

Automation game

Rapport

Description du projet

Objectifs

Créer un jeu d'automation (comme Factorio, satisfactory, shapez, etc.) dans le thème de la musique. Le joueur doit livrer des notes individuelles, puis des accords et enfin des suites de notes pour créer des courtes mélodies.

Exigences fonctionnelles

-

Exigences non-fonctionnelles

-

Architecture préliminaire

- Rédaction rapport : Typst, et l'extension VSCode tinymist Typst
- Création mockups et schémas : Figma
- Game Engine : Godot .Net 4.3.0.0
- IDE : JetBrains Rider 2024.2

Mockups / Landing page préliminaire

(Figma ici)

Description des choix techniques

- Typst pour de plus grandes possibilités de mise en page
- Figma pour une collaboration plus facile dans un premier temps
- Godot pour sa nature open source et sa montée en popularité très récente.
 - Pour la facilité d'intégration dans un pipeline non CI/CD à l'opposé de Unity, qui était l'autre solution envisagée pour un jeu à petite échelle en 2D. PixiJS était également une option mais nous avons préféré Godot en raison de sa popularité.
 - Nous avons décidé de partir sur .Net en raison de l'utilité de C# pour le futur, de notre familiarité avec Java qui partage beaucoup de similarité, et la réticence d'apprendre un langage unique à Godot tel que GDScript.
 - 2D
 - Comptabilité
- Rider pour l'habitude de l'équipe à travailler avec les produits JetBrains et sa comptabilité avec Godot.

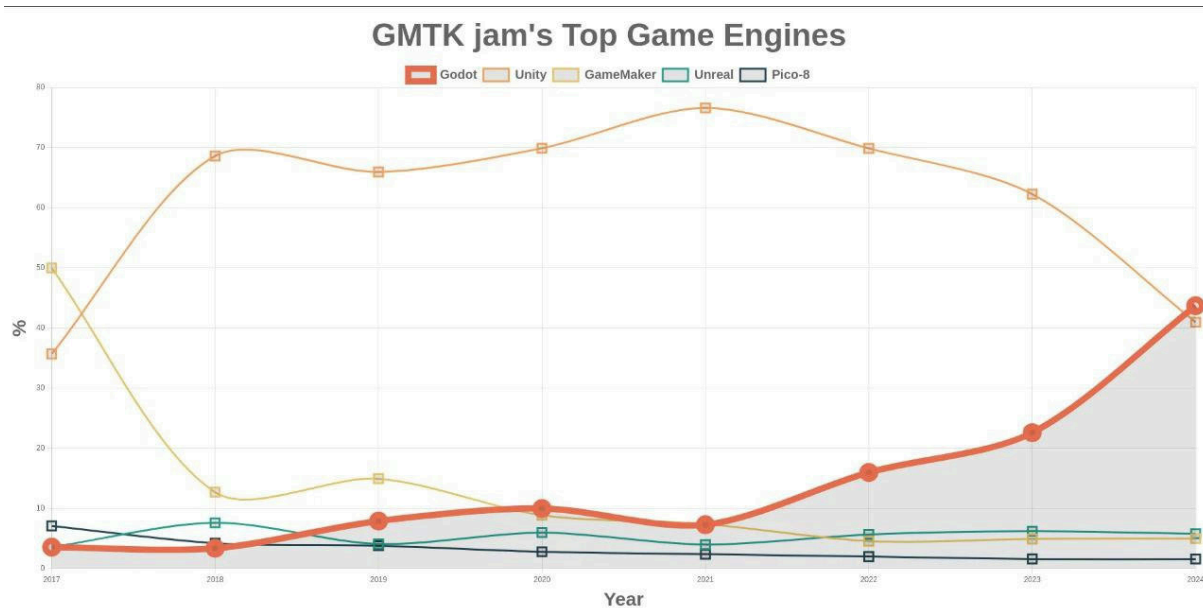


Figure 1: pourcentage de game engines utilisés lors de la GMTK game jam

Description du processus de travail

- Organisation du git: branche main avec le workflow pour créer un build
- Les PR ouvertes pour main lancent des teste unitaires / de features qui doivent passer pour pouvoir merge la PR
- Branches individuelles pour les features suffixées avec les initiales de la personne en charge
- Branche pages pour la landing page