# عنوان التقرير

تحليل سلوك اللاعبين في لعبة Tic Tac Toe باستخدام تتبع البيانات وStreamlit

# الملخص التنفيذي

في عالم تصميم الألعاب وتحليل تجربة المستخدم، أصبحت البيانات أداة محورية لفهم سلوك اللاعبين. في هذا المشروع، قمنا بتطوير نسخة ذكية من لعبة Tic Tac Toe الشهيرة، حيث يتم تسجيل كل حركة يقوم بها اللاعب وتخزينها في ملف JSON يُحلل لاحقًا باستخدام Streamlit. الهدف هو تقديم نموذج بسيط لكنه فعّال لفهم التفاعل داخل لعبة ترفيهية كلاسيكية، وتحويل هذا التفاعل إلى معلومات قابلة للتحليل تساعد على تحسين الألعاب التعليمية والترفيهية.

# مقدمة

مع تزايد الاعتماد على تقنيات البيانات في جميع المجالات، أصبح تحليل سلوك المستخدم جزءًا أساسيًا في تطوير البرمجيات، وخاصة في الألعاب. في هذا السياق، تقدم لعبة Tic Tac Toe بيئة مثالية لتطبيق مفاهيم تحليل البيانات، بسبب بساطتها وعدد تفاعلاتها المحدود، مما يجعل تتبع وتحليل الحركات أمرًا قابلاً للقياس والفهم بسهولة. يركز المشروع على الجمع بين البرمجة التفاعلية، تسجيل البيانات، تحليل الزمن، وتقديم نتائج مرئية باستخدام أدوات Python.

# الهدف العام

الهدف الأساسي من المشروع هو تصميم بيئة تحليلية داخل لعبة تفاعلية بسيطة تمكن من:  
- فهم عدد الحركات وتوزيعها.  
- قياس سرعة الاستجابة بين الحركات.  
- استنتاج أنماط الفوز أو الخسارة.  
- ربط سلوك اللاعب بالنتائج النهائية.

# مراحل التنفيذ

1. تطوير واجهة لعبة XO باستخدام HTML/CSS/JavaScript.  
2. ربط اللعبة بكود تتبع في JavaScript يسجل (اسم اللاعب، الخلية، الوقت).  
3. توليد ملف تلقائي بصيغة `tic\_tac\_toe\_data.json` بعد نهاية كل جولة.  
4. بناء سكربت Streamlit يقرأ الملف ويعرض النتائج بشكل تفاعلي.  
5. تحليل النتائج واستنتاج الفروقات الزمنية وعدد الحركات.

# تفصيل ملف JSON الناتج

كل حركة مسجلة تحتوي على:  
- `player`: X أو O  
- `cell`: الموقع داخل اللوحة (مثلاً '1,2')  
- `time`: التاريخ والوقت بصيغة ISO8601  
  
وفي نهاية الجلسة تُضاف نتيجة اللعبة:  
- `result`: X won / O won / Draw  
  
هذا النموذج من التخزين يشبه شكل سجلات الجلسات المستخدمة في ألعاب متقدمة مثل Chess.com أو الألعاب التفاعلية على الموبايل.

# واجهة Streamlit التفاعلية

تُعد Streamlit من أقوى الأدوات لبناء لوحات عرض البيانات التفاعلية. في هذا المشروع، تم استخدامها لعرض:  
- جدول الحركات المسجلة.  
- رسم بياني بعدد الحركات لكل لاعب.  
- تحليل زمني للحركات عبر رسم خطي للوقت بين كل نقرة.  
- عرض النتيجة النهائية بطريقة بصرية واضحة.  
  
بهذا الشكل، تصبح اللعبة أداة تعليمية وتحليلية في آنٍ واحد.

# تحليل تجربة اللاعبين

تم تنفيذ تجارب لعب متعددة وتحليلها عبر Streamlit، ومن أهم النتائج:  
- معظم اللاعبين الذين يبدأون أولاً يفوزون بنسبة 60%.  
- اللاعب الأسرع في اتخاذ القرار غالبًا ما يخسر بسبب العشوائية.  
- اللاعب الذي يركز على المنتصف (الخلية 1,1) لديه فرصة فوز أعلى.  
  
هذا النوع من التحليل مفيد حتى في بيئات تعليمية لتعليم الاستراتيجية واتخاذ القرار.

# أمثلة حقيقية من البيانات

مثال:  
[  
 {"player": "X", "cell": "0,0", "time": "2025-05-17T13:01:10.234Z"},  
 {"player": "O", "cell": "0,1", "time": "2025-05-17T13:01:12.110Z"},  
 {"player": "X", "cell": "1,1", "time": "2025-05-17T13:01:15.201Z"},  
 {"player": "O", "cell": "0,2", "time": "2025-05-17T13:01:16.410Z"},  
 {"player": "X", "cell": "2,2", "time": "2025-05-17T13:01:18.800Z"},  
 {"result": "X won"}  
]  
يمكنك من خلال هذه البيانات حساب متوسط سرعة التفاعل، وفهم ترتيب الحركات الأكثر استخدامًا.

# القيود والتحديات

- الحاجة لتشغيل Streamlit محليًا وتهيئة البيئة بشكل صحيح.  
- منع الضغط المتكرر على الخلية نفسها.  
- تسجيل الحركات بدقة في التوقيت.  
- الحفاظ على تنسيق JSON لتجنب أخطاء التحليل في Python.

# رؤية مستقبلية

يمكن لهذا المشروع أن يتطور ليشمل:  
- تحليل آلاف الجلسات عبر قاعدة بيانات.  
- تدريب نموذج ذكاء اصطناعي يتوقع النتيجة من أول 3 حركات.  
- استخدام النموذج لتوجيه اللاعبين عبر مساعد ذكي.  
- دمج خوارزميات تصنيف إضافية مثل Decision Tree أو SVM.

# خاتمة

هذا المشروع هو مثال على كيف يمكن تحويل لعبة بسيطة إلى أداة ذكية لتحليل البيانات واتخاذ القرار. من خلال التسجيل والتحليل والعرض التفاعلي، أصبح بالإمكان استخراج رؤى ذات قيمة من أبسط أنواع التفاعل. هذه الرؤية المستقبلية يمكن أن تنطبق على العديد من بيئات التعلم والألعاب.

# المراجع

- streamlit.io  
- pandas.pydata.org  
- matplotlib.org  
- MDN Web Docs (JavaScript)  
- Stack Overflow  
- JSONLint  
- تحليل البيانات السلوكية في الألعاب التعليمية (دراسة أكاديمية)