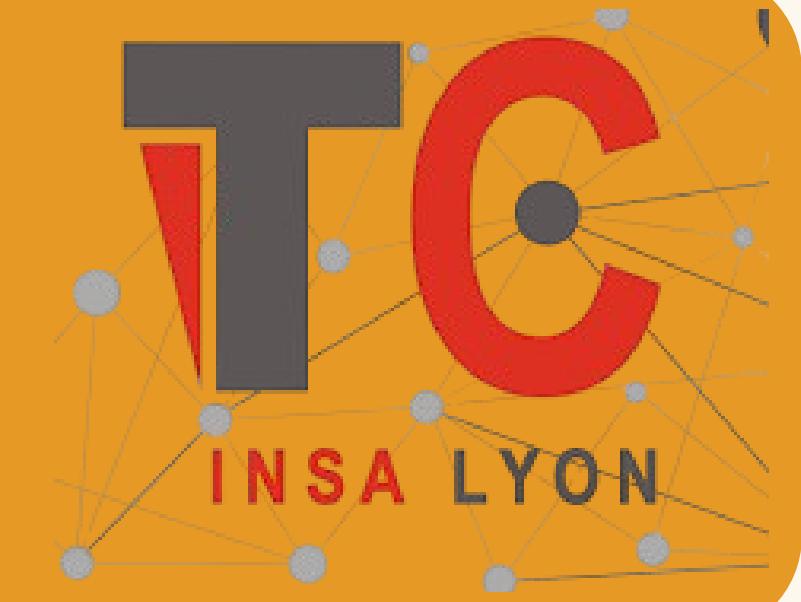


Shout 2 Play

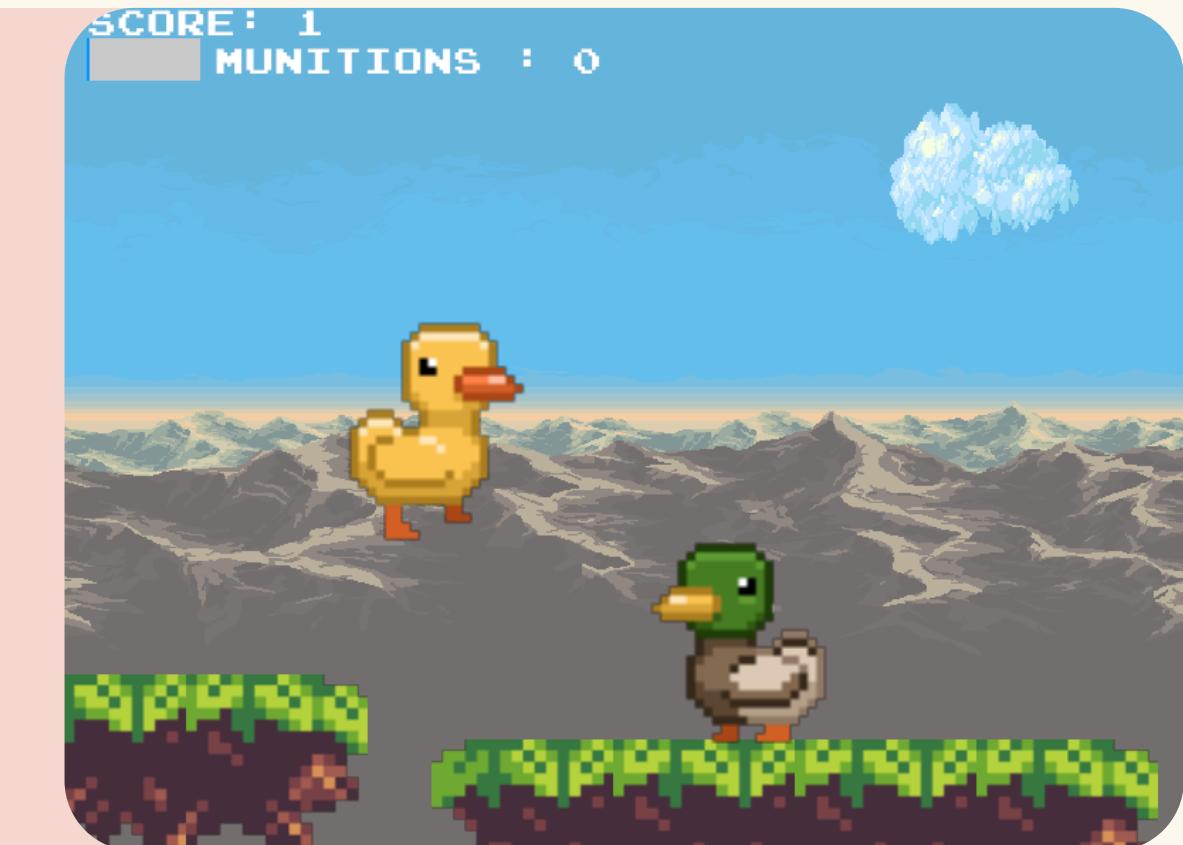
Hurler, c'est gagner !

Département Télécommunications
Services & Usages - Projet SON 2025
William Lawansch
Mathéo Daumas

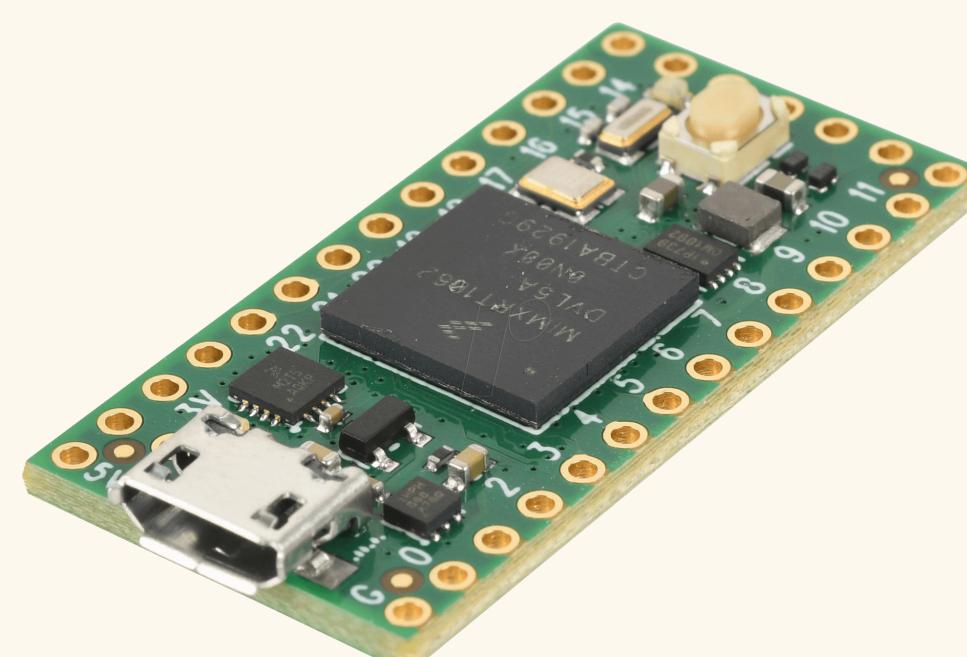


Présentation du projet

Shout 2 Play combine traitement du signal audio et contrôle de jeu vidéo.
Ce système capture et analyse la voix et les consignes de l'utilisateur grâce aux différents composants, puis les envoie en Serial. Un programme python sur PC interprète les données reçues pour contrôler un jeu de plateforme.

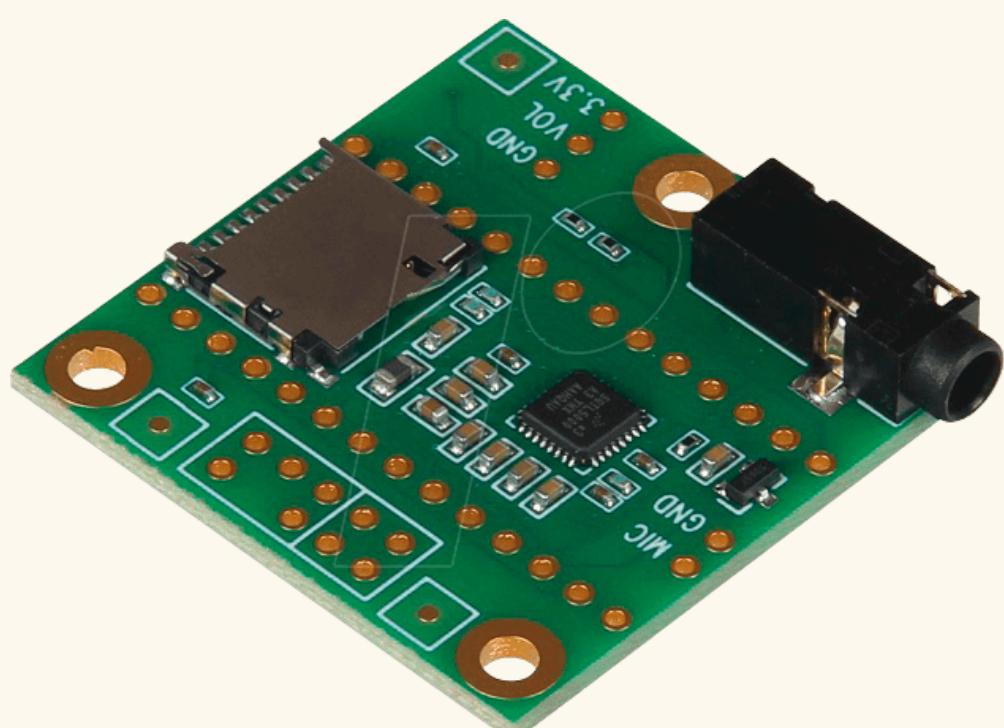


Cartes électroniques



Teensy 4.0

Audio Shield
SGTL5000



Autres composants

Microphone



Potentiomètres

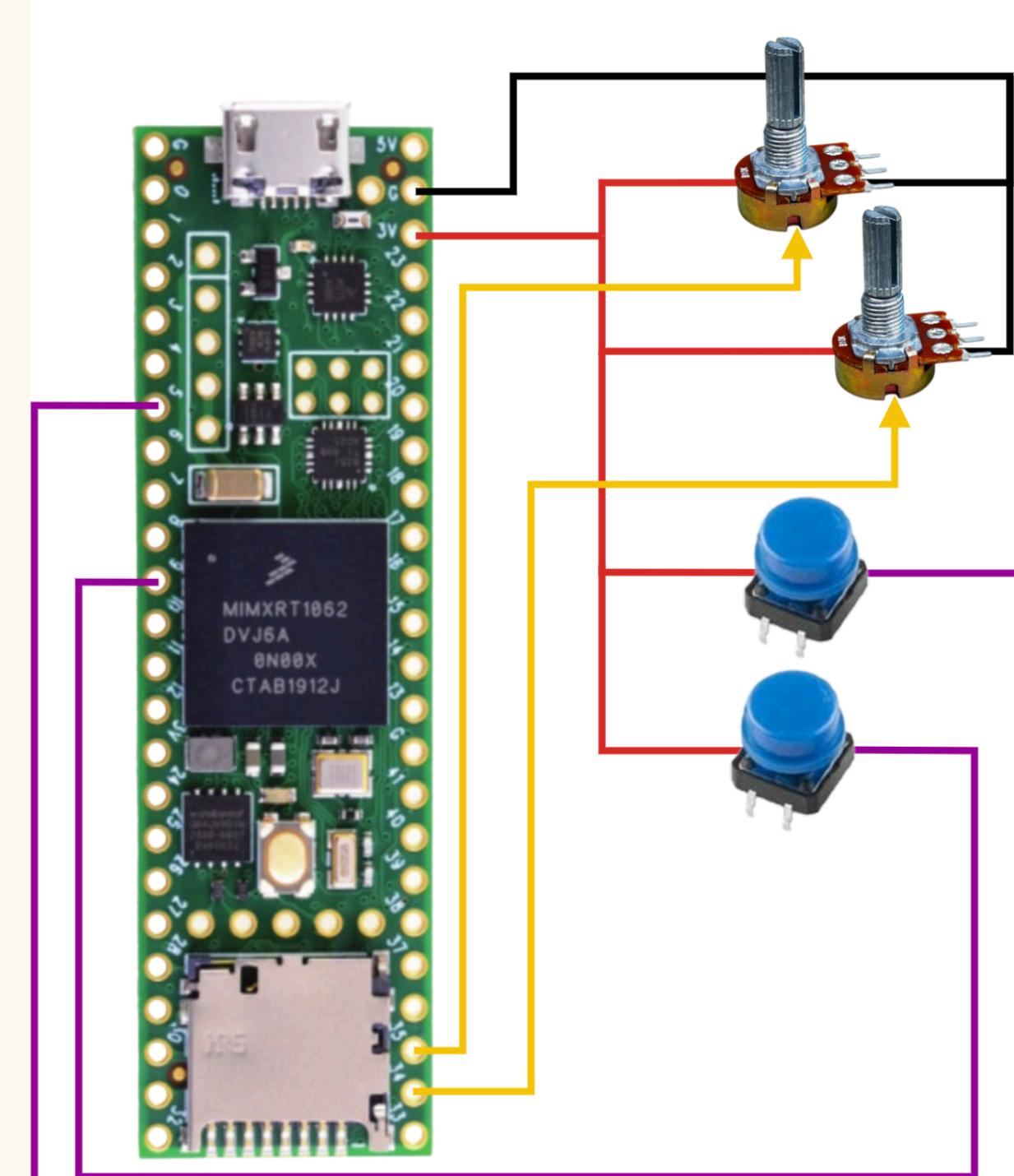


Boutons poussoir



Casque audio

Schéma électrique



1 Acquisition & Traitement

- Acquisition de la consigne de contrôle audio et exécution
- Échantillonnage du signal audio via le microphone par une FFT pour déduire la fréquence avec la plus grande contribution ainsi que la magnitude moyenne du signal échantillonné
- Relevé des valeurs potentiomètres et des boutons
- Changement d'échelle pour l'interprétation des valeurs des potentiomètres
- Fonction de lancement Wav en fonction des consignes reçues par SERIAL.
- Conversion de la magnitude en dB relatifs

SERIAL

2 Encodage & Transmission

- Structuration des états des composants au format JavaScript Object Notation (JSON)
- Encodage de la chaîne JSON en base64
- Envoi sur un lien en série (Baudrate de 115200 bits par seconde)

SERIAL

4 Évents & Audio

- Détection de collision du personnage avec les plateformes : réinitialisation des paramétrages de saut
- Génération des munitions si le joueur appuie sur le bouton de tir
- Mise en pause si le joueur presse le bouton pause
- Détection de collision d'une munition avec un ennemi
- Déterminer les commandes de contrôle de son à envoyer au Teensy



3 Décodage et exécution

Exploitation des données

- Gain + Threshold potentiomètre → Hauteur du saut
- Fréquence + Facteur potentiomètre → Vitesse de recharge des munitions
- Bouton de tir → Déclenchement du tir
- Bouton de pause → Mise en pause du jeu

Boucle de jeu

- Acquisition des données dans un Thread dédié
- Interprétation du JSON :
 - Recharger les munitions (en fonction de la vitesse)
 - Pause
 - Tir
 - Seuil de bruit ambiant
- Mise à jour des éléments du jeu (positions des éléments graphiques, des physiques, des parallaxes, des tilesets)
- Construction visuelle des éléments puis affichage sur le moniteur
- En cas de saut, suivi d'un PID (initialisé par la hauteur du saut)
- Création procédurale de plateformes et d'ennemis

Axes de développement possibles

- Déterminer automatiquement le seuil adapté
- Travailler sur une mécanique de saut plus avancée
- Optimisation de la complexité des algorithmes
- Présentation du breadboard sous forme d'un micro
- Ajout de potentiomètres pour gérer les volumes audio