

Player Insight – ML-Powered Game Analytics

🌟 الملخص التنفيذي

بهدف تحليل سلوك اللاعبين من خلال Unreal Engine في هذا المشروع، قمت بتطوير لعبة استكشاف بسيطة باستخدام جمع بياناتهم أثناء اللعب وتطبيق خوارزميات تعلم آلي بسيطة على هذه البيانات. بعد ذلك، صممت لوحة تحكم لعرض النتائج، مع تقديم توصيات تصميم مبنية على JavaScript و CSS و HTML تفاعلية باستخدام (Dashboard) التحليل. هذا المشروع يُظهر كيف يمكن لتقنيات الذكاء الاصطناعي أن تسهم في تحسين تجربة المستخدم من خلال الفهم الأفضل لأنماط اللعب وسلوك اللاعبين.

📌 المقدمة

صناعة الألعاب لم تعد تعتمد فقط على التصميم الفني والبرمجة، بل أصبحت تعتمد أيضًا على تحليل البيانات لفهم اللاعبين واتخاذ قرارات تصميم مدروسة. هو محاولة أولى لبناء نظام بسيط يجمع بيانات من اللعبة، يحللها باستخدام خوارزميات تعلم "Player Insight" مشروع آلي، ويعرض نتائجها على شكل لوحة تحكم تساعد مطوري اللعبة على فهم الأداء العام للاعبين.

🔧 المنهجية

🎮 تصميم اللعبة

- **Unreal Engine 5.** صممت لعبة ثلاثية الأبعاد للاستكشاف باستخدام
- لا تحتوي على ألغاز أو أعداء، بل تركز فقط على جمع العناصر واستكشاف البيئة.
- اللاعب لديه عدد معين من الحركات ووقت محدود للوصول إلى نهاية المشهد.

📊 جمع البيانات

- يتم تسجيل بيانات كل جلسة لعب
 - رقم اللاعب

- عدد الحركات
 - الزمن المستغرق
 - النجاح أو الفشل
 - توقيت الجلسة
- داخل المتصفح **LocalStorage** لتخزين البيانات في **JavaScript** استخدمت

لوحة التحكم

- **HTML و CSS** أنشأت صفحة ويب تعرض البيانات في جدول تفاعلي باستخدام
- يقوم بـ **JavaScript** كتبت كود:
 - حساب المتوسطات
 - **K-Means** تصنيف اللاعبين باستخدام خوارزمية
 - عرض توقعات (منطقية) لنجاح اللاعب بناءً على عدد الحركات والزمن

خوارزميات تعلم الآلة

- مكتوبة يدويًا لتقسيم اللاعبين إلى مجموعتين **K-Means** استخدمت خوارزمية:
 - مجموعة سريعة
 - مجموعة بطيئة
- لتوقع نجاح اللاعب (**Threshold Logic**) أضفت شرطًا بسيطًا:
 - إذا كان عدد الحركات > 50 والوقت > 60 ثانية \rightarrow متوقع النجاح

النتائج والتحليل

- "عدد من اللاعبين أظهروا أداءً جيّدًا وتم تصنيفهم ضمن "المجموعة السريعة".
 - التوقعات التي ظهرت في لوحة التحكم كانت منطقية ومطابقة لنتائج النجاح في اللعبة.
 - التوصيات التي أظهرها النظام كانت مفيدة، مثل:
 - اللعبة صعبة على معظم اللاعبين "إذا كانت التوقعات قليلة".
 - الأداء ممتاز "إذا نجح اللاعبون بسهولة".
 - هذا النوع من التحليل يساعد على اتخاذ قرارات تصميم أفضل في المستقبل.
-

⚙️ التحديات التي واجهتني

- لم أستخدم قاعدة بيانات خارجية أو سيرفر حقيقي، لذلك واجهت صعوبة في ربط اللعبة بالتحليلات.
 - وهو حل مؤقت ولكنه مناسب لحجم المشروع، LocalStorage اضطررت لاستخدام.
 - Scikit-learn يدويًا كان تحديًا، خاصة بدون استخدام مكتبات تعلم آلي مثل K-Means كتابة خوارزمية.
 - الواجهة البرمجية للوحة التحكم استغرقت وقتًا كبيرًا لضبط التنسيق، الألوان، والنتائج بشكل واضح ومفهوم.
-

🎮 التحسينات المستقبلية

- لحفظ البيانات تلقائيًا من داخل اللعبة SQLite أو Firebase استخدام قاعدة بيانات حقيقية مثل.
 - لنتائج أدق scikit-learn واستخدام مكتبات مثل Python نقل جزء تعلم الآلة إلى.
 - إضافة المزيد من المقاييس مثل عدد المحاولات، الأماكن التي توقف فيها اللاعب، وغيرها.
 - أثناء اللعب في المستقبل (real-time) دمج التحليلات بشكل حي.
 - تحسين تصميم واجهة المستخدم لتكون أكثر تفاعلية وسهولة في التنقل بين البيانات.
-

✓ الخاتمة

هذا المشروع أعطاني تجربة عملية في دمج الذكاء الاصطناعي مع تصميم الألعاب. رغم أن اللعبة بسيطة والبيانات محدودة، إلا أن النتائج كانت مفيدة وفتحت لي آفاقاً جديدة للتطوير في المستقبل. تعلّمت كيفية جمع البيانات، تحليلها، وعرضها بطريقة واضحة تساعد في اتخاذ قرارات تصميمية أفضل. كما زاد هذا المشروع من ثقّتي في استخدام أدوات الويب والبرمجة لتوسيع إمكانيات الألعاب التي أعمل عليها.

المراجع

- Coursera – Machine Learning by Andrew Ng
- MDN Web Docs – JavaScript & LocalStorage
- الرسمية Unreal Engine وثائق
- التعليمية K-Means مصادر خوارزمية
- YouTube tutorials on Dashboards and Game Analytics