الجمهورية العربية السورية
جامعة دمشق
كلية الهندسة المعلوماتية

مشروع مبادئ الذكاء الصنعي

Connect four game

❖ تنفيذ

- أسامة عرقسوسي
- ماهر القصير
- مجد العجلاني
- محمد خير دمشقي

أولاً: الحقائق المستخدمة

الحقيقة	شرح عنها
score(+Color,-Value)	من أجل معرفة قيمة تموضع الأقراص من أجل كل لاعب
depth(-Dep)	من أجل معرفة عمق شجرة البحث وذلك في خوارزمية ال minimax (تحدد في بداية اللعبة ومن قبل المستخدم)
piece(+Col,+Row,-Color)	من أجل معرفة لون القرص الموجود في العامود Col والسطر Row
size(-Col,-Row)	حقيقة من أجل معرفة عدد الأعمدة والأسطر الكليين
win(-Color)	حقيقة تولد عندما ينتصر أحد الطرفين على الأخر
lose(-Color)	حقيقة تولد عندما يخسر أحد الطرفين من الأخر
top(+Col,-Height)	حقيقة من أجل معرفة ارتفاع العامود Col
eval(+Row,+Col,-Value)	حقيقة من أجل معرفة قيمة التموضع (Row,Col)، تستخدم في خوارزمية البحث الذكية minimax

ثانياً: القواعد المستخدمة

1-قواعد تهيئة اللعبة:

init(+Col,+Row)	تقوم هذه القاعدة بحذف جميع الحقائق التي اضيفت مسبقاً وتقوم بتهيئة الحقائق لبداية لعبة جديدة.
init(+Col)	تقوم بتهيئة الحقائق (top) بقيمها البدائية حيث تقوم القاعدة init باستدعائها.

2-قواعد فحص النتيجة (فوز تعادل خسارة):

checkStatus(+Col,+Row,-Color)	تقوم باستدعاء القاعدة check/2 ومن ثم وضع الحقائق المناسبة (win/1،lose/1) حسب القاعدة الأخيرة
resetStatus()	تقوم بإزالة الحقائق (win/1،lose/1)
draw()	تقوم بالتحقق من الوصول لنتيجة التعادل وذلك عن طريق المرور على جميع الاعمدة والتحقق من قمة كل عمود.
check(+Col,+Row)	تقوم باختبار الفوز انطلاقا من القطعة الموجودة في الموقع (Row,Col) و ذلك عن طريق التجوال أفقياً و عمودياً و قطرياً باستخدام القاعدة traverse/5.
traverse(+C,+R,+IncC,+IncR,-V)	تقوم بإيجاد عدد القطع التي تحقق حالة فوز انطلاقاً من الموقع (R,C) و بأي اتجاه و ذلك حسب قيم الوسطاء (IncR,IncC) فاذا كانت

	قيمهم (1,0) فسيقوم بالتحقق من حالة الفوز بشكل افقي.

3-قواعد الدخل والخرج:

print(+Col,+Row)	لطباعة رقعة اللعب
scan(+Color)	لقراءة حركة الاعب والتحقق من صحتها
insert(+Col,+Color)	لإضافة قطعة من اللون Color إلى أعلى العمود Col
remove(+Col)	لإزالة قطعة من أعلى العمود Col

4-قواعد بداية لعبة جديدة:

newGame(+Col,+Row)	تأخذ حجم الرقعة الجديدة كوسطاء وتقوم بتهيئة حقائق اللعبة وبداية لعبة جديدة باستخدام القاعدتين init/2 وgame/0
game()	قاعدة عودية شرط توقفها هو الوصول لحالة تعادل أو فوز أحد الطرفين، المهمة الأساسية لها هي تنظيم الأدوار بين الاعب والحاسوب.

5-قواعد خوارزمية اللعب الطموحة:

مبدأ عمل الخوارزمية:

هدف الخوارزمية في كل دور هو الفوز بالخطوة الحالية إن أمكن وإلا فإنها ستبحث عن أفضل حركة تقوم من خلالها بمنع اللاعب من الوصول لحالة فوز.

get_max(+Col,+Row,-V)	عمل هذه القاعدة هوة الانطلاق من القطعة ذو الموقع (Row, Col) وإرجاع أكبر عدد من القطع المتتالية و التي تحمل نفس اللون افقيا او عمودياً او قطرياً.
can_win(+Col,-Res)	عمل هذه القاعدة هوة التحقق من امكانية الوصول لحالة فوز عن طريق حركة واحدة فقط.
best_move(+Col,-ResValue,-ResCol)	عمل هذه القاعدة هو ايجاد المكان الافضل لوضع القطعة في دور الحاسوب لمنع اللاعب من الوصول لحالة فوز لذلك ستوجد العمود الذي يحوي أكبر عدد من القطع المتتالية للخصم وتضع القطعة فيه لإيقاف التتالي وبالتالي منعه من الوصول لحالة فوز.
greedy(-Col)	قاعدة ترد أفضل عمود ليضع الحاسب فيه القطعة وذلك بالاستناد للقواعد السابقة

6-قواعد خوارزمية اللعب باستخدام خوارزمية الـ minimax:

قبل البدء بشرح قواعد هذه الخوارزمية سنتحدث عن هذه الخوارزمية بشكل عام.

خوارزمية الـ minimax: هي خوارزمية عودية من خوارزميات البحث الذكية، تُستخدم من أجل الحركة المثالية للاعب بافتراض أن الخصم يلعب بالشكل الأمثلي، وهي تشابه كيف يفكر البشر عند اللعب "إذا قمت بهذه الحركة، خصمي سيتاح له هذه الحركات، وهكذا "

سميت الخوارزمية بذلك لأنها تساعد الاعب الحالي ب "تقليل" كمية خسارته وذلك عندما يلعب الخصم بطريقة يريد بها أن يَربح ب "أكبر" قدر ممكن.

كيف تم تطبيقها في لغة ال prolog؟

تقوم القاعدة doBest/4 بتشكيل لائحة من أرقام تعبر عن أرقام الأعمدة للرقعة الحالية، ومن ثم تستدعي القاعدة minimax/4 والتي ترد سلسلة من سلاسل حيث أن كل سلسلة عبارة عن عنصرين الأولى قيمة الحركة والثانية العامود الذي نتجت منه هذه الحركة، بعد ذلك تقوم القاعدة بترتيب هذه السلسلة حسب القاعدة sort/2 المعرفة مسبقاً في مكاتب البرولوغ وتأخذ أول عنصر (إذا كانت العقدة Min) أو أخر عنصر (إذا كانت العقدة Max) وذلك بمساعدة القاعدة nth0/3 المعرفة أيضاً بشكل مسبق في مكاتب لغة البرولوغ.

doBest(+Color,+Depth,-BestScore,-BestMove)	حسب قيمة البارامتر Color نعلم نوع العقدة (max أو min) حيث من أجل العقد ذو اللون الأحمر تكون min واللون الأصفر max باعتبار أن الاعب يلعب بالأحمر والحاسب يلعب بالأصفر
minimax(+MoveList,+Depth, +Color, -ResList)	

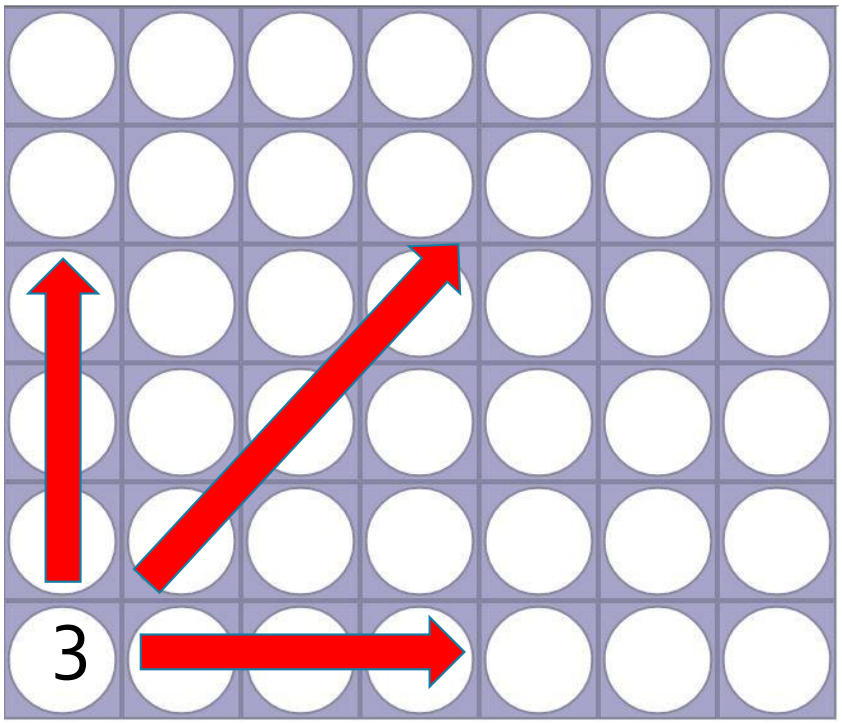
7-قواعد تابع التقييم Evaluation function من أجل خوارزمية ال minimax:

من أجل تقييم صحيح لتموضع الأقراص في الرقعة وذلك عند انتهاء العمق المحدد من قبل المستخدم في خوارزمية ال minimax، استخدمنا تابع يقوم على التالي:

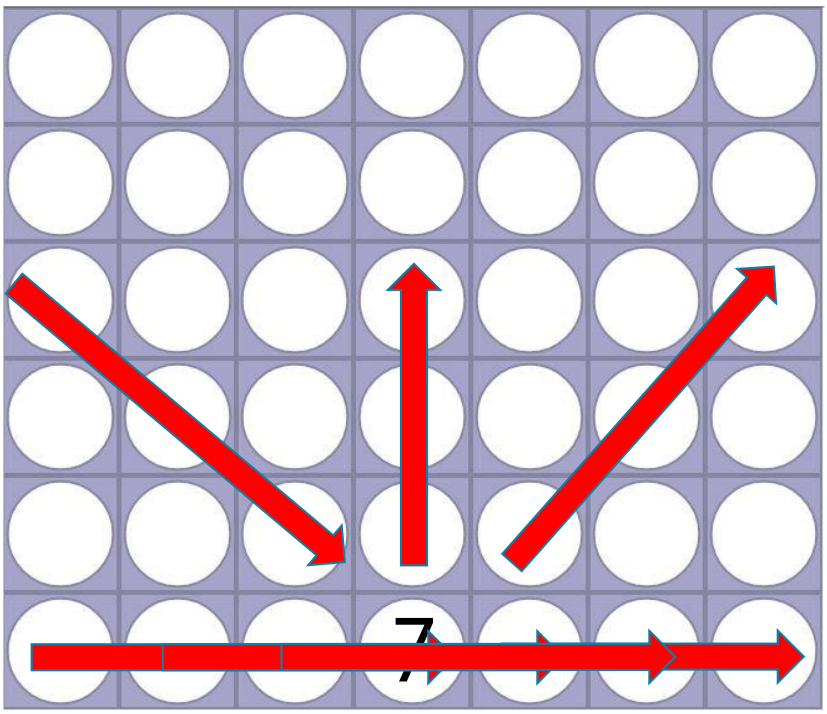
من أجل كل موقع في الرقعة هناك قيمة مقابلة لها وتساوي إلى عدد القطع المستقيمة التي تنتمي لها الموقع وطول هذه القطع هي 4.

3	4	5	7	5	4	3
4	6	8	10	8	6	4
5	8	11	13	11	8	5
5	8	11	13	11	8	5
4	6	8	10	8	6	4
3	4	5	7	5	4	3

فمثلاً من أجل الموقع الموجود بالزاوية اليسارية السفلى فهنالك ثلاث قطع مارة به وهي:



ومن أجل الموقع الموجود في منتصف السطر الأخير:



createEvalTable()	تقوم بإزالة جميع الحقائق eval/3 واستدعاء القاعدتين initEvalRow/1 وcalcEvalRow/1
initEvalRow(+Row)	المرور على جميع الأسطر واستدعاء القاعدة initEvalCol/2 من أجل كل سطر
initEvalCol(+Row,+Col)	إضافة الحقيقة eval(Row,Col,0)
calcEvalRow(+Row)	المرور على جميع الأسطر واستدعاء القاعدة calcEvalCol/2 من أجل كل سطر
calcEvalCol(+Row,+Col)	استدعاء القواعد التي في الأسفل
calcHor(+Row,+Col)	تزيد من قيم الحقيقة eval/3 من أجل القطعة المستقيمة التي على يمين الموقع (Row,Col) وذلك بمقدار زيادة 1.
calcVer(+Row,+Col)	تزيد من قيم الحقيقة eval/3 من أجل القطعة المستقيمة التي تقع فوق الموقع (Row,Col) وذلك بمقدار زيادة 1.
calcUpDia (+Row,+Col)	تزيد من قيم الحقيقة eval/3 من أجل القطعة المستقيمة التي على يمين القطر الثانوي للموقع (Row,Col) بمقدار 1.
calcDownDia (+Row,+Col)	تزيد من قيم الحقيقة eval/3 من أجل القطعة المستقيمة التي على يمين القطر الرئيسي للموقع (Row,Col) بمقدار 1.