

Case Enigma Documentation

None

DOMINGUES PEDROSA Samuel

None

Table of contents

1. Case Enigma	3
1.1 Présentation du jeu	3
1.2 Niveau 2 : Scanner de paume	4
1.3 Niveau 3 : Code morse	5
2. Plannings	7
2.1 Prévisonnel	7
2.2 Réel	7
3. Cahier des charges: Case Enigma	8
3.1 Sujet	8
3.2 But	8
3.3 Spécifications	8
3.4 Restrictions	8
3.5 Environnement	8
3.6 Livrables	8
3.7 Rédditions	9
3.8 Planning	9
4. Journal de bord	10
4.1 2022.09.15	10
4.2 2022.09.22	10
4.3 2022.09.29	10
4.4 2022.10.06	10
4.5 2022.10.20	10
4.6 2022.11.03	10
4.7 2022.11.10	11
5. 2022.11.17	12
6. 2022.11.19	13
6.1 2022.11.20	13

1. Case Enigma

Case Enigma est un jeu de puzzle qui utilise le CrowPi pour ses énigmes.

Le jeu se présente comme une série d'écrans sur lequel est affiché l'énigme. Si le joueur se trompe 3 fois, il peut découvrir un indice pour cette énigme.

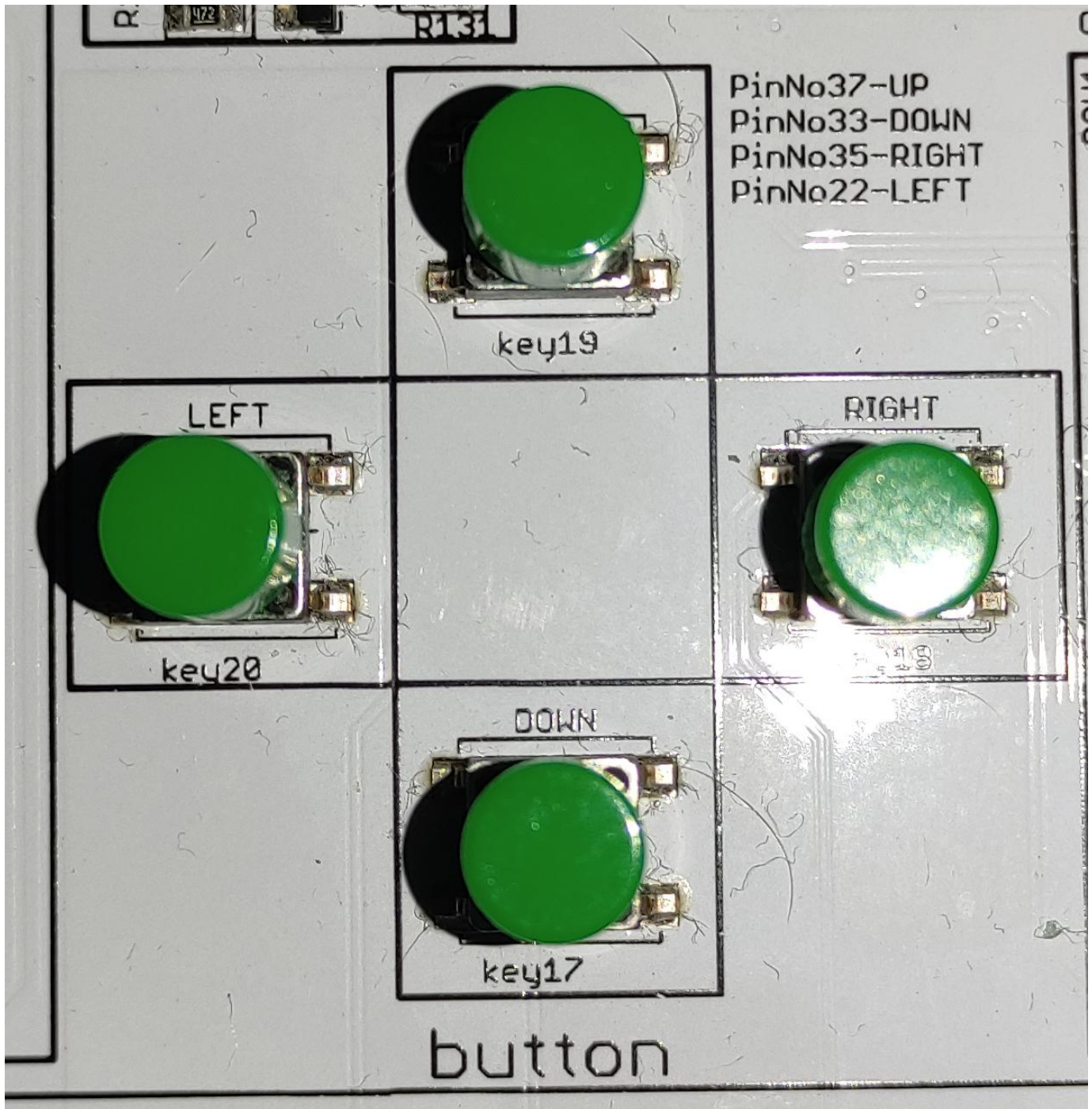
J'ai choisi cette idée car j'avais déjà vu le CrowPi par le passé et cette boîte m'a intrigué. J'ai donc pris l'occasion de pouvoir non seulement utiliser le CrowPi tout en améliorant mes compétences en Python et en gestion d'entrées/sorties en prévision de mon travail de diplôme

1.1 Présentation du jeu

1.1.1 Niveau 1 : Code au D-Pad

Cette énigme demande de faire un code à l'aide de la croix de bouton située en bas du CrowPi.

En interne, on regarde si on reçoit un input sur l'une des PINs du d-pad. On ajoute une lettre correspondant à la tentative de code, représentée par une chaîne, et on la compare au code. Si elle est fausse, on recommence.



J'ai choisi cette énigme parce que les boutons sont un input très basique, donc simple à utiliser quand on commence à utiliser les GPIO, et si on demande à quelqu'un de faire un code avec un d-pad, il y a de grandes chances qu'il fasse le Konami Code.

Solution: `↑ ↑ ↓ ↓ ← →` Explication de l'indice: Contra, le jeu qui a rendu populaire le Konami Code

1.2 Niveau 2 : Scanner de paume

Cette énigme "scanne" la paume de la main mais demande d'être à une distance précise

En réalité, le scanner est un capteur de distance à ultrason. On envoie un très court ultrason, on prend le temps, ce qu'on refait quand l'ultrason revient, et on soustrait les deux pour obtenir la distance.

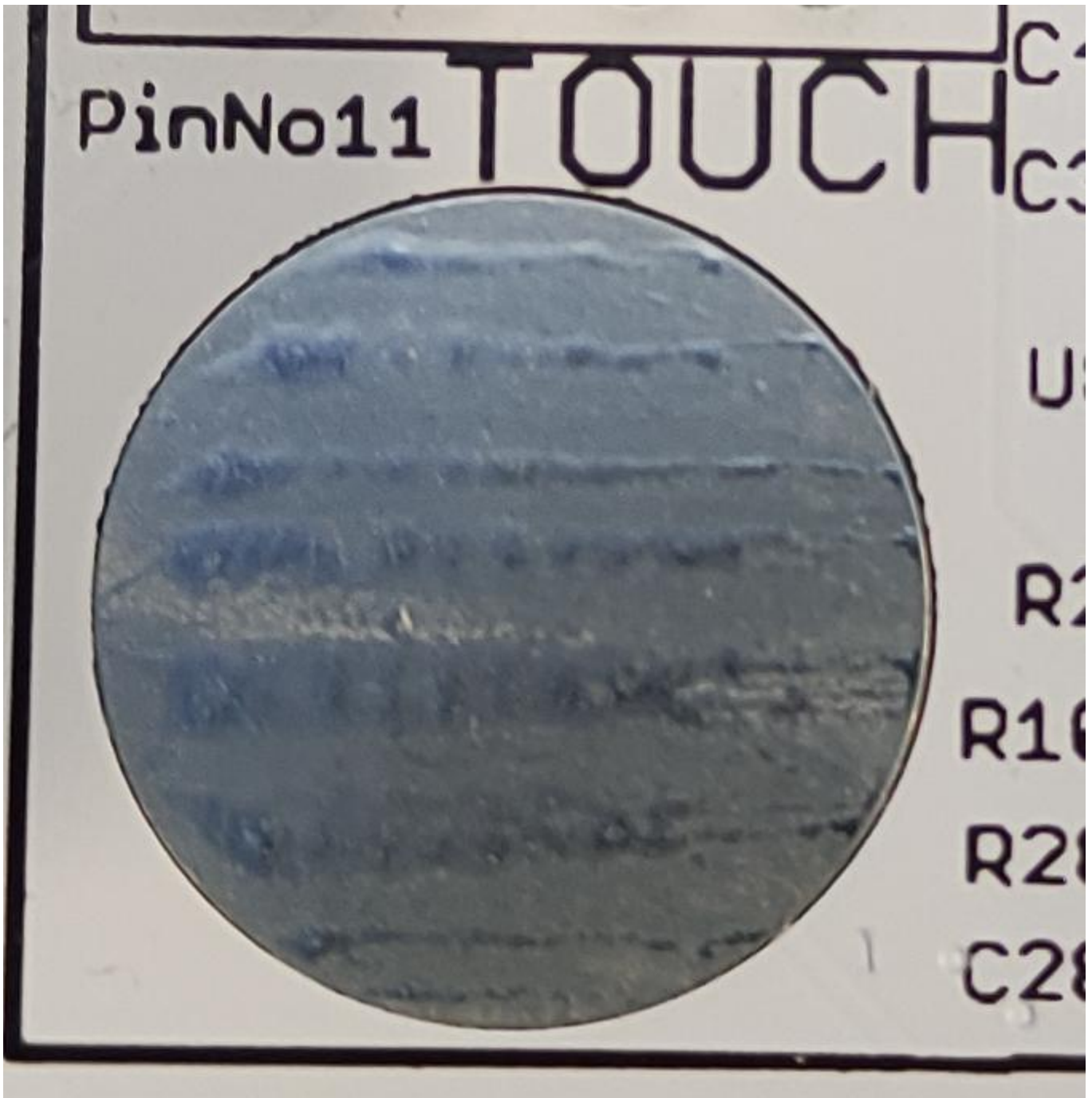


J'ai fait cette énigme car je trouve intéressant l'idée d'utiliser du son pour calculer des distances. L'idée d'en faire un scanner vient du fait que quand je testais le code, je mettais ma main à plat pour donner le plus de surface pour réfléchir les ultrasons. Solution: mettre 15 cm de distance au capteur Explication de l'énigme: 1 minute = 60 secondes, / 4 = 15 secondes donc 15 cm

1.3 Niveau 3 : Code morse

Cette énigme demande de faire un code Morse avec le capteur de toucher

En interne, cette énigme est très similaire à la première. Pour calculer le temps et déterminer le caractère, on ajoute un event listener à la PIN avec un callback à une fonction qui enregistre le temps pour quand on appuie et quand on relâche. Puis dans la fonction du niveau, on regrade le temps maintenu pour choisir le caractère.



J'ai choisi cette énigme car je voulais savoir comment faire pour tester l'état un input en Python et vu que j'avais déjà utilisé les boutons, j'ai eu l'idée d'utiliser le capteur de toucher et donc d'avoir l'idée de faire passer ça pour un test d'empreintes digitales.

Solution: .../---/... (SOS) Explication de l'indice: SOS est le code Morse le plus facile à faire (c'est pour ça que c'est SOS), donc si du morse est utilisé pour un mot de passe, ce serait équivalent à utiliser 1234

2. Plannings

2.1 Prévisonnel

	15/09/2022	22/09/2022	29/09/2022	06/10/2022	13/10/2022	20/10/2022	27/10/2022	03/11/2022	10/11/2022	17/11/2022
Documentation							Vacances			
Journal de bord										
Cahier des charges										
Plannification										
Mise en place environnement										
Famillisation avec Python										
Crésation des énigmes										
Poster										
Tests										
Préparation présentation										

2.2 Réel

	15/09/2022	22/09/2022	29/09/2022	06/10/2022	13/10/2022	20/10/2022	27/10/2022	03/11/2022	10/11/2022	17/11/2022
Documentation							Vacances			
Journal de bord										
Cahier des charges										
Plannification										
Mise en place environnement										
Famillisation avec Python										
Crésation des énigmes										
Poster										
Tests										
Préparation présentation										

3. Cahier des charges: Case Enigma

DOMINGUES PEDROSA Samuel

3.1 Sujet

Crée un jeu de puzzle avec un CrowPi

3.2 But

Le but du jeu est de résoudre une série d'énigmes en utilisant les divers composants d'un CrowPi, une mallette avec un écran et un circuit avec des composants utilisables avec un Raspberry Pi.

3.3 Spécifications

Le jeu se découpe en plusieurs énigmes qui devront utiliser un ou plusieurs composants du CrowPi. Si le joueur réussit l'énigme, il passe à la suivante jusqu'à la fin du jeu. S'il échoue, il recommence et à droite à un indice s'il recommence 3 fois. Le jeu commence dans un menu que l'on peut revenir à tout moment en appuyant sur un bouton dédié avec l'écran tactile et où l'on peut choisir une énigme (qui sont toutes bloquées de base sauf pour la première).

3.4 Restrictions

Pas tous les composants du CrowPi seront utilisés

3.5 Environnement

- Language : Python
- Environnement
- Windows 10
- Raspberry Pi OS
- Visual Studio Code
- Matériel :
- Raspberry Pi 4
- CrowPi
- Versionning : Git
- Documentation : MkDocs

3.6 Livrables

- Poster
- Documentation
- Journal de bord
- Code source du projet

3.7 Réditions

2022.10.13 : Rendu du poster

2022.10.23 : Rendu du rapport intermédiaire

2022.11.20 : Rendu final

2022.11.24 : Exposé oral

3.8 Planning

	15/09/2022	22/09/2022	29/09/2022	06/10/2022	13/10/2022	20/10/2022	27/10/2022	03/11/2022	10/11/2022	17/11/2022
Documentation										
Journal de bord										
Cahier des charges										
Plannification										
Mise en place environnement										
Famillisation avec Python										
Crésation des énigmes										
Poster										
Tests										
Préparation présentation										

Vacances

4. Journal de bord

4.1 2022.09.15

Vu que je ne fais plus le projet VR avec Rami, je dois refaire un cahier des charges pour mon projet solo.

Je prépare déjà Raspberry Pi Imager pour la semaine prochaine.

4.2 2022.09.22

J'ai installé une installation clean de Raspberry Pi OS. Je vais surtout regarder quelques tutos pour voir comment interagir avec le CrowPi et commencer à faire le poster.

J'ai essayé le code de démo du LCD mais on me dit que je n'ai pas la librairie et j'en installe une mais, soit j'ai installé la mauvaise, soit la librairie a beaucoup changé, mais je n'arrive pas à le faire fonctionner.

4.3 2022.09.29

Ce qui est bizarre c'est que les pins de la croix marquée dans le circuit du CrowPi et celles qui faut utiliser dans le programme sont différentes. Par exemple, pour le bouton haut ça dit qu'il faut utiliser la PIN 37 mais dans le programme c'est la PIN 26.

Il y a le même problème que chez moi ou pygame sur Linux à l'air de crasher quand on quitte.

Problème réglé, c'est que je reset le GPIO et après j'en utilise une.

Après avoir vu la doc, RPi.GPIO.BCM met les Pin dans un ordre différent.

J'ai réussi à faire interagir les GPIO et pygame

4.4 2022.10.06

J'ai pu faire en sorte qu'un bouton ne puisse pas faire constamment une action quand un bouton est maintenu.

4.5 2022.10.20

J'ai décidé de mettre le code de l'énigme que j'ai fait dans une fonction à part et en laissant la variable qui contient la tentative dehors, mais ça me retourne une erreur qui me dit que la variable est référencée avant d'être assignée. Ce qui est fait exprès.

Je pense savoir où est le problème, j'ai passé la variable en local mais maintenant une autre variable qui sert à empêcher le code de prendre en compte le bouton s'il est maintenu revoit la même erreur. Ce qui veut dire que le code ignore les variables globales.

J'ai trouvé la solution. Je devais juste déclarer que ces variables étaient globales au début de la fonction.

J'ai essayé d'exécuter le script de démo de la matrice LED, mais même après avoir installé les librairies qui faut, il n'arrive pas à utiliser le SPI pour communiquer avec la matrice.

4.6 2022.11.03

Aujourd'hui mon objectif est de créer une énigme où il faut crier au bon volume.

En regardant j'ai vu qu'à partir de Python 3.10, le langage possède un équivalent au switch case avec match, mais la version de Raspberry Pi OS est encore en 3.9.2, donc je vais plutôt utiliser les elif en cascade.

J'abandonne l'idée du micro car les GPIO ne sont que digitale. A la place j'aurais une énigme qui utilise le capteur de distance.



J'ai pu afficher du texte avec pygame. C'est facile mais il y a un peu de boilerplate code. Il faut créer un objet pour la police, puis pour le texte et un rectangle pour le texte.

```
font = pygame.font.SysFont('freesansbold.ttf', 32) # Initialiser la police
text = font.render('Vies : 3', True, (0, 0, 0)) # Créer l'objet texte
text_rect = text.get_rect() # Obtenir la boîte pour afficher le texte
screen.blit(text, text_rect) # Afficher le texte
```

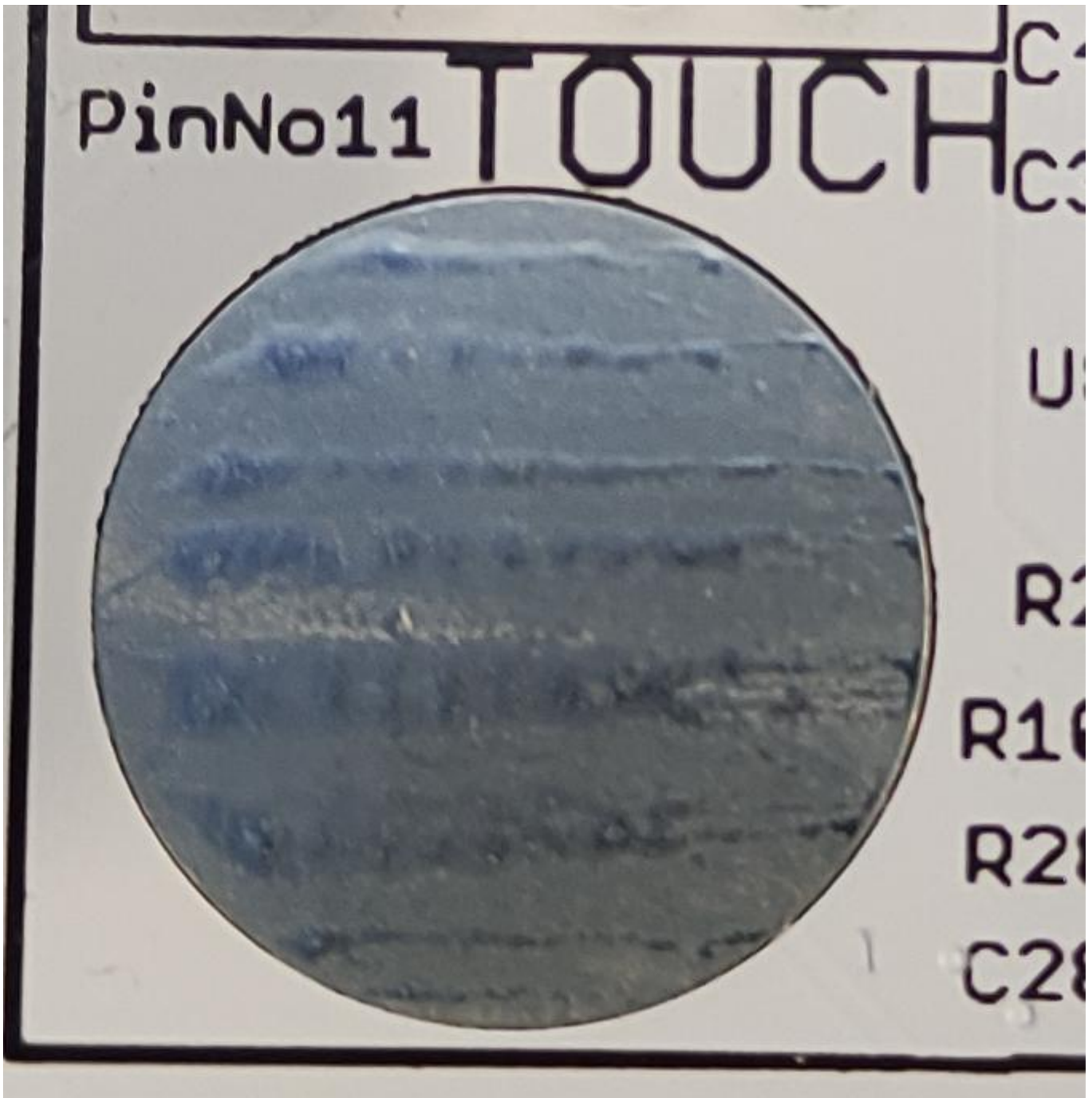
En essayant de push les changements d'aujourd'hui je me rends compte que j'ai oublié de pull avant de commencer et j'ai un conflit que j'ai besoin de résoudre.

4.7 2022.11.10

Honnêtement résoudre un conflit avec Vim était plutôt simple.

Aujourd'hui, mon objectif est de créer une troisième voir une quatrième énigme.

Pour ma troisième énigme, je pensais à devoir écrire un message en Morse avec le détecteur de toucher.



J'ai un peu modifié le code de la deuxième énigme pour affiche un vague message qui dit si on est trop haut ou trop bas au lieu de la distance elle-même.

5. 2022.11.17

J'ai un peu regarder comment calculer un intervalle de temps qu'un bouton est pressé, et quand je rajoute un bool pour ne pas inonder la console, ça me met des résultats incorrectes.

J'ai mis le code dans un callback et ça fonctionne

6. 2022.11.19

Vu que je dois vérifier deux chaînes de la même manière, j'ai mis le code dans une fonction séparée.

Pour gérer le callback, j'ai mis la variable en globale que je vérifie pour voir si la valeur est supérieure à 0, puis je regarde la durée pour déterminer si c'est un cours (.) ou un long (-)

Quand je fais un long, il est reconnu mais quand je fais un cours, il ne se passe rien.

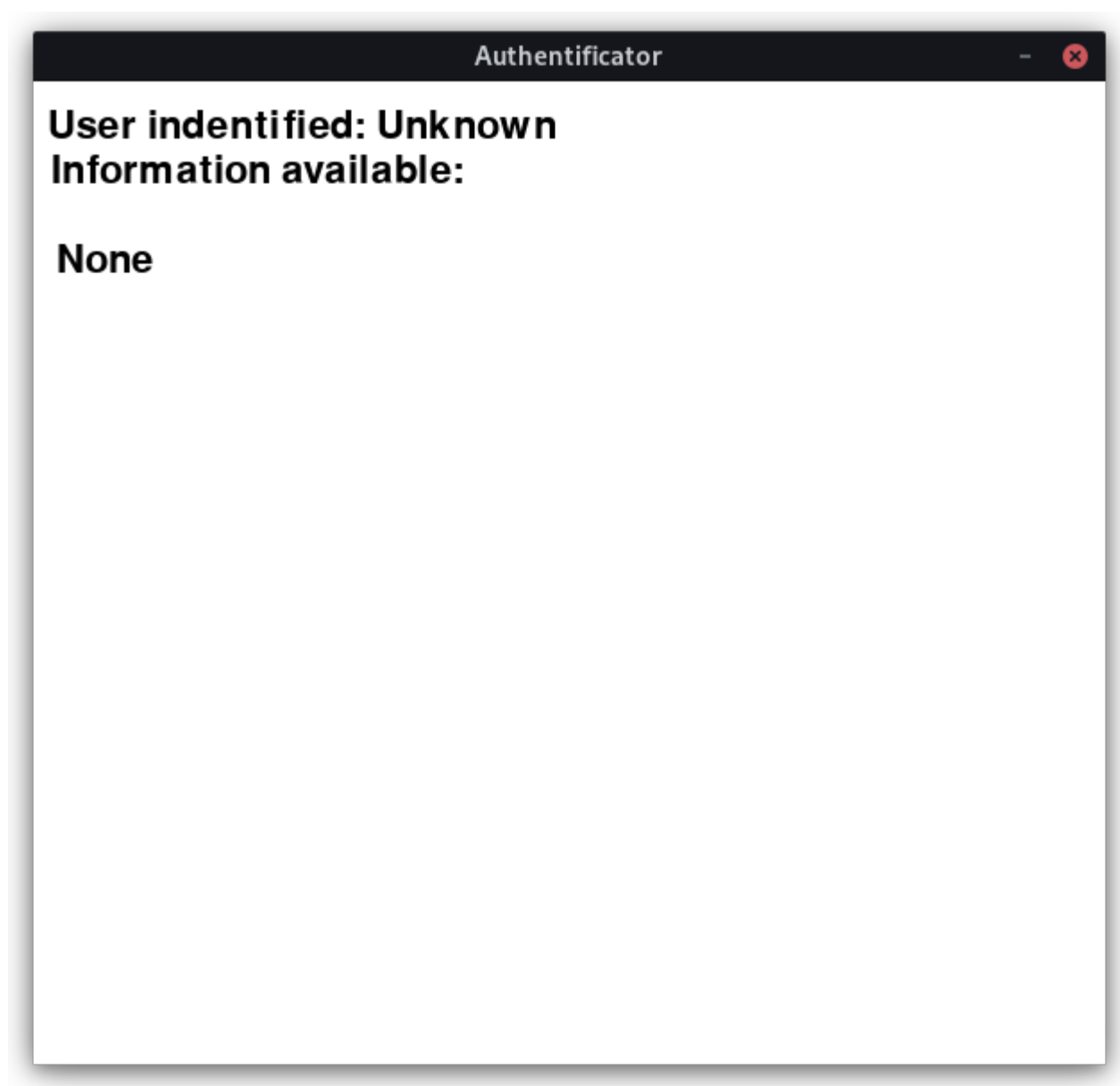
J'avais mis un temps pas assez intuitif pour un cours, pour finir, maintenir ~200 ms est un appui court, et ~ 600 ms est un appui long.

6.1 2022.11.20

Je vais me concentrer sur l'interface, la qualité du code et la documentation car je ne pense pas pouvoir faire tout ça et une nouvelle énigme.

J'ai ajouté un texte pour donner le contexte de l'énigme et un indice si on échoue trop de fois.

J'ai ajouté un écran de fin.



Je viens de me rendre compte que chaque lignes ou j'appelle `get_rect()` se finies par un ;