Arduino

1. Pourquoi Arduino ?
2. Relation Software-Hardware
3. La partie programmation

Figure -Logo Arduino

1. Grove
2. Amélioration : La création d’un parseur

*Pourquoi Arduino ?*

L’utilisation de la technologie Arduino était une nécessité. Afin de familiariser l’utilisateur avec les concepts initiaux de la programmation de composants électroniques, nous étions à la recherche d’un système adaptable et ludique. En tenant compte de ces attentes, nous avons entamé une recherche de périphériques facilement programmables et compatibles avec la carte Genuino Uno que nous allions utiliser pour le projet.

Ce choix était, avant tout, une décision commune. Durant notre parcours pré-universitaire, nous avons tous côtoyé à un moment ou un autre le monde merveilleux d’Arduino. Le projet tutoré du semestre 1 était pour la plupart des membres du groupe une première approche au sein de l’IUT à cet environnement électronique que nous voulions absolument découvrir en profondeur.

Jérémy et Valentin avaient déjà travaillé avec cette technologie sur des projets pré-baccalauréat et avaient par conséquent une première expérience et certains réflexes de programmation qui se sont avérés forts utiles lors de la phase de développement de la partie électronique. Pour toutes ces raisons, Arduino nous semblait un choix judicieux et plus particulièrement une carte : Genuino 101. Elle a pour avantage d’intégrer nativement un gyroscope mais aussi une possibilité de connexion Bluetooth. Pour son côté pratique et son aspect « plug-and-play », ce microcontroleur s’est imposé comme une évidence.

*Relation Software-Hardware*

Nous étions guidés par cet objectif de transmettre la connaissance nécessaire au joueur afin de s’amuser comme nous l’avons fait avec ces différents outils Arduino. Nous n’avions pas pour ambition de le porter à un haut niveau mais de lui permettre d’acquérir un niveau débutant suffisant pour commencer à tester les différents composants.

Pour parvenir à cet objectif-là, notre plan d’action était le suivant :

* Donner la possibilité de lire des cours
* Appliquer l’apprentissage à travers des questions en jeu
* Les bonnes réponses permettront de débloquer des parties du jeu

La finalité étant de pouvoir demander à l’utilisateur de saisir un petit programme qui permettrait par exemple d’allumer et éteindre une LED.

A travers la démonstration, nous avons permis un premier contact entre ces deux univers. Effectivement, le joueur doit pouvoir répondre à une question portant sur un programme Arduino afin de débloquer l’inversion de gravité et lui permettre de terminer le premier niveau.

Notre second objectif était de développer l’expérience en jeu pour surprendre l’utilisateur et lui montrer que les possibilités apportées par ce microcontroleur sont infinies. Nous avons alors mis en place une liaison en continue avec les périphériques Arduino qui permettent de :

* Tenir informé de la vie de Genuini1
* Afficher sur l’écran LED l’écran actif
* Permettre une interaction immédiate entre le monde physique et logiciel

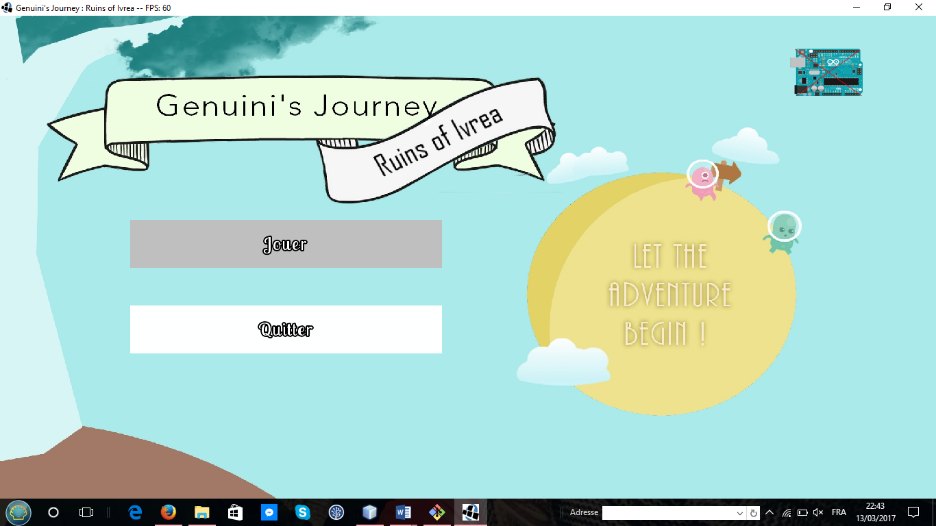
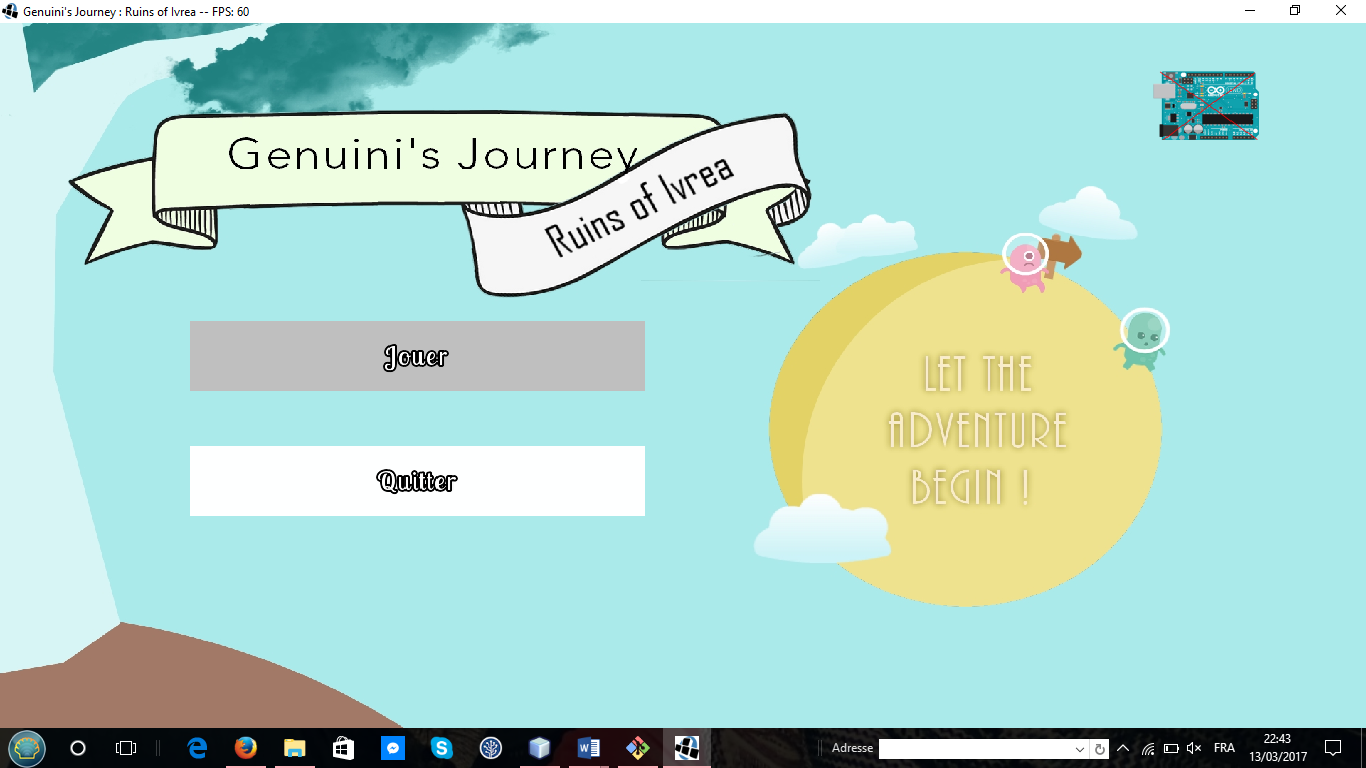
Le jeu devait pouvoir être également utilisable sans connexion entre les composants et l’ordinateur, nous avons alors implémenté la détection des périphériques et les actions qui seront effectuées en conséquence.

Figure -Liaison Arduino-Jeu non détectée

*Partie programmation*

Comme expliqué précédemment, l’environnement Arduino était un univers à approfondir pour certains et à découvrir pour d’autres. L’objectif premier était une collaboration sur toutes les parties du projet par tous les membres du groupe. Le développement Hardware n’a donc pas dérogé à la règle et a interessé de par son aspect ludique toutes les parties.

Le langage de programmation utilisé dans l’univers Arduino est le C/C++. Ayant déjà suivi des cours sur ce langage, la manière de développer n’a donc pas été une surprise pour nous.

Pour une réalisation optimale, nous avons travaillé sur l’IDE du constructeur nommé Arduino IDE. Son apparence ne diffère guère de nombreux autres environnements de développement, il a pour avantage de mettre en avant une interface claire et épuré permettant une compréhension rapide des différentes fonctionnalités du logiciel.

Son principal inconvénient reste la non présence de coloration syntaxique. A ce problème, une solution existe. Il est possible de coder sur un environnement différent appelé SublimeText qui lui prend en compte de nombreuses aides aux développeurs. Un plugin présent dans sa librairie de contenu additionnel permet à l’utilisateur de compiler et téléverser son programme par l’intermédiaire de ce logiciel à son microcontroleur.

<http://eskimon.fr/2224-arduino-mini-tuto-utiliser-sublime-text-ide>

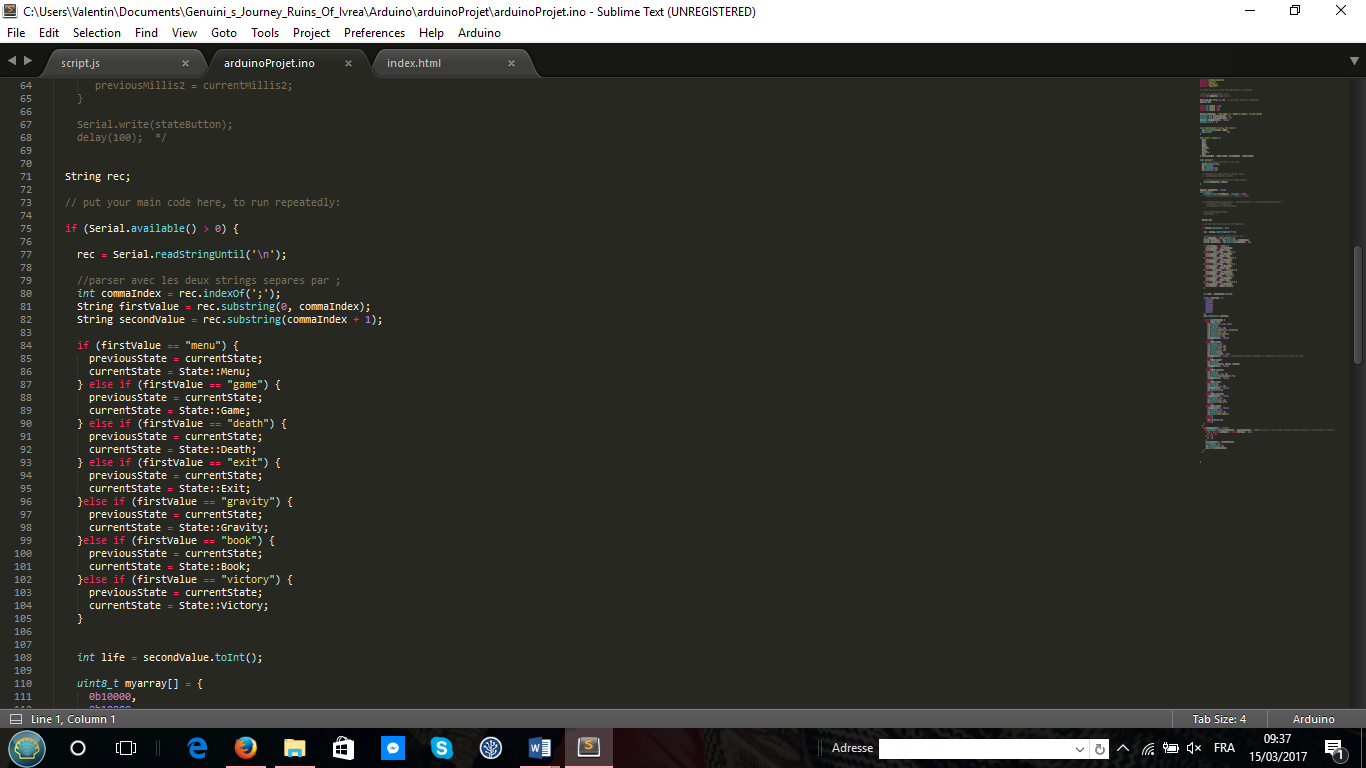
Le point essentiel du développement était la réussite de l’interfacage entre notre code Arduino et le jeu exécuté sur un ordinateur. Il s’est alors posé la question de la manière d’obtenir un résultat satisfaisant entre ces deux appareils.

Un port permettant la liaison entre le microcontroleur et le PC existe, il se nomme « Port COM ». La connexion série permet une relation d’entrée-sortie entre les deux appareils aux extrémités de la liaison. C’est une norme disponible sur tous les périphériques actuels et est utilisable via les connectiques USB. Une machine comporte plusieurs ports COM mais en attribue un seul pour le périphérique désiré.

*« Notre programme devait pouvoir faire face à tous types de situations. »*

Une de ces difficultés était la détection de la connexion entre le module Arduino et l’ordinateur. Pour parer ce problème, nous avons donc parcouru un tableau contenant tous les ports COM de la machine et avons vérifié si le microcontroleur était présent. La suite allait de soi, si le module est connecté nous le préparons pour recevoir et envoyer des données sinon le jeu informe l’utilisateur de la non détection du périphérique.

Par conséquent, afin de transmettre des informations de vie du personnage ou bien d’écran actif, nous utilisons une méthode : « InstanceDeLArduino.write(« infos ») » qui envoie sur le port série une séquence de bits correspondant à l’action que le microcontroleur va devoir effectuer. Nous devions rendre possible l’envoi de plusieurs informations en simultanée à l’Arduino. Malheureusement, la liaison série ne peut contenir qu’une seule suite de données, une chaîne de caractères. Il a donc fallu diviser notre chaîne à l’aide de point virgule afin que le programme côté microcontroleur puisse l’interpréter et effectuer les actions en conséquence.

Du côté Hardware, une énumération des écrans ainsi qu’une liste des différentes instructions permettent au code Arduino d’entretenir un contact réel avec le gameplay. Pour traiter les informations reçues, nous avons séparé la chaîne de caractère afin de pouvoir par exemple récupérer la valeur de la vie du personnage et ensuite l’écran actif du jeu.

Afin d’illustrer notre propos, soumettons un exemple concret qui est l’affichage de l’écran de mort. Une fois la vie valant 0 la partie logicielle s’occupe, de changer d’écran en indiquant au joueur la fin de la partie mais également, de lancer une musique particulière. A ce moment précis, un petit écran LED relié à la carte Arduino inscrit un message de mort et change de couleur pour passer en rouge afin d’augmenter le côté dramatique de la situation.

Finalement, de la même manière que la programmation JAVA, le développement Arduino a constitué une part importante dans le processus de réalisation du projet. Nous avons dû nous coordonner afin de parvenir à effectuer toutes les actions voulues.

Pour conclure, le projet à travers un univers Arduino s’est avéré être une excellente idée et ce microcontroleur a su convaincre tous les membres du groupe.

*Grove*

Afin de rendre les branchements électroniques plus faciles pour l’utilisateur, nous cherchions des composants additionnels compatibles avec notre solution Arduino et obéissant à ce principe que nous avions fixé : « plug-and-play ». Après de multiples recherches, nous avons découvert l’univers Grove. Il s’agit de modules offrant la possibilité d’accroitre les capacités de la carte en y ajoutant pour l’exemple des capteurs ou bien des écrans. L’avantage de ces outils est la librairie prête à l’utilisation fournie par l’entreprise SeedStudio comportant toutes les fonctions nécessaires à l’exploitation complète de ces modules.

<http://wiki.seeed.cc/Grove_System/>

Afin de proposer au joueur une multitude d’interaction avec le milieu Hardware, nous avons opté pour un pack comprenant les capteurs que nous souhaitions puis nous avons fait l’acquisition également d’une bargraphe LED pour permettre un affichage sur une Interface Homme-Machine de la vie du personnage.



Voici une liste exhaustive des différents modules présents dans ce pack :

* Base shield : A connecter à l’Arduino pour rendre disponible Grove
* LCD RGB BlackLight : Ecran Led pour l’écriture de textes
* Sound sensor : Capte le son ambiant
* Touch sensor : Petite interface tactile
* Temperature sensor : Retranscrit la température extèrieure à l’Arduino
* LED : Une Led de chaque couleur
* Button : Un bouton poussoir pour affecter des actions spécifiques

<http://wiki.seeed.cc/Grove_Starter_kit_for_Arduino_101/>

*Finalement, Grove offre de nombreuses possibilités d’interactions ludiques et immédiates entre les programmes Hardware et Software.*

*Amélioration : La création d’un parseur*

Nous avions un idéal dans ce projet, mettre en œuvre un parseur du côté Arduino.

Parseur : « Parser » signifie effectuer une action spécifique adéquate à partir d’un texte reçu dans un langage pré-établi.

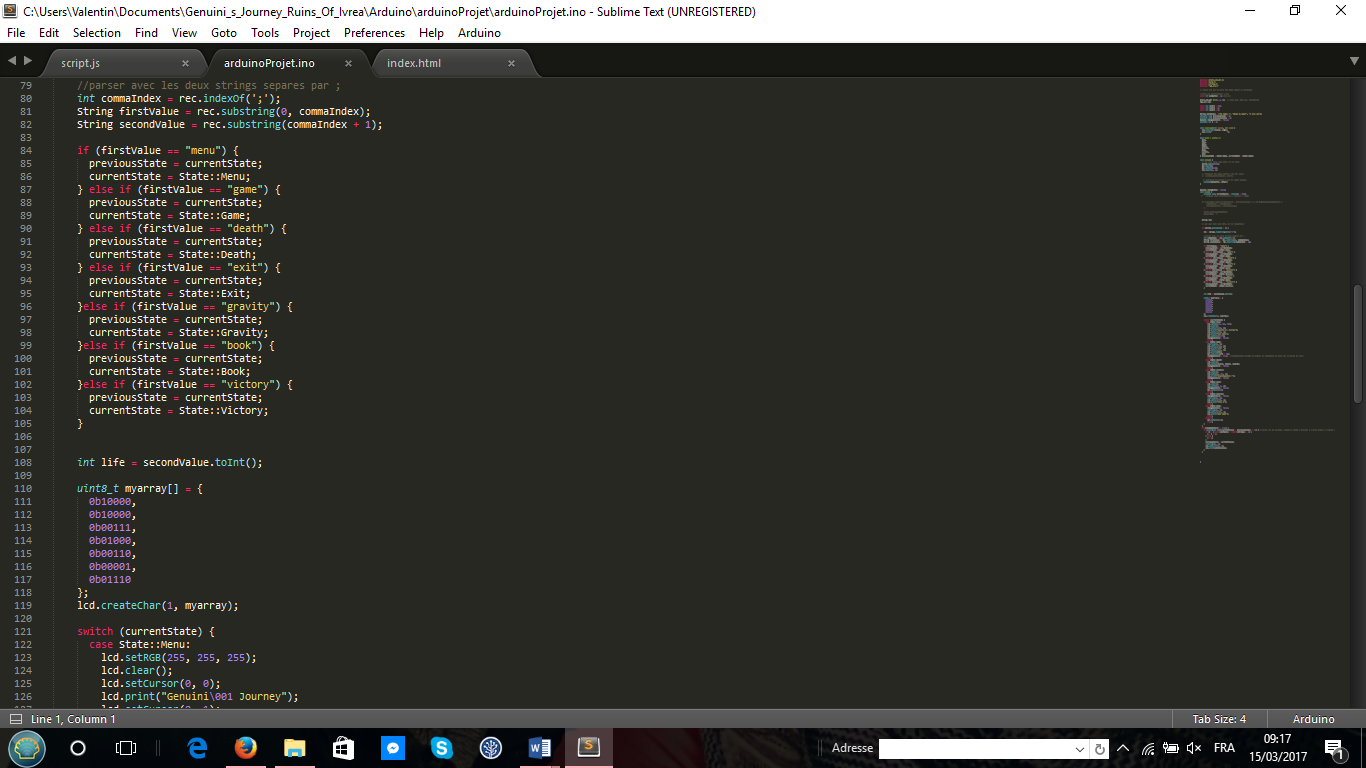
Par la suite, dans la continuité de notre développement, nous mettrons en place dans notre programme Arduino un système permettant la compréhension du langage C/C++ écrit dans le jeu par le microcontroleur. Afin de créer les premières activités d’apprentissage de l’utilisateur, nous avons commencé par vérifier si une ligne saisie par le joueur correspond à la ligne attendue pour compléter le niveau. De cette manière, notre réalisation est en adéquation avec le cahier des charges mais avant tout avec l’univers que nous voulions créer.

Figure -Conditions liées à la réception des données

*Afin de conclure pour la partie Arduino, nous pouvons tous affirmer avoir apprécié la conception, la programmation mais également l’interaction que nous a procuré cette liaison entre une partie logicielle et une partie Hardware. Dans un futur même proche, nous aimerions travailler dans un secteur proche de celui-ci. Et si jamais cela ne s’avérait pas être le cas, nous expérimenterons de nouveau ce lien à travers des projets personnels qui permettront de maitriser de plus en plus le monde Arduino.*

*De plus, nous aimerions poursuivre ce jeu par la suite après la soutenance.*

*Bilan personnel Valentin*

Lors de la première réunion, nous cherchions une idée commune, une ambition qui pourrait nous apporter l’envie de programmer en continue durant une année. Nous nous étions tous retrouvés sur la possibilité de jouer et d’apprendre dans un même temps. N’y a-t-il pas de meilleur moyen d’assimiler de nouvelles notions ?

Depuis mes débuts dans le milieu de l’informatique, la programmation de composants électroniques s’est imposée à moi, et plus particulièrement les systèmes Arduino ainsi que leurs dérivés. Notre but était de faire ressentir aux utilisateurs ces sensations que nous-mêmes avons éprouvées et éprouvons encore.

Je retire de cette réalisation de projet un sentiment positif. Dès la phase de conception, nous étions tous les quatres motivés pour parvenir à arriver au bout de cet objectif que nous nous étions fixé. Dans toute la continuité du projet, nous nous sommes heurtés à de nombreux problèmes (Travis, Bug de librairie…etc). J’ai pû tirer des enseignements de tous ces soucis et je saurai désormais réagir efficacement à ce type de situation dans un contexte professionnel.

Le semestre 3 a été, pour nous, le début de la phase de développement. J’avais hâte de programmer dans le langage JAVA. J’ai pour objectif dans un futur proche d’en faire mon langage de programmation quotidien, j’apprécie son côté haut niveau et le développement orienté objet. Son utilisation dans ce projet a donc été une excellente nouvelle d’un point de vue personnel.

Il est maintenant l’heure des conclusions. En ce qui me concerne, j’éprouve certains regrets, j’ai plutôt eu l’habitude de coder dans la précipitation. Sans paraitre arrogant et en tenant compte de l’avis de mes collègues, il me semble avoir bien avancé dans mes parties tout de même. Si je devais revenir en arrière, je programmerais dans la durée en étalant le travail. Mais, est-ce que cela aurait été possible ? Nous ne le serons pas. Toutefois, je suis fier de ce que j’ai réussi à accomplir et de ce que NOUS avons réussi à réaliser.

Pour conclure, cette aventure a été une réussite collective bien plus que personnelle. J’ai adoré travailler, coder, jouer aux côtés de mes trois compagnons et pouvoir communiquer avec notre tutrice.

Ce projet ne pouvait pas mieux correspondre à ce que j’espérais d’un travail en commun et j’ai énormément appris.

Je remercie donc mes collègues ainsi que Mme Cordier mais aussi toutes les personnes qui nous ont permis de mener à bien ce projet.