

# Apprentissage automatique pour Flappy Bird

## De la collecte de données à l'inférence native

Projet IA - Ivo TALVET, Timothée M'BASSIDJE

2025-12-15

# Objectifs

- Reproduire un comportement humain sur Flappy Bird
- Apprendre une politique de saut à partir de démonstrations
- Déployer un modèle léger, embarqué, sans dépendances
- Intégration temps réel dans Godot via une librairie C

# Collecte des démonstrations

- Enregistrement des parties humaines en CSV
- Une ligne par frame de jeu
- Données en unités de jeu (pas normalisées)

Exemple de colonnes :

- hauteur du joueur
- vitesse verticale
- distance au prochain obstacle
- position de l'ouverture
- action : saut ou non

# Features d'entrée du modèle

- Vitesse verticale
- Distance au plafond
- Distance horizontale à l'obstacle
- Distance au bas de l'ouverture
- Distance au haut de l'ouverture
- Temps avant impact (TTI)

## Exclusion volontaire :

- Nombre de portes franchies (passes)

# MLP (Multi-Layer Perceptron)

- Réseau entièrement connecté
- Deux couches cachées
- Activation  $\tanh$
- Sortie sigmoïde (probabilité de saut)

Entrée →  $\tanh$  →  $\tanh$  →  $\sigma$

# Pipeline d'entraînement

- Chargement de tous les CSV
- Équilibrage des classes (saut / non-saut)
- Normalisation par z-score
- Descente de gradient batch
- Régularisation L2

Fonction de perte :

$$\mathcal{L} = \text{Binary Cross Entropy}$$

## Export vers un format natif

- Poids exportés dans un fichier texte
- Moyennes et écarts-types inclus
- Format lisible par un chargeur C

Structure :

- En-tête MLP
- Statistiques de normalisation
- Matrices de poids
- Biais

- Chargement dynamique du modèle
- Calcul en double précision
- Aucune allocation pendant l'inférence
- Décision binaire par seuil

Avantages :

- Très faible latence
- Portable
- Déterministe

- Librairie C chargée dynamiquement
- Appel via P/Invoke en C#
- Le modèle décide du saut à chaque frame

Possibilité :

- Mode humain
- Mode Play like me

# Limites du MLP

- Sensible au bruit
- Décision trop globale
- Difficulté sur des cas rares
- Apprentissage de raccourcis si mauvaise feature

Exemple :

- Utiliser passes conduit à du shortcut learning

# Réseau à fonctions de base radiale (RBF)

- Activation locale (gaussienne)
- Centres dans l'espace d'état
- Décisions plus stables

$$\phi(x) = \exp\left(-\frac{\|x - c\|^2}{2\sigma^2}\right)$$

Avantages :

- Meilleure généralisation locale
- Très adapté à Flappy Bird

# Conclusion

- Pipeline complet : données → modèle → jeu
- Modèle simple mais maîtrisé
- Forte compréhension des mécanismes internes
- Base solide pour des architectures plus avancées

Message clé :

- Le modèle apprend à survivre
- Le moteur impose les règles du jeu