



الجمهورية العربية السورية

جامعة دمشق

كلية الهندسة المعلوماتية

# خوارزميات البحث الذكية

إشراف :

م.زينة دلال

إعداد الطلاب :

محمد أنس اللحام

آية بقلة

آية ملا

آية ساريج

رغد حواط

## شرح اللعبة:



بداية قمنا بفهم اللعبة و لعبها بهدف تحليل اجزائها و جميع الحالات الممكنة فيها و بعد المناقشة خرجنا بعدة classes مبدئية و ثم تطوريها و تعديلها فيما بعد .

بداية يمكن لعب اللعبة من شخصين الى 4 أشخاص بحيث يكون لكل شخص 4 أحجار غير موضوعة على الرقعة بعد يهدف كل شخص لايصال جميع أحجاره الى منطقة ال Target قبل باقي اللاعبين .

يوجد حجر نرد يرمى على الطاولة حيث يحرك اللاعب احجاره التي على الرقعة بنفس القيمة على الوجه العلوي للنرد .

يبدأ كل شخص بتفعيل حجره عند ظهور الرقم 6 على النرد .

يمكن للشخص ان يقتل احجار خصومه اذا توقف حجره عند نفس الخلية التي يقف عليها الخصم .

						#	#	#						
						#	#	#						
		A2	B2			#	#	#			A3	B3		
		C2	D2			#	#	#			C3	D3		
						#	#	#						
						#	#	#						
#	#	#	#	#	#	#	#	#	#	#	#	#	#	#
#	#	#	#	#	#	#	#	#	#	#	#	#	#	#
#	#	#	#	#	#	#	#	#	#	#	#	#	#	#
						#	#	#						
						#	#	#						
		A1	B1			#	#	#			A4	B4		
		C1	D1			#	#	#			C4	D4		
						#	#	#						
						#	#	#						

## بنية اللعبة :



قمنا بداية بتمثيل الرقعة ككل بمصفوفة ثنائية (grid) بحجم 15\*15 و قمنا بتمثيل المسار المخصص للأحجار (المسار المحدد باللون البيج) ببنية cyclic graph, حيث وقع الاختيار عليها تسهيلا للتعامل مع الخلايا المربوطة بأكثر من خلية أخرى إضافة الى أن أولوية اختيار الطريق لن تكون وفق ترتيب معين بل تعتمد على كل حجر على حدا , اذ أنه تم انشاء class Node الذي تكمن وظيفته بتمثيل كل خلية ضمن ال graph ويحوي attributes أهمها :

➤ sheer (الشيرة) ليميز كل خلية اذا كانت شيرة أم لا (Boolean)

➤ Children تمثل الخلايا التي تتصل بها الخلية الحالية (List)

➤ Pawns الأحجار التي تقف على الخلية الحالية (List)

قمنا بانشاء class Board المسؤول عن انشاء الرقعة الابتدائية للعبة و انشاء مسار الاحجار حيث يحوي مصفوفة Node للتعبير عن كل خلية من خلايا المسار .

تم تمثيل الأحجار ب class Pawn ليعبر عن مكان كل حجر بالاضافة الى عدة معلومات تم تمثيلها ب :

➤ Is\_playing للتمييز فيما اذا كان موجود على الرقعة (Boolean)

➤ Is\_win للتعبير عما اذا كان قد وصل لل Target zone (Boolean)

➤ EntryBlock لتخزين رقم الخلية التي ستمثل المدخل لل Safe zone (int)

تم تمثيل اللاعبين ب class Player حيث لكل لاعب مصفوفة حجمها 4 لتمثيل الاحجار الخاصة به .

ثم قمنا بإنشاء كلاس عام لتخزين المتغيرات و التوابع العامة التي يجب الوصول اليها من اي كلاس في الproject وسمي class Statics حيث خزنا فيه قيم الالوان العامة المستخدمة في الواجهات و خزنا فيه تابع throw\_dice الذي يمثل رمي النرد حيث يرجع List حجمها على الاكثر 3 لتمثل ماهي القيم التي اظهرتها رميات النرد .

و اخيرا تم انشاء **class Level** الذي يقوم بجمع مكونات اللعبة و دمجها من رقعة الى  
توابع الى لاعبين ...الخ لتمثيل اللعبة ككل و امكانية التفاعل معها حيث يحتوي على  
التوابع المسؤولة عن تطبيق حركات اللاعب مع مراعاة قواعد اللعبة بالاضافة للتوابع  
المسؤولة عن تحريك احجار الكمبيوتر اذا ما كان ضمن اللاعبين و اختيار افضل  
الحركات المتاحة مع الاخذ بعين الاعتبار الحركات التالية التي قد يلعبها المنافس و احتمال  
ظهورها . اهم توابع الكلاس :

➤ **Update** التابع المسؤول عن تطبيق حركة اللاعب و تنفيذها و قتل احجار  
المنافس المتاحة بالاضافة الى الى مراعاة جميع الحالات الواردة

➤ **Checker** التابع الذي يأخذ رقم الحجر و القيمة المراد تطبيقها و يقوم بارجاع  
قيمة بوليانية تمثل اذا ما كان بالامكان تطبيق هذه الحركة على الحجر المراد

➤ **Evaluation** التابع الذي يقوم بارجاع قيمة عددية مسؤولة عن تمثيل الوضع  
الحالي للعبة و و يمكن المقارنة بين هذه القيم للتمييز بين الحركات المتاحة حيث  
يتم اختيار سلسلة الحركات التي تحقق أكبر قيمة **Evaluation**

➤ **Recursion** التابع الذي يطبق خوارزمية ألفا بيتا لتحقيق أفضل نتيجة ممكنة  
للحاسب

➤ **GetFinalBoards** التابع الذي يأخذ سلسلة من القيم و يعيد جميع الالعب التي  
يمكن الحصول عليها بعد تطبيق هذه السلسلة بشتى الطرق على اللعبة الحالية

➤ **EndGame** الذي يقوم بفحص اللعبة فيما اذا كانت قد انتهت بفوز اللاعب  
الحالي أم لا

## : Technics and algorithms



تم الاستعانة بعدة خوارزميات و تقنيات لتحقيق اللعبة و تطبيقها بأسرع الاشكال الممكنة  
أهم هذه التقنيات :

➤ Deep copy تكنيك يستخدم لاسناد object الى object آخر دون حدوث Assign by reference كما يحدث بطبيعة الامر في بعض اللغات مثل الجافا حيث يقوم بالوصول لكل المتغيرات وفي هذا الكلاس و اسنادهم واحدا واحدا

➤ Dfs & Bfs تم استخدام هذه الخوارزميات لانشاء المسار بالاضافة الى استخدامهم في تابع ال update , checker لتحقيق حركات الاحجار بسهولة

➤ Alpha & Beta ساعدت الخوارزمية بإيجاد أفضل تطبيق لسلسلة القيم التي حصل عليها الحاسب لتطبيقها على اللعبة الحالية اعتمادا على depth معين يحدده المستخدم مع الاخذ بعين الاعتبار احتمال ظهور شتى القيم للخصم

➤ ExpectMiniMax و تطويرها اذ تكمن أهميتها في الالعب الاحتمالية الغير الحتمية التي تعتمد على الحظ كالعب النرد و سحب الورق ... الخ تقوم بحساب جميع القيم التي من الممكن أن تظهر لي و للخصم على مدى عمق معين و ضربه باحتمال ظهوره .  
تم توليد جميع الاحتمالات مسبقا لتوفير الوقت على المعالج تجنبنا لحساب الاحتمالات كل مرة

➤ Swing لتمثيل الواجهات الرسومية في جافا

➤ Threads لوضع Timer للحاسب لمحاكاة حركته و تقريبها للواقع

➤ RGB\_system للتحكم بألوان الخلايا و الاحجار



## مزايا لعبتنا التي تم تطويرها :

- إمكانية اللعب مع اصدقائك او اللعب ضد الحاسب مع التحكم بالصعوبة استنادا الى قيمة العمق الموضوعة لخوارزمية  $\alpha\&\beta$
- إمكانية وضع من لاعبين الى 4 لاعبين و التحكم بعدد اللاعبين الحقيقيين و الافتراضيين
- اضافة واجهة رسومية GUI لسهولة التحكم و مراقبة اللعبة مع وجود واجهات مميزة و توضيحية للعبة اضافة لوجود اللعبة بنسخة ال Terminal
- تطوير اللعبة باستخدام لغة جافا التي تعتبر من اللغات سريعة التنفيذ اضافة الى ذاكرتها الافتراضية الضخمة بالاضافة لدعمها لخاصية ال Threads

---

## للتنويه :

- وبناء على طلب حضراتكم في ذكر المصادر التي استفدنا واقتبسنا أشياء منها و حتى لا يكون هناك لبس او تشابه نوضح لحضرتكم ما يلي :
- تم الاستعانة بمصادر من الانترنت للحصول على تمثيل الالوان في كلا نظامي ال RGB و ANSI لاستخدامهما في الواجهات الرسومية و في ال console .
  - تم الاستعانة بموقع **Geeks for Geeks** للحصول على توابع ال Hashcode للكلاسات التي تفيد في القيام بعمليات المقارنة بين ال objects بشكل صحيح .