Logic Programming

Section #6

Eman Safwat

11/18/2009

Game Project

المرة دي احنا اخدنا Project جاهز بال Design بتاعه عشان لو حد عايز يستعمله في ال Project بعد كده عشان احنا مش دايسين في ال Visual Prolog أوي.

هوا عبارة عن Window مرسوم عليها Grid (Columns و Rows) ومتقسمة Cells أما تدوس على أي Cell فيها بتظهر جوه ال Cell دي Coin بلون معين. وطبعا دي ممكن نستعملها في Games كتيرة زي Connect 4، Queens 8 كده.. بس هيا ال Coins هنا مش عليها أي Restrictions ممكن لما تعمل ال Game بتاعتك تحط Rules و Restrictions زي ما أنت عايز.

المرة اللي فاتت كنا عملنا Dialog لل Calculator المرة دي هنشتغل على Window هما الاتنين زي بعض مفيش فرق كبير بينهم.

ال Window اللي عندي فيها Button واحد اللي هوا Draw اللي بيرسم ال Grid.

Window → select GameWin → Code Expert → from 'Event Type' select Mouse → e_MouseDown → Edit Clause

MouseDown دي اللي هيا MouseClick، لما ادوس Edit Clause هيفتحلي File.. GAMEWIN.PRO وده اللي فيه ال Code كله هنبدأ بقى نشرح ال Code من أوله.

Domains

ilist2=ilist *

database-gameDB

game(ilist2)

player(integer)

هنا عرفنا ilist2 على إنها List Of Lists وال Lists اللي جواها بتبقى List Of Integers. ilist دي Built-in في ال Visual Prolog متعرفة على إنها List Of Integers حتى هتلاحظوا إنها مكتوبة بالتفصيل في ال Comment فوق Domains.

بعدين عملنا Database اسمها gameDB هيا دي اللي شايلة ال Values بتاعت ال Grid وبتاخد List Of Lists وعندي player (بتاخد integer) دي Table جديد في ال Database د. عمرو زودو في ال **Code** عشان يعمل Switch بين 2 Players (لو دوست مرة على Cell تطلع Red Coin تدوس تاني تطلع Blue Coin على أساس إنه 2 Players بيلعبوا قدام بعض) وعمل ليها Predicate اسمها switch_player هنشوفها قدام في ال Code.

طيب أنا ليه عملت أصلا Database ومخلتهاش Predicates وخلاص مثلا؟

عشان زي ما احنا عارفين مينفعش أعرف Variables في ال Prolog وهنا أنا عاوزة Global Variables عشان كل ال Events تشوفها، فالطريقة الوحيدة عشان أعمل كده هيا ال Database. عشان اغير في ال Database دي باعمل Retractall لل Values القديمة و Assert للجديد.

Constants

x1=100

y1=100

cellsize=50

rowcount=8

colcount=8

epsilon=5

Constants دي Region جديدة بعملها أما أكون عاوزة أعرف Constants معينة في البرنامج بحيث أكتب اسم ال Constant وال Prolog تعوض عنه بال Value بتاعته (أكتب x1 في أي حتة في ال Code هوا يبقى عارف إنها 100 أكتب epsilon يعوض عنها ب 5 كده..) ولاحظوا إنها بتبدأ ب Small Letter عشان هيا مش Variable.

x1 و y1 دي ال Coordinates اللي هايبدأ من عندها رسم ال Grid، هتلاقوا ال Grid مرسومة بعد جزء من ال Window كده مش من أولها لو خلتوهم 10 و 10 مثلا هتلقي ال Grid (دخلت شمال شوية) قربت من بداية ال Window، طبعا البداية عند Coordinates 0 و 0.

CellSize دي ال Size لكل Cell هتترسم، RowCount و ColCount عدد ال Rows وال Columns اللي هيترسمو في ال Grid. Epsilon دي المسافة بين ال Coin اللي هترسم جوه ال Cell والمربع بتاع ال Cell نفسها (الفراغ الأبيض اللي بيبقى موجود بعد مال Coin تترسم بالنسبة لل Cell اللي ال Coin موجودة فيها).

Predicates

nondeterm drawGame(window)

بتاخد ال Handle بتاع ال Window ودي ال Predicate اللي هتشيل جواها drawBoard و drawCoins.

drawBoard(window)

دي اللي بترسم ال Board (Grid) وبتاخد ال Handle لل Window برضو.

drawCoins(window,ilist2)

دي بترسم ال Coins جوه ال Grid وبتاخد ال Handle و List Of Lists.

الباقي تقريبا نفس الكلام وهنشوف بالتفصيل في ال Clauses بتاعتهم.

Clauses

```
switch_player:-

player(1), !,

retractall(player(_)),

assert(player(2)).


switch_player:-

retractall(player(_)),

assert(player(1)).
```

زي ماقلنا player ده Table في ال Database و switch_player دي Predicate بت Check لو هوا Player 1 بتشيله و تعمل Assert ل Player 2 والعكس لو 2 هتشيله وتقلب على Player 1.

```
drawGame(A):-

drawBoard(A),

game(G),

drawCoins(A,G).
```

drawGame بتاخد Handle لل Window (ال A ده Variable بيعمل Matching مع ال Handle) و drawBoard نفس الكلام وبعدين بنده على ال Database (game) واعمل Loading لل Values ال List) اللي فيها في Variable .. G. وبعدين أنده على drawCoins بال Window Handle(A) وب Values ال Database اللي في G.

```
drawBoard(Win):-

X2=x1+colcount*cellsize,

Y2=y1+rowcount*cellsize,

draw_rect(Win,rct(x1,y1,X2,Y2)),

N1=colcount-1,

drawVLines(Win,x1,y1,Y2,N1),

N2=rowcount-1,

drawHLines(Win,x1,y1,X2,N2).
```

drawBoard دي اللي هترسم ال Grid كلها بإستخدام draw_rect (اللي هترسم ال Outer Rectangle بس، ودي Built-in Predicate في ال Visual) و drawVLines (اللي هترسم ال Vertical Lines في ال Grid، ودي احنا اللي عملينها وهنشوف ال Code بتاعها كمان شوية) و drawHLines (بترسم ال Horizontal Lines في ال Grid وهيا زي drawVLines بالضبط).

شغالة إزاي drawBoard؟

باديها ال Handle لل Window اللي هترسم فيها،

```
X2=x1+colcount*cellsize,

Y2=y1+rowcount*cellsize,
```

أنا لما باجي أرسم Rectangle ببقى محتاجة 2 Points (x1 و y1 دي أول Point اللي هيا ال Upper Leftmost Point في ال Rectangle) وال Point التانية (x2 و y2 وهيا ال Lower Rightmost Point في ال Rectangle)

فأنا هنا عندي x1 (ال Constant اللي مكتوب في أول ال File ومديالله Value بـ 100) و y1 (Constant برضو) ودي كده أول Point اللي هبدأ أرسم ال Rectangle من عندها.

X2 و Y2 دول ال Point التانية بتاعت نهاية ال Rectangle ودول أنا هحسبهم على حسب ال Constants اللي في ال File (طبعا ممكن اغير ال Values بتاعتهم في أي وقت).

X2 بترسم ال Rows فلما أضرب ال Size لكل Cell في عدد ال Columns اللي أنا عاوزها هتديني Width ال Rectangle وبزود عليهم x1 عشان ميعدش من أول ال Window لأ أنا عاوزاه يعد من بعد المسافة بتاعت x1.

Y2 نفس الكلام بس بترسم ال Columns.

draw_rect(Win,rct(x1,y1,X2,Y2)),

- draw_rect دي Built-in Predicate بأديها ال Handle لل Window وباديها X,Y-Coordinates لأول وتاني Point هيترسم بيهم ال Rectangle.

- وعامة في ال Project ده أي Predicate تلاقوا فيها Underscore كده تبقى Bulit-in (ما عدا switch_player احنا اللي عاملينها) لو دستوا Right Click في ال File:

Insert → Predicate Call → VPI Predicate

هتلاقوا كل ال Built-in Predicates وكلها على نفس Style.. draw_rect.

N1=colcount-1,

drawVLines(Win,x1,y1,Y2,N1),

N2=rowcount-1,

drawHLines(Win,x1,y1,X2,N2).

لو عدينا عدد ال Columns أو ال Rows اللي جوه ال Outer Rectangle (ال Rectangle الكبير) هنلاقيهم أقل من ال Count اللي في ال Constants ب 1 (colcount و rowcount) فعشان كده بقللهم واحد قبل ما أبعت لل Predicates اللي هترسم ال Lines.

drawVLines(Win,x1,y1,Y2,N1),

دي اللي هترسم ال Vertical Lines فبأديها ال Handle لل Window و x1 و y1 (بداية ال Rectangle) و Y2 اللي حسبناها فوق (طول كل Column) و N1 اللي هيا عدد ال Columns جوه ال Outer Rectangle.

نفس الكلام على drawHLines بس هترسم ال Rows.

---

drawVLines(_,_,_,_,0):- !.

هنا هيقف لما عدد ال Lines يبقى ب 0 عشان بكون خلصت رسم ال VLines كلهم.

drawVLines(Win,X1,Y1,Y2,N):-

بتاخد ال Handle بتاع ال Window. بعدين ال X اللي هترسم من عندها وال Y اللي هتبدأ منها وبعدين ال Y اللي هتنتهي عندها. وال N عدد ال Columns اللي هتترسم.

NX1=X1+cellsize,

كل مرة بيزود ال X1 (اللي هيا بت Match مع x1 ال Constant في drawBoard) بال cellsize عشان تنقل علي ال Cell اللي بعديها (لاحظوا إنها بدأت من x1 يعني من عند أول Line في ال Outer Rectangle فمش محتاجة ترسم حاجة عشان أول Line مرسوم وبتروح على تاني Cell).

OldPen = win_GetPen(Win),

GetPen ده بتجيب ال Handle بتاع ال Window يعني بتاخد لون ال Pen اللي في ال Window (ال Default إنه Black) وبتخزنه في OldPen.

win_SetPen(Win,pen(2,ps_Solid,color_Green)),

SetPen بت Set ال Color لل Window وبتخليه Green عشان احنا عايزين نرسم ال Vertical Lines.. Green.

كل Window ليها ال Pen بتاعها ال Color بتاعه Black ..Default فممكن اغيره لأي Color عاوزاه.

draw_Line(Win,pnt(NX1,Y1),pnt(NX1,Y2)),

هنا بقى هارسم Line عادي خالص بأدي ل draw_Line ال Handle بتاع ال Window و 2 Points اللي هارسم بيهم، pnt(NX1,Y1) دي نقطة البداية NX1 اللي حسبتها فوق و Y1 اللي جايلي أصلا وهنتهي عند NX1 برضو و Y2.

NN=N-1,

بعدين بنقص عدد ال Lines ب 1 عشان لما أنده على نفسي تاني.

win_SetPen(Win,OldPen),

بت Set ال Pen بال OldPen (ال Black) عشان لو دي آخر مرة أنده على نفسي فيها وهخش في ال Base Step وانهي ال Recursion ده وأروح على drawHLines يبقى ال Color اللي هارسم بيه رجع Black (هو ده اللي احنا عاوزينه هنا) لو جربته تشيلوا ال Line ده من ال Code هتلاقوه رسم كل ال Lines.. Vertical و Horizontal بـ Green.. ليه؟ عشان مش هيبقى رجع ال OldPen وهيروح على drawHLines بال Pen اللي هيا Set ل Green.

drawVLines(Win,NX1,Y1,Y2,NN).

بعدين بينده على نفسه تاني بعدد ال Columns الجديدة.

- نفس الكلام على drawHLines بس هيرسم ال Rows بقى ومن غير ما ي Set ال Pen بحاجة عشان هيا رايحه له بال OldPen (ال Black)، بس الفرق في drawHLines إنها هتتحرك بال Y من Row ل Row فوق كنت بتحرك بال X من Column لل Column اللي بعده.

أنا كده رسمت ال Lines اللي جوه ال Outer Rectangle وبقت ال Grid عندي متقسمة Cells وكله تمام، فاضل بقى إني أرسم ال Coins اللي هتبقى في ال Grid دي وده اللي بتعمله drawCoins.

drawCoins(Win,G):-
drawAt(Win,x1,y1,G).

drawCoins بأديها ال Handle بتاع ال Window وال G (اللي ه يا فيها ال Values Database اللي هيا ال List Of Lists). بتنده drawAt اللي هتبدأ من عندها رسم لل Coins وباديها ال Handle و x1 و y1 ال Constants (بداية ال Grid) وال G ال List Of Lists اللي على اساسها هترسم ال Coins في الأماكن اللي ال User بيدوس عليها في ال Grid.

drawAt بقى بتعمل إيه؟ بتاخد ال List Of Lists من ال DB وبتقسمها Head و Tail فبيبقى ال Head عندي عبارة عن List كاملة وال Tail باقي ال Lists في ال DB وبتبعت ال Head ل drawRow عشان ترسم ال Coins بقى وبعدين تتحرك بال Y لل Row اللي بعده وتنده على نفسها بال Tail وتفضل شغالة لحد ال List ال كبيرة ماتبقى فاضية وتدخل في ال Base Step.

drawAt(_,_,_, []):-!.
هتقف لما تخلص رسم كل ال Coins اللي في ال List الكبيرة يعني لما ال List تبقى فاضية.

drawAt(Win,X1,Y1,[H|T]):-
بتاخد ال Handle وال Start Point ال هترسم من عندها (ال Constants بداية ال Grid) وبتاخد List (G) اللي هيا عبارة عن List Of Lists.
drawRow(Win,X1,Y1,H),
تنده على drawRow وتديلها ال Handle وال Start Point وال Head اللي هيا List كاملة زي ماقلنا.

NY1=Y1+cellsize,
بعدين بتتحرك بال Y1 لتحت بمقدار ال cellsize (هتنقل على ال Row اللي بعده).

drawAt(Win,X1,NY1,T).
وتنده على نفسها تاني بال Y الجديدة وال Tail (باقي ال Lists من ال List الكبيرة بعد مشيلنا أول List).

نشوف بقى drawRow اللي هترسم ال Coins في كل Row شغالة إزاي..
هيا بتاخد List وتقسمها كل مرة ل Head و Tail بس ال Head هنا عبارة عن Integer عادي وال Tail باقي ال Integers اللي في ال List (يعني احنا الأول قسمنا ال List Of Lists ل Head و Tail وال Head ده كان عبارة عن List فبعتنا ال List بتاعت ال Head دي ل drawRow وهيا بقى هتقسمها Head و Tail برضو بس ال مرة دي هيبقوا Integers عادية).
واحنا عملين 3 Cases للـ drawRow:

1) ال Head تبقى ب Zero وكده يبقى مفيش Coin في ال Cell دي (محدش داس على ال Cell دي قبل كده).
2) ال Head تبقى ب One معناها إنها ترسم Red Coin في ال Cell دي (Player 1 كده لعب مثلا).
3) ال Head تبقى ب Two معناها إنها ترسم Blue Coin في ال Cell (Player 2 لعب).

- Case 3 مكنتش مت Handle في ال Code اللي معظمكم اخده في ال Section بس switch_player عملت Handling ليها فلما تعملوا Run هتظهر Red Coin وبعديها Blue Coin وهكذا..

drawRow(_,_,_,[]):- !.

برضو نفس الكلام هتقف لما ال Coins تخلص وال List تبقى فاضية.

drawRow(Win,X1,Y1,[H|T]):-

بتاخد ال Handle بتاع ال Window وال Start Point زي فوق وبتاخد List تقسمها Head و Tail.

H=0,!,

NX1=X1+cellsize,

بيت Check إذا كانت ال Head ب 0 يبقى مفيش Coin في ال Cell يبقى مش هيعمل حاجة (ال Cut هنا عشان لو كمل في ال Predicate دي ميدخلش في ال 2 Predicates الجايين ل drawRow) وهينقل على ال Cell اللي بعديها (بيتحرك بال X للـ Column اللي بعده) طبعا بإنه يضيف ال cellsize على ال Value القديمة زي معملنا قبل كده.

drawRow(Win,NX1,Y1,T).

بعدين بينده على نفسه بال X الجديدة اللي هيا ال Cell اللي بعدها.

- ال 2 Cases اللي جايين (لما ال Head تبقى ب 1 أو 2) زي بعض بالضبط فأنا هشرح واحدة منهم ومش هاكتب التانية هتلاقوها في ال Code في ال Game Project.

drawRow(Win,X1,Y1,[H|T]):-

H=1,!,

بي Check لو ال Head بتساوي 1 كده هيرسم Red Coin.

هوا إيه الفرق بين ال Pen وال Brush؟

ال Pen بترسم ال Lines أما ال Brush بترسم ال White Spaces اللي جوه حاجة (جوه Triangle مثلا أو Rectangle أو جوه ال Cell اللي هيا Rectangle صغير يعني) فال Line عشان مفيش جوه حاجة فمش محتاج Brush.. عشان كده استعملنا ال Pen في drawVLines و drawHLines.

OldB=win_GetBrush(Win),

GetBrush دي بتجيب ال Handle لل Window وتخزنه في OldB (عشان لما ألعب في لون ال Brush وأبقى عايزة ارجعه تاني.. يبقى موجود في OldB Variable، زي معملنا في ال Pen).

win_SetBrush(Win,brush(pat_Solid,color_red)),

كده غيرت لون ال Brush ل Red عشان أرسم ال Red Coin جوه ال Cell (Solid عشان اخلي ال Red اللي هرسمه Solid Red مش Gradient مثلا أو أي حاجة تانية).

XX1=X1+epsilon,

YY1=Y1+epsilon,

كده بحرك أول Point للمكان اللي هبتدي منه، بزود epsilon على ال X عشان اسيب مسافة جوه ال Cell فاضية (تخيلوا معايا كده.. لما ازود epsilon اللي هوا space ما (ممكن اغير قيمتها براحتي في ال Constants) على ال X اللي هيا بداية ال Cell فكده هيسيب حتة فاضية في الأول قبل مايرسم ال Coin.. باعمل الكلام ده لل X وال Y عشان أسيب مسافة Horizontally و Vertically فاضية جوه ال Cell (المسافة اللي أنا بتكلم فيها دي صغيرة أوي مش حاجة كبيرة لحد يفتكر إنها حاجة فظيعة يعني.. اعملوا Run ودوسوا علي ال Cells هتلقوا ال Coins بترسم فعلا بعيدة شوية عن Boarders ال Cell).. طيب لو معملتش السطرين دول هيحصل حاجة؟ لأ عادي ال Coins بس هترسم لامسة ال Boarders بتاعت ال Cell من ناحية الشمال.

XX2=X1+cellsize-epsilon,

YY2=Y1+cellsize-epsilon,

ده كده عشان احسب نقطة النهاية لل Coin عشان اسيب مسافة فاضية برضو بس على ناحية اليمين من ال Coin.. بجمع X1 (بداية ال Cell) على ال cellsize كده أنا وصلت لآخر ال Cell وأنقص منهم ال epsilon (يبقى هخش جوه شوية في ال Cell بس من ناحية يمين ال Coin) كده أنا وصلت ل Limit ال Coin من اليمين اللي أنا عاوزاه (اخرها اللي هتقف عنده).

الفكرة إن ال Circle أو ال Coin أما بتيجي تترسم بأتعامل معاها على إنها Rectangle بحسب ال Start Point بتاعته وال End Point فلما ادي ال 2 Points دول لل Function اللي هترسم بتبقى عارفة ال Limits بتعتها من اليمين ومن الشمال.

draw_Ellipse(Win,rct(XX1,YY1,XX2,YY2)),

دي ال Function اللي هترسم ال Circle بأديها ال Handle لل Window و ال 2 Points اللي حسبتهم.

win_SetBrush(Win,OldB),

بارجع ال Brush لل Default (اللي هوا ال Black) عشان لو مارجعتهاش وجت ندهت على نفسها وكانت ال Head ب 2 فهيدخل في ال Predicate اللي بعد دي (التالتة من drawRow) بعد ما هي Set ال Brush ل Blue ويرسم ال Coin هيجي يرجعها تاني لل OldB اللي هوا في الحالة دي بقى Red (عشان GetBrush بترجع آخر Color حصله Set في ال Window وهوا حصله Set آخر مرة ب Red) فلما تدوس تاني عال Grid هتلاقيها بقت RED كلها!

مش عارفة انتوا فاهمني ولا إيه بس شيلوا آخر سطر ده من ال Code و اعملوا Run وإلعبوا في ال Grid شوية هتفهموا قصدي.

ولو جيت شلت نفس ال Line ده من ال Predicate التالتة بتاعت H = 2 ولعبت في ال Grid هتلاقيها بقت Blue كلها لنفس السبب اللي قلنا عليه.. موضوع ظريف فعلا :d

NX1=X1+cellsize,

كده هتحرك يمين لل Cell اللي بعد كده.

drawRow(Win,NX1,Y1,T).

وأنده على نفسي تاني بال X الجديدة وال Tail وباقي الحاجات.

- Case ال 2 = Head زي اللي قلناه بالضبط بس الفرق إني ب Set ال Color فيها ل Blue عشان ترسم ال Blue Coin.

- الحمد لله خلصنا Drawing وكده ال Grid بقت جاهزة بال Coins بتاعتها نشرح بقى ال e_Create وال e_Update وال e_MouseDown وخلي set و get للاخر.

e_Create دي Event بيتنده أول مال Application ت Run.

```
win_gamewin_eh(_Win,e_Create(_),0):- !,

win_CreateControl(wc_PushButton,rct(470,12,516,52),"draw",_Win,[wsf_Gro
up,wsf_TabStop],idc_draw),


retractall(game(_)),
```

دي كده هتشيل أي game من ال Database ودايما بترجع True.

ليه مستعملتش هنا retract بس؟ عشان retract ممكن ت Fail لو مفيش Values في ال DB ولو عملت Fail يبقى مش هتكمل بقيت ال Code وده احنا مش عايزينه.

```
assert(game([

[0,0,0,0,0,0,0,0],

[0,0,0,0,0,0,0,0],

[0,0,0,0,0,0,0,0],

[0,0,0,0,0,0,0,0],

[0,0,0,0,0,0,0,0],

[0,0,0,0,0,0,0,0],

[0,0,0,0,0,0,0,0],

[0,0,0,0,0,0,0,0]

])),
```

بعدين هي Assert.. List Of Lists كلها Zeros في ال Database.

```
retractall(player(_)),

assert(player(1)),

!.
```

وبعدين برضو هشيل كل Values.. player من ال DB وهـ Assert أول Player.

---

e_Update

دي ال Event اللي بيعمل Update كل شوية لل Application وهتلاقوا جواها:

drawGame(_Win)

بتنده على drawGame عشان ترسم نفسها بال Values اللي فيها.. وحطيتها هنا عشان لما أعمل Minimize لل Form وبعدين Maximize ال Values متختفيش من عليها (ال Coins).

---

e_MouseDown ده ال Event اللي بيتنده كل أما ادوس بال Mouse في أي Point في ال Grid وهوا اللي فيه كل الشغل عشان على أساس ال Click هتحدد ال Cell اللي هتضاف فيها ال Coin سواء Red أو Blue.

win_gamewin_eh(_Win,e_MouseDown(_PNT,_ShiftCtlAlt,_Button),0):- !,

_PNT=pnt(X,Y),

أما بدوس في أي حتة في ال Grid بترجع point (X,Y-Coordinates) للمكان اللي دست فيه وكده أنا بطلع ال X و Y من ال point اللي رجعت دي.

X>=x1,X<=x1+colcount*cellsize,

Y>=y1,Y<=y1+rowcount*cellsize,

بت Check إذا كانت ال point دي جوه ال Grid ولا لأ بإن ال X تعدي ال x1 (ال Constant أول ال Grid) متعديش ال Width بتاع ال Grid ونفس الكلام على الY.

X1=X-x1,Y1=Y-y1,

هنا بجيب ال point بس بالنسبة لل Grid نفسها فبطرح منها المسافة ال Constant الفاضية في ال Window.

C=X1 div cellsize+1,

R=Y1 div cellsize+1,

Div دي بتقسم وترجع الرقم الصحيح للقسمة. وانا هنا بجيب رقم ال Column (C) ورقم ال Row (R) لل point بتاعتي فلو افترضنا إن:

X1 = 70

Y1 = 130

(70/50) + 1 = 2 → Column

(130/50) +1 = 3 → Row

فلما بداية ال Grid تبقى عند (0 ,0) بالنسبة لل Grid نفسها فيبقى تاني Column عند X = 50 وتالت Column بيبدأ من عند X = 100 فكده فعلا X1 = 70 هتبقى في ال Column التاني.. ونفس الكلام علي ال Y.

game(G), !,

بعدين بجيب ال List بتاعتي من ال DB.

get(R,C,G,V),

وابعتها ل Get و Get هترجعي ال V.

V=0,

لو V ب 0 معنى كده إن مفيش Coin مرسومة فممكن أرسم. لو كانت بـ 1 أو 2 معناها إن فيها Coin مرسومة فمينفعش أرسم عليها تاني.

retract(game(_)),

باشيل كل ال Values أو ال List القديمة من ال DB.

player(P),

بعدين هجيب رقم ال player من ال DB.

set(R,C,P,G,NG),

وأبعته ل Set و Set هترجعلي ال List وفيها ال Values الجديدة.

assert(game(NG)),

هضيف ال Values الجديدة في ال DB.

switch_player,

بعدين هاغير ال player لو كان 1 يبقى 2 والعكس عشان تبقى Red Coin وبعدين Blue والعكس.

win_Invalidate(_Win),

!.

Invalidate دي بتعمل Refresh لل Board يعني بتخليها ترسم نفسها تاني بال Values الجديدة.. ممكن أحط بدالها drawGame(_Win) نفس الحاجة برضو.

- كده خلصنا MouseDown فاضل بقى ال Get وال Set وخلاص..

```
set(1,C,X,[H|T],[H1|T]):- !,
setrow(C,X,H,H1).
```

Base Step بيدخلها أما ال R = 1 وينده على setrow.

```
set(R,C,X,[H|T],[H|T1]):-
R1=R-1,
set(R1,C,X,T,T1).
```

Set بتعمل إيه؟

Set بتمشي علي ال Rows لحد ماتجبلي ال Row اللي أنا عاوزاه.

بأديها رقم ال Row ورقم ال Column وال Value سواء 1 لو Red Coin أو 2 لو Blue Coin وبأديها ال List الكبيرة وهيا بترجعلي ال List متعدلة بال Value الجديدة.

كل مرة بتقلل ال Row ب 1 وتشيل ال Head من ال List وتنده على نفسها بال Tail (لاحظوا إن ال Head دي List Of Integers كاملة) فلما بقلل ب 1 بينزل على ال Row اللي بعده في ال Grid وهكذا لحد ما يوصل لل 1 Row =

اللي هوا ال Row المطلوب فيروح داخل في ال Base Step وينده على setrow برقم ال Col وال Value وال Head (اللي هيا آخر Head وصلها اللي هيا ال Row اللي هيعدل فيه) وبترجع ال Head List متضاف فيها ال X.

- نشوف بقى setrow..

```
setrow(1,X,[_|T],[X|T]):- !.
```

Base Step بيقف أما ال C = 1 وبيضيف ال Value على ال Tail اللي جايله.

```
setrow(C,X,[H|T],[H|T1]) :-
C1=C-1,
setrow(C1,X,T,T1).
```

قبل ما أنده على setrow بكون وصلت لل List اللي هعدل فيها وبأديها ل setrow عشان تمشي عليها Column وراء Column لحد ماوصل لل Cell المطلوبة واعدل ال Value بتاعتها (X).
ال Recursive Step زي ال set بتقلل ال Columns ب 1 كل مرة وتشيل ال Head (Integer Value)
وتنده على نفسها بال Tail لحد ماتوصل ل C = 1 (يبقى كده هيا وصلت لل Cell اللي عاوزها) وتضيف ال X ك Head في ال List الجديدة.

- ال Grid عبارة عن 2-Dimensional Array الاول بمشي Row By Row عشان أجيب ال Desired Row وبعدين على ال Row ده بامشي Column By Column لحد ماوصل لل Desired Column برضو وأبقى كده وصلت لل Cell اللي هعدل فيها سواء ب 1 أو 2 (Red or Blue Coin).

get نفس فكرة set بس بأديها رقم ال Row وال Column وال List وترجعلي ال Value في ال Cell (X).. امشوا معاها واحدة واحدة زي ال set وهتبقى مفهومة إن شاء الله.

Note:

ال Project خلص الحمد لله.. هتلاقوا File ال Project في ال Topic نزلوه بدل القديم عشان شرح ال Section ده ماشي معاه بترتيب ال Code وعشان تشوفوا ال Blue Coins.

Good Luck