جامعة دمشق

كلية الهندسة المعلوماتية

التقرير النهائي

مشروع الحسابات العلمية

إعداد:

محمد أنس دالاتي

رضا الكويفاتي

براء الرفاعي

بهاء برهم

معتز أبو عاصي

حسين الغزالي

بإشراف:

م. بلال عزيزية

المقدمة

تُعدّ لعبة البلياردو من أبرز الأمثلة التطبيقية على مبادئ الميكانيكا الكلاسيكية، إذ تختزل في حركة كرة واحدة منظومة كاملة من القوانين الفيزيائية المتشابكة: من قوانين نيوتن للحركة الخطية، إلى ميكانيكا الأجسام الصلبة والحركة الدورانية، وصولاً إلى ديناميكا التصادمات وقوانين حفظ الطاقة والزخم.

تهدف هذه الدراسة إلى تقديم تحليل فيزيائي شامل ودقيق لحركة كرة البلياردو في مختلف مراحلها: من لحظة الضرب بالعصا، مروراً بمراحل الانزلاق والتدحرج، وصولاً إلى التصادمات الثنائية والجماعية وارتداد الكرة عن الجدران المطاطية. ويُولي التحليل اهتماماً خاصاً بالاستنتاج الرياضي الصارم للمعادلات الحاكمة لكل ظاهرة، مع ربطها بالتفسير الفيزيائي المباشر.

تكمن أهمية هذه الدراسة في كونها نموذجاً تعليمياً متكاملاً يجمع بين التجريد الرياضي والتطبيق العملي الملموس، كما تُشكّل أساساً نظرياً متيناً لتطوير محاكيات فيزيائية واقعية (Physics Simulations لألعاب الطاولة. إن فهم هذه الديناميكا يُتيح للاعب المحترف أو المبرمج التنبؤ بمسارات الكرات بدقة رياضية عالية.

1. الأسس الفيزيائية العامة للقوى والعزوم المؤثرة على كرة البلياردو

قبل الخوض في تفاصيل حركة الكرة ومراحلها، لا بد من تأسيس قاعدة فيزيائية متكاملة تجمع كل القوى والعزوم والمعادلات التي تحكم سلوك كرة البلياردو. هذا الإطار النظري هو المرجع الأساسي لكل ما يأتي من تحليل.

1.1. القوى المؤثرة على الكرة

تتعرض كرة البلياردو أثناء حركتها لمجموعة من القوى المتداخلة. يلخص الجدول التالي هذه القوى مع قوانينها وخصائصها:

القوة	الرمز	القانون	الاتجاه	المصدر
قوة العصا (النبضة)	J	$J = m.v_0$	أفقي → أمام	عصا البلياردو
وزن الكرة	W	$W = m.g$	رأسي ↓	الجاذبية الأرضية
قوة رد الفعل	N	$N = m.g$	رأسي ↑	سطح الطاولة
الاحتكاك الحركي	f_k	$f_k \leq \mu_k.m.g$	معاكس للحركة	تلامس الكرة بالسطح
الاحتكاك الساكن	f_s	$f_k \leq \mu_k.m.g$	معاكس للإزاحة	عند بدء الحركة
مقاومة التدحرج	F_r	$a_{rolling} = \mu_r.g$	معاكس للتدحرج	تشوّه ألياف القماش
قوة التصادم (كرة-كرة)	F_{ij}	$F_{ij} = -F_{ji}$	على خط المراكز	تلامس الكرات
قوة ارتداد الجدار	J_n	$v_{out,n} = -e_c.v_{in,n}$	عمودي على الجدار	المطاط المرن

1.2. العزوم المؤثرة

العزم (Torque) هو المقياس الدوراني المكافئ للقوة في الحركة الخطية. يتحكم في التسارع الزاوي للكرة وفق قانون نيوتن الثاني للدوران: $\tau_G = I_G.\alpha$

العزم	الرمز	القانون / المعادلة	الملاحظات
عزم قوة العصا	τ_G	$\tau_G = F.(h-r)$	h: ارتفاع الضربة عن المركز
عزم القصور الذاتي (كرة مصمتة)	I_G	$I = \frac{2}{5}mR^2$	للكرة الكروية الصلبة

عزم احتكاك السطح	$\tau_{friction}$	$\tau_{friction} = f_k \ .R = \mu_k.m.g \ .R$	يسبب تسارعاً زاوياً
عزم احتكاك الجدار	τ_{wall}	$\tau = R \times f_t$	بعد الارتدادωيغير

1.3. معادلات الحركة الانسحابية

تصف هذه المعادلات حركة مركز كتلة الكرة في الاتجاه الأفقي وفق قوانين نيوتن الخطية:

قانون نيوتن الثاني (الحركة الخطية):

$$-\boldsymbol{F} = \boldsymbol{m} \, . \, \boldsymbol{a_{Gx}}$$

السرعة الخطية الابتدائية (بعد الضربة):

$$\boldsymbol{v_0} = \frac{\boldsymbol{J}}{\boldsymbol{m}}$$

السرعة الخطية خلال مرحلة الانزلاق:

$$\boldsymbol{v(t)} = \boldsymbol{v_0} - (\boldsymbol{\mu_k}.\boldsymbol{g}).\boldsymbol{t}$$

التسارع الخطي (تباطؤ) خلال الانزلاق:

$$\boldsymbol{a} = \boldsymbol{\mu_k} \, . \, \boldsymbol{g} \quad — \textit{التسارع الخطي أثناء الانزلاق}$$

تسارع مقاومة التدحرج (بعد التدحرج الطبيعي):

$$\boldsymbol{a_{rolling}} = \boldsymbol{\mu_r} \, . \, \boldsymbol{g}$$

1.4. معادلات الحركة الدورانية

تصف هذه المعادلات دوران الكرة حول مركز كتلتها وفق قانون أويلر للعزوم:

قانون أويلر للحركة الدورانية:

$$\tau_G = I_G . \alpha$$

عزم القصور الذاتي للكرة الكروية المصمتة:

$$I = \frac{2}{5} . m . R^2$$

السرعة الزاوية الابتدائية (بعد الضربة):

$$w_0 = \frac{J . h}{I}$$

ارتفاع الضربة عن مركز الكرة h *حيث*

تغير السرعة الزاوية خلال الانزلاق:

$$w(t) = w_0 + \left(\frac{f_k . R}{I}\right) . t$$

1.5. قيد التدحرج النقي

التدحرج النقي (Pure Rolling): هو الحالة التي تكون فيها سرعة نقطة تلامس الكرة مع السطح مساوية للصفر. يُعبَّر عنه بالعلاقة:

$$\boldsymbol{v} = \boldsymbol{w} \,.\, \boldsymbol{R}$$

سرعة نقطة التماس بالنسبة للطاولة = صفر

أو بصيغة التسارع:

$$a_{Gx} = -\alpha \,.\, r$$

(إشارة سالبة للدوران مع اتجاه عقارب الساعة)

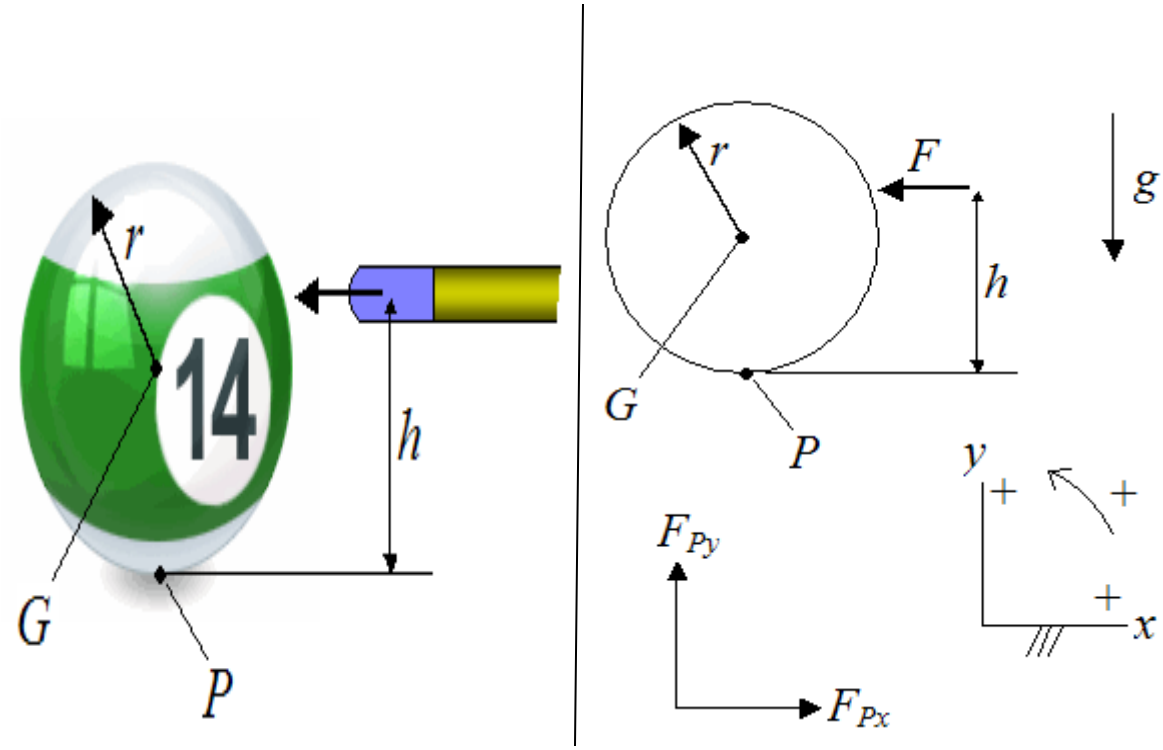
ويتبقى فقط احتكاك التدحرج الأضعف بكثير. هذه العلاقة هي المفتاح لاستنتاج f_kعند تحقق هذا الشرط، يتلاشى الاحتكاك الانزلاقي النقطة المثالية للضرب.

2. النقطة المثالية للضرب (The Sweet Spot): استنتاج مركز القرع

) الذي يجب أن تضرب فيه العصا الكرة بحيث تتدحرج فوراً دون أن ينشأ أي احتكاك أفقي h "النقطة المثالية" هي الارتفاع المحدد (Pبين الكرة وسطح الطاولة عند نقطة التلامس).

. نفرض قوة r ونصف قطرها m)، نطبق قوانين نيوتن على كرة كتلتها Free-Body Diagramبتمثيل النظام بمخطط الجسم الحر (h. تتجه يساراً على ارتفاع F ضربة العصا

) كشرط للنقطة المثالية، نحصل على F_{Px})، ومع افتراض انعدام قوة الاحتكاك (x بتطبيق قانون نيوتن الثاني على المحور الأفقي (المعادلة الأولى:



$$-\boldsymbol{F} = \boldsymbol{m}\,.\,\boldsymbol{a_{Gx}}$$

).$m.g$) وزن الكرة (F_{Py}وعلى المحور الرأسي، تلغي قوة رد الفعل (

) دون انزلاق Pure Rolling). بما أننا نريد حالة "تدحرج نقي" (G لحساب الحركة الدورانية، نكتب معادلة العزم حول مركز الكتلة (
) عبر العلاقة : a_{Gx}) بالتسارع الخطي (αنسبي عند نقطة التلامس، نربط التسارع الزاوي (

$$\alpha = -\frac{a_{Gx}}{r}.$$

)، يصبح لدينا: $M_G = I_G\,\alpha$ $\sum$بالتعويض في معادلة عزم الدوران (

). ينص القانون الأساسي على أن محصلة G من **قانون نيوتن الثاني للحركة الدورانية (قانون أويلر للعزوم)** حول مركز الكتلة (
العزوم المؤثرة على جسم تساوي عزم القصور الذاتي للجسم مضروباً في تسارعه الزاوي:

$$\sum \quad \tau_G = I_G \cdot \alpha$$

للوصول إلى معادلة "النقطة المثالية"، نقوم بتفكيك طرفي هذه المعادلة بناءً على معطيات الضربة:

1τ_G): صياغة الطرف الأيسر (العزم الميكانيكي المؤثر

- يُحسب العزم الدوراني بضرب القوة الأفقية المطبقة بواسطة عصا البلياردو (F) في **ذراع القوة** (**Lever Arm**).
- بما أن القوة تؤثر على ارتفاع (h) من سطح الطاولة، ومركز كتلة الكرة يقع على ارتفاع يوازي نصف القطر (r)، فإن المسافة العمودية الصافية بين خط عمل القوة ومحور الدوران هي ($h - r$).

وعليه، يكون العزم الناشئ عن الضربة:

$$\sum \quad \tau_G = F \cdot (h - r)$$

2$I_G\alpha$): صياغة الطرف الأيمن (القيد الديناميكي للتدحرج النقي

- لحظة التصادم (أي انتقال الكرة للأمام دون أي انزلاق على سطح الطاولة)، يجب أن **(Pure Rolling**لتحقيق حالة **التدحرج النقي (**) بعلاقة هندسية صارمة. يرتبط التسارع الخطي لمركز الكتلة (a_{Gx}) بالتسارع الزاوي (α).
- فإن التدحرج يتطلب دوراناً مع اتجاه عقارب الساعة. في $+x$ بافتراض أن التقدم للأمام على المحور الأفقي يمثل الاتجاه الموجب ( )، الاصطلاح الرياضي القياسي، الدوران مع عقارب الساعة يُعبر عنه بإشارة سالبة، مما يجعل معادلة القيد الحركي:

$$a_{Gx} = -\alpha \cdot r$$

- بإعادة ترتيب المعادلة رياضياً لاستخراج التسارع الزاوي (α):

$$\alpha = -\frac{a_{Gx}}{r}$$

- بتعويض هذه القيمة في الشق الأيمن من قانون العزوم العام، نصل إلى تعبير الاستجابة الدورانية:

$$I_G\alpha = I_G\left(-\frac{a_{Gx}}{r}\right)$$

بمساواة العزم المطبق (الطرف الأيسر) بالاستجابة الحركية المقيدة (الطرف الأيمن) بناءً على القانون التأسيسي، تتشكل لدينا المعادلة الفيزيائية التي تعبر عن التوازن الدقيق بين القوة الدافعة وعزم الدوران اللازم لمنع الانزلاق:

$$\boldsymbol{F\,.\,(h-r)\ =\ I_G\,(-\frac{a_{Gx}}{r})}$$

)، نحصل على ارتفاع الضربة:$a_{Gx}\ =\ -\frac{F}{m}$بقسمة هذه المعادلة على المعادلة الأولى بعد استخراج (

$$\boldsymbol{h\ =\ \frac{I_G}{mr}\ +\ r}$$

. بتعويض هذه القيمة، نصل إلى $I_G\ =\ \frac{2}{5}mr^2$بما أن الكرة البلياردو هي جسم كروي مصمت، فإن عزم قصورها الذاتي هو الاستنتاج الرياضي النهائي للنقطة المثالية:

$$\boldsymbol{h\ =\ \frac{7r}{5}}$$

ضعف نصف قطرها من سطح الطاولة، فإنها ستنطلق بتدحرج نقي فوراً 1.4هذا يعني أنه إذا ضُربت الكرة على ارتفاع يعادل بالضبط دون أي انزلاق، بغض النظر عن قوة الضربة. إذا ضُربت أعلى أو أسفل هذا الارتفاع، سيتطلب الأمر قوة احتكاك لمنع الانزلاق، وإذا) حيث لا تتوازن السرعة المماسية مع سرعة مركز الكتلة.Relative Slippingكانت الضربة قوية، سيحدث انزلاق فعلي (

لقد أتممنا الآن التأسيس الفيزيائي الكامل للنقطة المثالية للضرب.

الخطوة التالية: فيزياء تصادم كره مع العصا

---

3. فيزياء تصادم كره مع العصا

3.1. من النبضة إلى السكون

تعتبر حركة كرة البلياردو نموذجاً مثالياً لدراسة **الديناميكا الكلاسيكية،** حيث تتداخل القوى الخطية مع العزوم الدورانية لتشكّل سلوكاً معقداً يختلف تماماً عن مجرد " انزلاق جسم على سطح " .

3.1.1. مرحلة التصادم اللحظي (The Impact Phase)

تبدأ الرحلة بضربة العصا. في الفيزياء، لا نتعامل مع هذه القوة كدفع مستمر، بل **كنبضة** (Impulse)، وهي قوة كبيرة جداً تؤثر خلال زمن متناهي الصغر (Δt).

القوانين الحاكمة:

السرعة الخطية الابتدائية:

$$v_0 = \frac{J}{m}$$

حيث J هو النبضة، و m كتلة الكرة

السرعة الزاوية الابتدائية:

$$w_0 = \frac{J.h}{I}$$

حيث h الارتفاع عن المركز، و $I = \frac{2}{5}.m.R^2$ عزم القصور الذاتي

الوصف الفيزيائي:

يحدد موضع الضربة "شخصية" الحركة. إذا كانت الضربة فوق المركز، تبدأ الكرة بالدوران للأمام (Topspin)؛ وإذا كانت تحت المركز، تدور للخلف (Backspin).

3.1.2. مرحلة الانزلاق (The Sliding Phase)

بمجرد انفصال العصا، تبدأ الكرة حركتها على الطاولة. في ٩٠٪ من الضربات، لا تبدأ الكرة بالتدحرج فوراً بل تنزلق، وهنا تظهر قوة الاحتكاك الحركي (f_k). كلاعب رئيسي

القوانين الحاكمة:

قوة الاحتكاك:

$$f_k = \mu_k . m . g$$

معامل احتكاك القماش μ_k *حيث*

تباطؤ السرعة الخطية:

$$v(t) = v_0 - (\mu_k . g) . t$$

تغير السرعة الزاوية بفعل الاحتكاك:

$$w(t) = w_0 + \left(\frac{f_k . R}{I}\right) . t$$

الوصف الفيزيائي:

الاحتكاك هنا يعمل معيقاً للسرعة الخطية ومحرّكاً للسرعة الزاوية. إذا كانت الكرة تنزلق دون دوران، يُبطئ الاحتكاك حركتها للأمام مع إجبارها تدريجياً على الدوران.

3.1.3. نقطة التحول: التدحرج الطبيعي (Natural Roll)

هذه هي اللحظة التي يتوقف فيها الانزلاق تماماً وتصبح الكرة في حالة "تناغم" مع سطح الطاولة.

شرط التدحرج الطبيعي:

$$v = w \cdot R$$

سرعة نقطة التماس بالنسبة للطاولة = صفر

الوصف الفيزيائي:

عند هذه النقطة تصبح سرعة النقطة السفلية من الكرة (الملامسة للطاولة) صفراً بالنسبة للطاولة هنا، ويتلاشى الاحتكاك الانزلاقي العنيف (f_k) ويختفي تماماً.

3.1.4. مرحلة التدحرج والتباطؤ النهائي (Rolling & Resistance)

بعد الوصول للتدحرج الطبيعي، تستمر الكرة في الحركة لكن بتباطؤ أبطأ بكثير من مرحلة الانزلاق.

القانون الحاكم:

مقاومة التدحرج :

$$a_{rolling} = \mu_r \cdot g$$

μ_k معامل مقاومة التدحرج، وهو أصغر بكثير من μ_r حيث

الوصف الفيزيائي:

السبب في توقف الكرة هنا ليس الاحتكاك السطحي، بل تشوّه بسيط في ألياف القماش تحت ثقل الكرة، مما يُنشئ تلة مجهرية أمامها تعيق حركتها.

3.1.5. حالة السكون التام (The Full Stop)

النهاية الفيزيائية للرحلة؛ تتوقف الكرة توقفاً تاماً.

القوانين الحاكمة: لا توجد قوانين حركة هنا، بل هي حالة "الصفر المطلق" للمتجهات.

3.2. الاستنتاج الفيزيائي لكل مرحلة

مرحلة التصادم اللحظي — الاشتقاق الرياضي 3.2.1.

، لكن هذا القانون يصف **الحالة** بعد بدء الحركة. في لحظة الضرب، نحن نستخدم قانون نيوتن الثاني بصيغته $v = \frac{d}{t}$ أنت تعرف أن الأصلية:

أ. السرعة الخطية الابتدائية (v_0):

قانون نيوتن: القوة تساوي معدل تغير كمية الحركة:

$$F = \frac{dp}{dt} = m.\frac{dv}{dt}$$

وتكاملهما: بضرب الطرفين في dt

$$\int F\,dt = m\int dv$$

بما أن الكرة كانت ساكنة: $m.(v - v_{initial})$ ، والطرف الأيمن هو J) يُعرَّف بالنبضة $F\,dt$ ∫الطرف الأيسر (

$$v_0 = \frac{J}{m} \Rightarrow J = m.v_0$$ — *السرعة الخطية الابتدائية*

ب. السرعة الزاوية الابتدائية (ω_0):

نفس المنطق ينطبق على الدوران. القانون الدوراني لنيوتن هو:

$$\tau = I.\alpha$$

حيث τ العزم، و α التسارع الزاوي

). بالتكامل: τ = F·h العزم ينتج من ضرب القوة في البعد عن المركز (

$$\int (F.h)dt = I \int dw$$

بما أن h: ثابتة تخرج خارج التكامل

$$h.\int F\,dt = I.w_0 \Rightarrow h.J = I.w_0$$

$$w_0 = \frac{J.h}{I}$$ — *السرعة الزاوية الابتدائية*

3.2.2. مرحلة الانزلاق — الاشتقاق الرياضي

هنا ننتقل من "الضربة" إلى "تأثير الطاولة".

أ. قوة الاحتكاك (f_k):

الاحتكاك لا يعتمد على السرعة، بل على "قوة التلامس".

الكرة تضغط للأسفل بثقل mg: القوة هو نسبة من هذه الاحتكاك والطاولة ترد بنفس القوة للأعلى. القانون التجريبي للاحتكاك

$$f_k = \mu_k.m.g$$ — *قوة الاحتكاك الحركي*

ب. التباطؤ الخطي (a):

من قانون نيوتن الثاني F = m·a: القوة الوحيدة التي تبطئ الكرة أفقياً هي الاحتكاك،

$$-f_k = m.a \Rightarrow -(\mu_k.m.g) = m.a$$

بتبسيط المعادلة (قسمة الطرفين على m):

التسارع الخطي أثناء الانزلاق — $\boldsymbol{a = \mu_k . g}$

ج. تغير السرعة الزاوية:

الاحتكاك يؤثر على حافة الكرة (على بعد R: عن المركز)، مما يُنشئ عزماً دورانياً

$$\tau_{friction} = f_k . R$$

من قانون τ = I·α: يكون التسارع الزاوي

$$\alpha = \frac{f_k . R}{I}$$

بما أن الاحتكاك يعمل على تدوير الكرة المنزلقة، تزداد السرعة الزاوية مع الزمن:

السرعة الزاوية أثناء الانزلاق — $\boldsymbol{w(t) = w_0 + \left(\frac{\mu_k . m . g . R}{I}\right) . t}$

3.2.3. نقطة التحول — اشتقاق شرط التدحرج الطبيعي

لماذا نستخدم شرط $v = \omega \cdot R$؟

لتحليل هذه المرحلة، نتأمل النقطة الملامسة للطاولة أسفل الكرة. هذه النقطة تمتلك مركّبتين للسرعة:

- v: سرعة للأمام بسبب حركة الكرة الانتقالية الكلية
- ω · R: سرعة للخلف بسبب دوران الكرة حول مركزها

:لكي تتوقف الكرة عن الانزلاق، يجب أن تكون محصلة سرعة هذه النقطة بالنسبة للطاولة **صفراً**

$$v_{point} = v - (w \cdot R) = 0$$

شرط التدحرج الطبيعي — $v = w \cdot R$

3.2.4. التباطؤ النهائي — مقاومة التدحرج (Rolling Resistance)

عندما تتدحرج الكرة، لا يوجد "انزلاق"، إذن لماذا تتوقف؟

-) تنزاح Nالسبب ليس احتكاكاً بالمعنى التقليدي، بل هو **تشوه مرن**. الكرة تضغط القماش قليلاً، مما يجعل القوة الضاغطة (قليلاً للأمام ولا تمر بالمركز تماماً. هذا الانزياح البسيط يخلق "عزماً معوقاً".

-) لوصف التباطؤ التدريجي:μ_rبسبب التشوه المرن في ألياف القماش، نستخدم معامل مقاومة التدحرج (

تسارع مقاومة التدحرج — $a_{rolling} = -\mu_r \cdot g$

.لقد أتممنا الآن التأسيس الفيزيائي الكامل لحركة الكرة المنفردة مع العصا

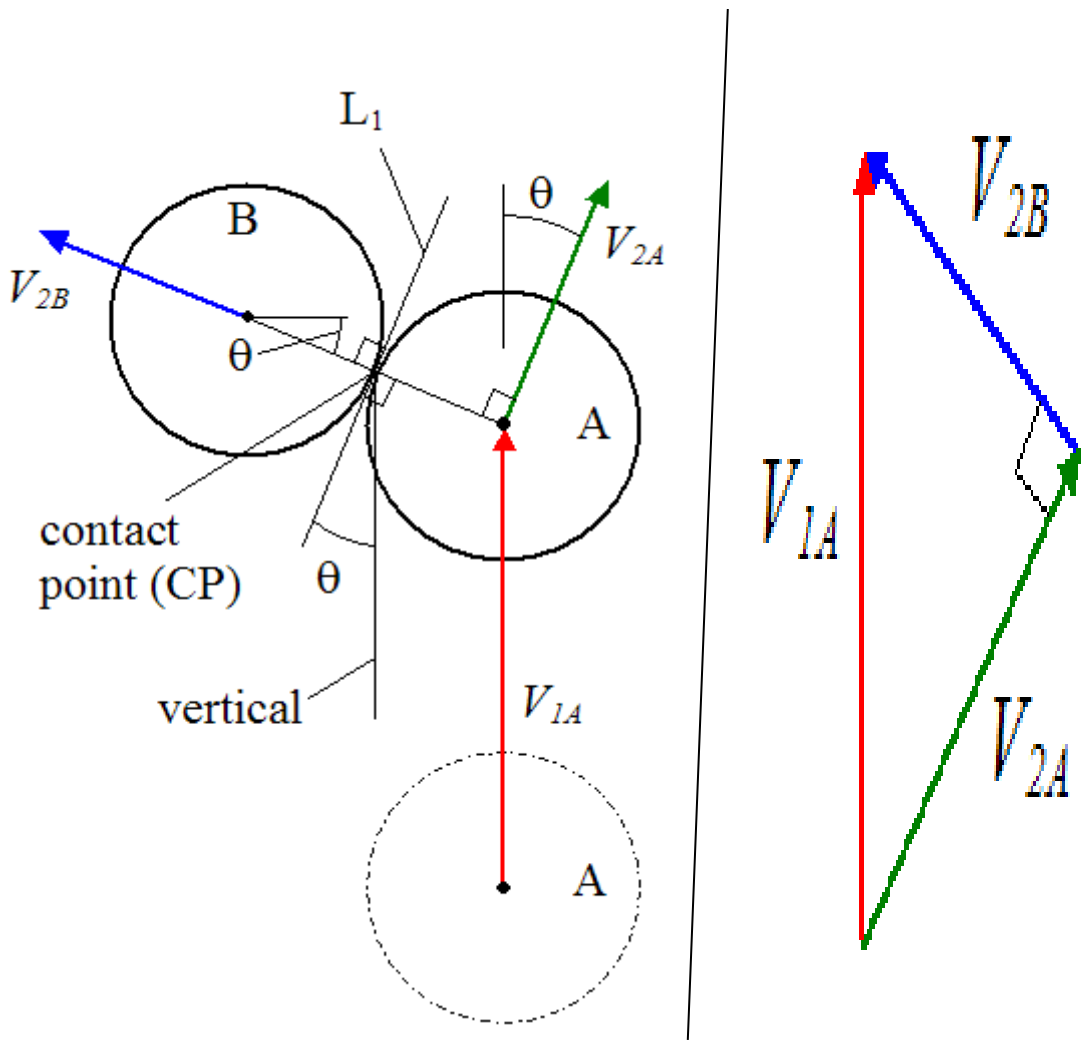
(Collision Physics)الخطوة التالية: فيزياء تصادم الكرات (

---

4. فيزياء تصادم الكرات

1. لغة الهندسة والزخم

) هي الجوهر الحقيقي للعبة البلياردو. بينما تتحكم القوى الفردية في حركة الكرة Ball-to-Ball Collisionتعتبر مرحلة تصادم الكرات (الواحدة، فإن التصادم هو اللحظة التي يتم فيها إعادة توزيع "الطاقة" و"كمية الحركة" بين الأجسام، مما يخلق شبكة معقدة من المسارات التبادلية.



2. هندسة التلامس: خط المراكز (The Line of Centers)

عندما تصطدم الكرة الضاربة (A) بالكرة الهدف (B) بالكرة الهدف (A) **هندسة التلامس**، فإن التفاعل لا يحدث بشكل عشوائي، بل يتحدد بالكامل من خلال **هندسة التلامس**.

- **التحليل الفيزيائي:** في لحظة التلامس، يتم رسم خط وهمي يمر بمركزي الكرتين ونقطة التلامس (CP). هذا الخط يسمى **المحور الناظم (Normal Axis)**. أما الخط العمودي عليه عند نقطة التلامس فيسمى **المحور المماسي (Tangential Axis)**.

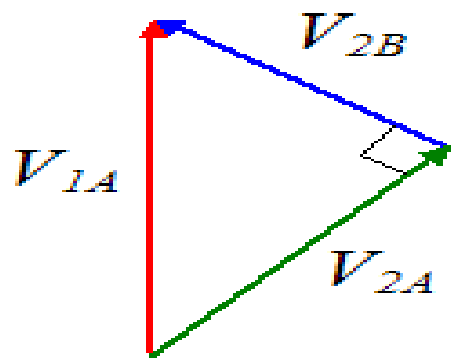
- **القاعدة الأساسية:** في النموذج المثالي (بدءاً من الصور التي أرفقتها)، نفترض عدم وجود احتكاك بين الكرتين. هذا يعني أن لا يمكنها التحرك إلا في الاتجاه الذي يدفعه فيه B"النبضة" أو القوة لا يمكن أن تنتقل إلا عبر **خط المراكز** فقط. أي أن الكرة نحو مركزها.Aمركز الكرة

3. قوانين الحفظ (Conservation Laws)

تقريباً، وهو ما يفتح الباب لاستخدام أقوى قانونين في الميكانيكا **(Elastic Collision**التصادم في البلياردو يُصنف كـ **تصادم مرن (** الكلاسيكية:

- **قانون حفظ كمية الحركة الخطية (Momentum)**

، فإن متجه السرعة قبل التصادم يجب أن يساوي مجموع متجهات السرعة بعد التصادم:$(m_A = m_B)$بما أن الكتل متساوية



$$(V_{1A})^2 = (V_{2A})^2 + (V_{2B})^2$$

- **قانون حفظ الطاقة الحركية (Kinetic Energy)**

باعتبار التصادم مرناً تماماً، فإن الطاقة لا تضيع (لا تتحول لحرارة أو تشوه دائم):

$$\frac{1}{2}m(V_{1A})^2 = \frac{1}{2}m(V_{2A})^2 + \frac{1}{2}m(V_{2B})^2$$

$(V_{1A})^2 = (V_{2A})^2 + (V_{2B})^2$بالتبسيط، نحصل على علاقة فيثاغورس:

4. قاعدة الـ 90 درجة (The 90-Degree Rule)

هذه هي النتيجة الأكثر إثارة في فيزياء البلياردو. من خلال دمج قانون حفظ الزخم مع قانون حفظ الطاقة، نستنتج رياضياً أن الزاوية بين بعد التصادم يجب أن تكون 90 درجة دائمًا (ما لم يكن التصادم رأسياً تماماً).B ومسار الكرة Aمسار الكرة

(فيثاغورس) مع معادلة المتجهات، يجب أن يكون المتجهان $(V_{1A})^2 = (V_{2A})^2 + (V_{2B})^2$ **الاستنتاج:** لكي تتحقق معادلة متعامدين. هذا هو "المفتاح" الذي يستخدمه المحترفون لتوقع مكان استقرار الكرة البيضاء بعد ضرب الكرة الهدف. V_{2B} و V_{2A}

5. الحالات الخاصة

- **التصادم المباشر (Head-on):** عندما تكون مراكز الكرتين على استقامة واحدة مع اتجاه الحركة. هنا تنتقل الطاقة بنسبة 100% من A إلى B. تتوقف A وتتحرك B تماماً بنفس سرعة A الابتدائية.

- **التصادم الزاوي (Glancing Blow):** وقلت A، زادت السرعة التي تحتفظ بها B عن مركز الكرة A كلما ابتعدت الكرة ، مع بقاء الزاوية بينهما 90 درجة. السرعة المنتقلة إلى B

لقد أتممنا الآن التأسيس الفيزيائي الكامل لصدم كرتين مع بعضهما البعض

الخطوة التالية: ديناميكا التصادمات المتعددة وسلاسل التفاعل (Multi-Body Dynamics)

عند الانتقال من تصادم كرتين فقط إلى تصادم مجموعة كرات متراصة — كما في ضربة البداية (The Break Shot — تتحول الحركة إلى شبكة معقدة تتوزع فيها الطاقة والزخم عبر نقاط التلامس بين الكرات.

1. التوزيع المتناظر للزخم (Symmetric Momentum Distribution)

في الحالة النظرية المثالية، عندما تضرب الكرة البيضاء قمة المثلث في المنتصف تماماً (تصادم مركزي)، فإن طاقة حركتها تتوزع على جميع الكرات الـ 15 بناءً على قانون **حفظ كمية الحركة (الزخم)**. هذا يعني أن طاقة الكرة البيضاء قبل التصادم يجب أن تساوي مجموع طاقات كل الكرات بعد التصادم.

الاستنتاج الفيزيائي (Deduction:)

لنبدأ من **قانون نيوتن الثالث لظاهرة الفعل ورد الفعل**. عند لحظة الصدمة بين أي كرتين (i و j) داخل المثلث، فإن القوة المتبادلة بينهما تكون متساوية في المقدار ومتعاكسة في الاتجاه:

$$\boxed{F_{ij} = -F_{ji}}$$

بإجراء عملية التكامل الزمني لهذه القوة عبر فترة التلامس المتناهية في الصغر ($\Delta t \; v \to 0$)، نحصل على النبضة الخطية المتبادلة (J):

$$J_{ij} = \int_0^{\Delta t} F_{ij}\, dt = -J_{ji}$$

) ينعدم تفاضلياً:Internal Forcesإذا نظرنا إلى المثلث كاملاً كـ **جملة ميكانيكية مغلقة**، فإن مجموع القوى الداخلية (

$$\sum_{i=1}^{15} \sum_{j=i} F_{ij} = 0$$

$\int f\, dt \approx$ تؤول إلى الصفر (Δt وبما أن النبضات الخارجية الناشئة عن الاحتكاك مع السطح خلال زمن الصدمة الميكروي :)، فإن المشتقة الزمنية للزخم الكلي للنظام تساوي صفراً، مما يعني أن **الزخم الخطي الكلي للجملة هو كمية محفوظة**0

$$\boldsymbol{\frac{dP_{total}}{dt} = 0 \Rightarrow P_{intial} = P_{final}}$$

)، فإن الزخم الابتدائي محصور في الكرة الضاربة (البيضاء) ذات الكتلة $v_i = 0$باعتبار أن جميع الكرات الخلفية في حالة سكون تام (. ومنه نستنتج دالة التوزيع الآني للزخم بعد التشتت:v_0 والسرعة m

$$\boldsymbol{P_{intial} = m\, v_0 = \sum_{i=1}^{15} m_i\, v_i}$$

- : الزخم الكلي الابتدائي للنظام. $\boldsymbol{P_{intial}}$
- : كتلة الكرة (بافتراض أن جميع الكرات متطابقة الكتلة). $\boldsymbol{m}$
- : سرعة الكرة البيضاء قبل الاصطدام مباشرة. $\boldsymbol{v_0}$
- : سرعة كل كرة بعد تفكك المثلث. $\boldsymbol{v_i}$

2. التصادم غير المركزي وسلاسل القوة (Force Chains)

)، يتغير سلوك الكرات بناءً على مبدأين:Off-centerفي الواقع، نادراً ما تكون الضربة في المركز تماماً. عندما تنحرف الضربة (

- **سلاسل القوة (Force Chains):** الطاقة لا تنتقل عشوائياً، بل تسير فقط في خطوط مستقيمة تمر بمراكز الكرات المتلامسة (الخط الناظم المشترك). الكرات التي تقع خارج هذه الخطوط الهندسية لن تتحرك مطلقاً لأنها معزولة مادياً عن مسار القوة.
- **عتبة الاحتكاك الساكن:** قماش الطاولة يقاوم الحركة. لكي تتحرك أي كرة من سكونها، يجب أن تكون القوة التي تصلها من الصدمة (F_{impact}) أكبر من أقصى قوة احتكاك ساكن يمكن أن يتحملها القماش:

الاستنتاج الفيزيائي (Deduction:)

في الضربات غير المركزية (Off-center Impacts)، يجب أن نخضع المنظومة لـ **شروط قيد الصلابة الهندسية (Signorini Contact Conditions)**. الكرة الصلبة لا يمكنها تداخل محيط كرتين؛ لذا فإن انتقال القوة محكوم بالمتجه الناظمي المشترك ($\widehat{n_{ij}}$) الواصل بين مراكز الكرات المتلامسة حصراً.

عندما تتوزع القوة هندسياً داخل المثلث متساوي الأضلاع، فإن زاوية التفرع بين مراكز الكرات المتراصة تكون دائماً $30°$ عن محور الدفع الأساسي. طبقاً لتحليل المركبات الهندسية، فإن القوة الناظمية المنقولة من كرة "أب" إلى كرتين "أبناء" تتوزع وفق المتطابقة المثلثية:

$$\boldsymbol{F_{transmitted} = F_{parent} \cdot cos(30°)}$$

وفق k ، فهذا يعني استنتاجياً أن القوة **تضمحل أسياً** كلما تعمقت عبر مستويات المثلث ($\boldsymbol{cos(30°) = \frac{\sqrt{3}}{2} \approx 0.866}$) بما أن الدالة الميكانيكية $F(k)\ \alpha\ (0.866)^k$.

وهنا نطبق **قانون كولوم للاحتكاك الجاف**. لكي تنتقل أي كرة طرفية من حالة السكون إلى الحركة، يجب أن تتغلب القوة الناظمية المتبقية $F(k)$ ومعامل احتكاك اللباد g على حد العتبة الأعلى للاحتكاك الساكن الناشئ عن الجاذبية الأرضية μ_s:

$$\boldsymbol{F_{impact} > f_{s(max)} = \mu_s \cdot m \cdot g}$$

- μ_s : معامل الاحتكاك الساكن للقماش.
- $\boldsymbol{g}$: الجاذبية الأرضية.

إذا كان عمق السلسلة الميكانيكية كبيراً بحيث تفرعت القوة وأصبحت $F(k) \leq \mu_s \cdot m \cdot g$ ، فإن الشغل المبذول يعجز عن إحداث أي إزاحة خطية ($\Delta x = 0$)، وتتحول القوة الصدمية بالكامل إلى طاقة انضغاط مرن مجهري وتبدد صوتي، وتظل الكرة ساكنة تماماً في إحداثياتها.

3. التبدد التراكمي للطاقة (Energy Dissipation)

رغم أننا نفترض أن التصادم مرن، إلا أن الكرات في الواقع تفقد جزءاً من طاقتها عند كل تلامس بسبب **معامل الارتداد (e)** الذي يكون دائماً أقل من الواحد الصحيح ($e < 1$).

عندما تنتقل الصدمة من صف إلى صف داخل المثلث، تضيع الطاقة تدريجياً بسبب عاملين:

1. **التشوه المرن المؤقت:** تنضغط الكرات مجهرياً بجزء من الملي ثانية ثم ترتد، هذا الانضغاط يستهلك طاقة تتحول إلى صوت (طقطقة الكرات) وحرارة مجهرية.
2. **مقاومة السطح:** الشغل المبذول للتغلب على مقاومة الالتصاق بين أسطح الكرات.

هذا الفقد المتراكم يفسر رياضياً لماذا تبطئ الكرات بشكل حاد كلما غصنا في عمق المثلث، وتتوقف المنظومة أسرع بكثير مما تتوقعه المعادلات المثالية.

الاستنتاج الفيزيائي (Deduction:)

لإثبات فقدان الطاقة، نبتعد عن فرضية المرونة التامة ونستنتج القوانين بناءً على **قانون نيوتن التجريبي للارتداد**. يُعرف معامل الارتداد (e) بأنه النسبة بين السرعة النسبية للأجسام بعد التصادم إلى سرعتها النسبية قبل التصادم على طول المحور الناظمي:

$$\boldsymbol{e = -\frac{v_{j(out)} - v_{i(out)}}{v_{j(in)} - v_{i(in)}}}$$

في تصادم ثنائي واحد لكرتين متماثلتي الكتلة (m) حيث في الأجسام الواقعية تكون ($e < 1$). لنستنتج مقدار الطاقة الحركية المتبددة ΔK:

من دمج معادلة معامل الارتداد مع معادلة حفظ الزخم، نجد أن التغير في الطاقة الحركية للنظام يُحسب استنتاجياً بالعلاقة الرياضية التالية:

$$\boldsymbol{\Delta K = K_{in} - K_{out} = \frac{1}{4} m(1 - e^2)(v_{rel-in} \cdot \hat{n})^2}$$

سيكون دائماً قيمة موجبة أكبر من الصفر، مما يثبت رياضيّاً أن **الطاقة الحركية الكلية** $(1 - e^2)$، فإن المقدار $e < 1$ بما أن **للنظام في تناقص مستمر مع كل تلامس**.

من التصادمات المتتالية، فإن الطاقة المنتقلة للصفوف الأخيرة تخضع لـ **تضاؤل هندسي** k وعندما تسير الصدمة عبر سلسلة مكونة من . هذا التبدد التراكمي هو الشغل الميكانيكي المستهلك في إعادة هيكلة الجزيئات السطحية للكرة مجهرياً أثناء e^{2k} **مركّب** قيمته ترتبط بـ)، وتحويله إلى تبدد حراري وصوتي. Δt زمن التلامس (

لقد أتممنا الآن التأسيس الفيزيائي الكامل لتصادم الكرات المتعدد

الخطوة التالية: ديناميكا التصادم مع الجدران المطاطية (Cushion & Rail Physics)

)، ننتقل من دراسة تصادم الأجسام الصلبة المتماثلة إلى تصادم جسم The Cushion عندما تصطدم الكرة بالإطار المطاطي للطاولة (صلب (الكرة) مع سطح مرن يمتص الطاقة (المطاط). هذا الاصطدام يعيد تشكيل متجهات السرعة الخطية والدورانية بناءً على ثلاثة عوامل أساسية:

1. التصادم العمودي وفقدان طاقة الانضغاط (Normal Cushion Impact)

)، ينضغط المطاط مرناً لفترة قصيرة جداً ويمتص جزءاً من v_{in} عندما تسقط الكرة بشكل عمودي تماماً على الجدار بسرعة ابتدائية ((e_c): يتم حسابها بناءً على معامل ارتداد المطاط (v_{out} طاقة الحركة. نتيجة لذلك، ترتد الكرة بسرعة أقل (

الاستنتاج الفيزيائي (Deduction:)

للحركة الخطية على المحور العمودي (الناظمي) على **(Impulse-Momentum Theorem** لنبدأ من **معادلة النبض وكمية الحركة (** الجدار $\hat{n}$:

$$\boldsymbol{J_n = \int_0^{\Delta t} F_n(t)dt = \Delta p_n = m(v_{out,n} - v_{in,n})}$$

). e_cبما أن الجدار يمثل قيداً ميكانيكياً ثابتاً، فإننا نطبق **قانون نيوتن للتصادمات التجريبي** عبر إدخال معامل ارتداد الحائل المطاطي (يُعرّف المعامل على أنه نسبة السرعة النسبية بعد التصادم إلى السرعة النسبية قبله:

$$\boldsymbol{e = -\frac{v_{out.n} - v_{cushion}}{v_{in,n} - v_{cushion}}}$$

)، يؤول القانون مباشرة إلى العلاقة الخطية التالية:$v_{out,n} = -e_c \cdot v_{in,n}$وباعتبار الجدار ثابتاً استاتيكياً في فضاء المحاكاة (

$$\boldsymbol{v_{out,n} = -e_c \cdot v_{in,n}}$$

) قبل وبعد الصدم العمودي:ΔK_nولإثبات الطاقة المتبددة ميكانيكياً، نحسب التغير في الطاقة الحركية الانسحابية (

$$\boldsymbol{\Delta K_n = \frac{1}{2} m v_{in,n}{}^2 - \frac{1}{2} m v_{out,n}{}^2 = \frac{1}{2} m v_{in,n}{}^2 (1 - e_c{}^2)}$$

. هذا الحساب يثبت رياضياً أن النقص $(1 - e_c{}^2) > 0$ دائماً، مما يجعل المقدار $e_c < 1$بما أن المطاط مادة ممتصة صدمياً، فإن) وتحويله Hysteresis Lossفي السرعة ليس مفقوداً في الفراغ، بل هو الشغل الديناميكي المستهلك في كبس الروابط الجزيئية للمطاط (إلى طاقة حرارية وتشوه لدن مؤقت.

2. انحراف زاوية الارتداد بسبب الاحتكاك (Oblique Impact & Rail Friction)

) بالنسبة للعمود المقام على الجدار. في الفيزياء النظرية المجرّدة، تكون زاوية السقوط θ_{in}في الضربات المائلة، تسقط الكرة بزاوية (مساوية لزاوية الانعكاس. أما في الواقع، فيفرض الجدار المطاطي قوة احتكاك مماسية تعاكس حركة الكرة الموازية له خلال زمن التلامس (Δt).

) وفي نفس الوقت يفرمل السرعة الموازية بفعل الاحتكاك، e_cولأن المطاط يمتص السرعة العمودية بشكل كبير (بفعل معامل الارتداد) أكبر دائماً من زاوية السقوط:θ_{out}فإن المحصلة تجعل زاوية الخروج الحقيقية (

$$\theta_{out} > \theta_{in}$$

هذا يعني برمجياً وفيزيائياً أن الكرة ترتد دائماً بزاوية "أكثر انفتاحاً واقتراباً من الجدار" نحو عمق الطاولة مقارنة بالمسار الهندسي المثالي.

Deduction:(الاستنتاج الفيزيائي (

v_n)، نقوم بتفكيك ناقل السرعة الابتدائية إلى مركبتين فوق محاور متعامدة: مركبة ناظمية Oblique Impactsفي التصادمات المائلة (
.v_tومركبة مماسية

) بالنسبة للمحور العمودي عبر العلاقة:θ_{in}تُحدد زاوية السقوط النظري (

$$tan(\theta_{in}) = \frac{v_{in,t}}{v_{in,n}}$$

موازية لسطح الجدار. **(f_t)**، يولد انضغاط المطاط قوة عمودية تنتج عنها فوراً **قوة احتكاك مماسية كولومبية (**Δtأثناء زمن التلامس
) كالآتي: J_tيُحسب نبض الاحتكاك المماسي (

$$J_t = \int_0^{\Delta t} f_t(t)dt = \mu_c.J_n$$

هذا النبض المماسي يفرمل السرعة المماسية للكرة لتصبح بعد الارتداد:

$$v_{out,t} = v_{in,t} - \frac{J_t}{m}$$

:(θ_{out}الآن، لنحسب زاوية الارتداد الحقيقية (

$$tan(\theta_{out}) = \frac{v_{out,t}}{|v_{out,n}|} = \frac{v_{in,t} - \frac{J_t}{m}}{e_c \cdot v_{in,n}}$$

، بينما 0.75 إلى 0.6 منخفضة تقارب e_cفي جدران البلياردو الواقعية، يكون امتصاص المطاط للمركبة العمودية كبيراً جداً (قيمة) يكون أقوى بكثير من انخفاض البسط بفعل e_cيكون كبح السرعة المماسية الناتجة عن الاحتكاك أقل نسبياً. رياضياً، انخفاض المقام (الاحتكاك، مما يؤدي إلى زيادة القيمة الإجمالية للكسر:

$$tan(\theta_{out}) > tan(\theta_{in}) \Rightarrow \theta_{out} > \theta_{in}$$

هذا الاشتقاق يثبت فيزيائياً أن الكرة مجبرة ميكانيكياً على الارتداد بزاوية أكثر انفتاحاً وانحرافاً نحو عمق الطاولة مقارنة بمسار السقوط.

3. تأثير الدوران المغزلي على استجابة الجدار (Spin & Cushion Coupling)

) لحظة اصطدامها بالجدار، فإن هذا الدوران يتداخل ميكانيكياً مع المطاط ليغير مسار وسرعة Spinإذا كانت الكرة تدور حول نفسها (الارتداد بشكل مفاجئ وحاد:

): تدور الكرة في نفس اتجاه الحركة، مما يقلل الاحتكاك مع المطاط. هذا يؤدي إلى Running Englishالدوران الجانبي المتوافق (تسريع الكرة خطياً بعد الارتداد وتوسيع زاوية الخروج بشكل ملحوظ (زاوية منفتحة جداً).

): تدور الكرة عكس اتجاه الحركة، مما يخلق قوة كبح مماسية هائلة تفرمل السرعة Reverse Englishالدوران الجانبي المعاكس (الخطية وتجبر الكرة على الارتداد بزاوية ضيقة جداً (حادة)، وفي بعض الحالات المتقدمة، قد ترتد الكرة على نفس خط مسار سقوطها.

) تُحسب عبر تكامل عزم الدوران الناشئ عن قوى الاحتكاك السطحي اللحظية المؤثرة على ω_{out}السرعة الزاوية الجديدة بعد الارتداد (محيط الكرة.

الاستنتاج الفيزيائي (Deduction:)

ميكانيكا الأجسام الصلبة المقترنة (Spin ωعندما تمتلك الكرة سرعة زاوية (**Coupled Rigid-Body**) لحظة التصادم، فإننا ننتقل إلى v_c). لحساب القوى بدقة، يجب أولاً تحديد السرعة النسبية للحافة الميكانيكية للكرة عند نقطة التلامس اللحظية مع الجدار ((**Dynamics**):

$$v_c = v_t + w \times R$$

f_t هو متجه نصف قطر الكرة الواصل لنقطة التلامس. بناءً على هذا القانون، فإن اتجاه ومقدار قوة الاحتكاك المماسية Rحيث إن يصبح تابعاً مباشراً لسرعة نقطة التلامس وليس للسرعة الخطية لمركز الكتلة فقط:

$$f_t = -\mu_c F_n \frac{v_c}{|v_c|}$$

يؤثر مباشرة على القشرة الخارجية للكرة وفق (τ_{Torque}هذه القوة الاحتكاكية لا تكتفي بفرملة الحركة الخطية، بل تنتج **عزم دوران (** معادلة أويلر للحركة الدورانية:

$$\tau = R \times f_t = I\frac{dw}{dt}$$

هو عزم القصور الذاتي للكرة الصلبة. ومن خلال إجراء التكامل الزمني للعزم، نستنتج التغير في الدوران $I = \frac{2}{5}mR^2$حيث إن (Δwالمغزلي):

$$w_{out} = w_{in} + \frac{1}{I}\int_0^{\Delta t} (R \times f_t)dt$$

-) في نفس اتجاه السرعة الخطية المماسية، مما يقلل من $w \times R$ يكون اتجاه الحد (**Running English):في حالة الدوران المتوافق (** ، وبالتالي يقل نبض الاحتكاك الكابح، فترتد الكرة بسرعة خطية أعلى وزاوية منفتحة جداً.$|v_c|$القيمة المطلقة لـ
-) معاكساً حاداً للسرعة المماسية، مما يضخم سرعة التلامس $w \times R$ يكون الحد (**Reverse English):في حالة الدوران المعاكس (** ليصبح قوة دفع خلفية تفرمل الكرة بعنف وتجبرها على الارتداد بزاوية حادة أو حتى العكس f_tالنسبية ويعكس اتجاه متجه الاحتكاك تماماً.

PhysicalBall Class

يمثل كلاس **PhysicalBall** النموذج البرمجي للكرة داخل محاكاة البلياردو الفيزيائية.
مسؤولية هذا الكلاس هي إدارة حالة الكرة الفيزيائية أثناء المحاكاة، مثل الموقع، السرعة الخطية، السرعة الدورانية، تطبيق الضربات، معالجة الاحتكاك، والانتقال إلى حالة السكون.

يعتمد التصميم على مبدأ Object Oriented Physics Simulation، حيث يتم تمثيل الكرة ككائن مستقل يمتلك بياناته الفيزيائية، بينما يتم تنفيذ الحسابات العامة داخل وحدة منفصلة PhysicsLaws.js

تم تصميم الكلاس وفق أسلوب **Modular Physics Architecture**، حيث تم فصل مسؤولية إدارة الكرة عن الحسابات الفيزيائية العامة.

الخوارزمية الأساسية المستخدمة في الكلاس هي **Discrete Time Simulation**، حيث يتم تحديث حالة الكرة على مراحل زمنية صغيرة، وفي كل مرحلة يتم حساب التأثيرات الخارجية ثم تحديث السرعة والموقع والحالة الداخلية للكائن.

هذا التصميم يسمح بإضافة تأثيرات فيزيائية جديدة مستقبلاً مثل التصادمات أو القوى الخارجية دون تعديل بنية الكلاس الأساسية.

التابع constructor

- **الاسم:** constructor (id, position, radius, mass)
- **الوظيفة:** إنشاء كائن كرة جديد وتجهيز جميع البيانات المطلوبة لبدء المحاكاة.
- **الخوارزمية المستخدمة:** Object Initialization Algorithm

خوارزمية تستخدم لتهيئة حالة الكائن عند إنشائه، حيث يتم تخزين الخصائص الأساسية وإنشاء القيم الابتدائية اللازمة لمعالجة الكائن لاحقاً.

- **آلية التنفيذ:**

يقوم التابع بتجهيز معرف الكرة، الموقع الابتدائي، السرعة الخطية والدورانية، الخصائص الفيزيائية مثل الكتلة ونصف القطر، حالة الكرة الحالية مثل النوم أو دخول الحفرة، كما يقوم بإنشاء المراجع اللازمة لاستخدامها أثناء تحديث الحركة.

التابع getKineticEnergy

- **الاسم:** getKineticEnergy()
- **الوظيفة:** إرجاع قيمة الطاقة الحركية الحالية للكرة.
- **الخوارزمية المستخدمة:** Energy Aggregation Algorithm

تعتمد على تقسيم المشكلة إلى أجزاء مستقلة، حساب كل جزء بشكل منفصل، ثم دمج النتائج للحصول على القيمة النهائية.

- **آلية التنفيذ:** يقوم التابع باستدعاء وظيفتين منفصلتين:

حساب تأثير الحركة الخطية، حساب تأثير الحركة الدورانية.
بعد ذلك يقوم بجمع النتائج وإرجاع الطاقة الكلية التي تمثل حالة حركة الكرة.

التابع applyAdvancedStrike

- **الاسم:** applyAdvancedStrike(impulseMagnitude, strikeDirection, offsetX, offsetY)
- **الوظيفة:** تطبيق ضربة عصا متقدمة على الكرة مع إمكانية التحكم بمكان تأثير الضربة لإنتاج حركة ودوران.
- **الخوارزمية الأولى:** Input Constraint / Clamping Algorithm

خوارزمية تستخدم لمنع القيم المدخلة من تجاوز الحدود المسموحة، وذلك للحفاظ على استقرار النظام ومنع الحالات غير الواقعية.

- **آلية التنفيذ:**

يتم فحص إزاحة نقطة الضرب عن مركز الكرة.

إذا تجاوزت القيمة الحد المسموح يتم تقليلها مع الحفاظ على اتجاهها.

- **الخوارزمية الثانية:** Impulse Based Velocity Update

خوارزمية تعتمد على تحويل تأثير الضربة إلى تغير مباشر في سرعة الجسم الفيزيائي.

- **آلية التنفيذ:**

يتم تحويل اتجاه الضربة إلى متجه معياري، ثم يتم تحديث سرعة الكرة بناءً على مقدار التأثير المطبق.

- **الخوارزمية الثالثة:** Local Coordinate Frame Construction

خوارزمية تستخدم لإنشاء نظام إحداثيات محلي مرتبط باتجاه الجسم أو القوة المطبقة.

- **آلية التنفيذ:**

يتم بناء محاور محلية حول اتجاه العصا لتحديد مكان نقطة التأثير بالنسبة للكرة بدقة.

- **الخوارزمية الرابعة:** Torque Impulse Calculation

خوارزمية لحساب التأثير الدوراني الناتج عن تطبيق قوة خارجية على نقطة بعيدة عن مركز الجسم.

- **آلية التنفيذ:**

يتم استخدام موقع تأثير الضربة واتجاهها لاستخراج التغير في السرعة الدورانية للكرة.

التابع applySweetSpotStrike

- **الاسم:** applySweetSpotStrike(impulseMagnitude, strikeDirection)
- **الوظيفة:** تنفيذ ضربة مثالية على الكرة باستخدام نقطة تأثير محسوبة مسبقاً.

- **الخوارزمية المستخدمة:** Wrapper Function Pattern

أسلوب برمجي يقوم بإعادة استخدام وظيفة موجودة مسبقاً مع تبسيط طريقة استدعائها.

- **آلية التنفيذ:**

يقوم التابع بحساب نقطة الضربة المثالية ثم يمررها إلى applyAdvancedStrike بدلاً من إعادة كتابة منطق الضربة.

التابع applyStandardFriction

- **الاسم:** applyStandardFriction(config, dt)
- **الوظيفة:** تحديث تأثير الاحتكاك على الكرة ومعالجة مراحل الحركة حتى التوقف، هذا هو التابع الرئيسي المسؤول عن تحديث الحالة الفيزيائية أثناء تشغيل المحاكاة.

- **الخوارزمية الأولى:** Early Exit Optimization

خوارزمية تحسين أداء تقوم بإيقاف تنفيذ الوظيفة مبكراً عندما لا تكون المعالجة مطلوبة.

- **آلية التنفيذ:**

إذا كانت الكرة متوقفة أو داخل الحفرة يتم إنهاء التابع مباشرة لتجنب العمليات الحسابية غير الضرورية.

- **الخوارزمية الثانية:** Contact Velocity Evaluation

خوارزمية تقوم بتحليل حالة نقطة التلامس لتحديد نوع حركة الجسم الحالية.

- **آلية التنفيذ:**

يتم حساب سرعة نقطة تماس الكرة مع الطاولة، ثم استخدامها لاتخاذ القرار بين حالة الانزلاق أو التدحرج.

- **الخوارزمية الثالثة:** Sliding Friction Integration Algorithm

خوارزمية محاكاة زمنية تقوم بتحديث حالة الجسم تدريجياً أثناء وجود احتكاك وانزلاق.

- **آلية التنفيذ:**

يتم حساب تأثير الاحتكاك ثم تعديل السرعة الخطية والدورانية خلال خطوة زمنية صغيرة.

- **الخوارزمية الرابعة:** Overshoot Prevention Algorithm

خوارزمية لمنع تجاوز قيمة التحديث المطلوبة أثناء العمليات العددية.

- **آلية التنفيذ:**

يتم مقارنة مقدار التغير الناتج عن الاحتكاك مع سرعة الكرة الحالية، وإذا كان أكبر يتم تقليله لمنع الاهتزازات.

- **الخوارزمية الخامسة:** State Transition Algorithm

خوارزمية تعتمد على تغيير طريقة المعالجة حسب حالة النظام الحالية.

- **آلية التنفيذ:**

يتم الانتقال بين نموذج الانزلاق ونموذج التدحرج بناءً على سرعة نقطة التلامس.

- **الخوارزمية السادسة:** Rolling Resistance Simulation

خوارزمية تحاكي فقدان الطاقة الناتج عن مقاومة التدحرج أثناء الحركة.

- **آلية التنفيذ:**

يتم تطبيق تأثير تدريجي يقلل سرعة الكرة أثناء التدحرج حتى تصل إلى حالة توقف.

- **الخوارزمية السابعة:** Damping Algorithm

خوارزمية تستخدم لتقليل قيمة متغيرة ديناميكية تدريجياً مع الزمن.

- **آلية التنفيذ:**

يتم تخفيض السرعة الدورانية بشكل مستمر لمحاكاة فقدان الطاقة الناتج عن الاحتكاك الداخلي.

- **الخوارزمية الثامنة:** Velocity Decay Algorithm

خوارزمية تستخدم لمحاكاة تأثير المقاومة الخارجية على سرعة الجسم.

- **آلية التنفيذ:**

يتم تعديل السرعة الحالية بنسبة صغيرة خلال كل خطوة زمنية لمحاكاة فقدان السرعة.

- **الخوارزمية التاسعة:** Numerical Time Integration

خوارزمية تستخدم لتحديث موقع الجسم اعتماداً على حالته الحالية والزمن.

- **<u>آلية التنفيذ:</u>**

يتم حساب الموقع الجديد بشكل تكراري في كل دورة محاكاة باستخدام السرعة الحالية والفاصل الزمني.

- **الخوارزمية العاشرة:** Sleep State Optimization

خوارزمية تستخدم في محركات الفيزياء لإيقاف تحديث الأجسام غير المتحركة.

- **<u>آلية التنفيذ:</u>**

عندما تصبح السرعات أقل من حدود معينة يتم تصفير الحركة وتحويل الكرة إلى حالة السكون لتحسين الأداء.

PhysicsWorld Class

يمثل كلاس PhysicsWorld البيئة الفيزيائية الرئيسية للمحاكاة، حيث يدير جميع العناصر الفيزيائية الموجودة داخل البيئة مثل الكرات، القوى الخارجية، التصادمات، الحفر، وحدود الطاولة.

يعتمد الكلاس على نموذج Physics Engine Architecture، حيث يتم تنفيذ دورة تحديث زمنية تقوم في كل خطوة بحساب التأثيرات الفيزيائية، تحديث حالة الأجسام، ثم معالجة التفاعلات بين العناصر.

تم تصميم PhysicsWorld كطبقة إدارة مركزية للمحاكاة، حيث يعمل كـ Simulation Controller مسؤول عن ترتيب تنفيذ الخوارزميات الفيزيائية وليس تنفيذ الحسابات التفصيلية لكل جسم.

يعتمد النظام على دورة تحديث Discrete Simulation Loop، حيث يتم في كل إطار زمني تطبيق التأثيرات الخارجية، حل التصادمات، تحديث الحالات، ثم الانتقال إلى الخطوة الزمنية التالية.

التابع constructor

- **الاسم:** constructor ()
- **الوظيفة:** تهيئة العالم الفيزيائي وإنشاء جميع الإعدادات الأساسية للمحاكاة.
    - **الخوارزمية المستخدمة:** World Initialization Algorithm

خوارزمية مسؤولة عن بناء البيئة الافتراضية قبل بدء المحاكاة، حيث يتم إنشاء الحاويات وتعريف الإعدادات والخصائص العامة للنظام.

- **آلية التنفيذ:**

يقوم التابع بإنشاء قائمة الكرات الموجودة في العالم، قائمة القوى الخارجية، إعدادات المحاكاة الفيزيائية، حدود الطاولة، مواقع الحفر، دوال الاتصال مع نظام اللعبة.

التابع addBall

- **الاسم:** addBall(ball)
- **الوظيفة:** إضافة كرة جديدة إلى العالم الفيزيائي وربطها مع النظام.
    - **الخوارزمية المستخدمة:** Object Registration Algorithm

خوارزمية تستخدم لتسجيل كائن جديد داخل نظام إدارة مركزي حتى يصبح جزءاً من دورة المعالجة.

- **آلية التنفيذ:**

يقوم التابع بربط الكرة مع مراجع العالم الفيزيائي، ثم يضيفها إلى قائمة الكرات التي تتم معالجتها أثناء التحديث.

التابع updateParameters

- **الاسم:** updateParameters(newParams)
- **الوظيفة:** تحديث إعدادات المحاكاة أثناء التشغيل.
- **الخوارزمية المستخدمة:** Configuration Merge Algorithm

خوارزمية تقوم بدمج الإعدادات الجديدة مع الإعدادات الحالية مع الحفاظ على القيم غير المعدلة.

- **آلية التنفيذ:**

يتم إنشاء نسخة جديدة من الإعدادات تحتوي على القيم القديمة والقيم الجديدة التي تم تمريرها.

التابع registerExternalForce

- **الاسم:** registerExternalForce(forceInstance)
- **الوظيفة:** إضافة قوة خارجية جديدة إلى النظام الفيزيائي.
  - **الخوارزمية المستخدمة:** Dynamic Force Registration Algorithm

خوارزمية تسمح بإضافة وحدات معالجة جديدة أثناء التشغيل بدون تعديل بنية النظام الأساسية.

- **آلية التنفيذ:**

يتم تخزين كائن القوة داخل قائمة، ثم يتم استدعاؤه لاحقاً أثناء تحديث العالم.

التابع getTotalKineticEnergy

- **الاسم:** getTotalKineticEnergy()
- **الوظيفة:** حساب مجموع الطاقة الحركية لجميع الكرات النشطة داخل العالم.
  - **الخوارزمية المستخدمة:** Iteration Aggregation Algorithm

خوارزمية تعتمد على المرور على مجموعة عناصر وتجميع قيمة مشتركة للحصول على نتيجة نهائية.

- **آلية التنفيذ:**

يقوم التابع بالمرور على جميع الكرات، يتجاهل الكرات المتوقفة أو الموجودة في الحفرة، ثم يجمع الطاقة الناتجة.

التابع update

- **الاسم:** update(dt)

- **الوظيفة:** تنفيذ دورة التحديث الفيزيائي الكاملة للعالم خلال خطوة زمنية واحدة.
  - **الخوارزمية المستخدمة:** Discrete Time Simulation Algorithm

خوارزمية تحاكي الحركة الفيزيائية عن طريق تقسيم الزمن إلى خطوات صغيرة يتم تحديث حالة النظام خلالها بشكل متكرر.

- **آلية التنفيذ:** يقوم التابع بتنفيذ مراحل المحاكاة بالترتيب:

تطبيق الاحتكاك على الكرات ← تطبيق القوى الخارجية ← فحص دخول الكرات إلى الحفر ← معالجة تصادم الكرات ← معالجة تصادم الجدران.

التابع resolveBallCollisions

- **الاسم:** resolveBallCollisions()
- **الوظيفة:** معالجة التصادمات بين جميع الكرات الموجودة في العالم.
- **الخوارزمية الأولى:** Pairwise Collision Detection Algorithm

خوارزمية تعتمد على مقارنة كل عنصر مع العناصر الأخرى لاكتشاف وجود تصادم.

- **آلية التنفيذ:**

يتم إنشاء أزواج من الكرات، ثم حساب المسافة بينها ومقارنتها مع حجمها لتحديد حدوث تصادم.

- **الخوارزمية الثانية:** Position Correction Algorithm

خوارزمية تستخدم لإزالة التداخل بين الأجسام بعد اكتشاف التصادم.

- **آلية التنفيذ:**

يتم تحريك الكرات بعيداً عن بعضها بمقدار التداخل حتى تعود إلى وضع فيزيائي صحيح.

- **الخوارزمية الثالثة:** Impulse Collision Resolution Algorithm

خوارزمية تستخدم لمعالجة التصادمات من خلال تغيير سرعة الأجسام بدلاً من تطبيق قوة مستمرة.

- **آلية التنفيذ:**

يتم حساب تأثير التصادم ثم تعديل سرعة كل كرة حسب اتجاه التصادم وخصائصها الفيزيائية.

التابع resolveCushionCollisions

- **الاسم:** resolveCushionCollisions()
- **الوظيفة:** معالجة تصادم الكرات مع حدود الطاولة.
  - **الخوارزمية المستخدمة:** Boundary Collision Detection Algorithm

خوارزمية تقوم بفحص موقع الجسم بالنسبة للحدود المسموحة وتطبيق المعالجة عند تجاوزها.

- **آلية التنفيذ:**

يتم مقارنة موقع الكرة مع حدود الطاولة، وعند خروجها يتم تصحيح الموقع وتطبيق ارتداد الجدار.

التابع applyCushionCollision

- **الاسم:** applyCushionCollision(ball, normal, e_c, mu_c)
- **الوظيفة:** تطبيق تأثير اصطدام الكرة مع الحاجز.
- **الخوارزمية الأولى:** Velocity Decomposition Algorithm

خوارزمية تقوم بتقسيم سرعة الجسم إلى مركبات مستقلة حسب اتجاه السطح.

- **آلية التنفيذ:**

يتم فصل السرعة إلى جزء عمودي على الجدار وجزء مماسي، ثم معالجة كل جزء بطريقة مختلفة.

- **الخوارزمية الثانية:** Elastic Reflection Algorithm

خوارزمية تستخدم لحساب تغير اتجاه الحركة بعد الاصطدام.

- **آلية التنفيذ:**

يتم عكس المركبة العمودية للسرعة وتقليل مقدارها حسب خصائص التصادم.

- **الخوارزمية الثالثة:** Friction Damping Algorithm

خوارزمية تقلل المركبة المماسية للحركة لمحاكاة فقدان الطاقة أثناء الاحتكاك.

- **آلية التنفيذ:**

يتم تخفيض السرعة المماسية والدوران تدريجياً أثناء اصطدام الكرة بالحاجز.

التابع checkPocketCollisions

- **الاسم:** checkPocketCollisions()
- **الوظيفة:** اكتشاف دخول الكرة إلى الحفر ومعالجة تأثير الجذب نحوها.
- **الخوارزمية الأولى:** Distance Based Collision Detection

خوارزمية تعتمد على حساب المسافة بين الجسم والهدف لتحديد القرب أو الاصطدام.

- **آلية التنفيذ:**

يتم حساب المسافة بين الكرة وكل حفرة، ثم مقارنتها مع مجال تأثير الحفرة.

- **الخوارزمية الثانية:** Attraction Force Simulation

خوارزمية تحاكي تأثير قوة جذب تدريجية باتجاه نقطة معينة.

- **آلية التنفيذ:**

عندما تقترب الكرة من الحفرة يتم تعديل سرعتها باتجاه مركز الحفرة حتى تدخل إليها.

- **الخوارزمية الثالثة:** State Change Algorithm

خوارزمية تقوم بتغيير حالة الكائن عند تحقق شرط معين.

- **آلية التنفيذ:**

عند دخول الكرة للحفرة يتم تغيير حالتها إلى isPocketted وإيقاف تحديث حركتها داخل العالم.

PhysicsLaws Class

يمثل كلاس **PhysicsLaws** طبقة الحسابات الفيزيائية الأساسية في النظام، حيث يحتوي على مجموعة من التوابع المستقلة التي تمثل العمليات الحسابية المستخدمة من قبل **PhysicalBall** و **PhysicsWorld**.

تم تصميم الملف وفق مبدأ **Modular Physics Functions**، بحيث يتم فصل الحسابات الفيزيائية عن منطق إدارة الكائنات، مما يقلل تكرار العمليات ويسمح بإعادة استخدام الخوارزميات في أكثر من جزء من المحاكاة.

التابع momentOfInertiaSolidSphere

- **الاسم:** momentOfInertiaSolidSphere(mass, radius)
- **الوظيفة:** حساب قيمة القصور الذاتي للكرة المصمتة لاستخدامها في الحركة الدورانية.
- **الخوارزمية المستخدمة:** Physical Property Calculation Algorithm

خوارزمية تقوم بحساب خاصية فيزيائية ثابتة للكائن اعتماداً على خصائصه الهندسية والكتلية.

- **آلية التنفيذ:**

يستقبل التابع كتلة الكرة ونصف قطرها، ثم يعيد قيمة القصور الذاتي التي تستخدم لاحقاً في تحديث الحركة الدورانية.

التابع linearKineticEnergy

- **الاسم:** linearKineticEnergy(mass, velocityVec)
- **الوظيفة:** حساب الطاقة الناتجة عن الحركة الخطية للكرة.
- **الخوارزمية المستخدمة:** Vector Magnitude Based Calculation

خوارزمية تعتمد على استخراج مقدار المتجه الفيزيائي واستخدامه في الحسابات العددية.

- **آلية التنفيذ:**

يتم حساب مقدار السرعة من المتجه ثم تحويلها إلى قيمة طاقة خطية مرتبطة بكتلة الجسم.

التابع rotationalKineticEnergy

- **الاسم:** rotationalKineticEnergy(inertia, angularVelocityVec)
- **الوظيفة:** حساب الطاقة الناتجة عن دوران الكرة حول مركزها.
  - الخوارزمية المستخدمة: Rotational State Evaluation Algorithm

خوارزمية تقوم بتحليل حالة الجسم الدورانية وتحويل سرعة الدوران إلى قيمة طاقة.

- **آلية التنفيذ:**

يتم استخدام مقدار السرعة الزاوية مع قيمة القصور الذاتي لإنتاج الطاقة الدورانية الحالية.

التابع linearVelocityFromImpulse

- **الاسم:** linearVelocityFromImpulse(impulseVec, mass)
- **الوظيفة:** تحويل تأثير النبضة الخارجية إلى تغير في السرعة الخطية.
  - **الخوارزمية المستخدمة:** Impulse Response Algorithm

خوارزمية تستخدم لتحديث سرعة جسم فيزيائي عند تعرضه لتأثير لحظي.

- **آلية التنفيذ:**

يتم تحويل متجه النبضة إلى تغير سرعة عن طريق تقسيم التأثير على كتلة الجسم.

التابع angularVelocityFromImpulse

- **الاسم:** angularVelocityFromImpulse(impulseVec, leverArmVec, inertia)
- **الوظيفة:** حساب التغير في السرعة الدورانية الناتج عن تطبيق قوة خارج مركز الجسم.
  - **الخوارزمية المستخدمة:** Angular Impulse Algorithm

خوارزمية تقوم بتحويل تأثير القوة ونقطة تطبيقها إلى حركة دورانية.

- **آلية التنفيذ:**

يتم حساب تأثير دوران القوة حول مركز الجسم ثم تحويله إلى سرعة زاوية باستخدام القصور الذاتي.

التابع slidingFrictionMagnitude

- **الاسم:** slidingFrictionMagnitude(muSliding, mass, gravity)
- **الوظيفة:** حساب مقدار تأثير الاحتكاك أثناء مرحلة انزلاق الكرة.
  - **الخوارزمية المستخدمة:** Friction Force Evaluation Algorithm

خوارزمية تقوم بحساب قيمة قوة مقاومة الحركة بناءً على خصائص السطح والجسم.

- **آلية التنفيذ:**

يتم استقبال معاملات الاحتكاك والكتلة والجاذبية وإرجاع مقدار القوة المستخدمة في تحديث الحركة.

التابع slidingLinearAcceleration

- **الاسم:** slidingLinearAcceleration(muSliding, gravity, directionVec)
- **الوظيفة:** حساب التسارع الناتج عن الاحتكاك أثناء الانزلاق.
  - **الخوارزمية المستخدمة:** Directional Vector Scaling Algorithm

خوارزمية تعتمد على تعديل متجه اتجاهي عن طريق معامل عددي لإنتاج تأثير موجه.

- **آلية التنفيذ:**

يتم عكس اتجاه الحركة ثم تطبيق مقدار التسارع الناتج عن الاحتكاك.

التابع slidingTorque

- الاسم: slidingTorque(radiusVec, frictionForceVec)
- **الوظيفة:** حساب تأثير الدوران الناتج عن قوة الاحتكاك.
- **الخوارزمية المستخدمة:** Cross Product Torque Algorithm

خوارزمية تعتمد على حاصل الضرب الاتجاهي لحساب تأثير قوة مطبقة على نقطة بعيدة عن مركز الجسم.

- **آلية التنفيذ:**

يتم استخدام متجه نقطة التأثير ومتجه القوة للحصول على متجه العزم الناتج.

التابع isPureRolling

- **الاسم:** isPureRolling(linearVelocity, angularVelocity, radiusVec, threshold)
- **الوظيفة:** تحديد ما إذا كانت الكرة في حالة تدحرج نقي.
- **الخوارزمية المستخدمة:** State Classification Algorithm

خوارزمية تقوم بتصنيف حالة النظام اعتماداً على مقارنة القيم الفيزيائية مع حد معين.

- **آلية التنفيذ:**

يتم حساب سرعة نقطة التلامس ثم مقارنتها مع قيمة صغيرة لتحديد حالة التدحرج.

التابع enforceRollingConstraint

- **الاسم:** enforceRollingConstraint(linearVelocity, radius, previousAngularVelocity)
- **الوظيفة:** تعديل السرعة الدورانية لتتوافق مع حركة التدحرج.
- **الخوارزمية المستخدمة:** Constraint Enforcement Algorithm

خوارزمية تستخدم لإجبار النظام على الالتزام بشرط فيزيائي محدد أثناء المحاكاة.

- **آلية التنفيذ:**

يتم تعديل مركبات الدوران المرتبطة بالحركة الخطية مع المحافظة على الدوران المغزلي.

التابع rollingResistanceForce

- **الاسم:** rollingResistanceForce(muRolling, mass, gravity, velocityVec)
- **الوظيفة:** حساب القوة الناتجة عن مقاومة التدحرج.
- **الخوارزمية المستخدمة:** Opposite Direction Force Algorithm

خوارزمية تقوم بإنشاء تأثير معاكس لاتجاه حركة الجسم لتقليل السرعة تدريجياً.

- **آلية التنفيذ:**

يتم استخراج اتجاه السرعة ثم إنشاء قوة عكسية بمقدار مقاومة التدحرج.

التابع collisionImpulseMagnitude

- **الاسم:** collisionImpulseMagnitude(relativeVelocityAlongNormal, e, invMass1, invMass2)
- **الوظيفة:** حساب مقدار التأثير الناتج عن تصادم كرتين.
- **الخوارزمية المستخدمة:** Collision Impulse Resolution Algorithm

خوارزمية تستخدم لحل التصادمات الثنائية عن طريق حساب تأثير لحظي يغير سرعات الأجسام.

- **آلية التنفيذ:**

يتم استخدام سرعة التصادم وخصائص الجسمين لإنتاج قيمة النبضة التي سيتم تطبيقها.

التابع kineticEnergyLossInCollision

- **الاسم:** kineticEnergyLossInCollision(reducedMass, e, relativeVelocityAlongNormal)
- **الوظيفة:** حساب مقدار الطاقة المفقودة أثناء التصادم.
- **الخوارزمية المستخدمة:** Energy Loss Estimation Algorithm

خوارزمية تستخدم لتقدير مقدار الانخفاض في الطاقة الناتج عن عدم مثالية التصادم.

- **آلية التنفيذ:**

يتم تحليل سرعة التصادم ومعامل الارتداد لإنتاج قيمة فقدان الطاقة.

التابع cushionNormalRebound

الاسم: cushionNormalRebound(velocityInNormal, e_cushion)

الوظيفة: حساب سرعة الكرة بعد ارتدادها عن الجدار.

الخوارزمية المستخدمة: Reflection Response Algorithm

خوارزمية تحاكي تغير اتجاه الحركة عند اصطدام الجسم بسطح صلب.

آلية التنفيذ:

يتم تعديل المركبة العمودية للسرعة بناءً على معامل الارتداد.

التابع tangentialFrictionImpulse

- **الاسم:** tangentialFrictionImpulse(v_tangential, normalImpulseMagnitude, mu_cushion, mass)
- **الوظيفة:** حساب تأثير الاحتكاك المماسي أثناء اصطدام الكرة بالجدار.
  - **الخوارزمية المستخدمة:** Friction Impulse Limiting Algorithm

خوارزمية تحسب تأثير الاحتكاك مع وضع حد أعلى لمنع تجاوز التأثير الفيزيائي الممكن.

- **آلية التنفيذ:**

يتم تحديد مقدار الاحتكاك المطلوب ثم تقييده بالقيمة القصوى المسموحة.

التابع applyTangentialImpulse

- **الاسم:** applyTangentialImpulse(velocityTangential, impulseVec, mass)
- **الوظيفة:** تحديث السرعة المماسية بعد تطبيق تأثير احتكاكي.
  - **الخوارزمية المستخدمة:** Velocity Correction Algorithm

خوارزمية تستخدم لتعديل حالة الحركة بعد تطبيق تأثير خارجي.

- **آلية التنفيذ:**

يتم تحويل النبضة إلى تغير سرعة ثم تطبيقه على المركبة المماسية للحركة.

التابع sweetSpotHeight

- **الاسم:** sweetSpotHeight(radius)
- **الوظيفة:** حساب ارتفاع نقطة الضربة المثالية على الكرة.
  - **الخوارزمية المستخدمة:** Geometric Parameter Calculation

خوارزمية تعتمد على استخراج قيمة هندسية مشتقة من أبعاد الجسم.

- **آلية التنفيذ:**

يتم استخدام نصف قطر الكرة للحصول على نقطة مرجعية تستخدم في نموذج الضربة.

التابع sweetSpotOffset

الاسم: sweetSpotOffset(radius)

الوظيفة: حساب إزاحة نقطة الضربة المثالية عن مركز الكرة.

الخوارزمية المستخدمة: Geometric Offset Calculation

خوارزمية تقوم بحساب موقع نسبي داخل جسم ثلاثي الأبعاد اعتماداً على أبعاده.

آلية التنفيذ:

يتم تحويل نصف قطر الكرة إلى قيمة إزاحة تستخدم عند تطبيق الضربة المثالية.

GameLogic Class

يمثل كلاس GameLogic طبقة إدارة قوانين اللعبة المنطقية في محاكاة 8-Ball Pool، حيث يكون مسؤولاً عن متابعة حالة المباراة، تبديل الأدوار، تعيين مجموعات الكرات، تحليل نتائج الضربات، وتطبيق قواعد الفوز والخسارة.

يعتمد التصميم على نموذج Finite State Machine (FSM)، حيث يتم تمثيل حالة اللعبة بمجموعة حالات محددة، ويتم الانتقال بينها بناءً على الأحداث الناتجة عن الضربات.

تم بناء GameLogic باستخدام نموذج State Driven Architecture، حيث يتم فصل منطق قوانين اللعبة عن المحاكاة الفيزيائية.

الكلاس لا يهتم بحركة الكرة أو التصادمات، بل يستقبل نتائج الأحداث الفيزيائية ويحولها إلى قرارات منطقية خاصة بقواعد اللعبة.

الخوارزمية الأساسية للكلاس هي Finite State Machine + Rule Evaluation System، حيث يتم التحكم بتدفق المباراة من خلال الحالات والأحداث

الثوابت GameState

- **الوظيفة:** تعريف الحالات الرئيسية التي يمكن أن تمر بها اللعبة.
- **الخوارزمية المستخدمة:** Finite State Machine Algorithm

خوارزمية تعتمد على تمثيل النظام كمجموعة حالات، بحيث يكون لكل حالة منطق معالجة وانتقال محدد.

- **آلية التنفيذ:**

يتم تحديد حالة اللعبة الحالية مثل: بداية المباراة، اللعب الطبيعي، الكرة باليد، نهاية اللعبة، ثم يتم تغيير الحالة حسب نتيجة الأحداث.

الثوابت BallGroup

- **الوظيفة:** تعريف أنواع مجموعات الكرات داخل اللعبة.
  - **الخوارزمية المستخدمة:** Enumeration State Representation

أسلوب برمجي يستخدم القيم الثابتة لتمثيل حالات أو تصنيفات محددة داخل النظام.

- **آلية التنفيذ:**

يتم استخدام القيم لتحديد مجموعة اللاعب من كرات مصمتة، كرات مخططة، الكرة الثامنة.

التابع constructor

- **الاسم:** constructor ()
- **الوظيفة:** إنشاء كائن نظام اللعبة وتجهيز الحالة الابتدائية.
  - **الخوارزمية المستخدمة:** Object Initialization Algorithm

خوارزمية تقوم بتهيئة جميع المتغيرات المطلوبة قبل بدء تشغيل النظام.

- **آلية التنفيذ:**

يقوم التابع باستدعاء reset () لإنشاء حالة نظيفة للمباراة، مع ضبط اللاعب الحالي والمجموعات وحالة الضربة.

التابع reset

- **الاسم:** reset ()
- **الوظيفة:** إعادة تهيئة اللعبة إلى الحالة الابتدائية.
  - **الخوارزمية المستخدمة:** State Reset Algorithm

خوارزمية تستخدم لإرجاع النظام إلى حالة ابتدائية معروفة لإعادة التشغيل أو بدء مباراة جديدة.

- **آلية التنفيذ:**

يتم تصفير بيانات الضربات السابقة، إزالة الفائز، إعادة اللاعب الأول، وإعادة حالة المباراة إلى بداية الكسر.

التابع onShotStart

- **الاسم:** onShotStart()
- **الوظيفة:** بدء تسجيل ضربة جديدة.
  - **الخوارزمية المستخدمة:** Event Tracking Algorithm

خوارزمية تستخدم لجمع بيانات الأحداث التي تحدث خلال فترة زمنية محددة.

- **آلية التنفيذ:**

يتم إنشاء سجل جديد للضربة يحتوي على الكرات التي دخلت الحفرة، حالة الكرة البيضاء، أول تلامس.

التابع onBallPocketed

- **الاسم:** onBallPocketed(ballId)
- **الوظيفة:** تسجيل دخول كرة إلى الحفرة أثناء الضربة الحالية.
  - **الخوارزمية المستخدمة:** Event Listener Algorithm

خوارزمية تعتمد على استقبال حدث خارجي وتحديث حالة النظام بناءً عليه.

- **آلية التنفيذ:**

يتم فحص رقم الكرة، ثم تحديث قائمة الكرات المدخلة أو تسجيل حالة خدش الكرة البيضاء.

التابع onFirstContact

- **الاسم:** onFirstContact(otherId)
- **الوظيفة:** تسجيل أول كرة تلامست معها الكرة البيضاء.

- **الخوارزمية المستخدمة:** First Event Capture Algorithm

خوارزمية تحفظ أول حدث يحدث وتتجاهل الأحداث التالية من نفس النوع.

- **آلية التنفيذ:**

إذا لم يتم تسجيل تلامس سابقاً، يتم حفظ معرف الكرة وتغيير حالة التسجيل.

التابع onShotEnd

- **الاسم:** onShotEnd(allPocketedIds)
- **الوظيفة:** تحليل نتيجة الضربة وتحديد نتيجة اللعبة والدور القادم.

- **الخوارزمية المستخدمة:** Rule Evaluation Decision Tree

خوارزمية تعتمد على شجرة قرارات لتحليل مجموعة شروط واختيار النتيجة المناسبة.

- **آلية التنفيذ:** يتم تحليل الضربة حسب ترتيب القواعد:

1. فحص ضربة الكسر.
2. فحص الكرة الثامنة.
3. فحص الخدش.
4. فحص المخالفات.
5. تحديد استمرار الدور أو انتقاله.

معالجة ضربة الكسر داخل onShotEnd

- **الخوارزمية المستخدمة:** Opening State Transition Algorithm

خوارزمية خاصة لمعالجة الحالة الأولى من اللعبة لأنها تمتلك قواعد مختلفة عن اللعب الطبيعي.

- **آلية التنفيذ:**

يتم تحديد نتيجة الكسر بناءً على الكرات الداخلة للحفرة وحالة الكرة البيضاء، ثم يتم الانتقال إلى حالة اللعب.

معالجة الكرة الثامنة داخل onShotEnd

- **الخوارزمية المستخدمة:** Terminal State Evaluation Algorithm

خوارزمية تحدد وصول النظام إلى حالة نهائية بناءً على شروط الفوز والخسارة.

- **آلية التنفيذ:**

يتم فحص توقيت دخول الكرة الثامنة وحالة اللاعب، ثم يتم تحديد الفائز وإنهاء اللعبة.

معالجة الخدش داخل onShotEnd

- **الخوارزمية المستخدمة:** Penalty State Transition Algorithm

خوارزمية تقوم بتغيير حالة النظام عند حدوث مخالفة.

- **آلية التنفيذ:**

يتم نقل الدور للاعب الآخر وتفعيل حالة الكرة باليد.

التابع getRemainingCount

- **الاسم:** getRemainingCount(group, allPocketedIds)
- **الوظيفة:** حساب عدد الكرات المتبقية في مجموعة معينة.
- **الخوارزمية المستخدمة:** Filtering and Counting Algorithm

خوارزمية تعتمد على تصفية مجموعة بيانات ثم حساب العناصر المطابقة لشرط معين.

- **آلية التنفيذ:**

يتم إنشاء قائمة المجموعة المطلوبة ثم إزالة الكرات الموجودة في قائمة الكرات المدخلة للحفرة.

التابع _other

- **الاسم:** _other ()
- **الوظيفة:** إرجاع رقم اللاعب المنافس.
  - **الخوارزمية المستخدمة:** Binary State Switching Algorithm

خوارزمية بسيطة لتبديل حالة بين قيمتين فقط.

- **آلية التنفيذ:**

إذا كان اللاعب الحالي هو الأول يتم إرجاع الثاني، والعكس صحيح.

التابع _groupOf

- **الاسم:** _groupOf(id)
- **الوظيفة:** تحديد مجموعة الكرة بناءً على رقمها.
  - **الخوارزمية المستخدمة:** Range Classification Algorithm

خوارزمية تصنف البيانات حسب المجال الذي تنتمي إليه.

- **آلية التنفيذ:**

يتم مقارنة رقم الكرة مع المجالات المحددة وإرجاع المجموعة المناسبة.

التابع _tryAssignGroups

- **الاسم:** _tryAssignGroups(pocketedIds)
- الوظيفة: تحديد مجموعات اللاعبين بعد الكسر.
  - **الخوارزمية المستخدمة:** Dynamic Classification Algorithm

خوارزمية تقوم بتحديد التصنيف النهائي للعناصر بناءً على أول حدث مؤثر.

- **آلية التنفيذ:**

يتم البحث عن أول كرة غير الكرة الثامنة دخلت الحفرة، ثم يتم تعيين المجموعة للاعب وتحديد مجموعة الخصم.

التابع _hasRemaining

- **الاسم:** _hasRemaining(group, allPocketedIds)
- **الوظيفة:** التحقق من وجود كرات متبقية للاعب.
- **الخوارزمية المستخدمة:** Existence Search Algorithm

خوارزمية تبحث عن وجود عنصر يحقق شرطاً معيناً بدون الحاجة لفحص جميع النتائج.

- **آلية التنفيذ:**

يتم استخدام البحث داخل مجموعة اللاعب لمعرفة إذا بقيت أي كرة غير مدخلة.

التابع _groupLabel

- **الاسم:** _groupLabel(group)
- **الوظيفة:** تحويل معرف المجموعة إلى نص قابل للعرض.
- **الخوارزمية المستخدمة:** Mapping Algorithm

خوارزمية تقوم بتحويل قيمة داخلية إلى تمثيل نصي مفهوم للمستخدم.

- **آلية التنفيذ:**

يتم مقارنة قيمة المجموعة وإرجاع الاسم والوصف المناسب لها.

ExternalForces Class

يمثل هذا الملف طبقة **القوى الفيزيائية الخارجية القابلة للإضافة** داخل نظام المحاكاة.
تم تصميمه باستخدام مفهوم **Extensible Physics System**، حيث يتم تعريف واجهة أساسية للقوة الفيزيائية ثم إنشاء أنواع مختلفة من القوى يمكن تسجيلها وتشغيلها داخل **PhysicsWorld** بدون تعديل بنية النظام الأساسية.

يعتمد التصميم على مبدأ **Polymorphism**، حيث تتشارك جميع القوى نفس الواجهة **(apply)** ولكن كل قوة تنفذ خوارزمية تأثير مختلفة على الجسم الفيزيائي.

الكلاس IExternalForce

- **الاسم:** IExternalForce
- **الوظيفة:** تعريف الواجهة الأساسية لجميع القوى الفيزيائية الخارجية.
  - **الخوارزمية المستخدمة:** Interface Abstraction Pattern

أسلوب تصميم يقوم بتحديد مجموعة عمليات مشتركة للكائنات المختلفة بدون تحديد طريقة التنفيذ الداخلية.

- **آلية التنفيذ:**

يتم تعريف الخصائص العامة من اسم القوة، حالة التفعيل، كما يتم توفير توابع أساسية يمكن للكلاسات المشتقة إعادة تنفيذها.

التابع constructor

- **الاسم:** constructor(name)
- **الوظيفة:** تهيئة كائن القوة الخارجية الأساسي.
  - **الخوارزمية المستخدمة:** Object Initialization Algorithm

خوارزمية مسؤولة عن تجهيز الحالة الابتدائية للكائن قبل استخدامه داخل النظام.

- **آلية التنفيذ:**

يتم تخزين اسم القوة وتفعيلها تلقائياً، مما يسمح للنظام بإدارتها أثناء دورة المحاكاة.

التابع apply

- **الاسم:** apply (ball, config, dt)
- **الوظيفة:** واجهة تنفيذ تأثير القوة على الكرة.
  - **الخوارزمية المستخدمة:** Polymorphic Method Dispatch

خوارزمية تعتمد على اختيار التنفيذ المناسب حسب نوع الكائن الفعلي الذي يستخدم التابع.

- **آلية التنفيذ:**

يتم ترك التابع الأساسي فارغاً، وتقوم الكلاسات المشتقة بتوفير طريقة التأثير الخاصة بها.

التابع setupGUI

- **الاسم:** setupGUI(folder)
- **الوظيفة:** توفير واجهة تحكم لإعدادات القوة.
  - **الخوارزمية المستخدمة:** Runtime Parameter Binding Algorithm

خوارزمية تربط خصائص الكائن بعناصر تحكم خارجية لتعديلها أثناء التشغيل.

- **آلية التنفيذ:**

يتم تمرير كائن واجهة المستخدم، ثم تقوم القوة بإضافة المتغيرات القابلة للتعديل إليه.

الكلاس WindBlowForce

يمثل نموذج قوة الرياح الجانبية التي تؤثر على حركة الكرة.

التابع constructor

- **الاسم:** constructor ()
- **الوظيفة:** تهيئة قوة الرياح وإعداد قيمة التأثير الابتدائية.
- **الخوارزمية المستخدمة:** Specialized Object Initialization

تهيئة نسخة متخصصة من كائن القوة الأساسي مع إضافة خصائص خاصة بها.

- **آلية التنفيذ:**

يتم استدعاء البناء الأساسي ثم إنشاء متغير شدة الرياح بقيمة ابتدائية.

التابع apply في WindBlowForce

- **الاسم:** apply (ball, config, dt)
- **الوظيفة:** تطبيق تأثير الرياح على حركة الكرة.
- **الخوارزمية الأولى:** Conditional Processing Algorithm

خوارزمية تقوم بفحص شروط التشغيل قبل تنفيذ العمليات لتجنب الحسابات غير الضرورية.

- **آلية التنفيذ:**

يتم تجاهل التنفيذ إذا كانت القوة غير مفعلة أو الكرة في حالة سكون.

- **الخوارزمية الثانية:** Velocity Integration Algorithm

خوارزمية تقوم بتعديل سرعة الجسم تدريجياً اعتماداً على تأثير خارجي والزمن.

- **آلية التنفيذ:**

يتم إضافة تأثير الرياح إلى مركبة السرعة الأفقية خلال الخطوة الزمنية الحالية.

التابع setupGUI في WindBlowForce

- **الاسم:** setupGUI(folder)
- **الوظيفة:** إضافة إعدادات التحكم الخاصة بالرياح إلى واجهة المستخدم.
- **الخوارزمية المستخدمة:** Interactive Parameter Exposure Algorithm

خوارزمية تسمح بعرض متغيرات النظام الداخلية كإعدادات قابلة للتعديل أثناء التشغيل.

- **آلية التنفيذ:**

يتم ربط تفعيل الرياح، شدة الرياح، مع عناصر تحكم مباشرة.

الكلاس MagneticCueBallForce

يمثل قوة جذب افتراضية تؤثر بين الكرة البيضاء وباقي الكرات.

التابع constructor

- **الاسم:** constructor ()
- **الوظيفة:** إنشاء كائن قوة المجال المغناطيسي.
  - **الخوارزمية المستخدمة:** Specialized Force Initialization

خوارزمية تهيئة لكائن قوة مخصص يرث خصائص النظام العام للقوى.

- **آلية التنفيذ:**

يتم تعريف اسم القوة وإنشاء معامل التحكم الخاص بقوة الجذب.

التابع apply في MagneticCueBallForce

- **الاسم:** apply (ball, config, dt)
- **الوظيفة:** تطبيق تأثير جذب الكرة البيضاء على الكرات الأخرى.

- **الخوارزمية الأولى:** Object Filtering Algorithm

خوارزمية تقوم بتحديد العناصر التي يسمح لها بالتأثر قبل تنفيذ المعالجة.

- **آلية التنفيذ:**

يتم تجاهل الكرة البيضاء نفسها، الكرات المتوقفة، الحالات غير الصالحة .

- **الخوارزمية الثانية:** Reference Object Retrieval Algorithm

خوارزمية تقوم بالوصول إلى جسم مرجعي مرتبط بالكائن الحالي.

- **آلية التنفيذ:**

يتم استخراج الكرة البيضاء من مرجع العالم الفيزيائي لاستخدامها كنقطة تأثير.

- **الخوارزمية الثالثة:** Distance Based Force Calculation

خوارزمية تعتمد على المسافة بين جسمين لتحديد مقدار التأثير بينهما.

- **آلية التنفيذ:**

يتم حساب الاتجاه والمسافة بين الكرة الحالية والكرة البيضاء، ثم تحديد مقدار القوة حسب القرب بينهما.

- **الخوارزمية الرابعة:** Directional Velocity Modification Algorithm

خوارزمية تقوم بتعديل سرعة الجسم باتجاه معين لمحاكاة تأثير قوة خارجية.

- **آلية التنفيذ:**

يتم تحويل اتجاه القوة إلى متجه حركة ثم إضافته إلى سرعة الكرة خلال الزمن.

التابع setupGUI في MagneticCueBallForce

- **الاسم:** setupGUI(folder)
- **الوظيفة:** إضافة إعدادات الحقل المغناطيسي إلى واجهة التحكم.

- **الخوارزمية المستخدمة:** Dynamic Configuration Interface Algorithm

خوارزمية تربط قيم النظام الداخلية بواجهة تعديل خارجية أثناء تشغيل المحاكاة.

- **آلية التنفيذ:**

يتم توفير عناصر تحكم لتفعيل المجال وتغيير مقدار قوة الجذب.

CueStickManager Class

يمثل كلاس CueStickManager طبقة إدارة عصا البلياردو والتفاعل معها داخل البيئة ثلاثية الأبعاد.
مسؤوليته هي إنشاء نموذج العصا، إدارة اتجاه التصويب، التحكم بحركة السحب والضربة، ثم تحويل الحركة الرسومية إلى تأثير فيزيائي على الكرة البيضاء.

يعتمد التصميم على نموذج Interactive State Machine، حيث يتم التحكم بحالة العصا من خلال حالات مثل: وضع التصويب، وضع انتظار الضربة، حركة تنفيذ الضربة، وغيرها.

التابع constructor

- **الاسم:** constructor (appContext, scene)
- **الوظيفة:** تهيئة مدير العصا وربطه مع سياق التطبيق والمشهد الرسومي.
  - **الخوارزمية المستخدمة:** Component Initialization Algorithm

خوارزمية تقوم بإنشاء وتجهيز مكونات النظام قبل بدء الاستخدام.

- **آلية التنفيذ:**

يتم تخزين المراجع الأساسية، تهيئة متغيرات الحالة، ثم استدعاء توابع إنشاء العصا وخط التصويب.

التابع initCueStick

- **الاسم:** initCueStick()
- **الوظيفة:** إنشاء النموذج الثلاثي الأبعاد لعصا البلياردو وإضافته إلى المشهد.
- **الخوارزمية المستخدمة:** 3D Object Construction Algorithm

خوارزمية مسؤولة عن بناء كائن رسومي من مكونات هندسية وربطه ضمن شجرة المشهد.

- **آلية التنفيذ:**

يتم إنشاء مجموعة تحكم للعصا، بناء الشكل الهندسي باستخدام Cylinder Geometry ، ثم إضافة المادة والكائن إلى المشهد.

التابع initAimLine

- **الاسم:** initAimLine()
- **الوظيفة:** إنشاء خط التصويب الذي يوضح اتجاه الضربة.
- **الخوارزمية المستخدمة:** Dynamic Visualization Algorithm

خوارزمية تستخدم للإنشاء عنصر مرئي مرتبط بحالة النظام لتوضيح معلومات داخلية للمستخدم.

- **آلية التنفيذ:**

يتم إنشاء خط ثلاثي الأبعاد باستخدام نقاط بداية ونهاية، ثم إضافته للمشهد وربطه باتجاه العصا.

التابع handlePointerDown

- **الاسم:** handlePointerDown(e, cueBall)
- **الوظيفة:** معالجة بداية تفاعل المستخدم مع العصا.

- **الخوارزمية المستخدمة:** Input Validation Algorithm

خوارزمية تقوم بفحص صلاحية الحالة قبل تنفيذ عملية إدخال.

- **آلية التنفيذ:**

يتم التحقق من حالة الكرة البيضاء، حالة العصا، نوع زر الإدخال.

إذا كانت الشروط صحيحة يتم تفعيل وضع التصويب وتخزين نقطة بداية الحركة.

التابع handlePointerMove

- **الاسم:** handlePointerMove(e)
- **الوظيفة:** تحديث زاوية العصا أثناء تحريك المؤشر.
  - **الخوارزمية المستخدمة:** Incremental Rotation Update Algorithm

خوارزمية تقوم بتحديث قيمة الدوران اعتماداً على التغير بين موقعين متتاليين للإدخال.

- **آلية التنفيذ:**

يتم حساب الفرق الأفقي لحركة المؤشر وتحويله إلى تغير زاوية ثم إضافته لاتجاه العصا الحالي.

التابع handlePointerUp

- **الاسم:** handlePointerUp()
- **الوظيفة:** إنهاء وضع التصويب عند تحرير الإدخال.
- **الخوارزمية المستخدمة:** State Transition Algorithm

خوارزمية تغير حالة النظام بناءً على حدث خارجي.

- **آلية التنفيذ:**

يتم تعطيل حالة التصويب وإعادة العصا إلى حالة الانتظار.

التابع updateState

- **الاسم:** updateState(cueBall)
- **الوظيفة:** تحديث موقع وظهور العصا وخط التصويب بناءً على حالة الكرة البيضاء.
  - **الخوارزمية المستخدمة:** State Synchronization Algorithm

خوارزمية تحافظ على توافق الحالة الرسومية مع الحالة الفيزيائية للكائن.

- **آلية التنفيذ:**

يتم فحص حالة الكرة، ثم تحديث موقع العصا, دورانها, ظهور خط التصويب.

التابع animateStrike

- **الاسم:** animateStrike(frameTime, cueBall)
- **الوظيفة:** تنفيذ الحركة الرسومية الكاملة لضربة العصا ثم تطبيق التأثير الفيزيائي.
- **الخوارزمية الأولى:** Time Based Animation Algorithm

خوارزمية تعتمد على الزمن لتغيير حالة الحركة بشكل تدريجي ومستقل عن معدل الإطارات.

- **آلية التنفيذ:**

يتم زيادة قيمة تقدم الحركة باستخدام الزمن ثم تحديد المرحلة الحالية من الضربة.

- **الخوارزمية الثانية:** Animation State Machine Algorithm

خوارزمية تقسم الحركة إلى مراحل مستقلة حسب قيمة التقدم.

- **آلية التنفيذ:** يتم تنفيذ ثلاث مراحل:

1. سحب العصا للخلف.
2. اندفاع العصا للأمام.
3. تنفيذ الاصطدام وتطبيق الضربة.

- **الخوارزمية الثالثة:** Physics Trigger Algorithm

  خوارزمية تربط حدثاً رسومياً بنتيجة فيزيائية.

- **آلية التنفيذ:**

  بعد وصول العصا لنقطة الاصطدام يتم حساب اتجاه الضربة واستدعاء applyAdvancedStrike على الكرة البيضاء.

TableGraphics Class

يمثل كلاس **TableGraphics** طبقة إنشاء البيئة الرسومية للطاولة باستخدام مكتبة **Three.js**.

مسؤوليته هي بناء المشهد، الكاميرا، الإضاءة، الهيكل الهندسي للطاولة، وربط العناصر الرسومية مع بيانات العالم الفيزيائي.

يعتمد التصميم على نموذج **Scene Graph Construction** حيث يتم بناء العناصر على شكل شجرة كائنات مترابطة داخل المشهد.

التابع constructor

- **الاسم:** constructor(appContext)
- **الوظيفة:** تهيئة نظام الرسومات وإنشاء مكونات الطاولة.
- **الخوارزمية المستخدمة:** Graphics Pipeline Initialization Algorithm

  خوارزمية تقوم بإعداد جميع المكونات اللازمة لعملية العرض الثلاثي الأبعاد.

- **آلية التنفيذ:**

  يتم تخزين سياق التطبيق ثم إنشاء المشهد, الإضاءة, نموذج الطاولة.

التابع initScene

- **الاسم:** initScene()
- **الوظيفة:** إنشاء بيئة العرض الأساسية.

- **الخوارزمية المستخدمة:** 3D Rendering Pipeline Setup

خوارزمية مسؤولة عن تجهيز مراحل عرض المشاهد ثلاثية الأبعاد.

- **آلية التنفيذ:**

يتم إنشاء Scene, Renderer, Camera, Orbit Controls, ثم يتم ربطها مع نافذة العرض.

التابع initLights

- **الاسم:** initLights()
- **الوظيفة:** إضافة مصادر الإضاءة إلى المشهد.
- **الخوارزمية المستخدمة:** Lighting Configuration Algorithm

خوارزمية تقوم بتحديد خصائص الإضاءة لتحسين تمثيل الأجسام ثلاثية الأبعاد.

- **آلية التنفيذ:**

يتم إضافة إضاءة عامة وإضاءة اتجاهية مع إعدادات الظلال.

التابع initTableStructure

- **الاسم:** initTableStructure()
- **الوظيفة:** إنشاء كامل الهيكل الهندسي لطاولة البلياردو.
- **الخوارزمية الأولى:** Hierarchical Object Construction Algorithm

خوارزمية تبني العناصر الرسومية على شكل مجموعات مترابطة.

- **آلية التنفيذ:**

يتم إنشاء مجموعة رئيسية للطاولة وإضافة جميع العناصر إليها.

- **الخوارزمية الثانية:** Procedural Geometry Generation Algorithm

خوارزمية تقوم بإنشاء المجسمات هندسياً باستخدام أبعاد ومعاملات قابلة للتغيير.

- **آلية التنفيذ:**

يتم توليد سطح الطاولة، الحواف، الجيوب، الهيكل السفلي، الأرجل، اعتماداً على أبعاد الطاولة.

- **الخوارزمية الثالثة:** Object Placement Algorithm

خوارزمية تحدد مواقع العناصر داخل الفضاء ثلاثي الأبعاد.

- **آلية التنفيذ:**

يتم حساب إحداثيات كل جزء ثم وضعه داخل مجموعة الطاولة.

التابع handleResize

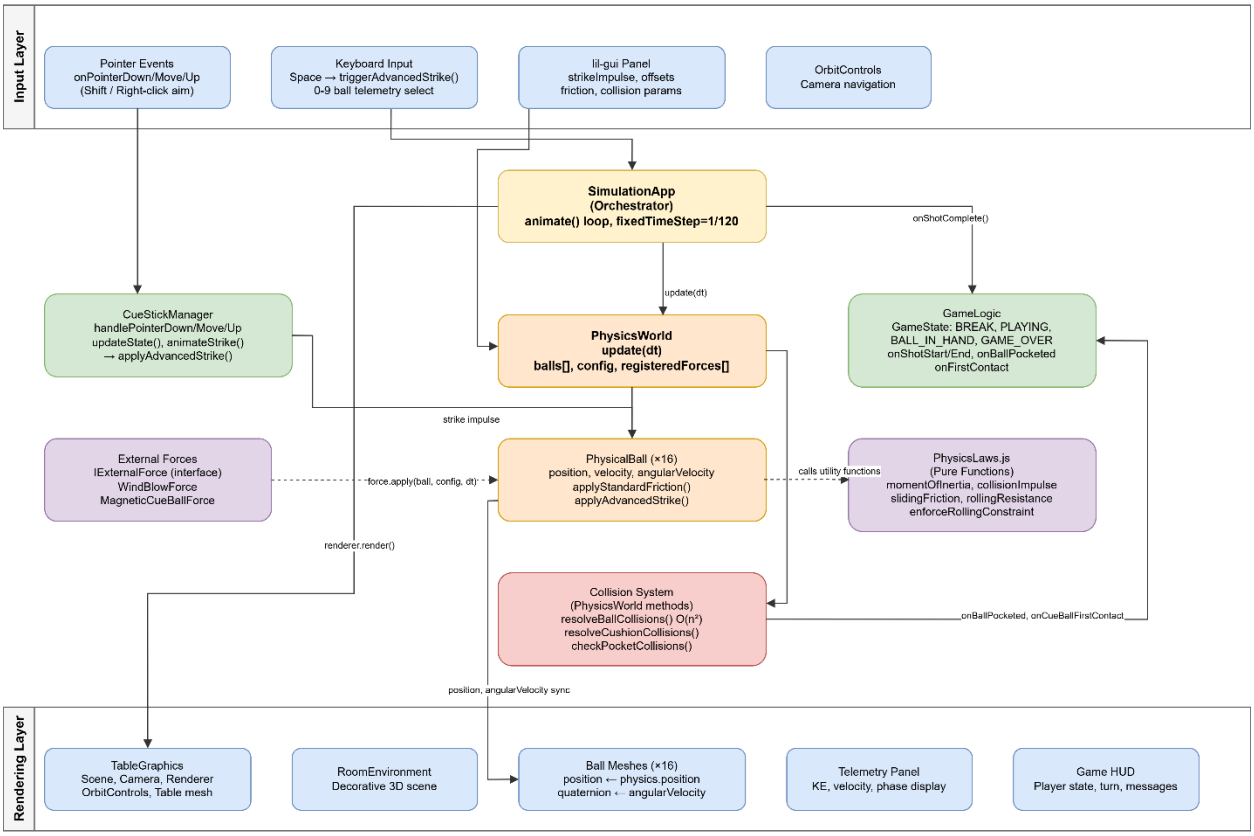
- **الاسم:** handleResize()
- **الوظيفة:** تحديث العرض والكاميرا عند تغير حجم النافذة.
- **الخوارزمية المستخدمة:** Viewport Adaptation Algorithm

خوارزمية تحافظ على توافق العرض ثلاثي الأبعاد مع أبعاد الشاشة الجديدة.

- **آلية التنفيذ:**

يتم تحديث نسبة الكاميرا، إعادة حساب الإسقاط، ثم تعديل حجم الـ Renderer.

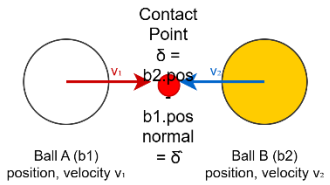
Physics System Architecture
Billiard Ball Physics Project



Solid arrows = direct calls / data flow
Dashed arrows = utility function dependencies
Entry point: main.js → new SimulationApp()

Collision Model Diagram
PhysicsWorld.resolveBallCollisions() + applyCushionCollision()

BEFORE COLLISION



Overlap detected:
distance < $r_1 + r_2$
penetration = (r_1+r_2) - distance

AFTER COLLISION

Balls separate along normal
Energy partially conserved (e=0.96)
onCueBallFirstContact callback if cue ball involved

DURING COLLISION (Processing)

1. Collision Detection
Nested loop: for i, for j=i+1
Skip if isPocketted

2. Position Correction
correction = normal × penetration × 0.8 / ($invM_1$+$invM_2$)
b1.pos -= correction × $invM_1$
b2.pos += correction × $invM_2$

3. Relative Velocity Check
v_rel = v_2 - v_1
velAlongNormal = v_rel · normal
If velAlongNormal > 0 → separating, skip

4. Impulse Calculation
collisionImpulseMagnitude(velAlongNormal, e_ball, $invM_1$, $invM_2$)
J = -(1+e) × velAlongNormal / ($invM_1$+$invM_2$)
e_ball = 0.96

5. Velocity Update
v_1 += impulse × (-$invM_1$)
v_2 += impulse × $invM_2$
isSleeping = false (both balls)

CUSHION COLLISION (resolveCushionCollisions)

1. Bounds Check
position ± radius vs tableBounds
Skip if near pocket (pocketRadius × 1.8)

2. Normal Rebound
vn_out = -e_cushion × vn
e_cushion = 0.75

3. Tangential Friction
vt_out = vt × (1 - μ_c)
μ_c = cushion_friction (0.2)

4. Spin Effect
ω_x, ω_z damped by μ_c
ω_y adds tangential kick to vt

5. Recompose velocity: v = vt_out + vn_out × normal

Note: Ball-ball collision uses impulse-based resolution (no angular velocity change).
Cushion collision modifies both linear and angular velocity. Pocket uses radial pull force then pocketed state.

Free Body Diagram — Billiard Ball on Table Surface
Based on PhysicalBall.applyStandardFriction() and IExternalForce

Legend

Solid arrows = active forces / motion vectors

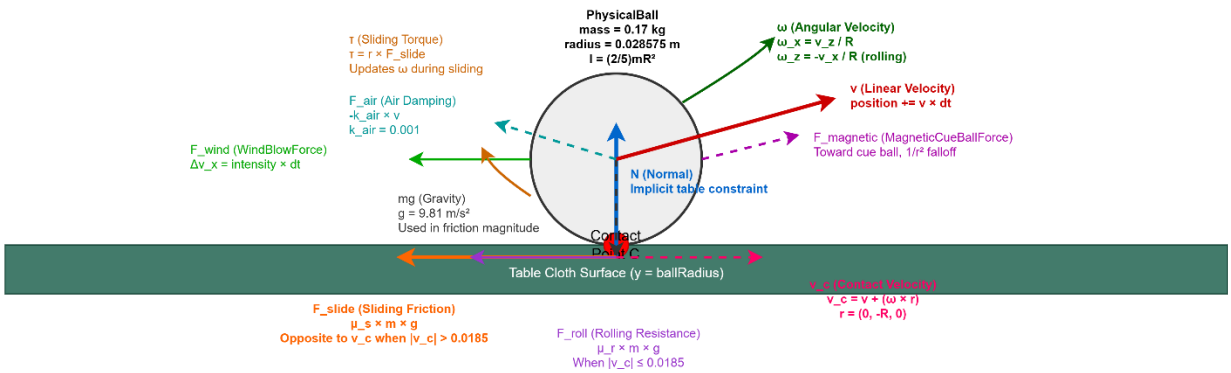
Dashed arrows = damping / external optional forces

μ_sliding = 0.15 (config)

μ_rolling = 0.005 (config)

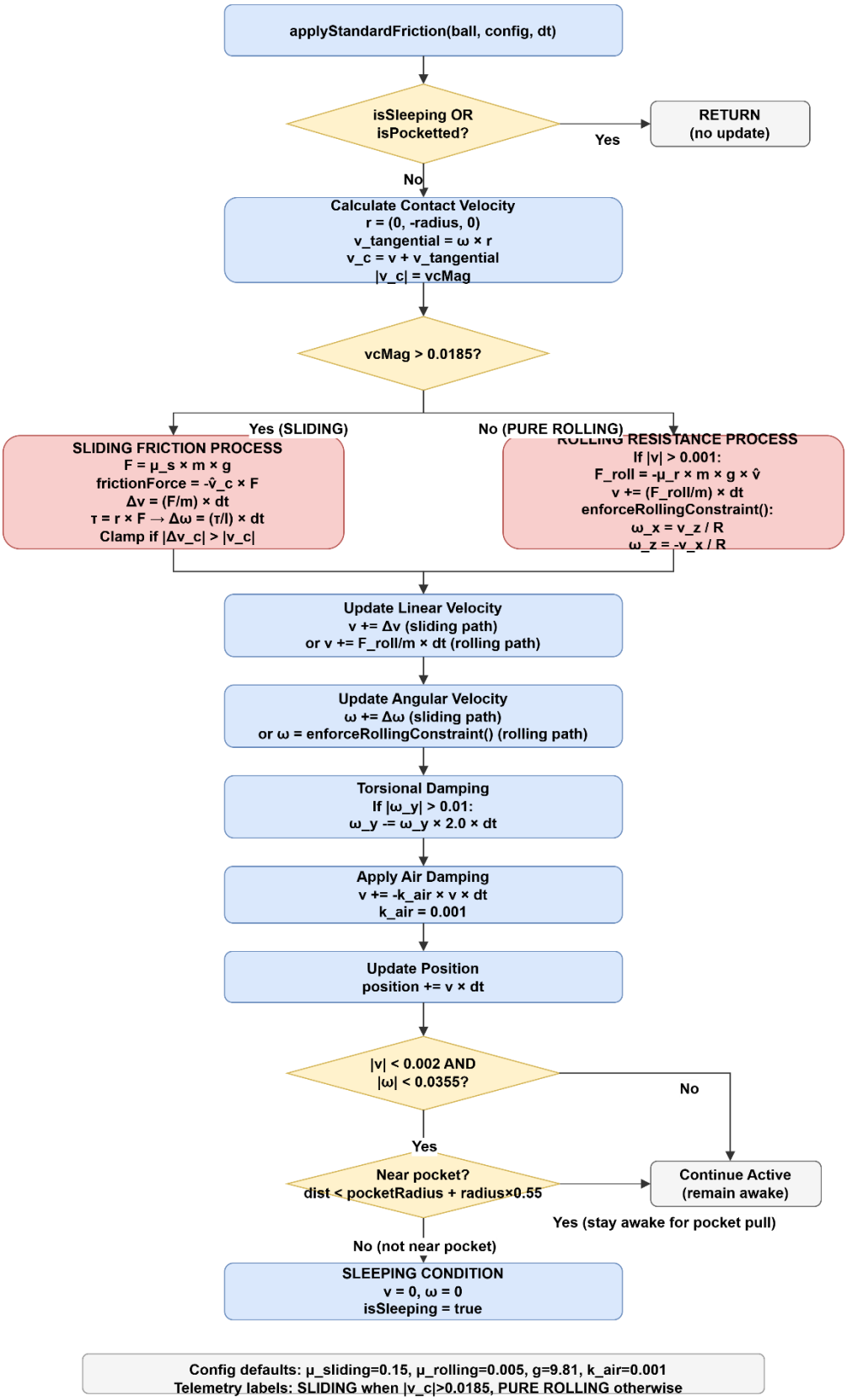
Torsional damping on ω_y: factor 2.0

Sliding/Rolling threshold: |v_c| = 0.0185 m/s



Key Equations (PhysicsLaws.js):
- F_slide = μ_s × m × g
- F_roll = -μ_r × m × g × v̂
- Δω from impulse: (r × J) / I
- Rolling: ω_x = v_z/R, ω_z = -v_x/R

Friction Algorithm Flowchart
PhysicalBall.applyStandardFriction(config, dt)

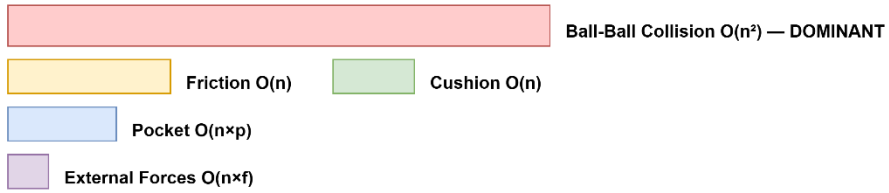


Config defaults: μ_sliding=0.15, μ_rolling=0.005, g=9.81, k_air=0.001
Telemetry labels: SLIDING when |v_c|>0.0185, PURE ROLLING otherwise

Algorithm Complexity Analysis
Billiard Ball Physics — n = 16 balls, p = 6 pockets, f = 2 external forces

Algorithm	Processing Method	Complexity	Performance Impact
Ball-Ball Collision Detection resolveBallCollisions()	Nested loop: for i=0..n, for j=i+1..n Distance check on XZ plane Impulse + position correction per pair	$O(n^2)$ 120 pairs/frame	HIGH Dominant cost
Friction Update applyStandardFriction()	Single pass over all balls Vector math per ball (v_c, friction, damping) Constant ops per ball	O(n) 16 balls/frame	MEDIUM Per-ball vector ops
External Forces registeredForces[].apply()	Double loop: f forces × n balls WindBlowForce, MagneticCueBallForce Early exit if disabled/sleeping	O(n × f) 32 ops/frame	LOW Only 2 forces
Pocket Collision Check checkPocketCollisions()	Nested loop: n balls × p pockets 2D distance check + pull force	O(n × p) 96 checks/frame	LOW-MEDIUM
Cushion Collision resolveCushionCollisions()	Single pass: bounds check per ball 4 wall checks + pocket proximity skip applyCushionCollision() on hit	O(n) 16 balls/frame	LOW Constant wall checks
Physics Update (Total) PhysicsWorld.update(dt)	Sum of all physics sub-steps Called 1-3× per render frame (accumulator) fixedTimeStep = 1/120 s	$O(n^2)$ dominated by collisions	HIGH ~120 Hz physics rate
Rendering Sync mesh position + quaternion	Loop ballMeshes: copy position, quaternion multiply if active Three.js render() — GPU bound	O(n) + O(1) GPU render	MEDIUM GPU handles scene
State Evaluation GameLogic.onShotEnd()	Rule checks on pocketed balls, first contact, group assignment Called once per shot (not per frame)	O(1) fixed rule set	NEGLIGIBLE Event-driven
Kinetic Energy Calc getTotalKineticEnergy()	Sum KE over active balls Called per render frame for telemetry	O(n) 16 balls	LOW Display only

Complexity Hierarchy (per physics step)



Summary: With n=16, the $O(n^2)$ ball-ball collision loop (120 pair checks at 120 Hz) is the bottleneck.
No spatial partitioning or broad-phase optimization is implemented. Performance is acceptable at this scale.
Fixed timestep accumulator may run physics 1-3× per visual frame (~60 FPS).

المشكله: ربط تطبيق القوة من العصا على الكرة البيضاء
عبر معادلات رياضية (PhysicalBall) وكلاس الكرة الفيزيائية (CueStickManager) من خلال تنسيق وثيق بين كلاس إدارة العصا دقيقة.

فيما يلي تحليل مفصل لكيفية الربط، والمشكلة الفيزيائية/البرمجية التي ظهرت أثناء التطوير، والحل الذكي الذي تم تطبيقه:

أولاً: كيفية ربط تطبيق القوة من العصا على الكرة

عملية الربط وتوليد الحركة تمر بثلاث مراحل أساسية تربط العناصر الرسومية بالمحرك الفيزيائي:

1. **التحضير الهندسي وتحديد الاتجاه (CueStickManager.animateStrike):**
يبدأ أنيميشن اندفاع العصا. وفور وصول العصا لنقطة الارتطام (المسافة صفر (Spacebar) عندما يضغط اللاعب على زر المسافة مع الكرة)، يتم ح**تحديد اتجاه الضربة الحقيقي ثلاثي الأبعاد** هندسياً بالاعتماد على جيب وجتا زاوية دوران العصا الحالية (θ) حول الكرة البيضاء:

$$\text{strikeDirection} = (-\sin(\theta), \, 0, \, -\cos(\theta))$$

3. **تمرير النبضة الفيزيائية:**
تقوم الدالة الممسكة بالأنيميشن باستدعاء الدالة المتقدمة داخل كلاس الكرة البيضاء:

```
cueBall.applyAdvancedStrike(config.strikeImpulse, strikeDirection, config.strikeOffsetX,
config.strikeOffsetY);
```

هنا يتم الربط الفعلي؛ حيث تُنقل قيم القوة (Impulse) كز**نسبة الانحراف عن المركز** والاتجاه، بالإضافة إلى (offsetX وoffsetY) التي يحددها اللاعب لعمل ضربات دورانية متقدمة (مثل تأثير الـ *Spins* أو *English*).

3. **تحويل الدفع إلى سرعات خطية وزاوية (PhysicalBall.applyAdvancedStrike):**
تنتج **:** لتقسيم الضربة إلى**سرعة خطية ابتدائية** (PhysicsLaws.js) تستقبل الكرة هذه المقادير وتستعين بملف القوانين الرياضية دائية:
:سرعة زاوية ابتدائية (دوران حول النفس) $\vec{v} = \frac{\vec{J}}{m}$) عن قسمة متجه الدفع على كتلة الكرة لنفس):
(المسافة من مركز الكرة إلى نقطة ضرب السن المنحرفة)، وتُطبق باستخدام الضرب التقاطعي "ذراع العزم الحقيقي"تُحسب بناءً على $\vec{\omega} = \frac{\vec{r} \times \vec{J}}{I}$) لتوليد الدوران المغزلي فوراً.

ثانياً: المشكلة وحلها (المعادلة الفيزيائية والواقعية) ##

(Miscue Problem) المشكلة: التسديدة الوهمية وعزم الدوران غير الواقعي .1 ###

لمنح اللاعبين المحترفين القدرة على التحكم في تدوير الكرة، (offsetX وoffsetY) عند إتاحة خيار ضرب الكرة بعيداً عن مركزها
:مشكلة:ظهرت مشكلة برمجية وفيزيائية **طبيعة المشكلة
"ذراع العزم الحقيقي"إذا قام اللاعب (أو الواجهة) بإدخال قيم انحراف ضخمة جداً وتتجاوز حدود نصف قطر الكرة الفعلي، فإن
المحسوب يصبح أطول من اللازم في الفضاء ثلاثي الأبعاد. رياضياً، هذا يؤدي إل**عزم دوران هائل وغير واقعي على الإطلاق**لإطلاق**
($\vec{\omega} = \frac{\vec{r} \times \vec{J}}{I}$). عند حساب النبضة الدورانية
عملياً في اللعبة، كان هذا يتسبب في طيران الكرة بشكل عشوائي، أو دورانها بسرعات جنونية تكسر استقرار المحرك الفيزيائي، وهو ما
.(انزلاق رأس العصا عن الكرة نتيجة الضرب عند طرف حرج جداً) **"iscue"**Miscue - الميسكيو**"يعرف في ال

(Offset Clamping) الحل المشروح في الكود: تقييد إزاحة نقطة الضرب .2 ###

لحماية استقرار اللعبة وضمان فيزياء منطقية تحاكي الواقع،**قيد أمان صارم**ن صارم** داخل دالة تطبيق الضربة المتقدمة في كلاس الكرة
(PhysicalBall.applyAdvancedStrike).

:خطوات الحل

1. تم تحديد حد أقصى مسموح به لإزاحة الضربة وهو 70% فقط من نصف قطر الكرة الفعلي (this.radius * 0.7)؛ لأن
.الضرب أبعد من هذه المساحة في الحقيقة يسبب انزلاق العصا

2. يقوم الكود بحساب طول متجه الانحراف الحالي للاعب باستخدام نظرية فيثاغورث:

$$\text{offsetMag} = \sqrt{\text{offsetX}^2 + \text{offsetY}^2}$$

3. إذا وجد الكود أن اللاعب حاول تجاوز حد الـ 70% (offsetMag > maxOffsetMag)، فإنه لا يرفض الضربة،**إعادة تقليص**
:لتقع عند الحافة الآمنة تماماً (Scale) ***وتعديل النسبة تلقائياً**ة تلقائياً

```
const scale = maxOffsetMag / offsetMag;

offsetX *= scale;

offsetY *= scale;
```

:النتيجة
بفضل هذا الحل، تم الحفاظ على نقطة التلامس ضمن سطح الكرة الفعلي دائماً. تلاشت تماماً متلازمة عزم الدوران اللانهائي أو
بشكل آمن وواقعي بنسبة *Draw* والـ *Follow* الاهتزازات المشوهة، وحصل التطبيق على محاكاة احترافية تسمح بتأثيرات الـ
100%.

أولاً: كيفية ربط تطبيق التصادم والقوة بين العصا والكرة البيضاء

:يتم الربط بين العصا والكرة البيضاء عبر سلسلة من الخطوات البرمجية والرياضية المتتالية التي تنتقل من كلاس إلى آخر كالتالي

1. **(SimulationApp): التحقق والبدء**
تقوم هذه الدالة بفحص شروط الأمان (مثل triggerAdvancedStrike. عندما يضغط اللاعب على زر المسافة، تستدعي اللعبة الدالة .التأكد من أن اللعبة لم تنتهِ، وأن الكرات توقفت تماماً عن الحركة). إذا تحققت الشروط، يتم تفعيل أنيميشن حركة العصا

2. **(CueStickManager): محاكاة حركة الاصطدام وحساب الاتجاه**
بتقسيم الحركة زمنيًا؛ تبدأ بمرحلة سحب العصا للخلف للتأهب، تليها مرحلة الاندفاع السريع للأمام نحو animateStrike تقوم الدالة الكرة. وعند لحظة الارتطام الفعلي (نهاية الاندفاع)، يتم إيقاف الأنيميشن وإخفاء العصا مؤقتاً. في هذه اللحظة، يتم حساب متجه اتجاه :باستخدام القانون (θ) الضربة ثلاثي الأبعاد هندسياً بناءً على زاوية الدوران الحالية

$$\text{strikeDirection} = (-\sin(\theta), \, 0, \, -\cos(\theta))$$

بعد ذلك، يتم استدعاء دالة تطبيق الضربة المتقدمة الموجودة في كلاس الكرة البيضاء، وتمرير مقدار القوة، الاتجاه، ونقاط الانحراف عن .(strikeOffsetX وstrikeOffsetY) المركز

3. **(PhysicalBall & PhysicsLaws): تحويل التصادم إلى حركة فيزيائية**
ابتدائية:** يتم حسابها applyAdvancedStrike تستقبل الكرة البيضاء هذه البيانات في الدالة **:لتوليد نوعين من **اسرعة خطية ابتدائية**
ة (دوران):** إذا كانت **:سرعة زاوية ابتدائية (دوران)**($\vec{v} = \frac{\vec{J}}{I}$) بقسمة متجه دفع الاصطدام على كتلة الكرة الضربة منحرفة عن المركز، يتم حساب ذراع القوة وتوليد سرعة دورانية ابتدائية عبر الضرب التقاطعي بين ذراع القوة ومتجه الدفع ($\vec{\omega} = \frac{\vec{r} \times \vec{J}}{I}$).

(Miscue - منع التسديدة الوهمية) ثانياً: المشكلة وحلها ##

:المشكلة .1 ###

، ظهرت (offsetX وoffsetY) عند إتاحة ميزة تدوير الكرة للاعبين المحترفين عن طريق ضرب الكرة في نقاط منحرفة عن مركزها .**(Miscue) "**التسديدة الوهمية"**(Miscue) مشكلة فيزيائ

تحدث هذه المشكلة عندما يحاول اللاعب إدخال قيم انحراف ضخمة جداً وتبعد كثيراً عن مركز الكرة. في الواقع الفيزيائي، هذا يؤدي إلى انزلاق سن العصا وتوليد عزم دوران غير واقعي ومشوه على الإطلاق يمنع محاكاة اللعبة بشكل منطقي.

2. الحل:

لحماية استقرار المحرك الفيزيائي وضمان واقعية اللعبة، تم وضع قيد صارم وتلقائي داخل دالة الضرب المتقدمة applyAdvancedStrike الخاصة بالكرة.

يتلخص الحل في أن الكود يقوم فورا وفور تلقي الضربة بالتحقق من أن نقطة التلامس المطلوبة ت**70% فقط من سطح الكرة الحقيقي**رة الحقيقي**. فإذا رصد الكود أن اللاعب حاول إدخال قيم انحراف ضخمة تتجاوز هذا النطاق،**بتقييد هذه القيم وتخفيضها تلقائياً** تلقائياً** لإعادتها إلى الحدود الآمنة. يضمن هذا الحل بقاء نقطة الضرب ضمن النطاق الفيزيائي المنطقي للكرة، مما يمنع الحركات العشوائية ويحافظ على دقة المحاكاة.

بناءً على الشرح المفصل لتوابع المشروع في المستند المرفق، إليك تحليل دقيق لكيفية ربط تصادم الكرات مع المشهد الرسومي ثلاثي الأبعاد، متبوعاً بمناقشة المشكلة الفيزيائية الشهيرة التي تحدث أثناء التصادم وكيفية حلها برمجياً:

أولاً: كيف تم ربط تطبيق التصادم بين الكرات مع المشهد الرسومي؟

يحدث الربط بين الحسابات الفيزيائية الجافة والمشهد البصري المرئي (Three.js Scene) عبر حلقة تحديث متزامنة ومستمرة (Game Loop) تربط كلاس SimulationApp بكلاس PhysicsWorld على النحو التالي:

1. **التهيئة والربط الابتدائي** (initPhysicsWorld):
عند بدء اللعبة، يقوم كلاس SimulationApp بإنشاء الكرات في العالم الفيزيائي أولاً، ثم يبني لها فوراً كائنات ومجسمات رسومية (مجسم) رسومي يتحرك معها Mesh. تصبح كل كرة فيزيائية مرتبطة بـ (Three.js Scene) ثلاثية الأبعاد داخل مشهد اللعبة.

2. **التحديث الزمني اللحظي** (PhysicsWorld.update):
في كل فريم (إطار)، تستدعي حلقة التحريك الدالة update(dt) ممررة الفارق الزمني الصغير جداً بين الفريمات. تقوم هذه الدالة بتطبيق قوانين الحركة، ثم تستدعي دالة معالجة التصادم بين الكرات resolveBallCollisions().

3. **عكس النتائج على المشهد (حلقة الـ animate):**
بعد أن تقوم دالة التصادم بتعديل مصفوفات الإحداثيات والسرعات لكل كرة رياضياً، يتم في نهاية الفريم نقل هذه الإحداثيات الفيزيائية الجديدة وتطبيقها على موقع المجسمات الرسومية (Mesh.position) داخل المشهد. نتيجة لذلك، يرى اللاعب الكرات وهي ترتد وتتصادم على الشاشة بشكل واقعي متزامن تماماً مع الحسابات الفيزيائية الخلفية.

ثانياً: المشكلة وحلها (Penetration Problem - مشكلة تداخل واندماج الكرات)

1. المشكلة:

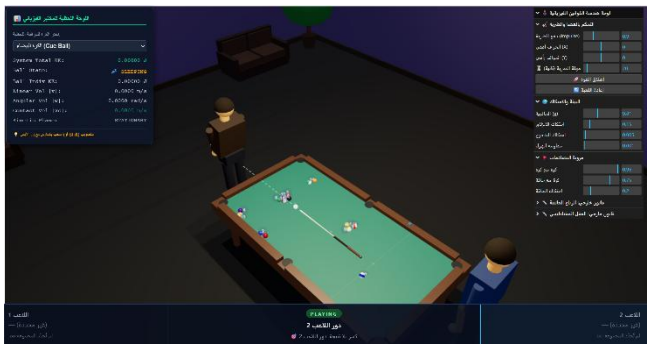
في المحركات الفيزيائية التي تعتمد على خطوات زمنية منفصلة (Discrete Time Steps - dt)، تظهر مشكلة شهيرة جداً عند تصادم الكرات؛ حيث تتحرك الكرات بسرعة كبيرة في فريم معين، وعند الفريم التالي يجد المحرك الفيزيائي أن الكرات قد تخطت حدود بعضها داخل بعضها **امتداخلة أو مندمجة داخل بعضها** (Overlapping / Penetration).

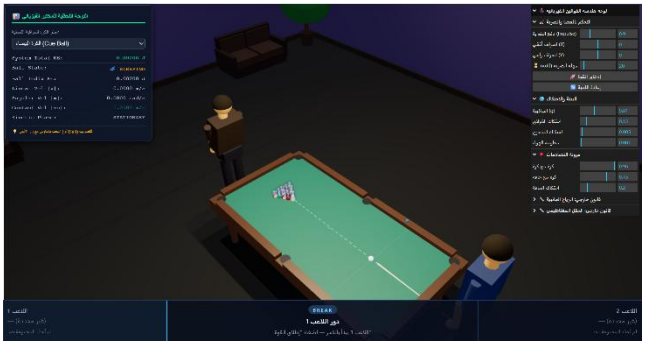
إذا لم تُعالج هذه المشكلة برمجياً، ستعلق الكرات داخل بعضها البعض، أو سيتولد عنها اهتزازات عشوائية وجنونية مشوهة للمشهد نتيجة لمحاولة المحرك حساب قوى ارتداد لانهائية من داخل مجسم الكرة.

2. الحل المشروح في الكود (resolveBallCollisions):

لإصلاح التصادم resolveBallCollisions() لحسم هذه المعضلة وضمان ارتداد نظيف وواقعي، طبق الكود حلاً ثنائياً ذكياً داخل الدالة على م**المرحلة الأولى: معالجة التداخل مكانياً (Positional Correction):** (rrection:**
يقوم الكود بعمل حلقة تكرارية مزدوجة لفحص المسافات بين كل كرتين . إذا وجد أن المسافة أقل من مجموع نصفي قطريهما (أي حدث تداخل بالفعل)، يق ما يدوياً **"دفع وإزاحة الكرتين بعيداً عن بعضهما يدوياً"** على طول خط التصادم بمقدار التداخل، وذلك لفصل أجسامهما تماماً ومنع اندماجهما قبل حساب أي **المرحلة الثانية: حساب الاندفاع الفيزيائي (Impulse Resolution):** (solution:**
بعد فصل الكرات مكانياً، يستدعي الكود معادلة الاندفاع لـ (collisionImpulseMagnitude) . الكرات المرنة يحسب السرعة النسبية بناءً على هذا الاندفاع، يتم منح كل كرة سرعتها الجديدة (e_ball). الكروي المخصص للكرات "معامل المرونة"بين الكرتين، ويطبق واتجاهها الصحيح المرتد للخارج بنعومة كاملة.

لتسجيل أول كرة لامستها الكرة (GameLogic) بالإضافة إلى ذلك، يستغل الكود لحظة التصادم النظيفة هذه لإرسال إشعار لمنطق اللعبة البيضاء، وهو أمر أساسي لتطبيق قوانين البلياردو الرسمية بدقة .





الخاتمة

توصلت هذه الدراسة إلى تحليل فيزيائي متكامل يكشف عن عمق الترابط بين مبادئ الميكانيكا الكلاسيكية وبساطة حركة كرة البلياردو الظاهرة. يمكن إجمال أبرز النتائج الفيزيائية في ما يلي:

أولاً: النقطة المثالية للضرب ($h = \frac{7}{5} r$): تُثبت الدراسة أن ضرب الكرة على ارتفاع 1.4 ضعف نصف قطرها يُلغي الانزلاق فوراً، مما يجعل الدوران والحركة الانتقالية في تناغم تام من اللحظة الأولى.

ثانياً: مراحل الحركة الخمس: تُظهر الدراسة أن الكرة تمر بتطور ديناميكي دقيق من النبضة اللحظية، مروراً بالانزلاق، وصولاً إلى التدحرج الطبيعي ثم السكون — وكل مرحلة محكومة بمعادلات واضحة.

ثالثاً: قاعدة الـ 90 درجة: النتيجة الأكثر أناقة رياضياً؛ فمن حفظ الزخم وحفظ الطاقة معاً ينبثق حتماً أن مساري الكرتين بعد التصادم متعامدان.

رابعاً: ديناميكا التصادمات المتعددة: سلاسل القوة والتبدد الأسي للطاقة عبر صفوف المثلث تفسر ظاهرة التوزيع غير المتساوي وتوقف الكرات الداخلية.

خامساً: الجدار المطاطي: يتحكم معامل الارتداد (e_c) والاحتكاك المماسي في تشكيل زاوية الخروج الحقيقية، بينما تُضاف طبقة إضافية من التعقيد القابل للتحكم. الدورانية (Spin) (< 1)

في المحصلة، تُقدم هذه الدراسة إطاراً نظرياً صارماً يُمكّن من بناء محاكيات فيزيائية دقيقة، ويُغني فهم أي لاعب محترف لأسباب استجابة الكرة لضرباته.