

TNM087 - MoS PROJEKT
GROUP 7 - ICE ICE BABY



Viktor Nilsson, vikni067@student.liu.se
Amaru Ubillus, amaub859@student.liu.se
Anders Nord, andno922@student.liu.se
Mirza Talovic, mirta980@student.liu.se

11 Mars 2012



Sammanfattning

Denna rapport kommer att behandla och fördjupa sig i simulering av curling. Syftet med projektet är att ta reda på hur olika parametrar påverkar curlingklots beteende. Rapporten skall ge läsaren förståelse för hur kollisioner mellan curlingklot går till med hjälp av en visualisering gjord i OpenGL och C++. Under projektets gång har en realistisk och fysikaliskt korrekt simulering av curlingklots grundläggande beteende tagits fram. De parametrar som studeras och varieras i det slutliga programmet är klotens hastighet, kollisionsvinklar samt olika rotationshastigheter.

Förord

Projektet Ice ice Baby genomfördes i kursen Modellbygge & simulering som är en del av en civilingenjörsutbildning i medieteknik. Ett tack riktas till: Ulf Sannemo och Anna Lombardi för god handledning.

Innehållsförteckning

1	Inledning	1
1.1	Bakgrund	1
1.2	Syfte	1
1.3	Curling	2
1.4	Spelregler	3
2	Metod	4
2.1	Beräkningar av friktionskoefficienten	4
2.2	Numerisk metod	4
2.3	Kollision	6
2.4	Klotets rotation	7
2.5	Simulering i MATLAB	10
2.6	Implementation av animationen	10
2.7	Vikten av visualiseringen i projektet	11
2.8	Förenklingar	12
3	Programmet	13
3.1	Programstruktur	13
3.2	Användargränssnitt	13
4	Resultat	14
5	Diskussion	15
6	Referenser	16

1 Inledning

Projektet innefattar ett modelleringsprojekt där uppgiften är att på valfritt sätt modellera och simulera ett fysikaliskt fenomen. Sporten Curling kommer behandlas där fysikaliska fenomen så som kollisioner och curlande klot kommer att modelleras för att sedan simuleras.

1.1 Bakgrund

Sporten curling bygger på många fysikaliska förlopp men är inte ett vanligt område att varken simulera eller visualisera. Det finns gott om vetenskapliga artiklar som behandlar fysiken bakom curling, dock existerar endast ett fåtal fysikaliskt korrekta simulationer. Detta gav upphov till idén om att försöka skapa en curlingsimulator med dessa rapporter som grund.

1.2 Syfte

Syftet med projektet var att skapa en fysikaliskt korrekt curlingsimulator.

1.3 Curling

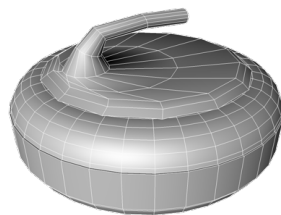
Curling är en precisionssport där man tävlar i lag och där de tävlande turas om att skjuta iväg klot mot en måltavla målrad på isen.

Curlingbanan är 45,72 meter lång och 5 meter bred. Curlingklotet som ska vara runt, är gjort av granit som har en medeldensitet mellan: 2.65 och 2.75 g/cm^3 . På ovasidan av klotet finns ett handtag.

Enligt internationell standard får stenen inklusive handtag och bult väga maximalt 19,96 kilo och som minst 17,24 kg. Curlingklotets omkrets får inte överstiga 914 mm och höjden får inte understiga 114 mm. Varje sten måste även ha en tydlig markering.

Den tunga massan samt temperaturen på isen gör det möjligt för stenen att färdas längs hela curlingbanan utan att hastigheten avtar drastiskt.

[1]



Figur 1: Tidig modell av curlingklotet.

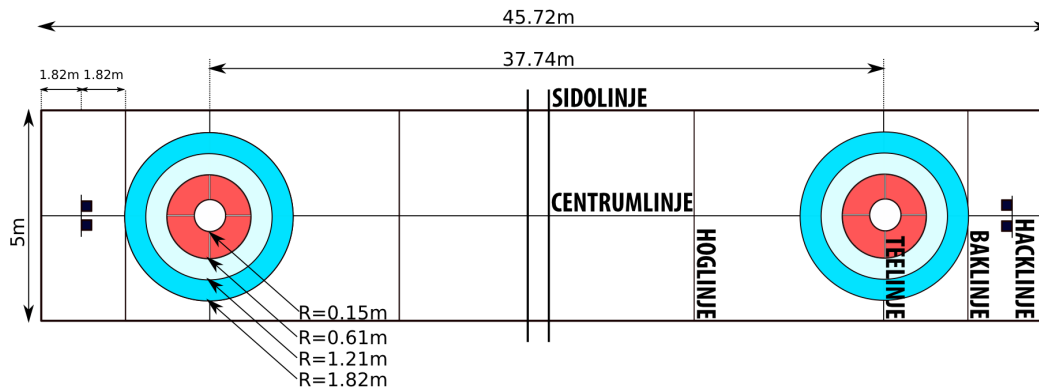
1.4 Spelregler

När en sten sätts i spel så är första linjen den passerar *baklinjen*, se Figur 2. Spelaren måste släppa stenen innan den nått fram till *hoglinjen*. För att en sten ska räknas som spelbar så måste den stanna mellan nästa *hoglinjen* och den *baklinjen* som är längst bort. Händer inte detta eller om kloten passerar sidolinjerna plockas kloten bort från spel.

Boet kallas den måltavla som är målad på isen och är ca 3,6 meter i diameter. Spelet går ut på att placera sina klot närmare cirkelns mittpunkt än motståndaren. Det ges ett poäng för varje sten som ligger närmare cirkelns mittpunkt än motståndarens vid spelomgångens slut. Det lag som vinner omgången startar nästa omgång.

Genom att sopa framför stenen påverkas stenens rörelsebana. Friktionen minskar vilket gör att kloten glider längre men även "curlar" mindre. Detta innebär att klotet aviker från sin bana.

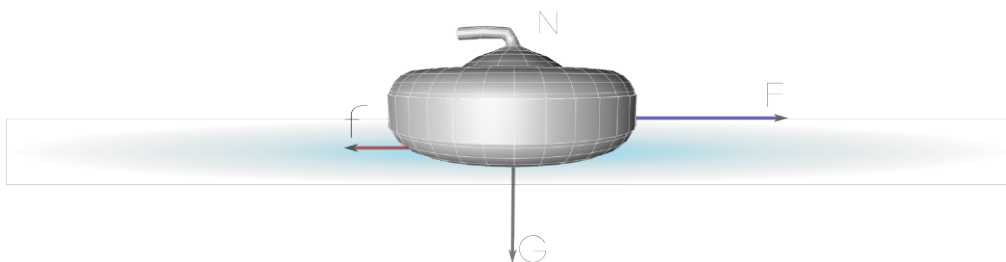
[1]



Figur 2: Curlingbanans utseende och mått

2 Metod

2.1 Beräkningar av friktionskoefficienten



Figur 3: Krafter som påverkar curlingklotet då det ligger på isen

För att uppskatta hur mycket friktion som finns mellan curlingklotet och isen beräknas en friktionskonstant. Beräkningarna baseras på Figur 5 där N = normalkraften, g = gravitationskraften, F = den applicerade kraften och f är friktionskraften.

Friktion: Friktionen är den kraft som är i motsatt riktning till rörelseriktningen. Den är proportionell mot den kraft som håller klotet mot isens yta, gravitationen i det här fallet. Normalkraften har motsatt riktning till gravitationen. Friktionskonstanten ges av kvoten mellan normalkraften och friktionskraften.

Beskrivning av friktionskoefficienten:

$$F_k = -\mu mg \quad (1)$$

F_k är friktionskraften, μ är friktionskonstanten, m är klotets massa och g är gravitationen

2.2 Numerisk metod

För att teoretiskt kunna simulera klotens rörelse i systemet användes den numeriska metoden Eulerapproximation. Det är en stegmetod som används för att lösa ordinära differentialekvationer med givna initialvärden.

Eftersom curling inte innefattar alltför varierande rörelser innebär detta att förloppet går att simulera via en Eulerapproximation med konstant steglängd.

För att beräkna curlingstenarnas position användes följande formler:

$$\vec{a} = -\mu\vec{v}g \quad (2)$$

$$\vec{v} = \vec{v} + \vec{a}\Delta t \quad (3)$$

$$p\vec{o}s = p\vec{o}s + \vec{v}\Delta t \quad (4)$$

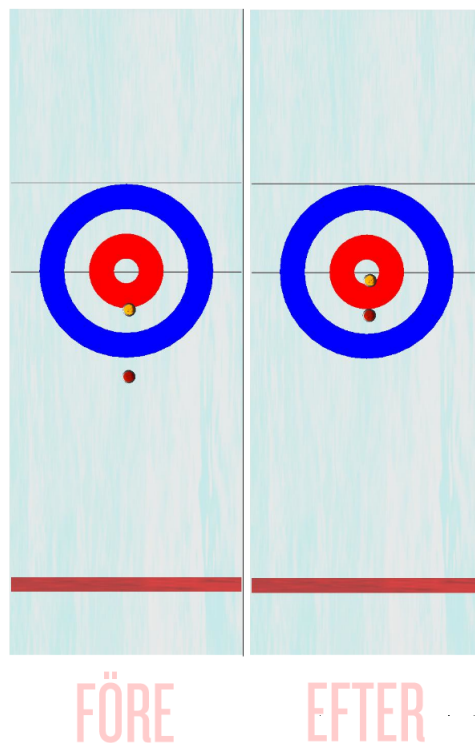
Där \vec{a} är accelerations-vektorn, μ är friktionskonstanten, \vec{v} är hastighets-vektorn, g är gravitationskonstanten, $p\vec{o}s$ är positions-vektorn och Δt är den konstanta steglängden.

Beräkningar utförs utifrån formlerna ovan, däremot modifieras (2) för att stämma överens med mätvärden utifrån vetenskapliga artiklar[2]. Ekvationen ändras enligt (5).

$$\vec{a} = -\mu(\vec{v} + \vec{v}_m\mu\omega)g \quad (5)$$

Där \vec{v}_m är \vec{v} med ombytta element i vektorn och ω är rotationshastigheten.

2.3 Kollision



Figur 4: Kloten före och efter kollision.

En kollision kan beskrivas som en händelse då två objekts ytor möts under en mycket kort tidsperiod. Under denna tid påverkas båda objekten av samma men motriktade krafter.

Det finns generellt två olika typer av kollisioner: elastiska och inelastiska. Den totala kinetiska energin då objekten kolliderar bevaras om kollisionen är elastisk, alltså då den kinetiska energin är densamma före och efter kollisionen. Om den kinetiska energin överförs till andra former av energi som exempelvis värme eller ljud så är inte den kinetiska energin bevarad. En sådan kollision kallas istället inelastisk.

Simuleringsproblemet kan generaliseras till kollision i två dimensioner mellan två cirklar. I början generaliserades beräkningarna genom att bara titta på kollision mellan två lika massor.

Vid kollisionshanteringen implementeras (6) för att beräkna det euklidiska avståndet mellan kloten.

$$d = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2} \quad (6)$$

1. Det första steget blev att undersöka kollision mellan två partiklar i en dimension, med givna initialhastigheter. Hela kollisionen sker med konstant friktion.
2. Detta utvecklas så att friktionen kan variera med klotets hastighet.
3. Kollision mellan två objekt i två dimensioner.
4. Kollision i två dimensioner mellan flera objekt.
5. Utöka systemet genom att lägga till varierande kollisionsvinklar, olika massa och implementera curlingklotets karaktäristiska rotation (Figur 3).

Beräkningarna för en *generell inelastisk kollision i två dimensioner*[3] sker via:

$$\alpha = \frac{y_2 - y_1}{x_2 - x_1} \quad (7)$$

$$Q = 2 \frac{v1_x - v2_x + \alpha(v1_y - v2_y)}{(1 + \alpha^2)(1 + \frac{m_2}{m_1})} \quad (8)$$

$$vx_{cm} = \frac{m_1 v1_x + m_2 v2_x}{m_1 + m_2} \quad (9)$$

$$vy_{cm} = \frac{m_1 v1_y + m_2 v2_y}{m_1 + m_2} \quad (10)$$

$$v1'_x = (v1_x - \frac{m_2}{m_1} Q - vx_{cm})R + vx_{cm} \quad (11)$$

$$v1'_y = (v1_y - \alpha \frac{m_2}{m_1} Q - vy_{cm})R + vy_{cm} \quad (12)$$

$$v2'_x = (v2_x + Q - vx_{cm})R + vx_{cm} \quad (13)$$

$$v2'_y = (v2_y + \alpha Q - vy_{cm})R + vy_{cm} \quad (14)$$

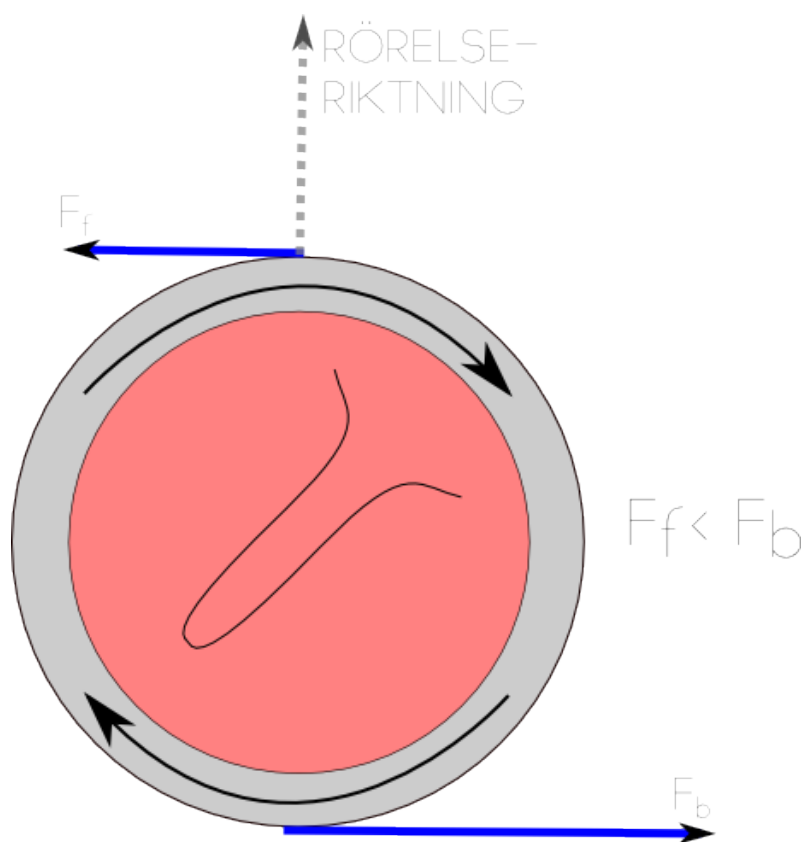
(α kollisionsvinkel, Q kvadratisk ekvation, vx_{cm} masscentrum x-led, vy_{cm} masscentrum y-led, $v1$ hastighet för första klotet innan stöt, $v1'$ hastighet för första klotet efter stöt. m_1 första klotets massa, R stötkoefficient)

Notera att formlerna ovan (6)-(13) är utformade så att energi och rörelsemängd alltid bevaras.

2.4 Klotets rotation

Orsaken till att klotet curlar är friktionen mellan isen och klotet, curlingklotets kontaktyta mot isen samt klotets rotation. Roterar curlingklotet medurs

vid "släppet" kommer det att curla höger och vänster om det roteras moturs. Rotationen hos curlingklotet är karaktäristisk då tex ett glas som roteras medurs på en plan yta svänger åt vänster istället. Sopandet framför curlingstenen genererar en reducerad friktion, detta leder till att klotet kan förflytta sig ca 5 meter ytterligare. Anledningen till att klotet färdas längre är att friktionen mellan borsten och isen genererar värme. Detta leder till att isen värms upp och på så sätt skapas en tunn film av vatten som gör att klotet färdas längre. Genom att sopa minskar man även stenens curlande då friktionen mellan stenen och isen blir lägre.



Figur 5: Krafter som påverkar curlingstenens rotation.

Klotets rotation räknas fram med hjälp av (2),(3) och (4). Friktionen mellan klotet och isen baseras på formler utifrån den vetenskapliga artikeln *The physics of sliding cylinders and curling rocks*[2] som tagit fram sina resultat med hjälp av experiment. Nedan presenteras de mest centrala

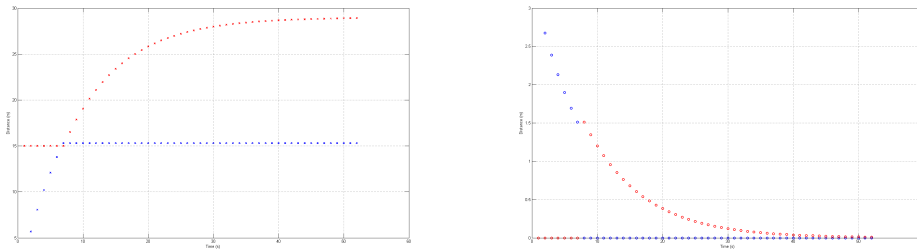
ekvationerna i resultatet:

$$\mu_k = \frac{c_f}{\sqrt{v}} \quad 0 \leq \theta < \pi \quad (15)$$

$$= \frac{c_b}{\sqrt{v}} \quad \pi \leq \theta < 2\pi \quad (16)$$

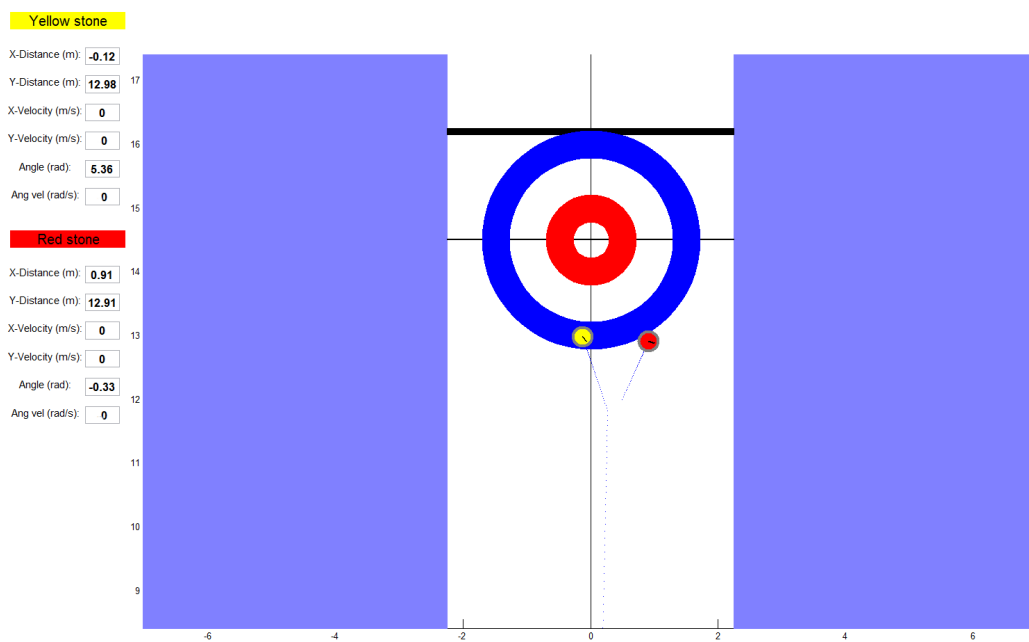
μ_k är friktionskonstanten, c_f och c_b är skalärer som representerar skillnaden mellan F_f och F_b i Figur 5. Ekvationerna (14) och (15) påverkar klotets translation och rotation indirekt i och med att μ_k används i stegmetoden.

2.5 Simulering i MATLAB



Figur 6: Positioner (vänster) samt hastigheter (höger) för kloten före och efter kollision.

Efter att modellen tagits fram implementerades den i MATLAB för att kontrollera resultatens relevans. I detta stadie generaliserades koden hela tiden för att underlätta framtida arbete och förhindra eventuella problem.



Figur 7: MATLAB-simulering

2.6 Implementation av animationen

Programmet skrevs i C++. Ramverket OpenGL och *freelut* används för att visualisera det fysikaliska förloppet som sker.

OpenGL har fördelen att det är ett lågnivåspråk som ger god kontroll över de operationer som krävs. *Freeglut* som är ett vertyg till OpenGL för att öppna fönster valdes för att det är *opensource* och innehåller många bra lättanvända funktioner. Båda dessa är väl dokumenterade vilket är viktigt för att eventuella problem ska kunna lösas snabbt. Det är även viktigt att använda ett plattformsoberoende format som stöds både i mac-miljö och pc-miljö eftersom båda är vanligt förekommande.

Med hjälp av OpenGL kan implementering av 3D-modeller, tangentbord och mus göras. Det finns även bra alternativ för att representera ljuskällor, göra geometriska transformationer och skapa önskvärda visuella effekter. Exempel är ljusets spegling i kloten.

Till en början skapades allt i en 2D miljö. Sedan utvecklades en 3D-miljö då det inte innebar några extra problem beräkningsmässigt. Anledningen till detta är att beräkningarna endast sker i 2D.

Stora delar av koden delas in i klasser för att bibehålla en ren och lättunderhållen programmeringsmiljö.

Wavefront Obj är valet av 3D-modellfilformat för att det innehåller möjligheterna att lagra normaler, face-definitioner, textur-koordinater och vertex-listor. Obj-formatet går även att exportera från Blender som är ett *opensource* 3D-modelleringsprogram.

GLM används för att läsa in objektfilerna. Det är ett bibliotek som kan rita ut modellerna på skärmen exakt så som de ser ut i Blender. GLM klarar även av att läsa in materialfiler som skapas i samband med exporteringen av obj-filen.

2.7 Vikten av visualiseringen i projektet

Fokus har lagts på visualiseringen då en väl genomtänkt sådan förtydligar resultaten. Valet att använda realistiska modeller stärker trovärdigheten i resultatet samt den visuella upplevelsen.

2.8 Förenklingar

Simuleringen är främst koncentrerad på kollisionshantering och rotation, därav har följande försumrats:

- Luftmotstånd, simuleringen utförs i en vakuum-miljö.
- Klotens karaktäristiska form kombinerat med deras tyngd gör att luftmotståndet är ytterst litet, och därmed påverkas inte resultatet märkvärt om detta försummas.
- Isens karaktäristiska yta, i curling brukar man innan matcher spruta vattendroppar på isen, dessa fryser och ger isen en speciell yta. (isen kan liknas vid texturen av ett apelsinskal)
- Putsande framför klotet
- Isens temperatur; denna parameter påverkar klotets beteende mycket, då den har en inverkan på både curlande och den tillryggalagda sträckan. Anledningen till detta är att stenens karaktäristiska beteende är att temperaturen på isen är låg och därmed lättsmält. Ju kallare isen är desto mindre kommer klotet att curla och sopandet framför kommer inte att påverka klotets sträcka.
- Simuleringen innefattar bara curlingklotets beteende efter att det lämnat spelarens hand (dvs vid hoglinjen).
- Kloten är perfekt runda; detta kan åtgärdas genom att lägga till en brusvariabel som simulerar små defekter i klotens kontaktyta

3 Programmet

3.1 Programstruktur

Nedanför följer en beskrivning av programmets funktioner och klasser.

- `curling_stone.cpp` I klassen skapas curlingkloten och en 3D modell ritas ut i programmet.
- `curling_collision.cpp` Hanterar kollisioner mellan curlingkloten där positioner och hastigheter ändras efter att kloten har kolliderat med varandra.
- `euler.cpp` Innehåller en stegfunktion som beräknar nya positioner, hastigheter och vinkelhastigheter för varje individuellt curlingklot.
- `environment.cpp` Skapar den bakgrundsmiljö som finns i programmet.
- `curling_main.cpp` Huvudfilen för programmet där information hämtas från alla andra klasser för att sedan starta programmet. Här hanteras även knapptryckningar från tangentbordet samt uppdatering av grafiken.

3.2 Användargränssnitt

Vid en ny curlingstöt kan parametrar så som initialhastighet, utgångsvinkel samt vinkelhastighet ställas in före varje nytt klot. Dessa värden anges och ändras med tangenterna enligt Tabell 1.

För att sedan skjuta iväg ett klot med valda inställningar trycker användaren på mellanslagstangenten. Kameravinkeln kan justeras med siffertangenterna 1-3.

Tabell 1: Tangenter för att interagera med programmet.

Händelse	Kommando
Skjuta ut ett nytt curlingklot	Mellanslag
Curlingklotets utgångshastighet	i,k
Curlingklotets utgångsvinkel	i,l
Rotationshastighet	u,o
Byt Kameravinkel	1,2,3
Ändra kamerans position	a,w,d,s
Ändra kamerans rotation	f,t,h,g
Ny spelomgång (kloten nollställs)	Enter

4 Resultat

Curling har modellerats utifrån en fysikalisk modell och en simulering har skapats i MATLAB. Simuleringen har sedan implementerats med C++ i OpenGL och en visualisering i 3D miljö har tagits fram.

Resultatet som har uppnåtts är trovärdigt då kollisionerna mellan curlingkloten visualiseras korrekt och beter sig enligt fysikens lagar. Utifrån beräkningarna och simuleringarna i MATLAB bevisas att sträckan kloten färdas är realistisk jämfört med resultatet från vetenskapliga artiklar. Detta gäller även för klotets rotation[3]. Resultatet är visuellt rimligt då modeller skapats utifrån verkliga utseenden och mått.

Det som utförts för att öka realismen i resultatet förutom de fysikaliska egenskaper som beskrivits är; texturering för att efterlikna verkliga material, samt ljussättning för att framhäva materialegenskaper samt belysa curlingbanan.

5 Diskussion

Prioriteringar kunde i ett tidigt stadium göras baserat på de beslut som nämnts i del 3.8, detta resulterade i att arbetet underlättades under projektets gång. Fokus kunde därför läggas på de mer väsentliga delarna av simuleringen som kollisionshanteringen.

Det finns utrymme för vidareutveckling och förbättring av programmet. Förslag på förbättringar, vilka skulle öka den visuella upplevelsen är att lägga på texturer på objekten som då skulle uppfattas mer realistiska. Att lägga till speglingar mellan is och curlingkloten är också ett sätt att få isen att se mer blank ut.

Det är också möjligt att utveckla programmet till ett komplett fungerande spel med poängberäkningssystem och regler enligt internationella curlingregler.

Att förstå sig på varför ett curlingklot ”curlar” har visat sig vara en svår uppgift då det är komplicerat och det inte finns mycket information dokumenterat kring detta. Faktum är att olika vetenskapliga artiklar presenterar olika svar på vad som orsakar fenomenet, vilket innebär att problemet är kontroversiellt. Beräkningarna i denna rapport har utgått ifrån en pålitlig källa [3] vilket har resulterat i relevanta resultat.

Problem har även uppstått i samband med implementering OpenGL. Att läsa in 3D modeller med material i OpenGL har visat sig vara problematiskt då många av de 3D program som finns sparar modeller på olika sätt. För att lösa problemet användes en färdig klass GLM för att läsa in 3D-modeller av typen OBJ samt en materialfil, MTL.

Att få kollisionen att fungera fysikaliskt korrekt mellan flera curlingklot i alla möjliga vinklar har varit ett återkommande problem då beräkningarna för kollisionshanteringen utfördes med hjälp utav loopar.

6 Referenser

- [1] Svenska Curlingförbundet, 2012, http://www.curling.se/Portals/0/Dokument/2011/Regelboken_2010.pdf.
- [2] The physics of sliding cylinders and curling rocks , A. Raymond Penner
Malaspina University-College, British Columbia, 2 Feb, 2000.
- [3] Elastic and Inelastic Collision in Two Dimensions , Smid Thomas
M.Sc. Physics, Ph.D. Astronomy. <http://www.plasmaphysics.org.uk/print/collision2d.htm>.

Bibliotek för att reproducera projektet:

GLM. Bibliotek för att läsa in 3D modeller
<http://www.3dcodingtutorial.com/Working-with-3D-models/Getting-GLM.html>

Grafiskt bibliotek till OpenGL.
Freeglut: <http://freeglut.sourceforge.net/>

Bibliotek för att hantera 3D grafik.
OpenGL: <http://www.opengl.org/>