

Toi Plus

Nicolas Marier, Guillaume Rousseau, Thomas Garneau-Hamel, Charles-Antoine Demetriade
420-204-RE, gr.1

Dossier de conception

Silence radio

Travail présenté à
M. Walid Boulabiar

Département d'informatique
Cégep Limoilou
Le 6 février 2019

Table des matières

1	Description du projet	2
2	Prototypes de l'application	3
3	Backlogs	5
3.1	Backlog de produit	5
3.2	Backlog de sprint #1	9
3.3	Backlog de sprint #2	12
3.4	Backlog de sprint #3	16
3.5	Backlog de sprint #4	23
4	Diagramme de classes	25
5	Échéancier	25

1 Description du projet

Objectifs : Le besoin auquel notre projet vise à répondre est la transmission de données sans fil. Pour répondre au besoin, il faut que la transmission soit sécuritaire et privée. De plus, il faut être en mesure de détecter et corriger les erreurs de transmission d'information, afin d'éviter la corruption de l'information. Notre projet répond également au besoin grandissant de comprendre en détail le fonctionnement d'Internet et les principes fondamentaux physiques et mathématiques qui permettent l'utilisation d'un tel réseau.

Description détaillée : Au plus haut niveau, l'application sera utilisée de la façon suivante : l'utilisateur commence par sélectionner un fichier de son choix, de n'importe quel type et de n'importe quel format, qu'il veut envoyer à un autre ordinateur. Ensuite, le logiciel se charge de convertir le fichier de l'utilisateur dans un format binaire sérialisé. Le fichier, sous forme binaire, est converti à nouveau sous un format analogue qui sera un son pour émettre du courant alternatif dans notre circuit. Le son sera sélectionné très précisément, car il est responsable de la modulation de la fréquence de notre onde ainsi que de l'amplitude de celle-ci. L'utilisation du son à cet effet permettra à l'utilisateur de choisir lui-même la fréquence sur laquelle il désire transmettre l'information. Le son sera transformé en onde électromagnétique par notre circuit RLC et reçu par notre autre circuit de réception. Une fois reçu par le récepteur, connecté à un second ordinateur, les ondes électromagnétiques seront reconverties en binaire, pour ultimement recréer l'information d'origine. S'il advenait qu'une erreur de transmission se produise, elle serait corrigée, si notre système de correction d'erreurs le permet. Sinon, l'ordinateur receveur avertira l'utilisateur de la corruption de son information et l'invitera à réinitialiser le transfert

Concepts :

- Circuits électriques
 - Calculs pour déterminer les composantes
 - Circuits RLC
 - Courant alternatif
- Ondes électromagnétiques
 - Fréquence
 - Amplitude
 - Ondes AM
- Correction et détection d'erreurs
- Sérialisation d'objets en blob (**b**inary **l**arge **o**bject)
- Conversion digital to analog et analog to digital
- Interprétation du flux binaire reçu pour reconstruire l'information

Nouveaux concepts : Tout ce qui a trait aux ondes est nouveau pour nous étant donné que nous sommes en train de compléter le cours d'ondes et physique moderne. Malgré le fait que nous avons fait des circuits dans le cours de physique précédent, ils étaient théoriques. Nous n'avons que très rarement assemblé des circuits, et nous ne les avons jamais élaborés nous-mêmes. La conversion digital to analog nécessaire à l'envoi d'ondes est quelque chose que nous n'avons jamais fait. Aussi, nous n'avons jamais transmis d'ondes entre deux circuits. Il nous faudra également apprendre les principes de correction d'erreurs pour détecter et corriger les erreurs.

Complexité : La complexité provient surtout du fait que nous avons beaucoup de choses à apprendre. Il est difficile d'évaluer correctement la complexité du projet étant donné notre inexpérience dans les nouveaux concepts. Cependant, nous pouvons dire avec certitude que la tâche ne sera pas si facile, car la contrainte de temps est considérable. De plus, les nombreuses tâches du projet vont nécessiter une attention particulière, étant donné que le succès de chaque étape est obligatoire pour l'élaboration de l'étape suivante. Nous sommes convaincus de la possibilité de réaliser le projet, car il est possible de retirer certains facteurs de complexité s'il s'avère que notre entreprise était initialement trop ambitieuse. Notre professeur de physique nous a confirmé qu'il allait pouvoir nous aider en cas de besoin, et qu'il est certain que la réalisation du projet est possible.

Concepts, rôles et justifications :

- Nicolas Marier, chef et développeur
- Thomas Garneau-Hamel, électronique, physicien et motivateur
- Guillaume Rousseau, documentation et physicien
- Charles-Antoine Demetriade, graphiques/interfaces

Nous avons choisi ces rôles en fonction de nos forces et nos goûts ainsi que nos faiblesses.

Séparation des tâches :

- Nicolas Marier : élaboration de la programmation des classes métier ainsi que des diagrammes UML.
- Thomas Garneau-Hamel : conception du circuit électrique et calculs.
- Guillaume Rousseau : documentation, recherche, contrôle de qualité.
- Charles-Antoine Demetriade : création de la vue et de ce qui se rapporte à l'environnement graphique.

Type d'application : Windows, Mac, Linux (exécutable par la JVM)

Langages utilisés, patrons de conception :

- Language : Java
- Patrons de conception : MVC

Technologies impliquées :

- IDE : Eclipse, IntelliJ, SceneBuilder, PlantUML
- Hardware : un ordinateur et un raspberry pi, des composantes électroniques (résistances, condensateurs, fils)

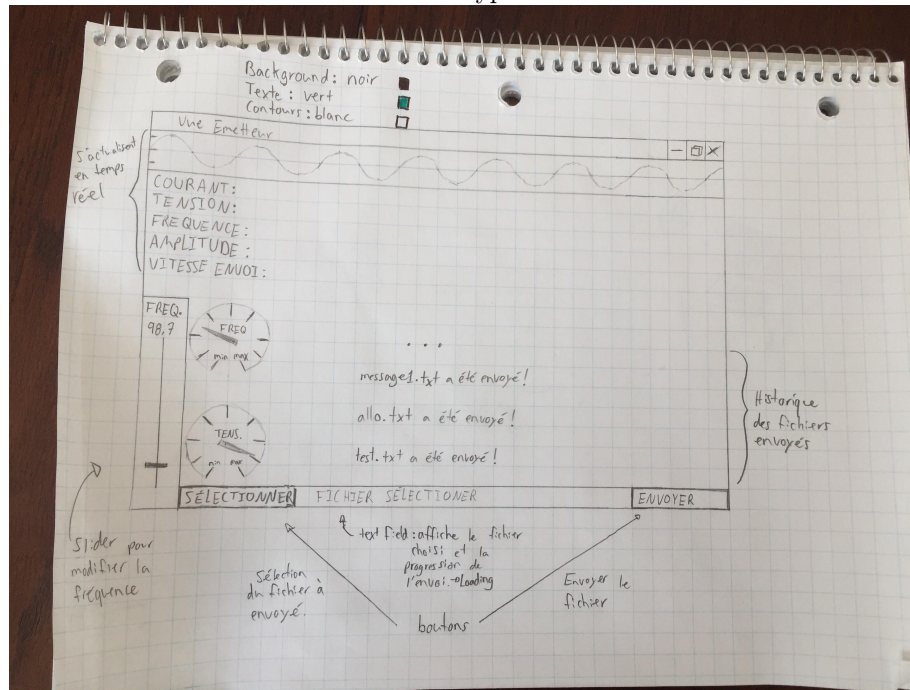
Références et documentation :

- <https://electronics.howstuffworks.com/radio.htm>
- <http://www.ta-formation.com/acrobat/am.pdf>
- <http://ekldata.com/I91iUA5sMDI3QLRiLD82FmviNF4/TX-AM.pdf>
- <http://www.cea.fr/comprendre/Pages/physique-chimie/essentiel-sur-ondes-electromagnetiques-communication.aspx>

2 Prototypes de l'application

Prototype : Nous aimerions que notre interface ait un thème de cockpit d'avion. Donc, les couleurs choisies ainsi que la police de caractère que nous sélectionnerons refléteront cette esthétique.

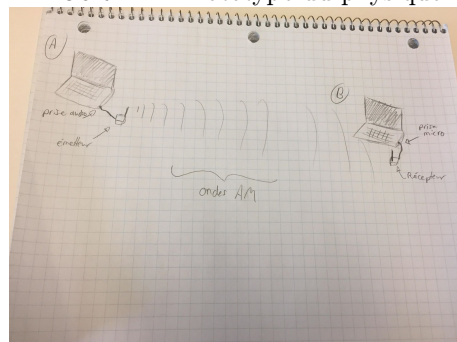
FIGURE 1 – Prototype de l'interface



Principaux objets :

- Visualisation des ondes envoyées
- Valeurs pour les différents paramètres
 - Courant dans le circuit
 - Tension sur le circuit
 - Fréquence sur laquelle on émet
 - Amplitude des ondes
 - Vitesse d'envoi (B/sec ou KB/sec)
- La liste des fichiers envoyés
- Des boutons pour choisir un fichier et l'envoyer
- Un contrôle pour modifier la fréquence
- Des visualisations pour la fréquence et la tension relative (pour savoir si on s'approche du maximum permis)
- Potentiellement à ajouter serait un indicateur de puissance utilisée par le circuit

FIGURE 2 – Prototype du physique



3 Backlogs

3.1 Backlog de produit

1	
Acteur ou rôle	Utilisateur
Scénario ou story	En tant qu'utilisateur, je veux pouvoir sélectionner un fichier afin de pouvoir l'envoyer.
Détail et description	Avoir accès à son explorateur de fichier dans le but de pouvoir sélectionner n'importe lequel fichier disponible dans les archives
Tests d'acceptation	Sélectionner un fichier et vérifier qu'il s'affiche correctement.
Complexité	1
Effort	1
Commentaires	
2	
Acteur ou rôle	Utilisateur
Scénario ou story	En tant qu'utilisateur, je souhaite que le logiciel soit capable de détecter si le transfert de fichiers s'est bien complété et que ce dernier puisse corriger l'information reçue si cette dernière comprend des erreurs.
Détail et description	En cas de corruption lors du transfert de données, le logiciel détecte l'emplacement du problème et offre à l'utilisateur de corriger les erreurs rencontrées afin d'avoir des données complètes.
Tests d'acceptation	Des bits de confirmation seront ajoutés à l'information. Il y aura un système de vérification inspiré du International Standard Book Number
Complexité	1
Effort	1
Commentaires	
3	
Acteur ou rôle	Programmeur
Scénario ou story	En tant que programmeur, je dois pouvoir convertir l'information contenue dans le fichier sélectionné en code binaire afin que ces dernières soient plus faciles à transmettre par ondes radio.
Détail et description	Pouvoir prendre n'importe quel type de fichier et le convertir en binaire, soit en suite de 1 et de 0, dans le but de faciliter la transmission par ondes radio puisque le binaire est beaucoup plus simple à représenter par des ondes.
Tests d'acceptation	Comparaison de la conversion en binaire avec un fragment du fichier connu.
Complexité	1
Effort	2
Commentaires	
4	
Acteur ou rôle	Développeur
Scénario ou story	En tant que développeur, je veux que le code binaire soit converti en son afin de permettre à l'information d'être envoyée par l'antenne de transmission.
Détail et description	Avec le son nouvellement transformé, il sera alors possible de créer un courant alternatif qui permettra par la suite de créer plus facilement des ondes.
Tests d'acceptation	Tests sonores ainsi que des tests à l'aide d'un multimètre.
Complexité	2
Effort	3
Commentaires	
5	
Acteur ou rôle	Physicien
Scénario ou story	En tant que physicien, je veux posséder un circuit émetteur afin de pouvoir envoyer les ondes radio contenant l'information du fichier sélectionné.
Détail et description	Un circuit électrique de courant alternatif composé de résistance, de condensateurs et de bobines inductives.
Tests d'acceptation	Tests techniques avec un multimètre et des tests pour les ondes avec une radio

Suite à la page suivante

Complexité	3
Effort	6
Commentaires	
6	
Acteur ou rôle	Utilisateur
Scénario ou story	En tant qu'utilisateur, je veux posséder un circuit récepteur afin de pouvoir capter les ondes radio produites par l'émetteur.
Détail et description	Un circuit électrique de courant alternatif composé de résistance, de condensateurs et de bobines inductives.
Tests d'acceptation	Tests techniques avec un multimètre
Complexité	3
Effort	6
Commentaires	
7	
Acteur ou rôle	Développeur
Scénario ou story	En tant que développeur, les courants électriques créés par la réception d'ondes radio doivent m'être convertis en binaire afin que ces derniers puissent être interprétés par l'ordinateur.
Détail et description	Selon les caractéristiques des ondes reçues, les reconvertir en binaire.
Tests d'acceptation	Test de conversion avec des valeurs connues prédéterminées
Complexité	3
Effort	4
Commentaires	
8	
Acteur ou rôle	Programmeur
Scénario ou story	En tant que programmeur, je dois être en mesure de reconstruire le fichier envoyé par l'émetteur à partir du code binaire reçu.
Détail et description	Effectuer la reconstruction du fichier d'origine en utilisant seulement ce qui a été reçu par l'antenne de réception
Tests d'acceptation	Tests avec des fichiers connus
Complexité	1
Effort	1
Commentaires	
9	
Acteur ou rôle	Utilisateur
Scénario ou story	En tant qu'utilisateur, je souhaite pouvoir avoir une représentation visuelle du progrès du transfert des fichiers afin de me permettre de savoir quand ce dernier est fini et afin d'ajouter un élément esthétique à l'interface.
Détail et description	Avoir accès visuellement à la progression de l'envoi du fichier à l'aide d'une barre de progrès
Tests d'acceptation	Tests avec des fichiers de grosseur connues
Complexité	1
Effort	2
Commentaires	
10	
Acteur ou rôle	Utilisateur
Scénario ou story	En tant qu'utilisateur, je dois pouvoir sélectionner la fréquence radio sur laquelle je souhaite faire mon transfert d'information afin de pouvoir choisir celle qui possède le moins d'interférences pouvant affecter négativement le transfert d'informations.
Détail et description	Être en mesure de choisir la fréquence désirée, celle avec le moins d'interférence, dans le but d'assurer un meilleur transfert de données
Tests d'acceptation	Tests avec une radio
Complexité	1
Effort	1
Commentaires	

Suite à la page suivante

11	
Acteur ou rôle	Utilisateur
Scénario ou story	En tant qu'utilisateur, je dois avoir une interface visuelle de l'émetteur afin de pouvoir visualiser et sélectionner le fichier.
Détail et description	Interface graphique suivant la thématique d'un cockpit, c'est-à-dire une interface comportant toutes les données du circuit et du transfert de données qui sont importantes.
Tests d'acceptation	Tests visuels
Complexité	2
Effort	3
Commentaires	
12	
Acteur ou rôle	Utilisateur
Scénario ou story	En tant qu'utilisateur, je dois avoir une interface dédiée à la réception des fichiers en transfert afin de pouvoir mieux gérer l'enregistrement de ces derniers.
Détail et description	Interface comportant l'historique des réceptions de fichiers ainsi qu'un moyen intuitif pour enregistrer ce qui a été reçu.
Tests d'acceptation	Tests visuels
Complexité	1
Effort	2
Commentaires	
13	
Acteur ou rôle	Utilisateur
Scénario ou story	En tant qu'utilisateur, je dois être en mesure de sélectionner où est enregistré le fichier que je reçois afin d'éviter de le perdre ou qu'il ne s'enregistre nulle part.
Détail et description	Avoir accès à l'explorateur de fichier dans le but de pouvoir enregistrer de manière simple et efficace.
Tests d'acceptation	Tests unitaires
Complexité	1
Effort	1
Commentaires	
14	
Acteur ou rôle	Administrateur
Scénario ou story	En tant qu'administrateur, je souhaite que la fréquence radio de l'émetteur se choisisse automatiquement afin qu'elle choisisse elle-même la meilleure fréquence radio pour faire le transfert.
Détail et description	Le logiciel de l'émetteur soit en mesure de trouver la meilleure fréquence par lui-même afin de choisir automatique cette fréquence dans le but d'effectuer le meilleur transfert de données possibles.
Tests d'acceptation	Tests pratiques et comparaison avec la fréquence connue
Complexité	3
Effort	6
Commentaires	
15	
Acteur ou rôle	Utilisateur
Scénario ou story	En tant qu'utilisateur, je dois être en mesure de décider laquelle des interfaces visuelles je souhaite utiliser afin de me permettre d'utiliser le logiciel adéquatement.
Détail et description	En démarrant l'application, il y a l'option de sélectionner le mode émetteur ou bien le mode de récepteur.
Tests d'acceptation	Tests visuels
Complexité	1
Effort	1
Commentaires	
16	
Acteur ou rôle	Administrateur

Suite à la page suivante

Scénario ou story	En tant qu'administrateur, je souhaite sécuriser la communication afin de rendre celle-ci privée.
Détail et description	Rendre la communication plus sécurisée en cryptant les données à envoyer afin d'éviter les vols d'informations
Tests d'acceptation	Tests visuels
Complexité	4
Effort	8
Commentaires	

3.2 Backlog de sprint #1

1	
Acteur ou rôle	Utilisateur
Scénario ou story	En tant qu'utilisateur, je veux pouvoir sélectionner un fichier afin de pouvoir l'envoyer.
Détail et description	<ol style="list-style-type: none"> 1. Permettre à l'utilisateur d'accéder à sa bibliothèque de fichiers. <ol style="list-style-type: none"> 1.1. Qui et temps : <ol style="list-style-type: none"> 1.1.1. (NM) 1.1.2. (30 minutes) 1.2. Préconditions : <ol style="list-style-type: none"> 1.2.1. La présence d'un bouton qui permet à l'utilisateur de procéder à l'action de sélectionner un fichier. 1.3. Règles d'affaires : <ol style="list-style-type: none"> 1.3.1. Doit sélectionner un fichier existant dans l'espace de donné de l'ordinateur utilisé. 1.4. Règles d'affaires alternatives : <ol style="list-style-type: none"> 1.4.1. S'il ne sélectionne pas un fichier existant ou que le fichier sélectionné n'existe plus une fois sélectionné, une alerte est envoyé à l'utilisateur afin qu'il en sélectionne un nouveau. 1.5. Tests d'acceptation de cet item : <ol style="list-style-type: none"> 1.5.1. Tester si le fichier sélectionné est existant dans la base de donnée de l'ordinateur et accessible par le logiciel. 1.5.2. Vérifier que le contenu du fichier est lisible par le logiciel. 1.6. Post-conditions : <ol style="list-style-type: none"> 1.6.1. L'utilisateur peut sélectionner le fichier qu'il souhaite émettre. 2. Présenter à l'utilisateur les informations du fichier qu'il s'appête à envoyer. <ol style="list-style-type: none"> 2.1. Qui et temps : <ol style="list-style-type: none"> 2.1.1. (NM) 2.1.2. (30 minutes) 2.2. Préconditions : <ol style="list-style-type: none"> 2.2.1. Le fichier doit être sélectionné. 2.3. Règles d'affaires : <ol style="list-style-type: none"> 2.3.1. Doit pouvoir envoyer l'information à la vue du nom du fichier. 2.4. Règles d'affaires alternatives : <ol style="list-style-type: none"> 2.4.1. Si le nom comportes des caractères non interprétables par l'ordinateur, ce dernier ne les affichera pas. 2.5. Tests d'acceptation de cet item : <ol style="list-style-type: none"> 2.5.1. vérifier chacun des caractère du nom d'un fichier afin de pouvoir afficher tous ceux possibles. 2.6. Post-conditions : <ol style="list-style-type: none"> 2.6.1. L'utilisateur peut voir le nom de son fichier.
Tests d'acceptation	Sélectionner un fichier et vérifier qu'il s'affiche correctement.
Complexité	1
Effort	1
Commentaires	
2	

Suite à la page suivante

Acteur ou rôle	Utilisateur
Scénario ou story	En tant qu'utilisateur, je dois posséder une interface visuelle pour l'émission des fichiers.
Détail et description	<ol style="list-style-type: none"> 1. Construire une interface avec la disposition souhaitée et les fonctions de bases permettant d'envoyer un fichier. <ol style="list-style-type: none"> 1.1. Qui et temps : <ol style="list-style-type: none"> 1.1.1. (CAD) 1.1.2. (2h) 1.2. Préconditions : <ol style="list-style-type: none"> 1.2.1. Avoir préparé un croquis de l'interface. 1.2.2. Avoir le logiciel Scene Builder pour la conception de l'interface. 1.3. Règles d'affaires : <ol style="list-style-type: none"> 1.3.1. Afficher l'interface de l'émetteur. 1.4. Règles d'affaires alternatives : <ol style="list-style-type: none"> 1.4.1. Il n'y a pas d'alternative, car sans interface, l'utilisateur ne peut rien faire. 1.5. Tests d'acceptation de cet item : <ol style="list-style-type: none"> 1.5.1. Les tests seront au niveau visuel. S'il y a un problème d'affichage, on pourra le voir. 1.6. Post-conditions : <ol style="list-style-type: none"> 1.6.1. L'interface devra pouvoir afficher les éléments suivant : Boutons pour la sélection, envois de fichier. 2. Ajouter à l'interface de la couleur et du style. <ol style="list-style-type: none"> 2.1. Qui et temps : <ol style="list-style-type: none"> 2.1.1. (CAD) 2.1.2. (1h) 2.2. Préconditions : <ol style="list-style-type: none"> 2.2.1. Avoir terminé le fxml. 2.3. Règles d'affaires : <ol style="list-style-type: none"> 2.3.1. Styliser l'interface en css. 2.4. Règles d'affaires alternatives : <ol style="list-style-type: none"> 2.4.1. Il n'y a pas d'alternative. 2.5. Tests d'acceptation de cet item : <ol style="list-style-type: none"> 2.5.1. Les tests seront au niveau visuel. S'il y a un problème d'affichage, on pourra le voir. 2.6. Post-conditions : <ol style="list-style-type: none"> 2.6.1. L'interface devra être en noir, vert et blanc comme une vue de console.
Tests d'acceptation	Les tests seront au niveau visuel. S'il y a un problème d'affichage, on pourra le voir.
Complexité	1
Effort	1
Commentaires	
3	
Acteur ou rôle	Utilisateur
Scénario ou story	En tant qu'utilisateur, je veux posséder un circuit émetteur afin de pouvoir envoyer les ondes radio contenant l'information du fichier sélectionné.

Suite à la page suivante

Détail et description	<ol style="list-style-type: none"> 1. Faire les plans du circuit à produire. <ol style="list-style-type: none"> 1.1. Qui et temps : <ol style="list-style-type: none"> 1.1.1. (TG-H) et (GR) 1.1.2. (1h) 1.2. Règles d'affaires : <ol style="list-style-type: none"> 1.2.1. Doit être un circuit permettant au courant de circuler et de se rendre à l'antenne. 1.2.2. le circuit doit pouvoir être alimenté par une prise d'écouteurs d'ordinateur. 1.2.3. Doit comprendre des composantes permettant d'amplifier le courant transmis par la source d'énergie. 1.3. Tests d'acceptation de cet item : <ol style="list-style-type: none"> 1.3.1. Valider ce dernier en consultant des sources montrant le fonctionnement d'émetteurs radio. 1.4. Post-conditions : <ol style="list-style-type: none"> 1.4.1. Nous allons avoir un plan de conception et pourrons procéder à la prochaine étape. 2. Accumuler le matériel nécessaire à la conception du circuit électrique. <ol style="list-style-type: none"> 2.1. Qui et temps : <ol style="list-style-type: none"> 2.1.1. (TG-H) et (GR) 2.1.2. (30 minutes) 2.2. Préconditions : <ol style="list-style-type: none"> 2.2.1. Avoir un schéma de construction fini et la liste du matériel. 2.3. Règles d'affaires : <ol style="list-style-type: none"> 2.3.1. doit respecter le matériel déterminé dans le schéma de construction. 2.4. Tests d'acceptation de cet item : <ol style="list-style-type: none"> 2.4.1. Vérifier si nous possédons bien chaque pièce nécessaire à la complétion de l'émetteur. 2.5. Post-conditions : <ol style="list-style-type: none"> 2.5.1. Nous allons pouvoir procéder au montage du circuit émetteur. 3. Procéder à la conception du circuit de l'émetteur. <ol style="list-style-type: none"> 3.1. Qui et temps : <ol style="list-style-type: none"> 3.1.1. (TG-H) et (GR) 3.1.2. (4h) 3.2. Préconditions : <ol style="list-style-type: none"> 3.2.1. Avoir les pièces nécessaires à la conception en main. 3.2.2. Avoir le plan de conception achevé et vérifié. 3.3. Règles d'affaires : <ol style="list-style-type: none"> 3.3.1. Suivre le schéma de conception et construire le circuit. 3.4. Tests d'acceptation de cet item : <ol style="list-style-type: none"> 3.4.1. Vérifier que le circuit émet bel et bien des ondes radio. 3.4.2. Vérifier que l'information transmise est exacte. 3.5. Post-conditions : <ol style="list-style-type: none"> 3.5.1. L'utilisateur peut émettre des ondes radio à partir de son ordinateur.
Tests d'acceptation	Émettre des ondes et en vérifier l'exactitude et bon fonctionnement.

Suite à la page suivante

Complexité	3
Effort	6
Commentaires	
4	
Acteur ou rôle	Utilisateur
Scénario ou story	En tant que programmeur, je dois pouvoir convertir l'information contenue dans le fichier sélectionné en code binaire afin que ces dernières soient plus faciles à transporter par ondes radio.
Détail et description	<ol style="list-style-type: none"> 1. Construire une interface avec la disposition souhaitée et les fonctions de bases permettant de décider l'emplacement d'enregistrement du fichier. <ol style="list-style-type: none"> 1.1. Qui et temps : <ol style="list-style-type: none"> 1.1.1. (NM) 1.1.2. (3h) 1.2. Préconditions : <ol style="list-style-type: none"> 1.2.1. Il faut que la sélection de fichier soit fonctionnelle. 1.3. Règles d'affaires : <ol style="list-style-type: none"> 1.3.1. Lire le fichier et le convertir le fichier en binaire. 1.4. Tests d'acceptation de cet item : <ol style="list-style-type: none"> 1.4.1. Les tests seront de comparer la conversion en binaire avec un fragment du fichier connu. 1.5. Post-conditions : <ol style="list-style-type: none"> 1.5.1. La conversion devra être complètement fonctionnelle. 2. Présenter à l'utilisateur les informations du fichier qu'il s'apprête à envoyer. <ol style="list-style-type: none"> 2.1. Qui et temps : <ol style="list-style-type: none"> 2.1.1. (NM) 2.1.2. (1h) 2.2. Préconditions : <ol style="list-style-type: none"> 2.2.1. Pouvoir sélectionner un fichier. 2.3. Règles d'affaires : <ol style="list-style-type: none"> 2.3.1. Extraction de données de type String. 2.4. Règles d'affaires alternatives : <ol style="list-style-type: none"> 2.4.1. Prendra une String avec comme valeur : Nom inconnu. 2.5. Tests d'acceptation de cet item : <ol style="list-style-type: none"> 2.5.1. Les tests seront au niveau visuel, car nous pourrons voir si l'affichage s'effectue. 2.6. Post-conditions : <ol style="list-style-type: none"> 2.6.1. L'information du fichier sera présenté à l'écran.
Tests d'acceptation	Les tests seront de comparer la conversion en binaire avec un fragment du fichier connu.
Complexité	1
Effort	2
Commentaires	

3.3 Backlog de sprint #2

1	
Acteur ou rôle	Utilisateur

Suite à la page suivante

Scénario ou story	En tant qu'utilisateur, je veux posséder un circuit récepteur afin de pouvoir capter les ondes radio produites par l'émetteur.
Détail et description	<ol style="list-style-type: none"> 1. Faire les plans du circuit à produire. <ol style="list-style-type: none"> 1.1. Qui et temps : <ol style="list-style-type: none"> 1.1.1. (TG-H) et (GR) 1.1.2. (1h) 1.2. Règles d'affaires : <ol style="list-style-type: none"> 1.2.1. Doit être un circuit permettant à l'antenne de réception de transmettre le courant créé par les ondes radio de se rendre à l'ordinateur. 1.2.2. le circuit doit pouvoir être relié à l'ordinateur par la prise d'écouteurs. 1.3. Tests d'acceptation de cet item : <ol style="list-style-type: none"> 1.3.1. Valider ce dernier en consultant des sources montrant le fonctionnement de récepteurs d'ondes radio. 1.4. Post-conditions : <ol style="list-style-type: none"> 1.4.1. Nous allons avoir un plan de conception et pourrons procéder à la prochaine étape. 2. Accumuler le matériel nécessaire à la conception du circuit électrique. <ol style="list-style-type: none"> 2.1. Qui et temps : <ol style="list-style-type: none"> 2.1.1. (TG-H) et (GR) 2.1.2. (30 minutes) 2.2. Préconditions : <ol style="list-style-type: none"> 2.2.1. Avoir un schéma de construction fini et la liste du matériel. 2.3. Règles d'affaires : <ol style="list-style-type: none"> 2.3.1. doit respecter le matériel déterminé dans le schéma de construction. 2.4. Tests d'acceptation de cet item : <ol style="list-style-type: none"> 2.4.1. Vérifier si nous possédons bien chaque pièce nécessaire à la complétion du récepteur. 2.5. Post-conditions : <ol style="list-style-type: none"> 2.5.1. Nous allons pouvoir procéder au montage du circuit récepteur. 3. Procéder à la conception du circuit du récepteur. <ol style="list-style-type: none"> 3.1. Qui et temps : <ol style="list-style-type: none"> 3.1.1. (TG-H) et (GR) 3.1.2. (4h) 3.2. Préconditions : <ol style="list-style-type: none"> 3.2.1. Avoir les pièces nécessaires à la conception en main. 3.2.2. Avoir le plan de conception achevé et vérifié. 3.3. Règles d'affaires : <ol style="list-style-type: none"> 3.3.1. Suivre le schéma de conception et construire le circuit. 3.4. Tests d'acceptation de cet item : <ol style="list-style-type: none"> 3.4.1. Vérifier que le circuit reçoit bel et bien des ondes radio. 3.4.2. Vérifier que l'information transmise à l'ordinateur est bien la même que celle envoyée. 3.5. Post-conditions : <ol style="list-style-type: none"> 3.5.1. L'utilisateur peut capter des ondes radio et les enregistrer sur son ordinateur.

Suite à la page suivante

Tests d'acceptation	Essayer de capter des ondes et en vérifier l'exactitude.
Complexité	3
Effort	6
Commentaires	
2	
Acteur ou rôle	Utilisateur
Scénario ou story	En tant qu'utilisateur, je dois posséder une interface visuelle pour la réception des fichiers.
Détail et description	<ol style="list-style-type: none"> 1. Construire une interface avec la disposition souhaitée et les fonctions de bases permettant de décider l'emplacement d'enregistrement du fichier. <ol style="list-style-type: none"> 1.1. Qui et temps : <ol style="list-style-type: none"> 1.1.1. (CAD) 1.1.2. (2h) 1.2. Préconditions : <ol style="list-style-type: none"> 1.2.1. Avoir préparé un croquis de l'interface. 1.2.2. Avoir le logiciel Scene Builder pour la conception de l'interface. 1.3. Règles d'affaires : <ol style="list-style-type: none"> 1.3.1. Afficher l'interface du récepteur. 1.4. Règles d'affaires alternatives : <ol style="list-style-type: none"> 1.4.1. Il n'y a pas d'alternative, car sans interface, l'utilisateur ne peut rien faire. 1.5. Tests d'acceptation de cet item : <ol style="list-style-type: none"> 1.5.1. Les tests seront au niveau visuel. S'il y a un problème d'affichage, on pourra le voir. 1.6. Post-conditions : <ol style="list-style-type: none"> 1.6.1. L'interface devra pouvoir afficher les éléments suivant : Boutons pour la sélection et pour l'envoi de fichier. 2. Ajouter à l'interface de la couleur et du style. <ol style="list-style-type: none"> 2.1. Qui et temps : <ol style="list-style-type: none"> 2.1.1. (CAD) 2.1.2. (1h) 2.2. Préconditions : <ol style="list-style-type: none"> 2.2.1. Avoir terminé le fxml. 2.3. Règles d'affaires : <ol style="list-style-type: none"> 2.3.1. Styliser l'interface en css. 2.4. Règles d'affaires alternatives : <ol style="list-style-type: none"> 2.4.1. Il n'y a pas d'alternative. 2.5. Tests d'acceptation de cet item : <ol style="list-style-type: none"> 2.5.1. Les tests seront au niveau visuel. S'il y a un problème d'affichage, on pourra le voir. 2.6. Post-conditions : <ol style="list-style-type: none"> 2.6.1. L'interface devra pouvoir nous diriger correctement sur la bonne vue selon la sélection de l'utilisateur.
Tests d'acceptation	Les tests seront au niveau visuel. S'il y a un problème d'affichage, on pourra le voir.
Complexité	1
Effort	1

Suite à la page suivante

Commentaires	
3	
Acteur ou rôle	Utilisateur
Scénario ou story	En tant qu'utilisateur, je dois posséder une interface visuelle pour le menu permettant d'accéder aux différentes vues.
Détail et description	<ol style="list-style-type: none"> 1. Concevoir une vue pour le menu. <ol style="list-style-type: none"> 1.1. Qui et temps : <ol style="list-style-type: none"> 1.1.1. (CAD) 1.1.2. (1h) 1.2. Préconditions : <ol style="list-style-type: none"> 1.2.1. Avoir préparé un croquis de l'interface. 1.2.2. Avoir le logiciel Scene Builder pour la conception de l'interface. 1.3. Règles d'affaires : <ol style="list-style-type: none"> 1.3.1. Afficher l'interface du menu. 1.4. Règles d'affaires alternatives : <ol style="list-style-type: none"> 1.4.1. Il n'y a pas d'alternative, car sans interface, l'utilisateur ne peut rien faire. 1.5. Tests d'acceptation de cet item : <ol style="list-style-type: none"> 1.5.1. Les tests seront au niveau visuel. S'il y a un problème d'affichage, on pourra le voir. 1.6. Post-conditions : <ol style="list-style-type: none"> 1.6.1. L'interface devra pouvoir afficher deux boutons pour les deux autres vues. 2. Relier la vue récepteur et la vue émetteur à la vue menu. <ol style="list-style-type: none"> 2.1. Qui et temps : <ol style="list-style-type: none"> 2.1.1. (CAD) 2.1.2. (1h) 2.2. Préconditions : <ol style="list-style-type: none"> 2.2.1. Avoir terminé le fxml des autres vues. 2.3. Règles d'affaires : <ol style="list-style-type: none"> 2.3.1. L'interaction se fera via le contrôleur en Java. 2.4. Règles d'affaires alternatives : <ol style="list-style-type: none"> 2.4.1. Il n'y a pas d'alternative. 2.5. Tests d'acceptation de cet item : <ol style="list-style-type: none"> 2.5.1. Les tests seront au niveau visuel. S'il y a un problème d'affichage, on pourra le voir. 2.6. Post-conditions : <ol style="list-style-type: none"> 2.6.1. L'interface devra être en noir, en vert et en blanc comme une vue de console.
Tests d'acceptation	Les tests seront au niveau visuel. S'il y a un problème d'affichage, on pourra le voir.
Complexité	1
Effort	1
Commentaires	
4	
Acteur ou rôle	Programmeur
Scénario ou story	En tant que programmeur, je veux convertir le code binaire en son afin de pouvoir prochainement le transférer par la prise audio.

Suite à la page suivante

Détail et description	<ol style="list-style-type: none"> 1. Convertir le code binaire en son. <ol style="list-style-type: none"> 1.1. Qui et temps : <ol style="list-style-type: none"> 1.1.1. (NM) 1.1.2. (2h) 1.2. Règles d'affaires : <ol style="list-style-type: none"> 1.2.1. Utiliser l'API MidiChannel de Java pour synthétiser le binaire en son. 1.3. Tests d'acceptation de cet item : <ol style="list-style-type: none"> 1.3.1. Valider ce dernier en diffusant ce son par la sortie audio de l'ordinateur (caisse de son). 1.4. Post-conditions : <ol style="list-style-type: none"> 1.4.1. Le son obtenu sera le résultat du fichier en binaire une fois converti.
Tests d'acceptation	Valider ce dernier en diffusant se son par la sortie audio de l'ordinateur (caisse de son).
Complexité	2
Effort	2
Commentaires	
5	
Acteur ou rôle	Utilisateur
Scénario ou story	En tant qu'utilisateur, je veux pouvoir émettre le son créé par la conversion de mon fichier afin de pouvoir l'envoyer à un récepteur.
Détail et description	<ol style="list-style-type: none"> 1. Faire jouer le son contenant l'information. <ol style="list-style-type: none"> 1.1. Qui et temps : <ol style="list-style-type: none"> 1.1.1. (NM) 1.1.2. (30 minutes) 1.2. Préconditions : <ol style="list-style-type: none"> 1.2.1. Posséder le son créé de la conversion du fichier. 1.3. Règles d'affaires : <ol style="list-style-type: none"> 1.3.1. Sélectionner le son créé de la conversion. 1.3.2. Faire jouer le son. 1.4. Règles d'affaires alternatives : <ol style="list-style-type: none"> 1.4.1. Si aucun son n'est créé, ne rien émettre. 1.5. Tests d'acceptation de cet item : <ol style="list-style-type: none"> 1.5.1. Faire un test sonore. 1.6. Post-conditions : <ol style="list-style-type: none"> 1.6.1. L'utilisateur peut jouer le son.
Tests d'acceptation	Faire un test sonore pour voir si un son est créé.
Complexité	1
Effort	1
Commentaires	

3.4 Backlog de sprint #3

Acteur ou rôle	Physicien
Scénario ou story	En tant que physicien, je veux me renseigner sur les antennes et la télécommunication
Détail et description	<ol style="list-style-type: none"> 1. Recherche d'informations sur Internet et au département de télécommunication au Cégep. <ol style="list-style-type: none"> 1.1. Qui et temps : <ol style="list-style-type: none"> 1.1.1. (TG-H) et (GR) 1.1.2. (1h) 1.2. Préconditions : <ol style="list-style-type: none"> 1.2.1. Avoir un circuit émetteur relativement fonctionnel. 1.2.2. Avoir un circuit récepteur relativement fonctionnel. 1.3. Règles d'affaires : <ol style="list-style-type: none"> 1.3.1. Recherche sur la concordance d'impédance dans un circuit doté d'une antenne. 1.3.2. Rencontre avec un professeur du département de télécommunication. 1.4. Tests d'acceptation de cet item : <ol style="list-style-type: none"> 1.4.1. Il n'y en pas vraiment. 1.5. Post-conditions : <ol style="list-style-type: none"> 1.5.1. Comprendre l'influence des antennes sur les circuits.
Tests d'acceptation	Vérifier nos recherches avec des profs de technique en télécommunications.
Complexité	1
Effort	1
Commentaires	
2	
Acteur ou rôle	Physicien
Scénario ou story	En tant que physicien, je veux optimiser le circuit de l'émetteur et du récepteur.

Suite à la page suivante

Détail et description	<ol style="list-style-type: none"> 1. Accumuler le matériel nécessaire à la conception de meilleures antennes <ol style="list-style-type: none"> 1.1. Qui et temps : <ol style="list-style-type: none"> 1.1.1. (TG-H) et (GR) 1.1.2. (30 minutes) 1.2. Préconditions : <ol style="list-style-type: none"> 1.2.1. Avoir un schéma de construction fini et la liste du matériel. 1.3. Règles d'affaires : <ol style="list-style-type: none"> 1.3.1. Doit respecter le matériel déterminé dans le schéma de construction. 1.4. Tests d'acceptation de cet item : <ol style="list-style-type: none"> 1.4.1. Vérifier si nous possédons bien chaque pièce nécessaire à la complétion des antennes. 1.5. Post-conditions : <ol style="list-style-type: none"> 1.5.1. Nous allons pouvoir procéder au montage des antennes. 2. Procéder à la conception de meilleures antennes <ol style="list-style-type: none"> 2.1. Qui et temps : <ol style="list-style-type: none"> 2.1.1. (TG-H) et (GR) 2.1.2. (4h) 2.2. Préconditions : <ol style="list-style-type: none"> 2.2.1. Avoir les pièces nécessaires à la conception en main. 2.2.2. Avoir le plan de conception achevé et vérifié. 2.3. Règles d'affaires : <ol style="list-style-type: none"> 2.3.1. Suivre le schéma de conception et construire les antennes 2.4. Tests d'acceptation de cet item : <ol style="list-style-type: none"> 2.4.1. Vérifier que l'antenne d'envoi transmet des ondes assez fortes. 2.4.2. Vérifier que l'antenne de récepteur reçoit assez de puissance. 2.5. Post-conditions : <ol style="list-style-type: none"> 2.5.1. L'utilisateur peut capter des ondes radio de manières efficaces.
Tests d'acceptation	Réception d'ondes pour en évaluer la puissance et l'exactitude.
Complexité	3
Effort	6
Commentaires	
3	
Acteur ou rôle	Développeur
Scénario ou story	En tant que développeur, je veux pouvoir animer les interfaces pour l'envoi et la réception.

Suite à la page suivante

Détail et description	<ol style="list-style-type: none"> 1. Ajouter des animations sur les interfaces pour visualiser l'envoi et la réception des fichiers . <ol style="list-style-type: none"> 1.1. Qui et temps : <ol style="list-style-type: none"> 1.1.1. (CAD) 1.1.2. (5h) 1.2. Préconditions : <ol style="list-style-type: none"> 1.2.1. Avoir les vue FXML à jour. 1.2.2. Avoir fait une recherche sur les animations JavaFx. 1.3. Règles d'affaires : <ol style="list-style-type: none"> 1.3.1. Afficher une animation dans le haut des vues. 1.4. Règles d'affaires alternatives : <ol style="list-style-type: none"> 1.4.1. Ne pas afficher d'animation. 1.5. Tests d'acceptation de cet item : <ol style="list-style-type: none"> 1.5.1. Les tests seront au niveau visuel. S'il y a un problème d'affichage, on pourra le voir. 1.6. Post-conditions : <ol style="list-style-type: none"> 1.6.1. Les interfaces devront pouvoir afficher une animation représentant l'envoi d'un fichier. 2. Modifier le slider de fréquence pour qu'il devienne un slider qui gère la vitesse de l'envoi d'un bit et le binder aux autres informations. <ol style="list-style-type: none"> 2.1. Qui et temps : <ol style="list-style-type: none"> 2.1.1. (CAD) 2.1.2. (2h) 2.2. Préconditions : <ol style="list-style-type: none"> 2.2.1. Avoir la vue FXML à jour. 2.3. Règles d'affaires : <ol style="list-style-type: none"> 2.3.1. Modifier la vitesse d'un bit avec la slider. 2.4. Règles d'affaires alternatives : <ol style="list-style-type: none"> 2.4.1. Éliminer le slider, comme il deviendra désuet. 2.5. Tests d'acceptation de cet item : <ol style="list-style-type: none"> 2.5.1. Les tests seront au niveau visuel. S'il y a un problème d'affichage, on pourra le voir. 2.6. Post-conditions : <ol style="list-style-type: none"> 2.6.1. Le slider pourra modifier la vitesse d'envoi d'un bit.
Tests d'acceptation	Les tests seront au niveau visuel. S'il y a un problème d'affichage, on pourra le voir.
Complexité	3
Effort	3
Commentaires	Puisque nous ne savons pas encore comment va fonctionner la réception des informations, il est difficile de paufiner la vue de cette dernière et donc de la finaliser.
4	
Acteur ou rôle	Développeur
Scénario ou story	En tant que développeur, je veux pouvoir optimiser les interfaces pour l'envoi et la réception.

Suite à la page suivante

Détail et description	<ol style="list-style-type: none"> 1. Modifier les dimensions des vues et les informations disponibles sur l'envoi du fichier. <ol style="list-style-type: none"> 1.1. Qui et temps : <ol style="list-style-type: none"> 1.1.1. (CAD) 1.1.2. (3h) 1.2. Préconditions : <ol style="list-style-type: none"> 1.2.1. Avoir la vue FXML à jour. 1.3. Règles d'affaires : <ol style="list-style-type: none"> 1.3.1. Modifier les dimensions pour avoir des interfaces plus grandes. 1.3.2. Modifier et ajouter des informations sur la vitesse et le son produit par l'envoi. 1.4. Règles d'affaires alternatives : <ol style="list-style-type: none"> 1.4.1. Il n'y en a pas. 1.5. Tests d'acceptation de cet item : <ol style="list-style-type: none"> 1.5.1. Les tests seront au niveau visuel. S'il y a un problème d'affichage, on pourra le voir. 1.6. Post-conditions : <ol style="list-style-type: none"> 1.6.1. Les vues seront plus grandes et l'information disponible sera plus spécifique. 2. Ajouter de la documentation aux classes des contrôleurs. <ol style="list-style-type: none"> 2.1. Qui et temps : <ol style="list-style-type: none"> 2.1.1. (CAD) 2.1.2. (1h) 2.2. Préconditions : <ol style="list-style-type: none"> 2.2.1. Avoir terminer les contrôleurs. 2.3. Règles d'affaires : <ol style="list-style-type: none"> 2.3.1. Ajouter la JavaDoc 2.4. Règles d'affaires alternatives : <ol style="list-style-type: none"> 2.4.1. Il n'y en a pas. 2.5. Tests d'acceptation de cet item : <ol style="list-style-type: none"> 2.5.1. Tester de générer la JavaDoc et de la lire. 2.6. Post-conditions : <ol style="list-style-type: none"> 2.6.1. Les méthodes des contrôleurs seront documentées.
Tests d'acceptation	Les tests seront au niveau visuel. S'il y a un problème d'affichage, on pourra le voir.
Complexité	2
Effort	1
Commentaires	
5	
Acteur ou rôle	Programmeur
Scénario ou story	En tant que programmeur, je souhaite acquérir des connaissances sur le fonctionnement des RaspberryPi et des librairies Java permettant son fonctionnement.

Suite à la page suivante

Détail et description	<ol style="list-style-type: none"> 1. Chercher une librairie en Java pour le GPIO d'un RaspberryPi. <ol style="list-style-type: none"> 1.1. Qui et temps : <ol style="list-style-type: none"> 1.1.1. (NM) 1.1.2. (30 min) 1.2. Règles d'affaires : <ol style="list-style-type: none"> 1.2.1. Trouver une librairie Java pour le GPIO. 1.3. Règles d'affaires alternatives : <ol style="list-style-type: none"> 1.3.1. Il n'y a pas d'alternative, car sans librairie, il est impossible de faire fonctionner le RaspberryPi. 1.4. Post-conditions : <ol style="list-style-type: none"> 1.4.1. Nous possédons une librairie utilisable pour le GPIO du RaspberryPi. 2. Comprendre la librairie et le GPIO trouvé. <ol style="list-style-type: none"> 2.1. Qui et temps : <ol style="list-style-type: none"> 2.1.1. (NM) 2.1.2. (1h45) 2.2. Préconditions : <ol style="list-style-type: none"> 2.2.1. Avoir trouvé une librairie utilisable et bien documentée. 2.3. Règles d'affaires : <ol style="list-style-type: none"> 2.3.1. Faire des lectures de la documentation. 2.3.2. Regarder des vidéos en ligne sur le sujet. 2.4. Règles d'affaires alternatives : <ol style="list-style-type: none"> 2.4.1. Il n'y a pas d'alternative puisqu'il faut apprendre ce avec quoi l'on travaille. 2.5. Post-conditions : <ol style="list-style-type: none"> 2.5.1. On comprend désormais le fonctionnement de la librairie. 3. Déterminer s'il existe un simulateur de raspberryPi virtuel. <ol style="list-style-type: none"> 3.1. Qui et temps : <ol style="list-style-type: none"> 3.1.1. (NM) 3.1.2. (15 min) 3.2. Règles d'affaires : <ol style="list-style-type: none"> 3.2.1. Faire des recherches en ligne pour en chercher l'existence. 3.3. Règles d'affaires alternatives : <ol style="list-style-type: none"> 3.3.1. S'il n'existe pas de simulateur de raspberryPi, nous devons nous en passer. 3.4. Tests d'acceptation de cet item : <ol style="list-style-type: none"> 3.4.1. Vérifier que le simulateur fonctionne de façon similaire au vrais. 3.5. Post-conditions : <ol style="list-style-type: none"> 3.5.1. Nous possédons, ou non, un simulateur de raspberryPi qui fonctionne pour faire nos tests.
Tests d'acceptation	Vérification pratique des connaissances.
Complexité	5
Effort	2
Commentaires	

Acteur ou rôle	Programmeur
Scénario ou story	En tant que programmeur, je souhaite pouvoir convertir le courant électrique reçu par le circuit en binaire.
Détail et description	<ol style="list-style-type: none"> 1. Faire un prototype d'application de réception. <ol style="list-style-type: none"> 1.1. Qui et temps : <ol style="list-style-type: none"> 1.1.1. (NM) 1.1.2. (5h) 1.2. Préconditions : <ol style="list-style-type: none"> 1.2.1. Avoir une librairie Java fonctionnelle et la comprendre. 1.3. Règles d'affaires : <ol style="list-style-type: none"> 1.3.1. Lire le voltage sur un pin du GPIO. 1.3.2. Coder le traitement de ce voltage. 1.3.3. Reconstruction des octets envoyés. 1.4. Règles d'affaires alternatives : <ol style="list-style-type: none"> 1.4.1. Il n'y a pas d'alternative, car sans ce code, il est impossible de recréer le fichier envoyé. 1.5. Post-conditions : <ol style="list-style-type: none"> 1.5.1. Il est possible de décoder l'information reçue. 2. Programmer l'implémentation finale de la réception. <ol style="list-style-type: none"> 2.1. Qui et temps : <ol style="list-style-type: none"> 2.1.1. (NM) 2.1.2. (2h) 2.2. Préconditions : <ol style="list-style-type: none"> 2.2.1. Avoir un prototype fonctionnel. 2.3. Règles d'affaires : <ol style="list-style-type: none"> 2.3.1. Lire le voltage sur un pin du GPIO. 2.3.2. Coder le traitement de ce voltage. 2.3.3. Reconstruction des octets envoyés. 2.4. Règles d'affaires alternatives : <ol style="list-style-type: none"> 2.4.1. Il n'y a pas d'alternative, car sans ce code, il est impossible de recréer le fichier envoyé. 2.5. Tests d'acceptation de cet item : <ol style="list-style-type: none"> 2.5.1. Ce sera un test fonctionnel. 2.5.2. Tester unitairement en simulant la réception d'un voltage. 2.6. Post-conditions : <ol style="list-style-type: none"> 2.6.1. Le programme est en mesure d'interpréter les voltages reçus e reconstruire un fichier.
Tests d'acceptation	Les tests seront unitaires en simulant la réception d'un voltage
Complexité	8
Effort	4

Suite à la page suivante

Commentaires	Cette story n'a pas été complètement réalisée, étant donné les problèmes rencontrés avec le Raspberry Pi. Tout ce qui pouvait être fait sans le Raspberry Pi a été fait, donc la story a tout-de-même avancé. Il reste alors à analyser ce que l'on reçoit pour déterminer les bits reçus. Nous n'utiliserons donc pas le Raspberry Pi dans notre projet. Nous utiliserons les librairies offertes par Java pour parvenir à analyser ce que nous recevons.
--------------	--

3.5 Backlog de sprint #4

1	
Acteur ou rôle	Développeur
Scénario ou story	En tant que développeur, je veux me renseigner sur les API et les librairies à disposition pour faire la réception du son.
Détail et description	<ol style="list-style-type: none"> 1. Recherche d'informations sur Internet et prendre le temps de lire les librairies. <ol style="list-style-type: none"> 1.1. Qui et temps : <ol style="list-style-type: none"> 1.1.1. (TG-H), (GR), (NM) et (CAD) 1.1.2. (2h) 1.2. Préconditions : <ol style="list-style-type: none"> 1.2.1. Avoir terminer l'émetteur et le récepteur physique. 1.2.2. Avoir terminer la programmation de l'émission du fichier. 1.3. Règles d'affaires : <ol style="list-style-type: none"> 1.3.1. Recherche sur les API Java son. 1.3.2. Rencontre sur les librairies Java son. 1.4. Tests d'acceptation de cet item : <ol style="list-style-type: none"> 1.4.1. Il n'y en pas vraiment. 1.5. Post-conditions : <ol style="list-style-type: none"> 1.5.1. Comprendre le fonctionnement de réception sonore avec java.
Tests d'acceptation	Appliquer nos apprentissages dans notre programme.
Complexité	2
Effort	3
Commentaires	
2	
Acteur ou rôle	Développeur
Scénario ou story	En tant que développeur, je veux appliquer des modifications à la vue correspondant à la réception de fichier.

Suite à la page suivante

Détail et description	<ol style="list-style-type: none"> 1. Lier la vue réception aux classes métiers permettant de gérer la réception <ol style="list-style-type: none"> 1.1. Qui et temps : <ol style="list-style-type: none"> 1.1.1. (CAD) et (TG-H) 1.1.2. (3h) 1.2. Préconditions : <ol style="list-style-type: none"> 1.2.1. Avoir terminer la vue réception. 1.2.2. Avoir terminer les classes métiers. 1.3. Règles d'affaires : <ol style="list-style-type: none"> 1.3.1. Avoir une vue permettant la réception et l'enregistrement de fichier. 1.4. Règles d'affaires alternative : <ol style="list-style-type: none"> 1.4.1. Aucune, il est essentiel pour l'accomplissement du projet. 1.5. Tests d'acceptation de cet item : <ol style="list-style-type: none"> 1.5.1. Vérifier si on est capable de réceptionner le message envoyé en comparant ce que l'on enregistre et ce que l'on a envoyé. 1.6. Post-conditions : <ol style="list-style-type: none"> 1.6.1. Le programme pourra enregistrer les fichiers envoyés. 2. Optimiser la configuration de l'interface récepteur. <ol style="list-style-type: none"> 2.1. Qui et temps : <ol style="list-style-type: none"> 2.1.1. (CAD) et (TG-H) 2.1.2. (4h) 2.2. Préconditions : <ol style="list-style-type: none"> 2.2.1. Avoir terminer la vue réception. 2.3. Règles d'affaires : <ol style="list-style-type: none"> 2.3.1. Avoir une vue avec les outils nécessaires pour visualiser et gérer la réception. 2.4. Tests d'acceptation de cet item : <ol style="list-style-type: none"> 2.4.1. Le tests se feront visuellement. 2.5. Post-conditions : <ol style="list-style-type: none"> 2.5.1. Avoir une vue avec les éléments nécessaires pour visualiser la réception.
Tests d'acceptation	Vérifier si on est capable de réceptionner le message envoyé en comparant ce que l'on enregistre et ce que l'on a envoyé.
Complexité	3
Effort	5
Commentaires	
3	
Acteur ou rôle	Développeur
Scénario ou story	En tant que développeur, je veux pouvoir analyser les données reçues par radio afin de déterminer si on reçoit un 1 ou un 0.

Suite à la page suivante

Détail et description	<ol style="list-style-type: none"> 1. Accéder aux données entrantes par la prise du microphone d'un ordinateur. <ol style="list-style-type: none"> 1.1. Qui et temps : <ol style="list-style-type: none"> 1.1.1. (NM) et (GR) 1.1.2. (2h) 1.2. Préconditions : <ol style="list-style-type: none"> 1.2.1. Avoir fait la recherche à ce sujet. 1.3. Règles d'affaires : <ol style="list-style-type: none"> 1.3.1. Pouvoir accéder aux données sonores du microphone. 1.4. Règles d'affaires alternatives : <ol style="list-style-type: none"> 1.4.1. Il n'y a pas d'alternative, cette étape est absolument nécessaire, car il est impossible de reconstruire le fichier si nous n'avons pas accès à ces informations. 1.5. Tests d'acceptation de cet item : <ol style="list-style-type: none"> 1.5.1. Les tests seront unitaires. 1.6. Post-conditions : <ol style="list-style-type: none"> 1.6.1. Nous avons maintenant accès aux informations provenant de la prise du micro et nous pouvons alors manipuler ces informations. 2. Analyser les données reçues par la prise du microphone et déterminer si nous recevons un 1 ou un 0. <ol style="list-style-type: none"> 2.1. Qui et temps : <ol style="list-style-type: none"> 2.1.1. (NM) et (GR) 2.1.2. (4h) 2.2. Préconditions : <ol style="list-style-type: none"> 2.2.1. Avoir accès aux informations provenant de la prise du microphone. 2.3. Règles d'affaires : <ol style="list-style-type: none"> 2.3.1. Déterminer si les données représentent un 1 ou un 0. 2.4. Règles d'affaires alternatives : <ol style="list-style-type: none"> 2.4.1. Il n'y a pas d'alternative, cette étape est absolument nécessaire, car il est impossible de reconstruire le fichier si nous n'avons pas accès à ces informations. 2.5. Tests d'acceptation de cet item : <ol style="list-style-type: none"> 2.5.1. Les tests seront fonctionnels, car on ne peut pas simuler la réception de données du microphone. 2.6. Post-conditions : <ol style="list-style-type: none"> 2.6.1. Il est possible de déterminer si nous recevons un 1 ou un 0.
Tests d'acceptation	Les tests unitaires passent et les tests fonctionnels sont réussis.
Complexité	7
Effort	4
Commentaires	

4 Diagramme de classes

5 Échéancier

Échéancier : Utilisation d'un échéancier suivant le modèle du diagramme de Gantt réalisé à l'aide de MS Project. Il est divisé en trois captures d'écran à des fins pratiques.

FIGURE 3 – Diagramme de classes

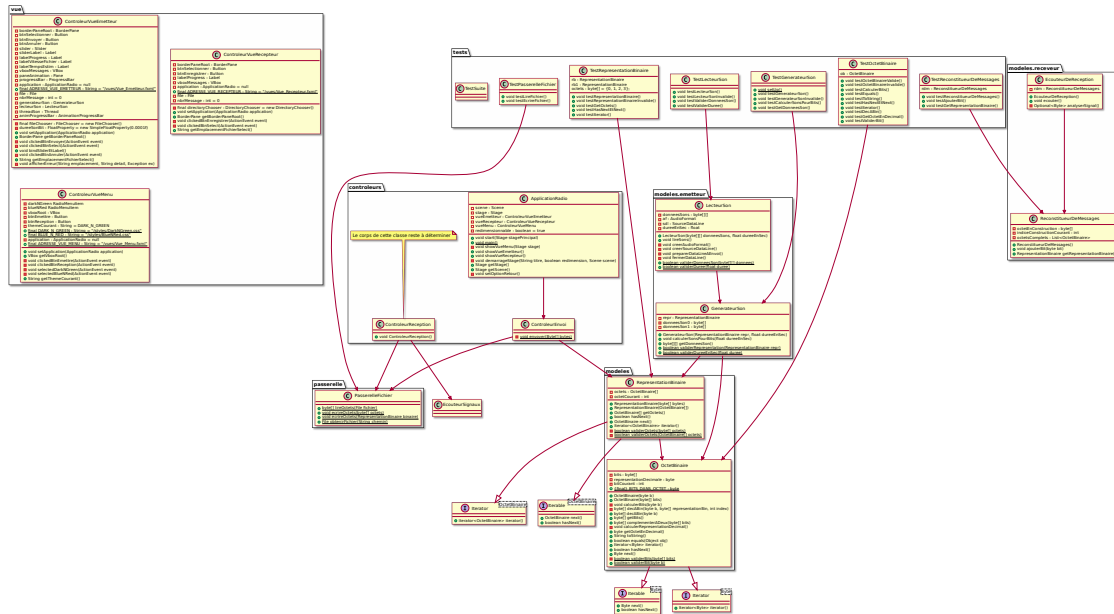


FIGURE 4 – Échéancier partie 1

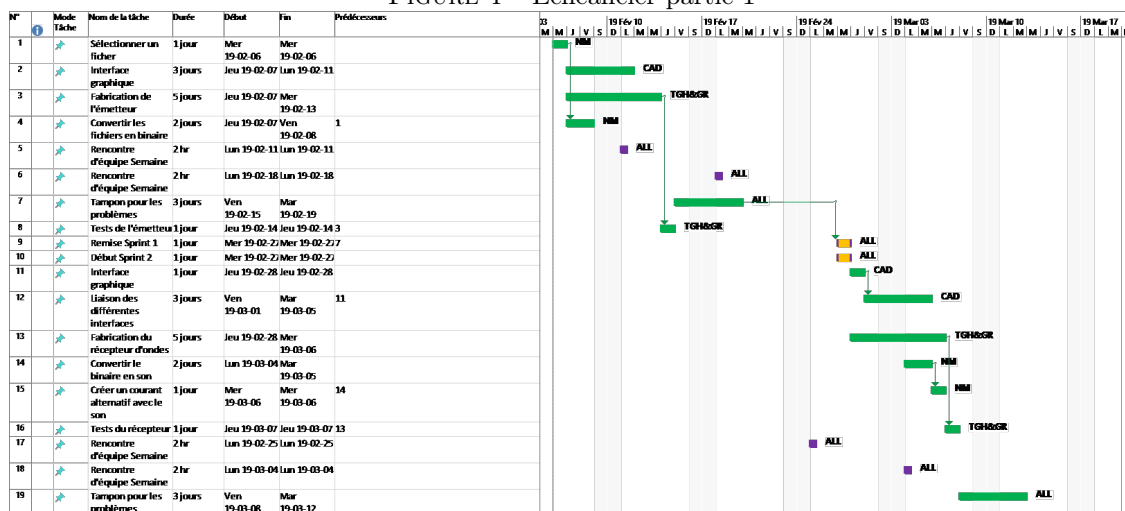


FIGURE 5 – Échéancier partie 2

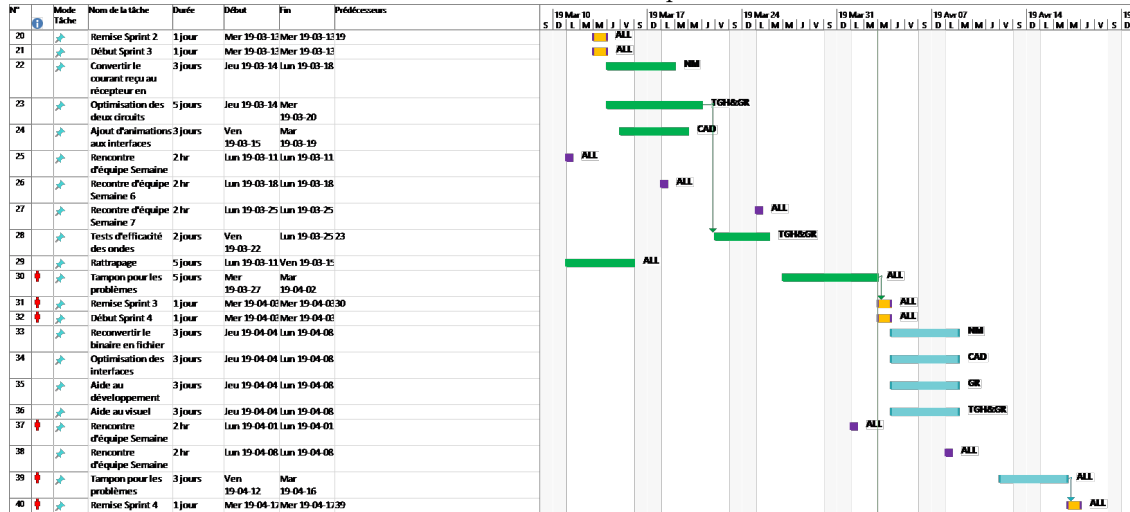


FIGURE 6 – Échéancier partie 3

