

SEPTEMBRE-NOVEMBRE 2021

Peppa Maths

PROPOSÉ PAR

POGU Louise & FORTINEAU Louis

ENSEIGNEMENT &
FORMATION

Electronique Numérique
2ème année Bachelor solutions numériques connectés

CAHIER DES CHARGES

Le projet est un jeu interactif intitulé Peppa Maths, destiné à un public plutôt jeune (ou pas).

Le jeu consiste à résoudre mentallement différentes opérations mathématiques qui seront affichées sur un écran lcd et où le temps de réponse sera limité par un chrono et signalé par un buzzer. La réponse sera envoyée à la carte grâce à un clavier matriciel.

Si la réponse de l'utilisateur est bonne, le programme demandera s'il veut continuer sa série ou s'il veut arrêter et retourner au menu principal, mais, si la réponse est fausse, game over, le jeu retournera au menu principal. Cependant, il n'est pas possible de mettre en pause la partie.

SCHÉMA ÉLECTRIQUE

COMPOSANT	BROCHE PINOUT	PIN COMPOSANT
Clavier matriciel	PB11 KEYBi0	1
	PB10 KEYBi1	2
	PB1 KEYBi2	3
	PB0 KEYBi3	4
	PB12 KEYBo0	5
	PB13 KEYBo1	6
	PB14 KEYBo2	7
	PB15 KEYBo3	8
UART	+5V	2
	GND	3
	PA3_U2RX	4
	PA4_U2TX	5
LED	PA15_LED	pas de Pin composant
BUZZER	PA11_BUZZER	pas de Pin composant
Ecran TFT	+3.3V	1
	GND	2
	PA12_TFT-CS	3
	PA9_TFT-RESET	4
	PA8_TFT-DC	5
	PA7_SPI-MOSI	6+12
	PA5_SPI-SCK	7+10
	+3.3V	8
	PA6_SPI-MISO	9+13

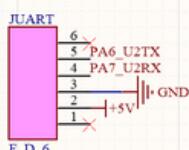
SCHÉMA ÉLECTRIQUE

IHM -> ENTREES/ COMMUNICATION

Clavier Matriciel

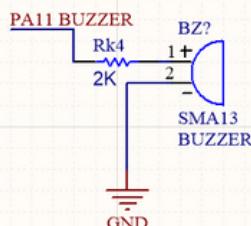
Communication

UART



SONORE

Buzzer

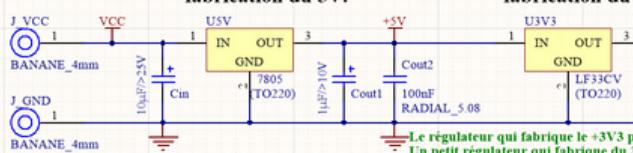


ALIMENTATION

Regulateurs de tension

7V < Vcc < 15V env.

fabrication du 5V.



fabrication du 3V3

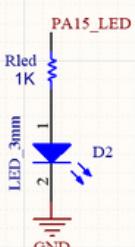
Le régulateur qui fabrique le +3V3 peut être supprimé si aucun composant n'en a besoin !
Un petit régulateur qui fabrique le 3V3 est disponible sur la mini carte du microcontrôleur
Ce potentiel est nommé sur ce schéma +3V3_BP (Blue Pill)

Alimentation sur la BluePill :
le +5V est une entrée
le +3V3_BP est une sortie

IHM -> AFFICHAGE

Ecran graphique tactile 2.4"

LED



ECRAN TFT

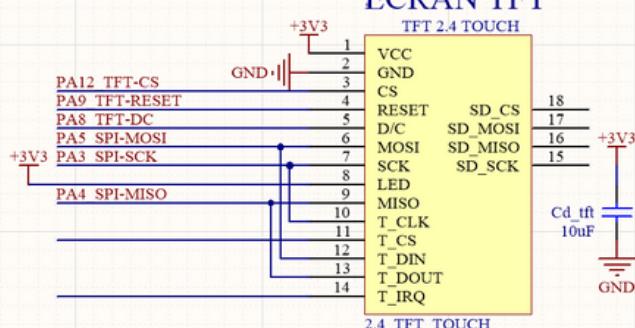
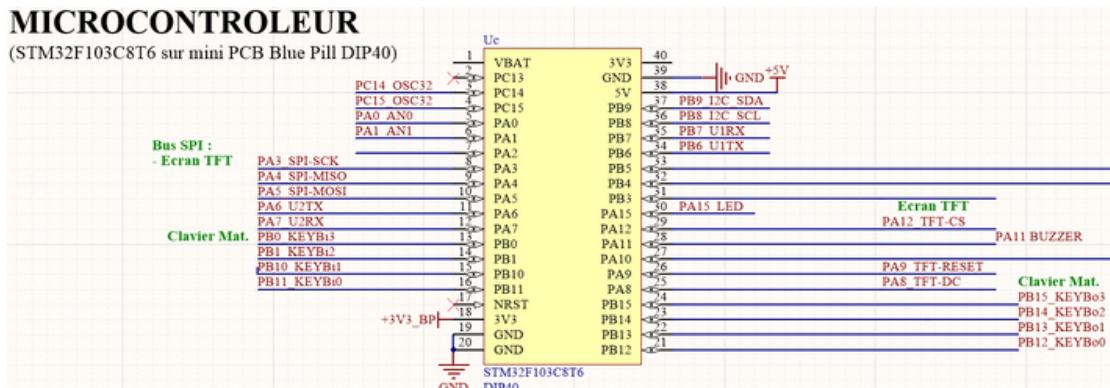


SCHÉMA ÉLECTRIQUE

MICROCONTROLEUR

(STM32F103C8T6 sur mini PCB Blue Pill DIP40)



NOM OU REF DU COMPOSANT

- Clavier matriciel
- Buzzer
- Ecran TFT

CARACTÉRISTIQUES PRINCIPALES DANS LE PROJET

Il permet de répondre aux opérations et a sélectionner les différents mode de jeu

Lorsque la réponse est fausse ou que le temps est écoulé, il émet du son. Un son différent est émit lorsque la réponse est bonne

Il permet d'afficher les opérations, et de suivre les indications

ROUTAGE

image en 2D du PCB

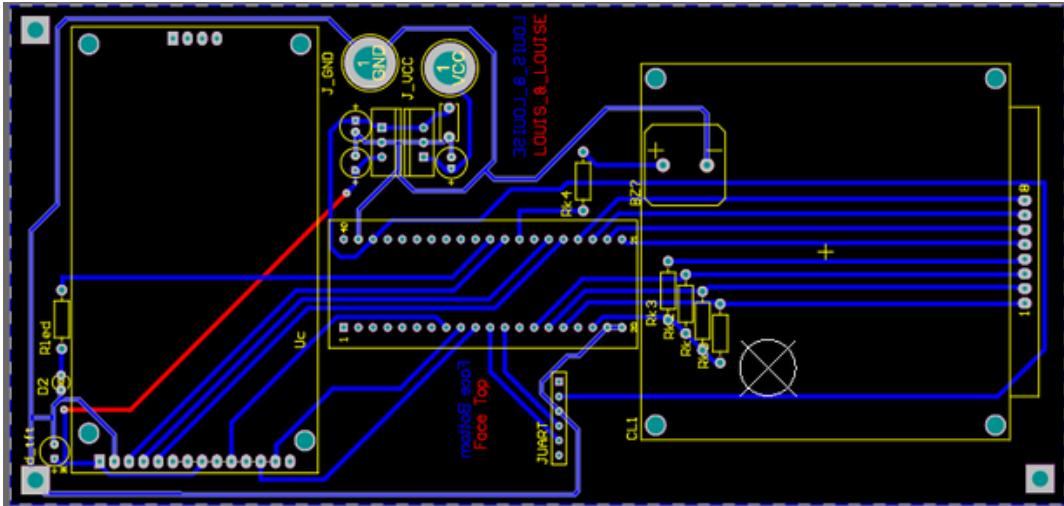
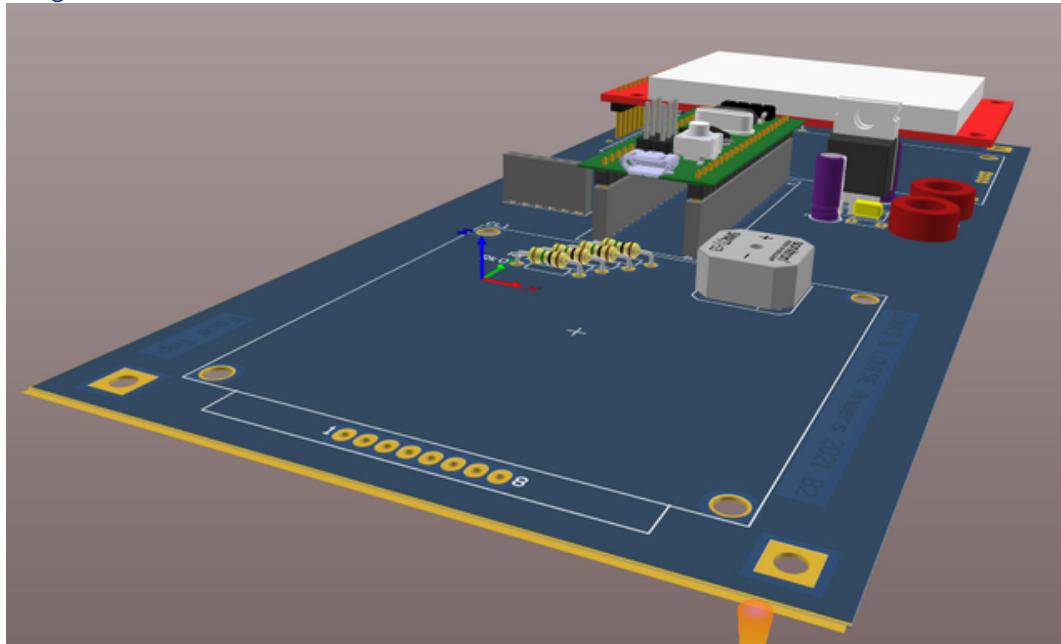
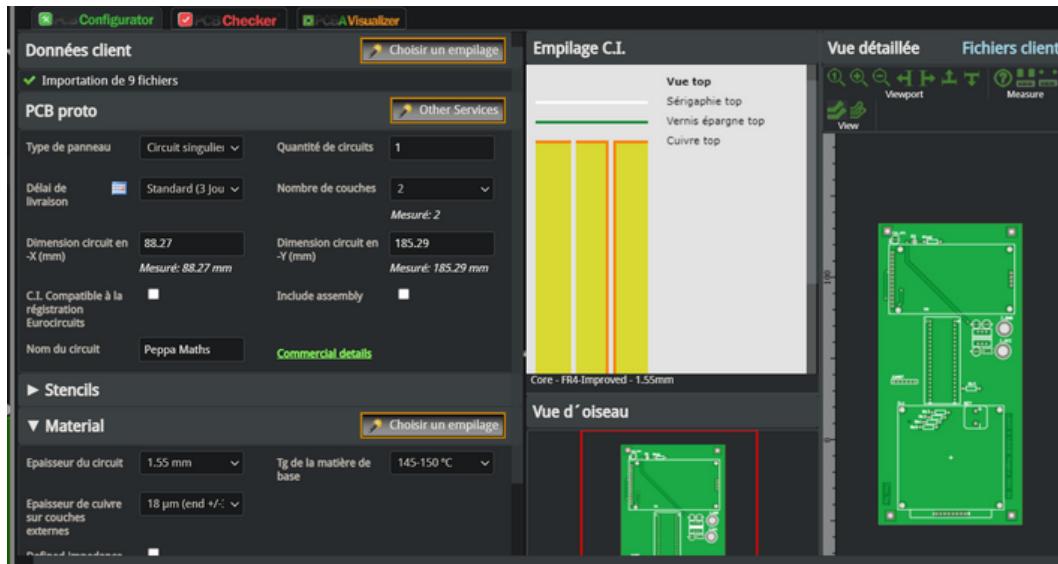


image en 3D du PCB



VALIDATION DU PCB



Résumé	
Service	PCB proto
Date d'expédition	08-10-2021
Quantité	1 circuits
Surface de la carte / Surface de la commande	1.64 dm ² / 1.64 dm ²
Prix	Prix TTC*
Prix unitaire	€ 71.94
Prix circuits	€ 71.94
Transport express	€ 0.00
Prix total	€ 71.94

* Les prix bruts incluent 20.00% TVA.

MANUEL D'UTILISATION

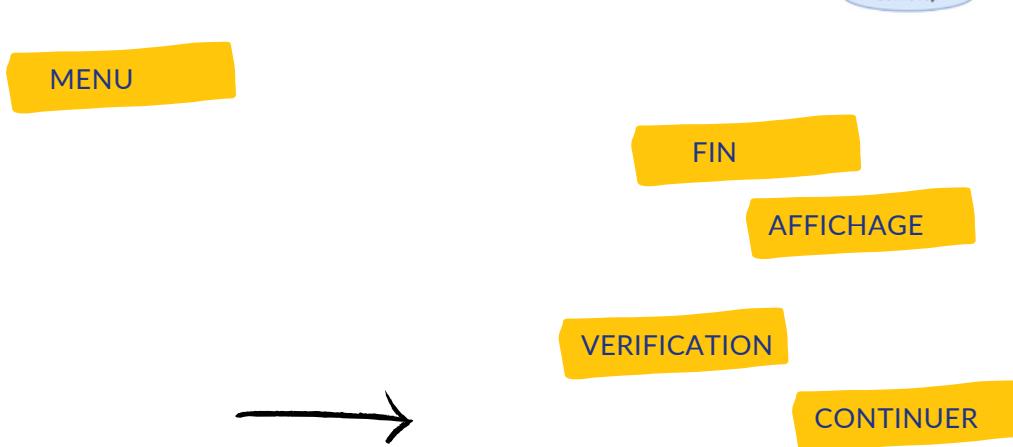
Bienvenue sur notre jeu Peppa Maths

Pour sélectionner le premier mode appuyer sur [1], le deuxième mode sur [2], le troisième mode sur [3] et le dernier mode sur [4].

Une opération est apparue, pour répondre, il vous faut taper la solution puis valider en appuyant sur [D] avant la fin du timer. Attention, il y a aucun moyen d'effacer ou d'annuler une réponse.

Ensuite(en cas de bonne réponse), nous vous demandons si vous voulez continuer en validant [#] ou en arrêtant [*]. En cas d'échec ou d'arrêt, vous retombez sur le Menu principal.

DESCRIPTION D'UN ALGORITHME DU PROGRAMME



Nous avons décidé de réaliser une machine à état pour rendre la compréhension plus claire et plus lisible.

Le schéma ci-dessus représente la structure de notre code. Des fonctions complémentaires ont été déclarés en dehors mais, elles sont utilisées dans la machine à état.

STRUCTURE DU PROGRAMME

Fichier "main.c"

FONCTION & DÉCLARATION

DÉVELOPPEUR

DESCRIPTION

state_e	Louise	Enumération pour le nom des cases de la machine à état
custom_keyboard_13_touches[16]	Louise & Louis	Création des touches utilisés dans le programme
KEYBOARD_get_int	Louis	Permet d'envoyer un int lorsque l'on appuie sur une touche de type char
decompte	Louise	Permet de savoir si le décompte est fini
nombreAleatoire	Louise & Louis	Permet d'avoir des nombres aléatoires lors d'un appui
main	Louise & Louis	Permet de gérer la machine à état

TESTS

NOM DU TEST	DESCRIPTION	RÉSULTAT
Test d'alimentation du microcontrôleur	Mesure au voltmètre ; les entrées d'alimentation doivent être à 5V	OK
Test de la led	Dès que le code est lancé, la led verte s'allume	OK
Test du clavier	Dès que l'on appuis sur une touche, il doit envoyer le bon caractère en fonction du clavier choisi (personnaliser ou non)	OK
Test écran	affichage de la démo pour voir s'il est bien connecté	OK
Test de la machine à état	appuis sur les bonnes touches pour changer d'état	OK
Test fonction decompte	lorsque le timer est à 0, le buzzer émet du son, on va dans VERIFICATION puis dans FIN	OK
Test des nombres aléatoires	lorsque l'on appuie sur une touche, la fonction nombreAleatoire est appelée et ensuite on va stocker cette valeur dans nb1 ou nb2	OK
Test de bien entrer ce que l'on tape	lorsque l'on va appuyer sur une touche, elle va s'afficher sur l'écran	OK
Test nombre de tour	on vérifie que l'on sort bien après avoir fait 5 tours	OK

CAHIER DE SUIVI

DATE	LOUISE	LOUIS
21/09	<ul style="list-style-type: none"> - Choix du sujet - Réflexion sur la machine à état en soft 	
27/09	<ul style="list-style-type: none"> - Réflexion sur la mise en place des éléments en hard - Appropriation du logiciel - Début de la mise en forme du PCB sur Altium 	
4/10	<ul style="list-style-type: none"> - Mise en place des lignes de connexion avec Louis sur le PCB (Altium) 	<ul style="list-style-type: none"> - Revu des sorties du microcontrôleur
5/10	<ul style="list-style-type: none"> - Fin du PCB et envoi du fichier 	<ul style="list-style-type: none"> - Vérification des lignes de connexion - Début du soft - Trouver pourquoi ca ne fonctionne pas
11/10	<ul style="list-style-type: none"> - Soudure 	
15/10	<ul style="list-style-type: none"> - Test de la carte - vérification des tensions d'entrées - mise en place de la carte - début du code 	
18/10	<ul style="list-style-type: none"> - Début de visualisation pour le clavier - Résolution de problème sur le pcb - recherche problème 2UART 	<ul style="list-style-type: none"> - Test de l'écran - Résolution de problème sur le pcb - recherche problème 2UART
8/11	<ul style="list-style-type: none"> - galère sur 2 bout de fils de merde (2h10) - début de la machine à état 	
15/11	<ul style="list-style-type: none"> - très gros avancement dans le code - test des différentes fonctions / états - continuation de debogg 	<ul style="list-style-type: none"> - test nombres aléatoires - mise en forme sur l'écran sur le code - continuation de debogg

ÉTAT D'AVANCEMENT ET ANALYSE DU PROJET RÉALISÉ

Si vous deviez refaire le projet, que feriez-vous autrement?

On aurait sans doute fait plus attention à notre PCB, au niveau de la création. Ca nous a poser des problèmes qui ne serait pas apparu si nous avions fait un peu plus attention

Quel moyen de débogage aurait-il fallu utiliser plus tôt ou différemment ?

- Il aurait fallu utiliser encore plus tot docklight avec plus de printf et de breakpoint ca nous aurait fait gagner plus de temps encore même si on les a utilisé frequemment sur la fin

Quelles limites gênantes et pour quels composants?

-L'implémentation d'images était compliqué en raison de la mémoire du microcontrôleur trop petite ce qui rend le visuel un peu fade malheureusement

Reste-t-il des points à terminer ?

-Régler quelques bugs sur la taille des reponses pour ne pas avoir des chiffres aleatoire
- implementation d'un timer et d'autre mode de jeu

CONCLUSION

Ce projet nous a beaucoup appris.

Pour la partie hard, le logiciel Altium nous a permis de réfléchir sur le passage des lignes de connexions. Le placement des composants devait être astucieux en fonction de la place qu'ils prenaient, des connexions à la carte et des lignes d'alimentation. Cette partie a vraiment été intéressante au niveau astuce et réalisation.

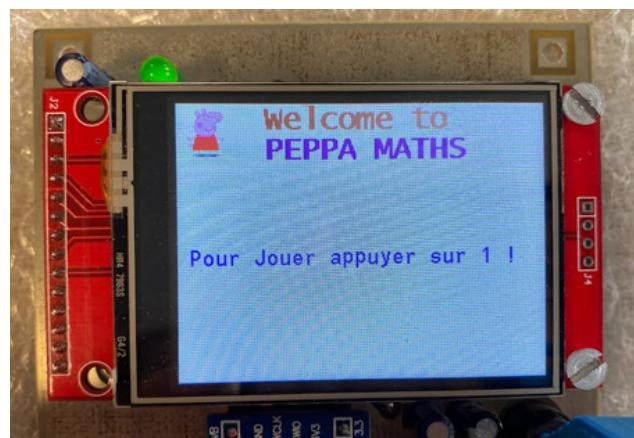
Pour la partie soft, le code parfois n'a pas été sympathique. Nous avons tout de même réussi a surmonter cet obstacle. Nous avons pu vraiment travailler en équipe sur la réalisation du code.

Dans l'ensemble, le projet a vraiment été enrichissant. Nous avons pu mener à bien et jusqu'au bout notre projet. Nous sommes tout de même content de notre travail même s'il n'est pas fini.

PEPPA MATHS

ANNEXES

MEMU



VIDEOS

NOVEMBRE - DÉCEMBRE 2021



BN
PL
MC

Micro-projet réseau S3

BACHELOR SOLUTIONS NUMRÉRIQUES
CONNECTÉES

Réseaux S3 2021

PROPOSÉ À

BAUDIQUEY Nicolas POGU Louise MAIGNAN Clémence

CONSIGNES:

Toutes les réponses doivent être argumentées.

- Le document est donné à titre indicatif. Charge à vous de le modifier pour le rendre le plus précis.
- Le document doit comprendre tout ce qui concerne les 2 sites et le cœur de réseau de la société.
- La partie interconnexion des trinômes n'est prise en compte que si les parties précédentes sont opérationnelles et validées.
- Agencé votre document de façon propre pour faciliter la lecture. (Pris en compte dans la note)

NOTATION:

Notation :

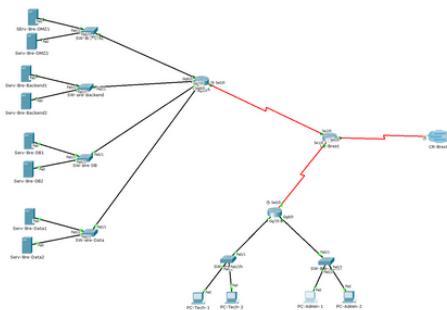
- 12 points pour ce document
- 8 points sur la validation

1- INFRASTRUCTURE DU RÉSEAUX (1 POINT)

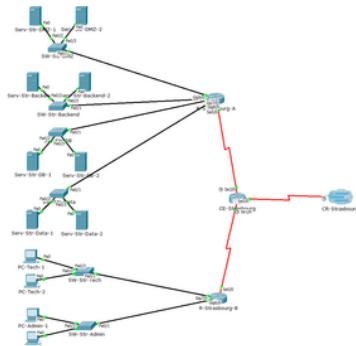
A. TOPOLOGIE LOGIQUE DE NOTRE RÉSEAUX

Export des images du Packet Tracer : File -> Print -> Print to file)

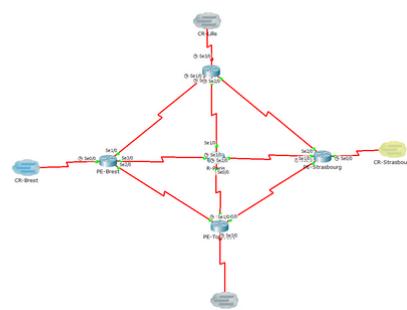
BREST



STRASBOURG



COEUR DE RESEAU



II. LES ADRESSES DES RÉSEAUX (4 POINTS)

A. RÉSEAU DE LA SOCIÉTÉ.

On part du réseau 12.34.0.0 / 18 que l'on divise en 8 pour que chacun des groupes. On obtient alors un réseau en /21. Etant le groupe 5, nous avons le réseau 5. Pour passer du réseau d'un groupe à un autre, il faut ajouter à l'adresse serveur 0.0.8.0 . En partant du réseau de départ, nous devons ajouter 0.0.32.0 à 12.34.0.0 /21 . Nous avons donc comme adresse de réseau principal, 12.34.32.0/21 .

B. MASQUES DES SOUS RÉSEAUX.

Le réseau DMZ étant composé de 24 machines, il nous fallait un masque pouvant contenir ces 24 machines + 2 pour l'adresse source et de broadcast. Nous avons trouvé qu'il nous fallait 5 bits pour la partie machine car $2^5 = 32$. On se retrouve donc avec 27 bits pour la partie réseau donc un masque /27.

Ville	Réseau	Masque /xx	Masque xxx.xxx.xxx.xxx
Brest	DMZ	/27	255.255.255.224
	Backend	/28	255.255.255.240
	DB	/29	255.255.255.248
	Data	/29	255.255.255.248
	Tech	/28	255.255.255.240
	Admin	/28	255.255.255.240
	Lien A-CE	/30	255.255.255.252
	Lien B-CE	/30	255.255.255.252
Strasbourg	DMZ	/27	255.255.255.224
	Backend	/28	255.255.255.240
	DB	/29	255.255.255.248
	Data	/29	255.255.255.248
	Tech	/28	255.255.255.240
	Admin	/28	255.255.255.240
	Lien A-CE	/30	255.255.255.252
	Lien B-CE	/30	255.255.255.252
Coeur de réseaux	Paris-Brest	/30	255.255.255.252
	Paris-Lille	/30	255.255.255.252
	Paris-Strasbourg	/30	255.255.255.252
	Paris-Toulouse	/30	255.255.255.252
	Brest-Lille	/30	255.255.255.252
	Lille-Strasbourg	/30	255.255.255.252
	Strasbourg-Toulouse	/30	255.255.255.252
	Toulouse-Brest	/30	255.255.255.252
	CR-Brest - PE-Brest	/30	255.255.255.252
	CR-Strasb - PE-Srasb	/30	255.255.255.252

C. LISTE DES RÉSEAUX DE NOTRE SOCIÉTÉ

Réseau	Adresse	Masque xxx.xxx.xxx.xxx
Brest	12.34.32.0	255.255.255.128
Brest – Bat A	12.34.32.0	255.255.255.192
Brest – Bat B	12.34.32.64	255.255.255.224
Brest – DMZ	12.34.32.0	255.255.255.224
Brest – Backend	12.34.32.32	255.255.255.240
Brest – DB	12.34.32.48	255.255.255.248
Brest – Data	12.34.32.56	255.255.255.248
Brest – Tech	12.34.32.64	255.255.255.240
Brest – Admin	12.34.32.80	255.255.255.240
R-Bre-A <-> CE-Bre	12.34.32.97 <-> 12.34.32.98	255.255.255.252
R-Bre-B <-> CE-Bre	12.34.32.103 <-> 12.34.32.105	255.255.255.252
Strasbourg	12.34.32.0	255.255.255.128
Strasbourg – Bat A	12.34.32.0	255.255.255.192
Strasbourg – Bat B	12.34.32.64	255.255.255.224
Strasbourg – DMZ	12.34.32.0	255.255.255.224
Strasbourg – Backend	12.34.32.32	255.255.255.240
Strasbourg – DB	12.34.32.48	255.255.255.248
Strasbourg – Data	12.34.32.56	255.255.255.248
Strasbourg – Tech	12.34.32.64	255.255.255.240
Strasbourg – Admin	12.34.32.80	255.255.255.240
R-Str-A <-> CE-Str	12.34.32.105 <-> 12.34.32.106	255.255.255.252 <-> 255.255.255.252
R-Str-B <-> CE-Str	12.34.32.102 <-> 12.34.32.101	255.255.255.252 <-> 255.255.255.252

D. EXPLIQUEZ LE RÔLE D'UNE PASSERELLE PAR DÉFAUT SUR UN RÉSEAU.

Quand on a affaire à un hôte appartenant à un autre réseau que l'hôte qui émet, le paquet doit passer par une passerelle par défaut. L'adresse de cette passerelle est l'adresse IP de l'interface réseau du routeur connecté au réseau local. Donc lors d'un envoi de paquets vers des hôtes de réseaux distants, la passerelle par défaut va donc être utilisée.

III. PLAN D'ADRESSAGE (3 POINTS)

1. SITE DE BREST

Lieu	Nom	Equipement	IP	Masque	Passerelle
Bat A					
DMZ	DMZ-1		12.34.33.1	/27	12.34.33.30
	DMZ-2		12.34.33.29	/27	12.34.33.30
	R-Brest-A - Gi 6/0		12.34.33.30	/27	12.34.33.30
Backend	Backend-1		12.34.33.33	/28	12.34.33.46
	Backend-2		12.34.33.45	/28	12.34.33.46
	R-Bre-A - Gi 7/0		12.34.33.46	/28	12.34.33.46
DB	DB-1		12.34.33.49	/29	12.34.33.54
	DB-2		12.34.33.53	/29	12.34.33.54
	R-Bre-A - Gi 8/0		12.34.33.54	/29	12.34.33.54
DATA	Data-1		12.34.33.55	/29	12.34.33.54
	Data-2		12.34.33.61	/29	12.34.33.54
	R-Bre-A - Gi 9/0		12.34.33.62	/29	12.34.33.54
Bat B					
Technicien	PC-Prod-1		12.34.33.65	/28	12.34.33.78
	PC-Prod-2		12.34.33.77	/28	12.34.33.78
	R-Bre-B - Gi 7/0		12.34.33.78	/28	12.34.33.78
Administ.	PC-Log-1		12.34.33.81	/28	12.34.33.94
	PC-Log-2		12.34.33.93	/28	12.34.33.94
	R-BatB - Gi 8/0		12.34.33.94	/28	12.34.33.94
Liaison					
Lien A-CE	R-CE - Se 2/0		12.34.33.98	/30	12.34.33.98
	R-BatA - Se 0/0		12.34.33.97	/30	12.34.33.98
Lien B-CE	R-CE - Se 1/0		12.34.33.105	/30	12.34.33.105
	R-BatB - Se 0/0		12.34.33.103	/30	12.34.33.105

2. SITE DE STRASBOURG

Lieu	Nom	Equipement	IP	Masque	Passerelle
Bat A					
DMZ	DMZ-1		12.34.32.1	/27	12.34.32.30
	DMZ-2		12.34.32.29	/27	12.34.32.30
	R-Str-A - Gi 0/9		12.34.32.30	/27	12.34.32.30
Backend	Backend-1		12.34.32.33	/28	12.34.32.46
	Backend-2		12.34.32.45	/28	12.34.32.46
	R-Str-A - Gi 0/8		12.34.32.46	/28	12.34.32.46
DB	DB-1		12.34.32.49	/29	12.34.32.54
	DB-2		12.34.32.53	/29	12.34.32.54
	R-Str-A - Gi 0/7		12.34.32.54	/29	12.34.32.54
DATA	Data-1		12.34.32.57	/29	12.34.32.62
	Data-2		12.34.32.61	/29	12.34.32.62
	R-Str-A - Gi 0/6		12.34.32.62	/29	12.34.32.62
Bat B					
Technicien	PC-Prod-1		12.34.32.65	/28	12.34.32.78
	PC-Prod-2		12.34.32.77	/28	12.34.32.78
	R-BatB - Gi 0/7		12.34.32.78	/28	12.34.32.78
Administ.	PC-Log-1		12.34.32.81	/28	12.34.32.94
	PC-Log-2		12.34.32.93	/28	12.34.32.94
	R-BatB - Gi 0/6		12.34.32.94	/28	12.34.32.94
Liaison					
Lien A-CE	R-CE - Se 2/0 <->	12.34.32.105 <->	/30	12.34.32.105 <->	
	R-BatA - Se 0/0	12.34.32.106	/30	12.34.32.106	
Lien B-CE	R-CE - Se 1/0 <->	12.34.32.101 <->	/30	12.34.32.101 <->	
	R-BatB - Se 0/0	12.34.32.102	/30	12.34.32.102	

3. CŒUR DE RÉSEAU

Lieu	Nom	Equipement	IP	Masque	Passerelle
Cœur réseau	R-Paris	Se 0/0	12.34.39.142	/30	Vers PE-Toulouse
		Se 1/0	12.34.39.134	/30	Vers PE-Lille
		Se 2/0	12.34.39.138	/30	Vers PE-Strasbourg
		Se 3/0	12.34.39.130	/30	Vers PE-Brest
	PE-Strasbourg	Se 0/0	12.34.39.170	/30	Vers CR-Strasbourg
		Se 3/0	12.34.39.154	/30	Vers PE- Toulouse
	PE-Brest	Se 0/0	12.34.39.162	/30	Vers CR-Brest
		Se 1/0	12.34.39.146	/30	Vers PE- Lille
		Se 3/0	12.34.39.130	/30	Vers R-Paris
PE-Toulouse	Se 2/0		12.34.39.158	/30	Vers PE-Brest
PE-Lille	Se 2/0		12.34.39.150	/30	Vers PE-Strasbourg

III. PLAN D'ADRESSAGE (3 POINTS)

A. R-BRE-A DE BREST

Connecté / Distant	Réseau	Masque	Next-Hop / if
Connecté	12.34.33.97	/30	Se 0/0

B. CE DE BREST

Connecté / Distant	Réseau	Masque	Next-Hop / if
Connecté	12.34.32.158	/30	Se 0/0

C. PE-BREST (COEUR DE RÉSEAUX)

Connecté / Distant	Réseau	Masque	Next-Hop / if
Connecté	12.34.39.144	/30	Se1/0
Connecté	12.34.39.160	/30	Se0/0
Connecté	12.34.39.156	/30	Se2/0
Connecté	12.34.39.128	/30	Se3/0

D. PE-STRASBOURG (COEUR DE RÉSEAUX)

Connecté / Distant	Réseau	Masque	Next-Hop / if
Connecté	12.34.39.136	/30	Se1/0
Connecté	12.34.39.148	/30	Se2/0
Connecté	12.34.39.152	/30	Se3/0
Connecté	12.34.39.168	/30	Se0/0

V. LISTE DES COMMANDES IOS. (1 POINTS)

La commande "ip address xx.xx.xx.xx 255.xxx.xxx.xxx" doit être tapée dans les routeurs et les PCs des réseaux de villes afin de les connecter entre eux par les routes statiques.

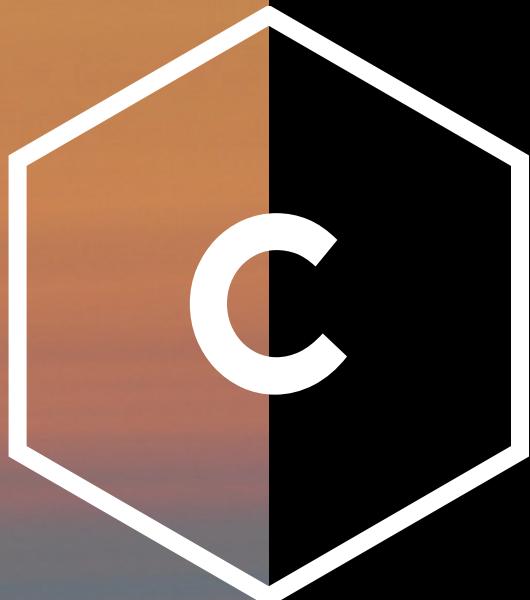
La commande "ip route xx.xx.xx.xx masque xx.xx.xx.xx" doit être tapé pour créer une route par défaut. Elle est alors utilisée lorsque la table de routage ne contient pas de chemin vers un réseau de destination. Pour la redondance, on utilise la même méthode.

La commande "enable secret cisco" créer un mot de passe pour accéder au mode privilégié.

La commande "line console 0" permet d'accéder aux paramètres de la console, puis "password class" de mettre un mot de passe pour accéder à la console.



MAI 2020



Restitution de Projet

PROJET REALISE

Jeu de Dames

REALISE PAR

Louise POGU & Clémence MAIGNAN

SOMMAIRE

1- PRESENTATION DU PROJET (page2)

A- INTRODUCTION

B- LE PROJET A REALISER

2- CAHIER DES CHARGES BASIQUES (page 3)

A- CE QUI A ETE REALISER

B- FONCTIONNALITE REALISE PAR LE SOFT

3- REPARTITION DES TACHES (page 4)

A- ORGANISATION AU SIEN DU BINOME

4- EXTENSIONS DU PROJET (page 4)

A- TRAVAIL SUR DES EXTENSIONS

5- CONCLUSION (page 5)

6- ANNEXE PHOTOS DU JEU (page 6)

A- POUR LANCER LE JEU

PRESENTATION DU PROJET

INTRODUCTION

Le projet est le jeu de Dame, jeu connu pour sa stratégie complexe à appliquer. Afin de pouvoir y jouer sur nos écrans, nous avons programmé ce jeu sous le langage C, un langage informatique utilisé dans plusieurs domaines comme les systèmes embarqués.

LE PROJET A REALISER

- Développement le jeu de dames en langage C
- Réalisé en autonomie sur plusieurs séances.
- Mise en application du cours

CAHIER DES CHARGES BASIQUES

Tout au long du projet, notre but était de rendre le programme le plus lisible et le plus logique possible.

De plus, nous avons utilisé des éléments tels que les structures pour rendre un code plus compacte. Pour prendre un exemple, nous en avons une qui nous permet avec un seul tableau, de stocker les données de tout les pions grâce à une structure qui contient les coordonnées du pion, sa couleur, s'il existe ou non et si c'est une dame ou un pion. Cela pour chaque joueur.

Tout d'abord, nous nous sommes penchés sur comment stocker (format) la grille de jeu. Ensuite, nous avons pu s'occuper de coder le jeu.

Nous avons blindé le code. Autrement dit, de prévenir des erreurs dû à une faute de la part de l'utilisateur.

En plus de blinder le code, nous avons couvert tous les cas possibles du jeu. Cette partie a pris énormément de temps. En effet, cela nous a demandé une recherche plus approfondie sur papier. Cependant, nous avons gagné du temps sur l'écriture du code. Tout était clair dans nos têtes.

Nous avons préféré découper notre code en plusieurs fonctions et les regrouper dans leurs sections (initialisation grille, initialisation tableau de pions, visuels, ...). La lisibilité du code est ainsi optimisée.

Dans nos grandes fonctions, nous avons utilisé les switch/case accompagnés d'une boucle while. Ça permet de séparer les différentes étapes qui prennent place durant le tour d'un joueur. Dans le but de traiter les cas de manière organisée et lisible lors de la lecture du programme.

A ce jour, aucun bug n'existe du moins que l'on sache. Pour le menu, ne soyez pas étonné de passer du numéro 1 au 3. Le 2 était réservé pour un jeu contre l'ordinateur (partie légèrement commencée mais pas présente dans le code).

Pour la sauvegarde de la partie, le binôme a eu des problèmes. Nous travaillons de manières différentes, ceci a créé des petits soucis lors de la mise en commun. L'autre personne a dû réadapté le code pour pouvoir l'insérer. Désormais, la sauvegarde est opérationnelle donc fonctionnelle.

FONCTIONNALITÉ RÉALISÉ PAR LE SOFT

Cahier des charges de base	Oui
Cahier des charges extensions	Non
Cahier des charges pour les plus forts	Non (début d'un travail SQLite)

REPARTITION DES TACHES

L'ORGANISATION AU SEIN DU BINÔME

Au tout départ du projet, nous avons à plusieurs reprise discuté de l'organisation du programme. Ceci nous a permis d'avoir plus de visibilité.

Au niveau de la répartition, ça n'a pas été définie. Au fur et à mesure, chacune est aller à son rythme et à son niveau.

Clémence s'est occupée de la sauvegarde, des règles du jeu, de la mise en beauté et du menu.

Louise s'est occupée de coder le jeu.

Ceci est dû à la rapidité de chacune. Nous savons et avons vu ce que l'une et l'autre ont fait. Cependant, des problèmes d'insertions sont remontés ce qui a posé quelques dysfonctionnements.

EXTENSIONS DU PROJET

TRAVAIL SUR DES EXTENSIONS

Nous avons travaillé sur une sauvegarde sous SQLite. Nous avons commencé à faire des tests fonctionnels pour le projet.

Le but de son utilisation était de pouvoir jouer sur deux terminaux différents. Ce qu'il faut savoir lors des recherches est la suivante, la sauvegarde actuelle n'existe pas donc un autre moyen de sauvegarde a été créé en plus de celui présent dans le code.

De plus, nous avons aussi un peu travailler sur la création d'une intelligence artificielle mais l'idée a été abandonnée par manque de temps.

Nous avons intégré une fonctionnalité de : "Reprendre là où je m'étais arrêté" (ou "Pick Up where left"), afin de permettre à l'utilisateur de reprendre sa partie de manière fluide et sans interruption durant le jeu.

CONCLUSION

Le projet s'est révélé très enrichissant dans la mesure où il a constitué une base solide pour de nombreuses matières.

L'apprentissage à son rythme est aussi très révélateur sur notre assiduité. Nous avons pu le vérifier grâce à ce projet. Notre propre avancé a pu être validée mais aussi le fait de revoir les notions moins bien assimilées.

Une fois le projet terminé, nous nous sommes rendu compte que celui-ci nous a fait progresser et améliorer notre niveau dans le langage.

Sa complexité au niveau des cas possibles a rendu le projet encore plus intéressant. Nous avons dû réaliser une structure à la fois claire et complète. Ceci a été vraiment très captivant.

Nous aurions voulu intégrer la possibilité de jouer sur deux fenêtres en même temps. Malgré le manque de temps, nos recherches pourront nous servir dans l'avenir sur des projets extérieurs.

En définitive, ce projet nous a beaucoup aider sur la maîtrise du langage C. Cette expérience a été très complémentaire au cours et nous a permis de s'entraîner sur un projet concret.

JEU DE DAMES

ANNEXE (PHOTOS DU JEU)

MENU DU JEU

```
=====
Bienvenue dans Jeu de dames
=====

=====

    --- MENU ---
1 : Nouvelle partie humain Vs humain
3 : Charger une partie
4 : Regles du jeu
5 : Quitter

Faites votre choix : |
```

DEBUT DU JEU

```
Phase de : Sélection du pion

 0 1 2 3 4 5 6 7 8 9
0 - r - r - r - r - r
1 r - r - r - r - r -
2 - r - r - r - r - r
3 r - r - r - r - r -
4 - - - - - - - -
5 - - - - - - - -
6 - b - b - b - b - b
7 b - b - b - b - b -
8 - b - b - b - b - b
9 b - b - b - b - b -
```

Le joueur qui joue est le :1
-> rouge
Entrer la colonne du pion choisi:
|

APPARITION D'UNE DAME

```
Phase de : Sélection du pion

 0 1 2 3 4 5 6 7 8 9
0 - r - r - r - B - r
1 r - r - r - - - r -
2 - r - r - - - r - r
3 r - - - - - r -
4 - - - - - - - -
5 - - - b - - - -
6 - r - - - - - - -
7 b - b - b - - - b -
8 - b - b - b - b - b
9 b - b - b - b - b -
```

Le joueur qui joue est le :1
-> rouge
Entrer la colonne du pion choisi:
|

POUR LANCER LE JEU

Pour compiler:

Le fichier "makefile" a été créé pour que l'on puisse compiler le code sur Linux et/ou sur Windows.

Pour lancer :

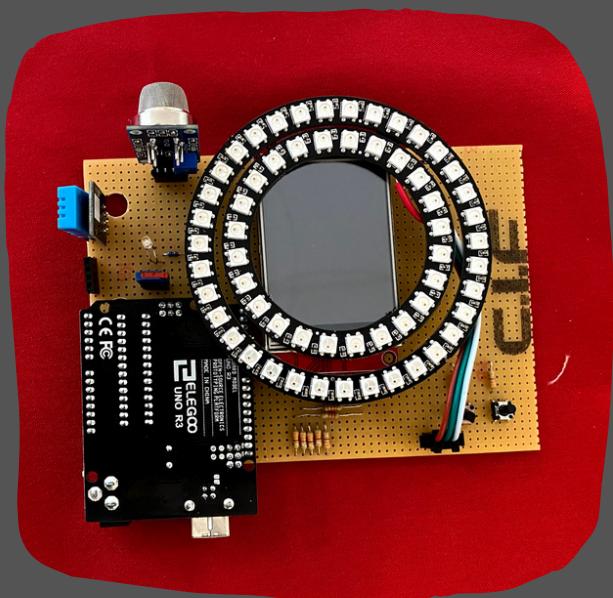
Soit vous recompilez CodeSource.c grâce à "makefile", soit vous lancer "Game.out" qui est déjà un fichier compilé de "CodeSource.c".

PROJET ARDUINO

STATION MÉTÉO PORTABLE

Matériel:

- Carte arduino uno
- Capteur MQ2
- Capteur DHT11
- Néopixel 24 leds et 35 leds
- 2 bouton-poussoir
- Ecran tft lcd
- Led
- Interrupteur à coulisse



But du projet:

Notre projet a pour but d'informer l'utilisateur sur la température, l'humidité et sur la qualité de l'air. Tout cela est donné en temps réel.

Ces données sont affichées sur l'écran. Nous avons aussi décidé grâce aux anneaux de montrer autrement les informations.

Pour la qualité de l'air nous avons également voulu informer selon les données, si l'air est de bonne, de moyenne ou de mauvaise qualité.

Les améliorations possibles:

Nous aurions aimé ajouter d'autres composants pour rendre le projet encore plus intéressant, par exemple, pouvoir remettre le BMP180, un capteur de luminosité, etc. Mais aussi faire un affichage plus élaboré ainsi qu'un boîtier.

Nous voulions aussi remettre notre machine à état de base qui fonctionnait sans le capteur de pression atmosphérique.

Problèmes rencontrés:

Le capteur BMP180 nous a posé problème pendant un long moment. En effet nous avons voulu l'intégré afin d'avoir la pression atmosphérique et l'altitude. Cependant, lorsque nous l'avons inclus, lors de son appel il nous faisait reset la carte uno. En réalité, le capteur tirait trop sur la carte. Nous avions également en amont, essayer le capteur et celui-ci fonctionnait. Après de nombreuses recherches, nous nous sommes rendu compte que le problème ne pouvait venir du code mais uniquement du composant. En le échangeant par un autre, le programme s'exécutait sans problème. Malheureusement il n'est pas possible de le prendre.

Ce que nous a apporté le projet:

Ce projet nous a permis d'utiliser de nouveaux composants. Leur différents fonctionnement sont enrichissants. A l'avenir, nous serons probablement confronté à de nouveaux composants et donc de voir comment approcher sereinement ce cas.