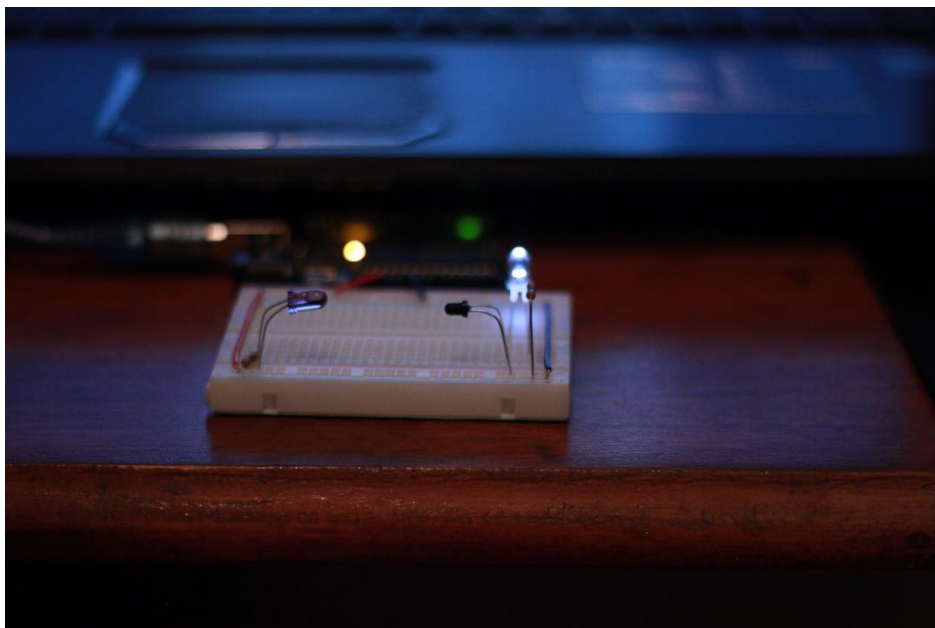
Concept:

Initialement, le concept de ce projet était de créer un contrôleur alternatif qui propose d'éviter l'utilisation des mains, conçu pour une expérience unique au jeu pour lequel il est associé. De plus, je souhaitais intégrer un système de communication à deux sens, dans le but de modifier l'état de la manette selon les résultats du programme (jeu).

Depuis ma proposition, j'ai sélectionné un émetteur et des capteurs de phototransistor pour diriger mon bateau de l'aquarium à l'écran. De plus, j'ai ajouté un module accéléromètre dans le but d'ajouter une mécanique de jeu qui limite la force du souffle des joueurs. Cependant, lors du développement, j'ai abandonné l'idée d'intégrer un système de communication à deux sens. J'ai concentré mes efforts sur la communication entre la plateforme Arduino et le moteur de jeu Unity, en plus de me concentrer sur la création d'un circuit électronique efficace et fonctionnel. Au final, il me semblait plus important d'intégrer une mécanique de jeu supplémentaire qui complimenterait l'interaction première, c'est-à-dire l'acte de souffler.

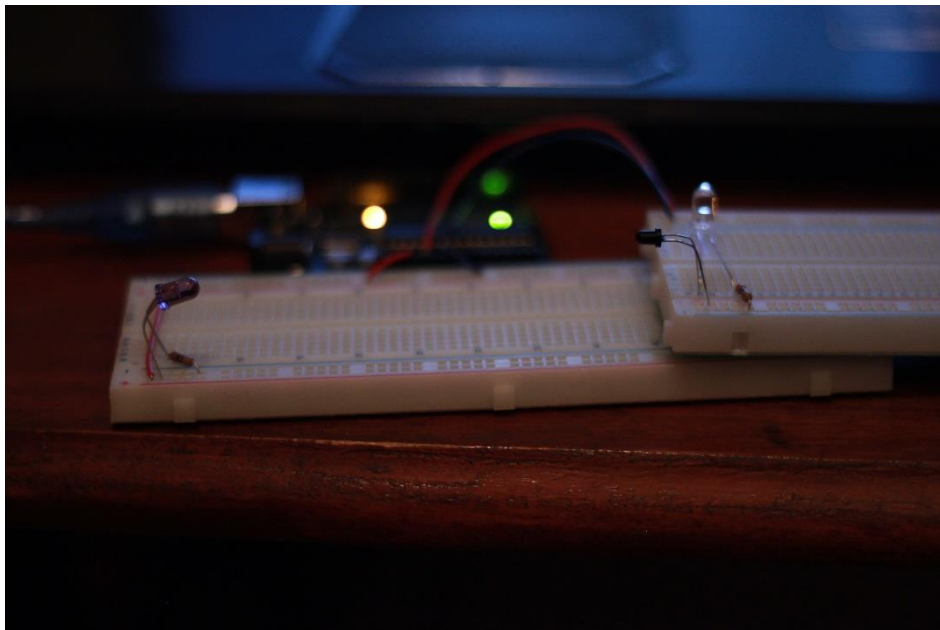
Développement :

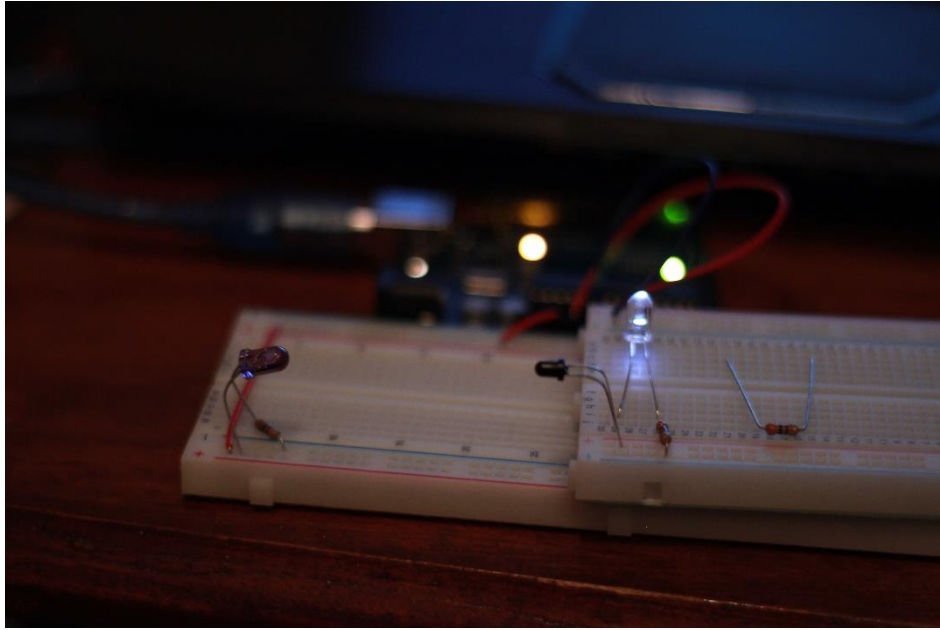
J'ai reproduit le circuit de l'exercice « Circuit Exercise 2 : Analog Switches, II. Infrared », tel que démontré en classe le 14 septembre 2018. Pour ce faire, j'ai utilisé un émetteur infrarouge ainsi qu'un capteur IR, chacun relié à des résisteurs de 220 Ohm. Ensuite, l'ensemble du capteur IR et son résistor ont été reliés à une diode lumineuse, dans le but de déterminer l'efficacité du circuit.



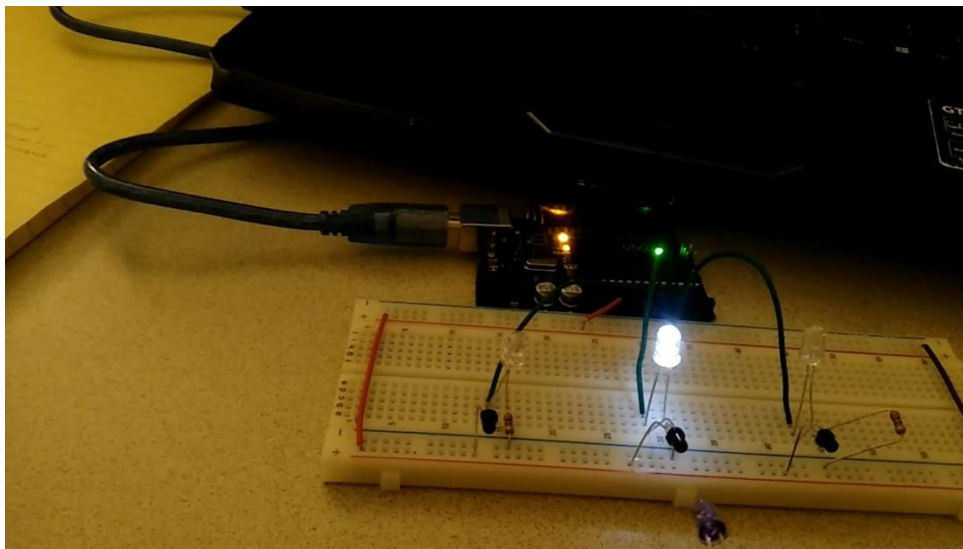


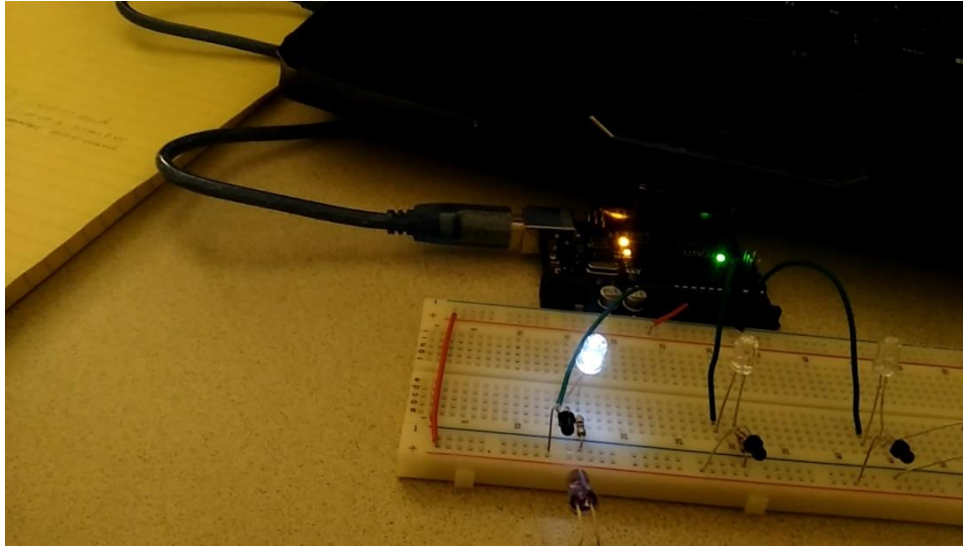
Par la suite, j'ai divisé ce circuit sur deux planches de montage expérimental. Cela m'a permis d'évaluer la distance optimale du capteur, en plus d'obtenir une distance maximale approximative. Ces tests m'ont permis de construire mon premier prototype de manette, composé d'une diode électroluminescente infrarouge, de résisteurs et d'une batterie, qui a la capacité d'être déplacé dans l'espace environnant.



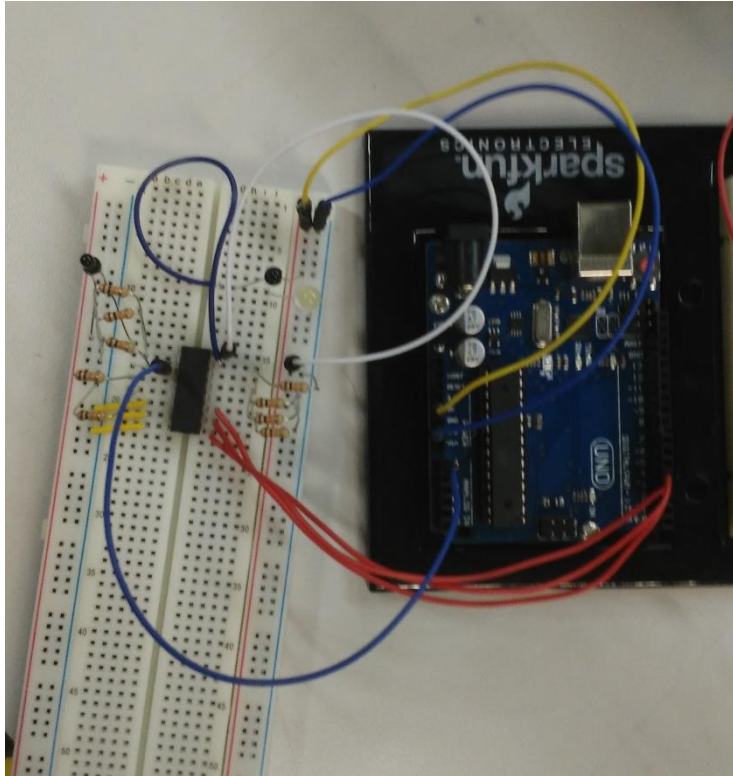


Ensuite, j'ai utilisé ma plus grande planche de montage expérimental pour construire un circuit composé de trois capteurs IR, trois résisteurs de 220 Ohm ainsi que trois diodes lumineuses, le tout directement relié à ma plate-forme Arduino. Chaque capteur est connecté à une entrée convertisseur analogique-numérique.





Par la suite, j'ai modifié mon circuit pour créer un troisième prototype, similaire à celui qui a été étudié lors du cours tenu le 10 novembre 2018. À la différence des prototypes précédents, celui-ci comporte un multiplexeur, capable de supporter plusieurs capteurs IR et de les concentrer sur la même voie de transmission. Ainsi, il m'est possible de brancher huit transistors de détection photos infrarouges sur un seul convertisseur analogique-numérique. L'ensemble est composé de neuf résisteurs de 10K Ohm et de huit capteurs IR.

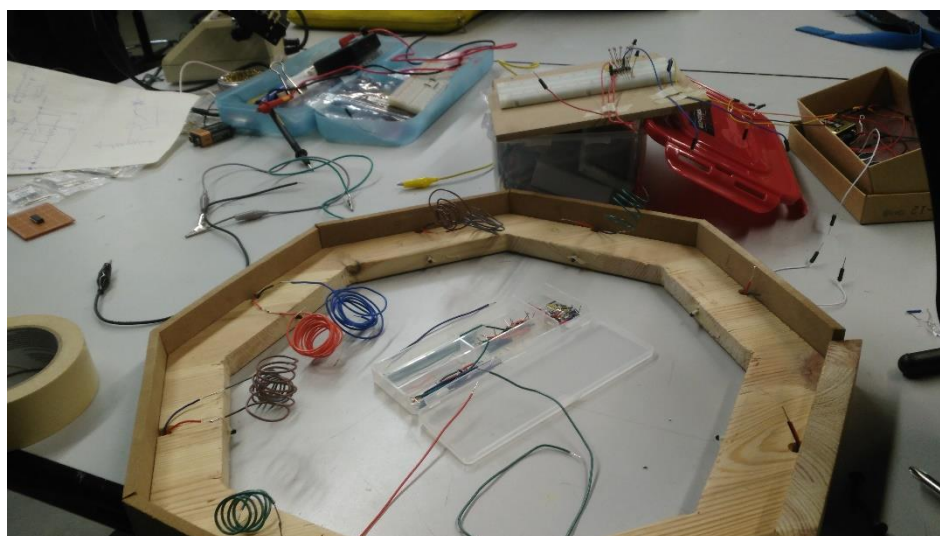
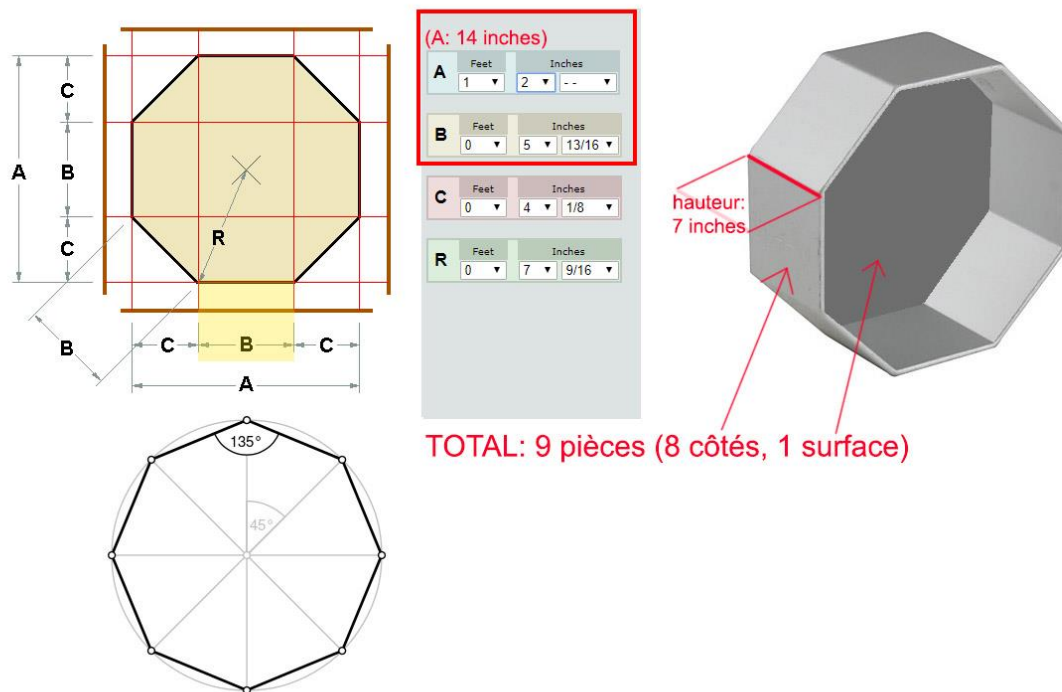


La difficulté de cette étape se rapporte surtout à la manière de communiquer correctement l'information reçue du convertisseur analogique-numérique au moteur de jeu Unity. C'est l'étape qui m'a demandé le plus de temps, puisque j'éprouvais de la difficulté à bien traduire cette information d'une plate-forme à une autre. À ma connaissance, Unity est uniquement capable de recevoir des strings de la part de la plateforme Arduino. L'ensemble des strings parvient du serial monitor, ce que Unity n'est pas capable de déchiffrer par lui-même. Ainsi, la difficulté du projet était d'envoyer un string d'information à Unity et trouver un moyen d'accéder à chaque élément individuellement. Pour résoudre la situation, j'ai utilisé l'élément « contains », qui retourne un boolean positif ou négatif en présence de la valeur contenue. De la plate-forme Arduino, j'ai envoyé un ensemble de strings contenant des lettres, chacune associée à des coordonnées dans Unity. La fonction dans Unity vérifie si le message contient un caractère particulier. Dans un cas positif, la fonction déplace le bateau selon les nouvelles coordonnées.

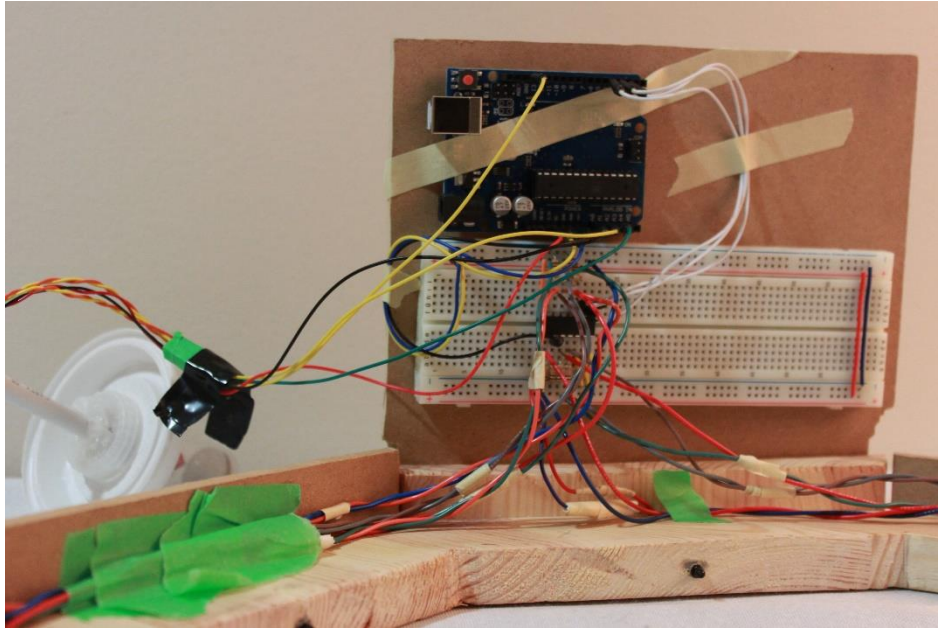


Tout d'abord, je tiens à noter que pour éviter toute blessure avec l'équipement, mon père m'a aidé lorsqu'il fût le temps de découper les matériaux.

L'aquarium est composé de 9 feuilles de plexiglass collées ensemble à l'aide de scellant silicone, formant ainsi un octogone. Une planche de bois découpée en forme octogonale sert de base pour soutenir l'aquarium. Un « couvercle » fait le contour de l'aquarium. La base mesure 16 pouces et possède une épaisseur de $\frac{3}{4}$ pouces, tandis que le couvercle occupe une surface de $16\frac{1}{2}$ pouces. L'épaisseur du couvercle mesure aussi $\frac{3}{4}$ pouces. Chaque côté mesure 2 pouces de large, par $6\frac{1}{2}$ pouces de largeur (à partir du milieu). La finition externe qui cache les fils mesure $1\frac{1}{2}$ pouces, et fait 7 pouces de longueur. L'aquarium occupe une surface de 14 pouces et fait 7 pouces de haut, chaque côté mesure $5\frac{13}{16}$ pouces. L'une des planches de finition externe a été allongée pour permettre de poser la planche de montage expérimental et la plate-forme Arduino. Un morceau de bois a été ajouté pour permettre d'obtenir l'espacement nécessaire pour passer l'ensemble des matériaux électroniques. Au départ, je voulais créer un aquarium assez profond pour pouvoir couler le bateau à l'aide un mécanisme. C'est pour cette raison que l'aquarium possède une profondeur de 7 pouces. Lors du développement, l'idée de couler le bateau a été retirée, par soucis de manque de temps. Je désirais me concentrer sur la communication entre Unity et Arduino, en plus de travailler davantage sur mon circuit.

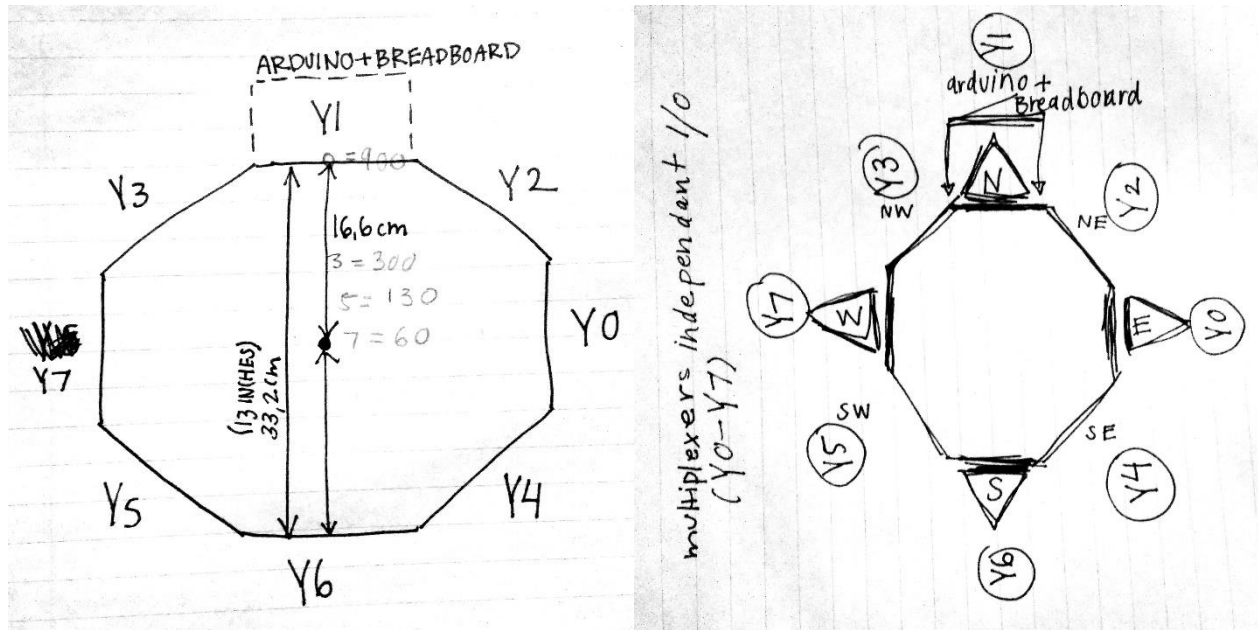


Au centre de chaque côté interne, un trou a été creusé pour pouvoir y insérer un capteur IR. La sortie de chaque trou se fait en dessous du couvercle, à l'extrémité du côté. Les fils sont par la suite alignés sur le contour du couvercle. L'ensemble des fils rejoignent le côté « Nord », là où se trouve l'extension de la finition externe. Ensuite, la plate-forme Arduino est directement connectée à un ordinateur portable.



Le bateau a été construit à partir d'une soucoupe de plastique, sur laquelle j'ai collé une tige verticalement pour créer le mat. La voile a été faite à partir d'un séparateur transparent pour cartable, dans lequel j'ai découpé un triangle. J'ai choisi cette forme, après quelques tests avec des variantes, dû à son inefficacité à propulser le bateau à travers l'aquarium. De cette manière, l'objet oblige le joueur à souffler davantage, avec plus de force, pour déplacer le bateau. De plus, cela a des chances d'activer le module accéléromètre et de créer davantage de vagues, conséquemment engageant une deuxième mécanique de jeu (voir fonctionnement du code). Pour maintenir la voile sur le mat, en plus d'avoir mis de la colle, j'ai apposé une attache à sac de pain (préalablement fondue dans une forme circulaire). La batterie ainsi que la diode électroluminescente ont été placées à l'intérieur d'une fiole pour médicaments. L'avantage de ce contenant est qu'il est imperméable et facilement malléable : j'ai percé un trou sur le couvercle de la fiole puis glissé la diode à l'extérieur et sceller le trou avec de la colle. La fiole a été collée à l'intérieur de la soucoupe. Pour balancer le poids et assurer que la diode dirigeait son rayon vers les capteurs IR, j'ai utilisé de la monnaie.

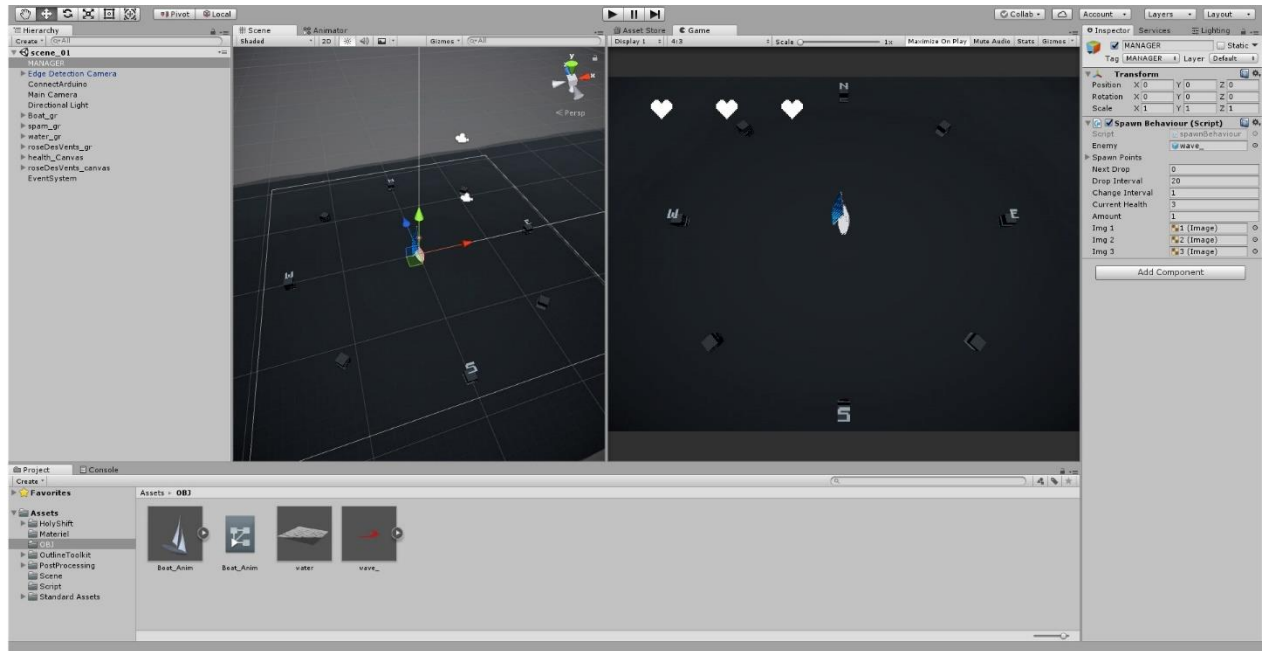
Pour la bouée, j'ai utilisé une technique similaire à celle de la création du bateau. J'ai pris une fiole pour médicaments puis fais un trou sur le couvercle. À travers le trou, j'ai placé une gaine thermorétractable, dans lequel les fils du module accéléromètre passent.



Chaque capteur IR a été connecté au multiplexeur puis identifié pour permettre de retracer les valeurs reçues à son capteur. De plus, pour obtenir un positionnement similaire dans le jeu, j'ai identifié l'intensité du courant perçu par les capteurs selon la distance puis créé des intervalles pour organiser ces valeurs. Cela m'a permis de recréer ces distances dans le moteur de jeu Unity.

Création numérique (Unity) :

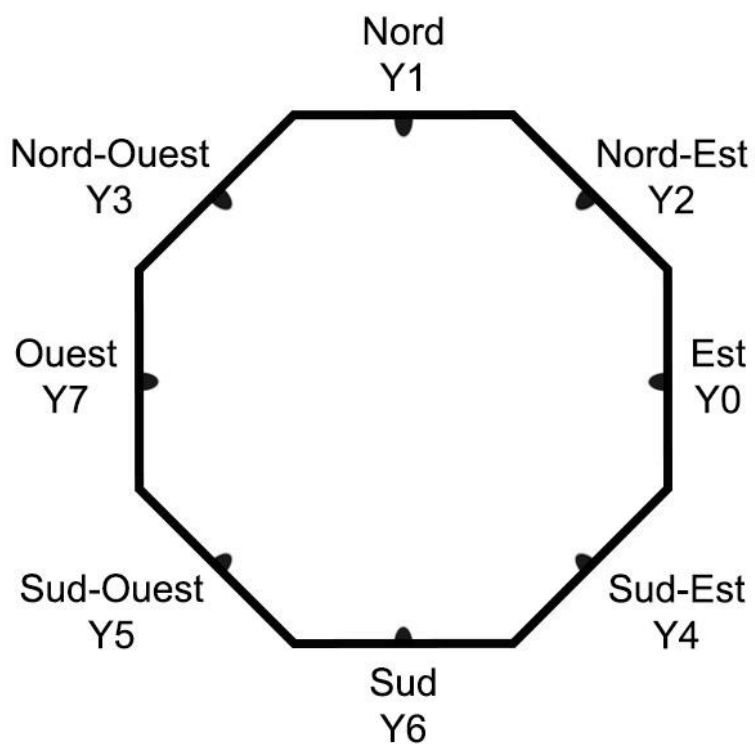
Pour la création du jeu associé à ce projet, j'ai utilisé le moteur de jeu multi-plateforme Unity, version 2017.4.2f2. Le modèle 3D du bateau a été téléchargé en ligne à partir du site internet TurboSquid. Bien que le modèle originale appartient à l'utilisateur Direct3DD, j'ai apporté des modifications pour convenir aux besoins de mon prototype.



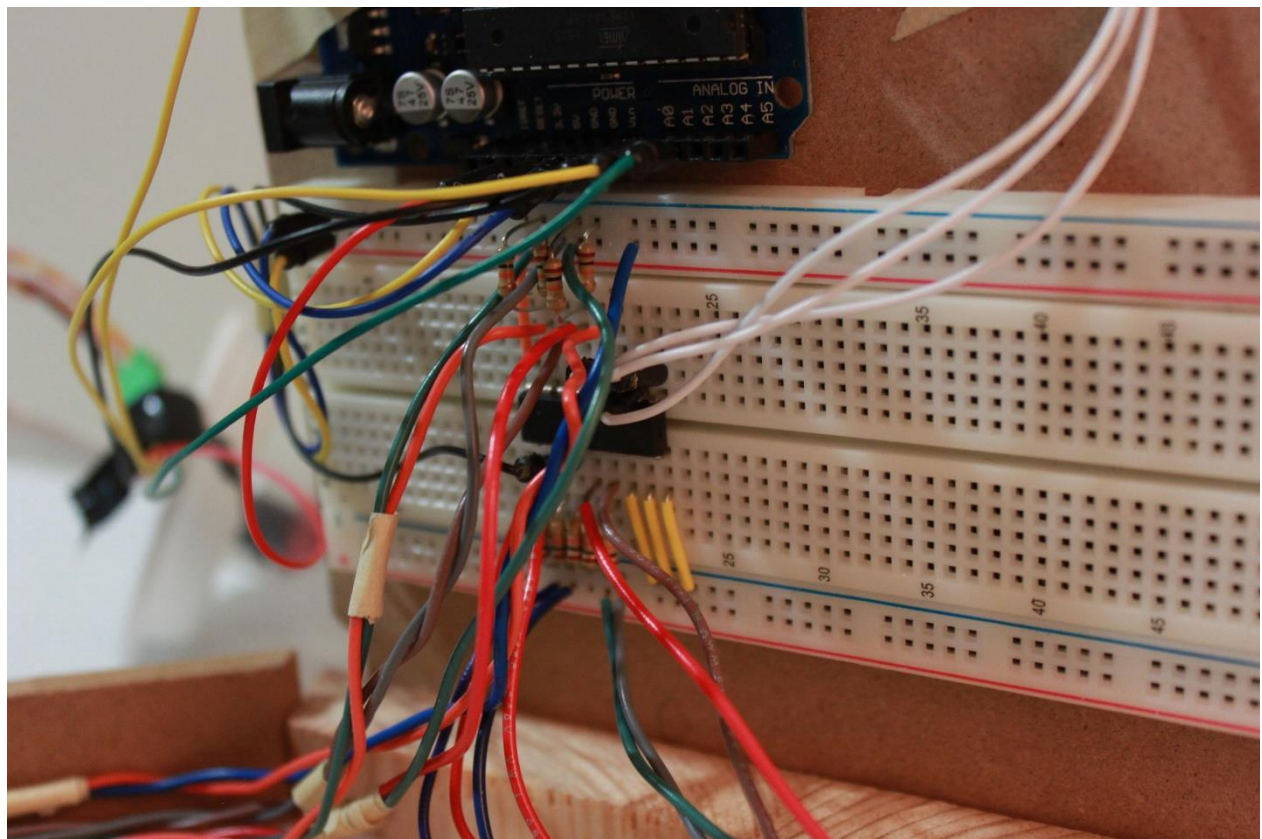
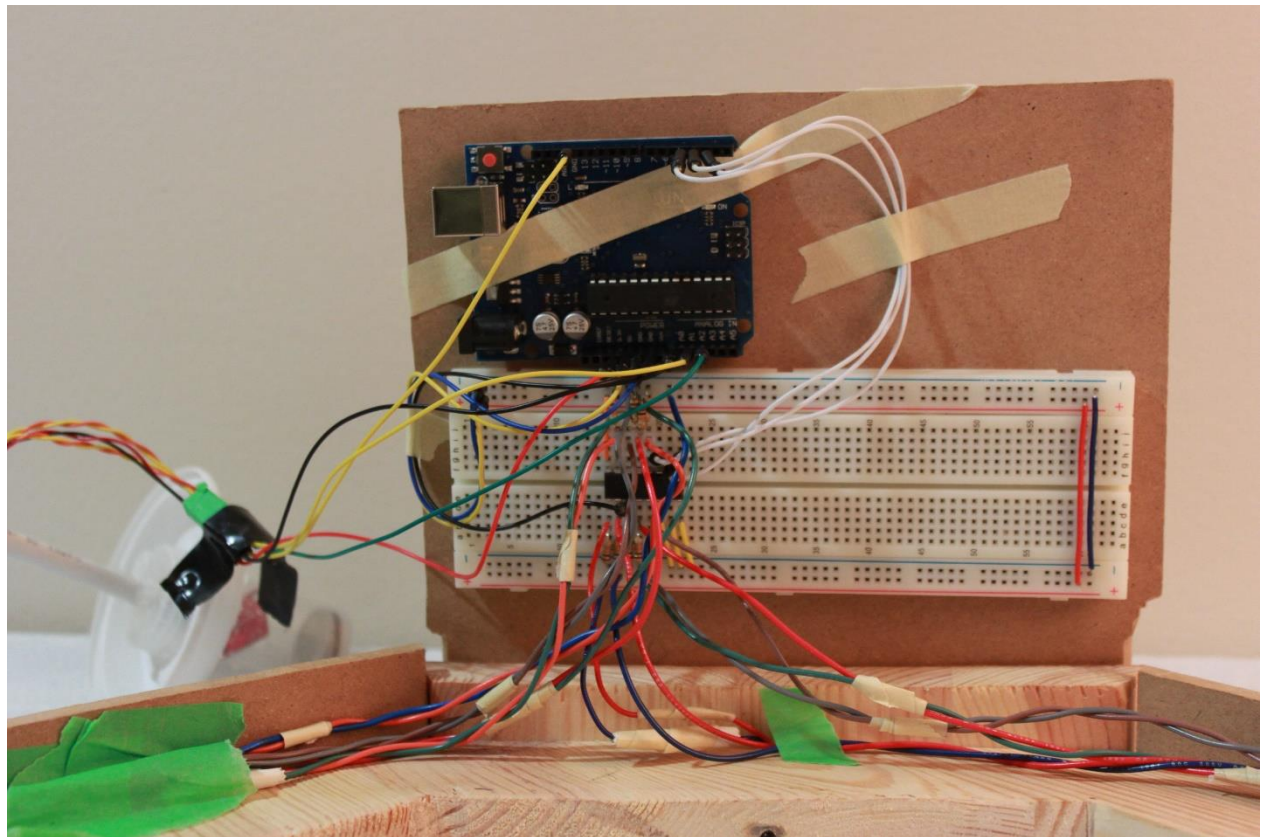
Dans l'ensemble, j'ai créé trois scripts différents pour rendre le prototype opérationnel; « spawnBehaviour » permet de contrôler la manière dont les vagues vont apparaître, tandis que « waveBehaviour » permet de gérer la manière dont les vagues vont réagir lors de leur apparition en jeu. Plus important, le script « untyArduinoComm » permet de communiquer avec la plate-forme Arduino. Chaque capteur IR a été associé avec un objet (cube) puis placé de manière à recréer la forme de l'aquarium. Des lettres représentant la rose des vents ont été placés pour aider les joueurs à reconnaître la direction dans laquelle l'aquarium est situé par rapport à celle virtuelle. L'interface de jeu est simple, ne comportant que le nombre de vie du bateau et que les lettres indiquant les points cardinaux de la rose des vents. Le *canvas* « health_Canvas » communique avec le script « spawnBehaviour », dans le sens où chaque image (cœur) disparaisse si le bateau est touché par une vague rouge. En tenant compte de la limite de temps, j'ai créé des éléments visuels dans le but d'obtenir une simulation crédible de l'environnement. Ainsi, j'ai créé une animation de mouvement sur l'eau pour le voilier et utilisé un matériel du package « HolyShift » pour recréer l'effet des vagues qui « caressent » les objets.

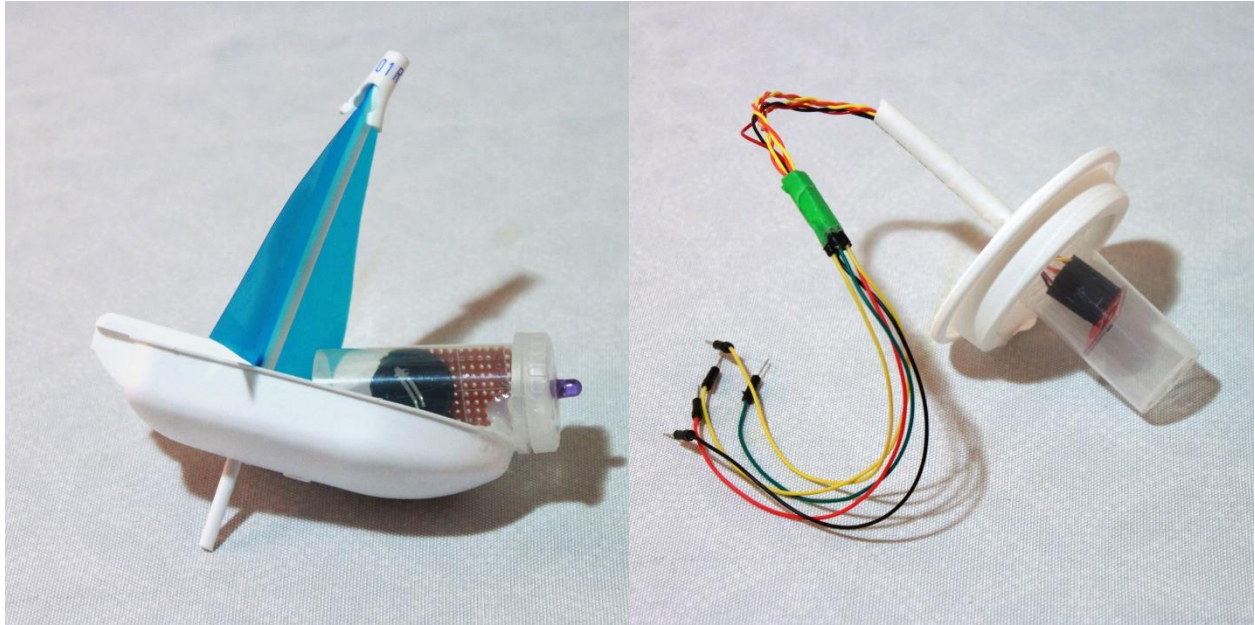
Je n'ai pas créé de condition de victoire à proprement parlé, le but étant d'éviter le plus de vagues pendant le plus de temps possible. La condition de défaite est atteinte lorsque le joueur prend trois dommages. Dans ce cas, un changement de scène s'effectue et affiche la phrase « game over ».

12



Circuit :





À gauche (le bateau), l'émetteur infrarouge a été soudé à une batterie. À droite (la bouée), un module accéléromètre analogique a été relié à des câbles. Ceux-ci sont directement connectés à l'arduino.

Fonctionnement du code :

Pour ce projet, j'utilise les capteurs IR pour déterminer la distance entre l'émetteur infrarouge (le bateau) et eux-mêmes. Ainsi, selon la distance ou l'intensité du rayon infrarouge, les capteurs IR sont en mesure de capter le courant. Plus l'émetteur est proche d'un capteur, plus le courant capté est important. Cela me permet de déterminer la distance approximative entre les deux éléments en créant des intervalles et en les associant à des déplacements/ distances dans l'engin de jeu Unity. Étant donné que Unity ne différencie pas les données perçues à travers le Serial Monitor, j'ai associé chaque intervalle à des lettres; De cette manière chaque capteur possède trois ensembles d'intervalles identifiés par des lettres de l'alphabet. Par exemple, lorsque la puissance du courant perçu par l'émetteur Y0 est entre 60 et 70, celui-ci va retourner la lettre « a ». Dans Unity, la lettre « a » a une coordonnée Vector3(0, -0.17f, 0). Bien que cela n'est pas efficace, c'est la manière que j'ai choisie pour communiquer entre les deux plateformes. La même technique a été utilisée pour obtenir des valeurs du module accéléromètre. Dès que l'accélération dépasse la puissance de

courant de 355, celui-ci retourne la lettre « z » pour l'axe X et la lettre « y » pour l'axe Y. Lorsqu'une de ces deux lettres sont captées par Unity, la fonction Vortex1 et/ ou Vortex2 est déclenchée. Celles-ci créent des instances de vagues réparties sur quatre points de la rose des vents. Cela sert à mettre au défi les joueurs en modulant leur souffle.