

Simulering av kastbana

Lars Bergman
Christoffer Engelbrektsson
Jens Jakobsson
Johan Nordin

Institutionen för teknik och naturvetenskap (ITN)
Linköpings universitet

12 mars 2015

Sammanfattning

Projektet har resulterat i en webbapplikation för simulering av kastbanor. Matematiskt och fysikaliskt korrekta modeller togs fram för hur tyngdacceleration, luftmotstånd och vind påverkar ett sfäriskt objekt. Projektgruppen tog även fram en simpel kollision med olika material för sfärer. Objekten som skjuts iväg har en utgångshastighet, vinkel, radie samt ett material. Detta simuleras sedan i både MATLAB och WebGL. Projektgruppen har arbetat i grupp med en tydlig projektplan och översiktlig tidsplanering. Projektlogg och möten med examinator och assisterande lärare har varit ett återkommande moment under projektet. Resultat blev en webbapplikation i 3D som kan rendera en kastbana i realtid. Projektgruppen är nöjd med resultatet och anser att de har tagit fram en tydlig och interaktiv simulering för kastbanor. En del avgränsningar har skett, däribland har både luftmotstånd och vind konstanta värden. Vilket inte är fysiskt korrekt för en realistisk simulation av en kastbana.

Innehållsförteckning

Sammanfattning	i
Figurer	iii
Tabeller	iv
1 Inledning	1
1.1 Syfte	1
1.2 Avgränsningar	1
1.3 Arbetsgång	1
2 Utförande	3
2.1 Matematiska modellen	3
2.1.1 Kastbana utan luftmotstånd	3
2.1.2 Kastbana med luftmotstånd	4
2.1.3 Kastbana med vind	5
2.1.4 Kollision med fast yta	5
2.2 Simulering	7
2.2.1 Numeriska metoder	7
2.2.2 MATLAB	7
2.2.3 WebGL	8
3 Resultat	10
3.1 Diskussion	11
3.2 Slutsats	12
Litteraturförteckning	13

Figurer

1.1	Gantt-schema för tidsplaneringen	2
2.1	Illustration av luftens påverkan med ekvationer	4
2.2	Illustration av vindens påverkan med ekvationer	5
2.3	Illustration av en kollision med en fast vägg	6
2.4	Jämförelse mellan stegmetoder i MATLAB	8
2.5	Första versionen av implementationen i WebGL	9
3.1	Slutversionen av implementationen	10

Tabeller

2.1	Tabell över material och dess tillhörande värden	9
2.2	Konstanter	9

Kapitel 1

Inledning

Projektgruppen har tagit fram en simulering för kastbanor under påverkan av tyngdacceleration, luftmotstånd och vind. Tanken är att simuleringen i realtid ska visualisera hur en kastbana för ett sfäriskt objekt med olika egenskaper ser ut.

Med hjälp av ett användargränssnitt kan användaren ändra olika parametrar. Exempel på parametrar är storleken på sfären, dess utgångshastighet, samt utgångsvinkel. Dessutom kan användaren ändra vinden genom att ändra dess kraft och vinkel.

Simuleringen använder sig av Eulers stegmetod för att räkna ut den nya positionen.

1.1 Syfte

Syftet med projektet är att ta fram en fysikaliskt korrekt modell för en kastbana och sedan simulera den i valfritt programspråk.

1.2 Avgränsningar

Alla modeller har begränsats så att beräkningar sker i två dimensioner och inte tar hänsyn till den tredje axeln. Att simuleringen är visualiserad i en 3D-miljö är enbart för att simuleringen ska vara mer visuellt tilltalande. Det förutsätts att luftmotståndet och vinden är konstant oavsett höjd eller längd på kastbanan. Simulationen tar inte hänsyn till rotation och alla värden för materialen är generella värden.

1.3 Arbetsgång

I inledningen av projektet började gruppen med att ta fram en projektplan. Planen innehöll en beskrivning av projektet och krav på vad som skulle göras. En tidsplanering skapades i form av ett Gantt-schema (se figur 1.1) och fungerande som en övergripande planering för projektet. Varje arbetsdag inleddes med ett kort möte om vad som gjorts och vad som skulle göras under dagen. Mötet kompletterades med en projektlogg som fylldes i efter varje dag.

Förslag											
Model											
Implementation											
MATLAB											
Språk											
Rapport											
Presentation											
Vecka	4	5	6	7	8	9	10	11	12	13	

Figur 1.1: Gantt-schema för tidsplaneringen

Projektgruppen tillämpade par-programmering, individuellt arbete samt arbete i helgrupp, där arbete i helgrupp var den dominerande arbetsformen. Som versionshanterare har programvaran Git används och för dokumenthantering har gruppen använt Google Drive. Det gav gruppen bra översikt över projektets progression och underlättade versionshantering av koden. Projektets milstolpar bestod av möten med examinator och assisterande lärare, där projektets status och fortsatta utveckling diskuterades. Mötena hölls regelbundet en gång per vecka.

Kapitel 2

Utförande

2.1 Matematiska modellen

För att beskriva en rörelse t.ex. en kastbana är det vanligt att man gör det med hjälp av ett tvådimensionellt koordinatsystem. Där y-riktningen förknippas med den vertikala rörelsen och riktningen i x-led är den horisontella rörelsen [1].

I beräkningen av vindens påverkan på bollen används funktionen *arctan*. Den kan vara problematisk eftersom olika programmeringsspråk kan använda olika metoder för att räkna ut den. Det kan ge skillnader i resultaten mellan programmen.

Vår matematiska modell utökades kontinuerligt under projektets gång. Genom att börja med en väldigt grundläggande kaströrelse och därefter lägga till flera externa krafter såsom luftmotstånd.

2.1.1 Kastbana utan luftmotstånd

Grunden för hela modellen består av de kinematiska rörelseekvationerna i två dimensioner. [2] I en dimension kan den allmänna rörelseekvationen skrivas på följande form.

$$p(t) = p_0 + v_0 t + \frac{1}{2} a(t) t^2 \quad (2.1)$$

I ekvation 2.1 beskrivs positionen $p(t)$ för ett objekt med hjälp av en startposition p_0 , en starthastighet v_0 , och en acceleration a .

Deriverar man ekvation 2.1 med avseende på tiden får man följande ekvation.

$$v(t) = v_0 + a(t)t \quad (2.2)$$

Ekvationen 2.2 beskriver objektets hastighet som en funktion av tiden. Deriverar man ekvation 2.2 ytterligare en gång med avseende på tiden fås följande ekvation.

$$a(t) = a \quad (2.3)$$

Ekvation 2.3 beskriver acceleration, som är ett mått på hur snabbt hastigheten varierar för objektet.

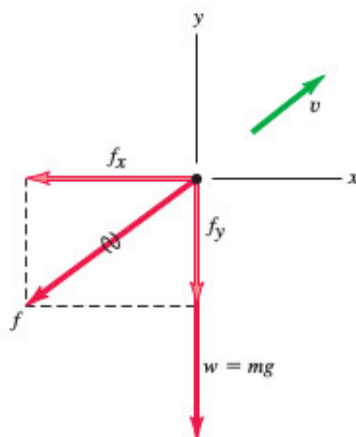
I den första modellen av kastbanan antas att objektet har en konstant tyngdacceleration i y-led och ingen acceleration i x-led. Då kan den generella ekvationen 2.1 slutligen skrivas på följande form.

$$\begin{aligned}x(t) &= x_0 + v_0 t \\y(t) &= y_0 + v_0 t - \frac{1}{2}gt^2\end{aligned}\tag{2.4}$$

Grundekvationerna för vår modell består alltså av en vågrät rörelse med konstant hastighet och en lodrät rörelse med konstant acceleration.

2.1.2 Kastbana med luftmotstånd

Vid påverkan av luftmotstånd kommer bollens densitet och radie ha en inverkan. En illustration av hur det ser ut går att se i 2.1.



Figur 2.1: Illustration av luftens påverkan med ekvationer

Ekvation 2.5 visar hur luftmotståndet räknas ut[3]. Där ρ är densiteten av bollens material, C_d är en koefficient som beror på objektets form och A är tvärsnittsarean av objektet. En generell och något förenklad formel för luftmotståndet ges av ekvation 2.5.

$$F_d = \frac{1}{2}\rho C_d A\tag{2.5}$$

Genom att förenkla $\rho C_d A$ med en konstant D , samt veta att $F_d = ma_d$ så går det att ta reda på bollens acceleration med ekvation 2.6. Sedan är det bara att stoppa in den nya accelerationen i ekvation 2.4.

$$a(t) = -\frac{1}{2} \frac{Dv(t)^2}{m}\tag{2.6}$$

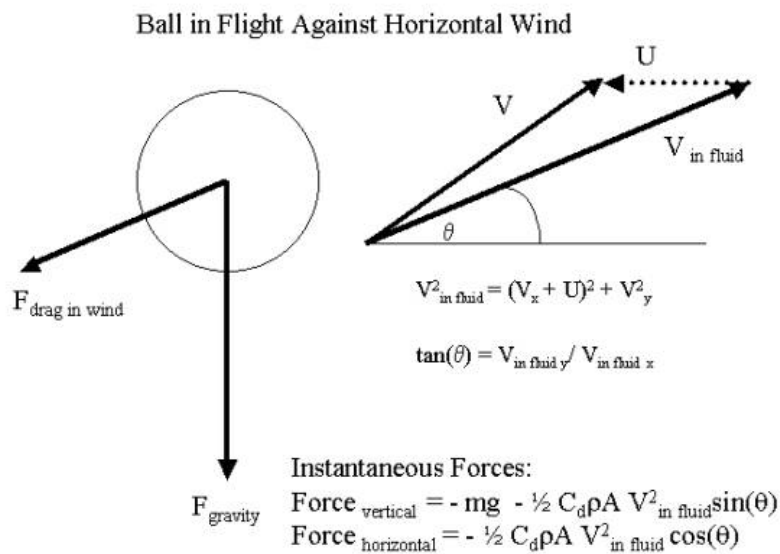
2.1.3 Kastbana med vind

Om en vind påverkar bollen utöver luftmotståndet behövs beräkningen för accelerationen modifieras för att ta hänsyn till vindens hastighet[4]. Då används ekvation 2.7, där U_x och U_y är vindens hastighetskomponenter och sedan beräknas den nya vinkeln med ekvation 2.8. För att beräkna den nya hastigheten av bollen, stoppas de nya värdena in i ekvation 2.9.

$$v^2(t) = (v_x(t) + U_x(t))^2 + (v_y(t) + U_y(t))^2 \quad (2.7)$$

$$v_{ang}(t) = \arctan(\text{abs}(\frac{(v_y(t) + U_y(t))}{(v_x(t) + U_x(t))})) \quad (2.8)$$

Figur 2.2¹ visar hur vinden påverkar ett föremål.



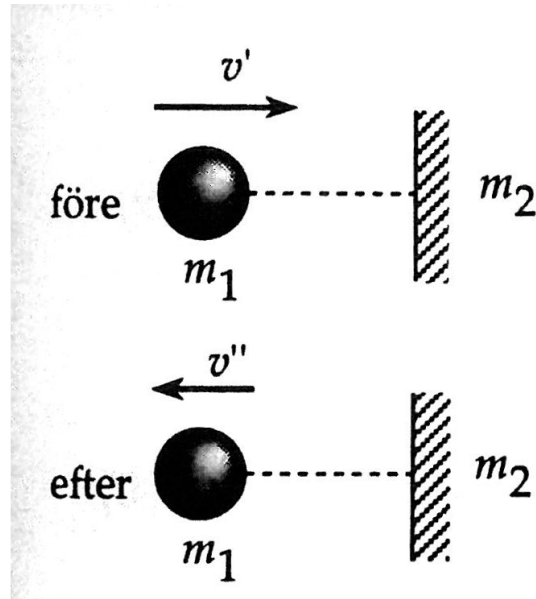
Figur 2.2: Illustration av vindens påverkan med ekvationer

$$\begin{aligned} a_x(t) &= -\frac{1}{2} \frac{Dv(t)^2 \cos(\theta)}{m} \\ a_y(t) &= -g - \frac{1}{2} \frac{Dv(t)^2 \sin(\theta)}{m} \end{aligned} \quad (2.9)$$

2.1.4 Kollision med fast yta

För att ta fram en stötkoefficient för en fast yta får man utgå ifrån ekvation 2.10. Hur en kollision kan se ut illustreras i figur 2.3.

¹<http://madsci.org/posts/archives/Mar2003/1046720933.Ph.r.html>



Figur 2.3: Illustration av en kollision med en fast vägg

$$m_1 v'_1(t) + m_2 v'_2(t) = m_1 v''_1(t) + m_2 v''_2(t) \quad (2.10)$$

v'_k är hastigheten före kollisionen och v''_k är hastigheten efter kollisionen. m_k är objektens tillhörande massor. Om ett objekt kolliderar med en fast yta låter man den massan gå mot oändligheten. Ekvationen delas därefter med massan. Det leder till ekvation 2.11 som är stötrelationen mellan ett rörligt objekt och en fast yta.

$$v'_2(t) = v''_2(t) = 0 \quad (2.11)$$

Sedan kan ekvationen slutligen förenklas till ekvation 2.12. Där v' är hastigheten före kollision och v'' är hastigheten efter kollisionen. e är stötkoefficienten som varierar beroende på materialet. Den varierar mellan 0 till 1, där 1 är helt elastiskt, det vill säga att all rörelseenergi bevaras.

$$v''(t) = e v'(t) \quad (2.12)$$

2.2 Simulering

2.2.1 Numeriska metoder

Vår modell för en kastbana representeras med ett system av differentialekvationer. För att kunna utföra simuleringen kommer dessa differentialekvationer lösas genom numeriska metoder. Numeriska metoder är ett kraftfullt verktyg för att lösa differentialekvationer. Den enklaste numeriska metoden för att lösa ett begynnelsevärdeproblem är Eulers stegmetod.

$$x_{n+1} = x_n + h_n \cdot f(t_n, x_n) \quad (2.13)$$

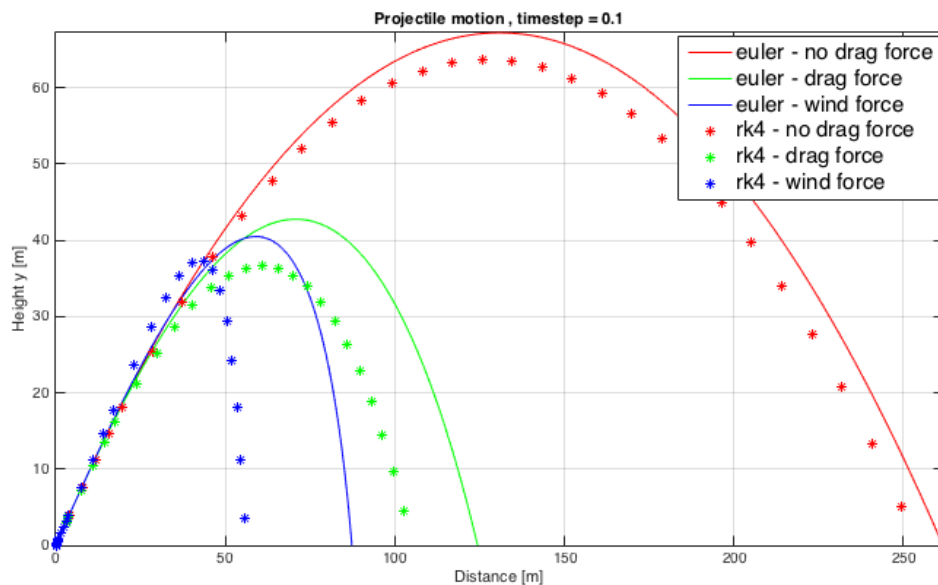
I ekvationen 2.13 ges n :te-stegstorleken ges av $h_n = t_{n+1} - t_n$

Eulers stegmetod är en rekursiv ekvation som kommer att beräkna ett nytt värde med en steglängd av h . Metoden approximerar lösningen på differentialekvation. Genom att beräkna nästa punkt på kurvan som följer tangentens riktning.

Eulers stegmetod kan betraktas som en Taylor serieutveckling av första ordningen. När man avslutar en Taylor serie vid något n så kommer n vara graden av denna metod. Det lokala felet skulle i sådana fall för Eulers stegmetod vara $O(h^{n+1})$ och det globala felet är $O(h^n)$ under ett antagande om en likformig stegstorlek på h .

2.2.2 MATLAB

Projektgruppen tog först fram en simulering i MATLAB. Den matematiska modellen implementerades och visualiserades i ett antal plottar. Simuleringen visualiserades med Eulers stegmetod och Runge-Kutta metoden. Metoderna jämfördes och det var främst vid stora tidssteg som Eulers stegmetod gav en dålig representation av kastbanan. Här nedanför visas ett exempel på skillnaden mellan de två olika numeriska metoderna, se figur 2.4.



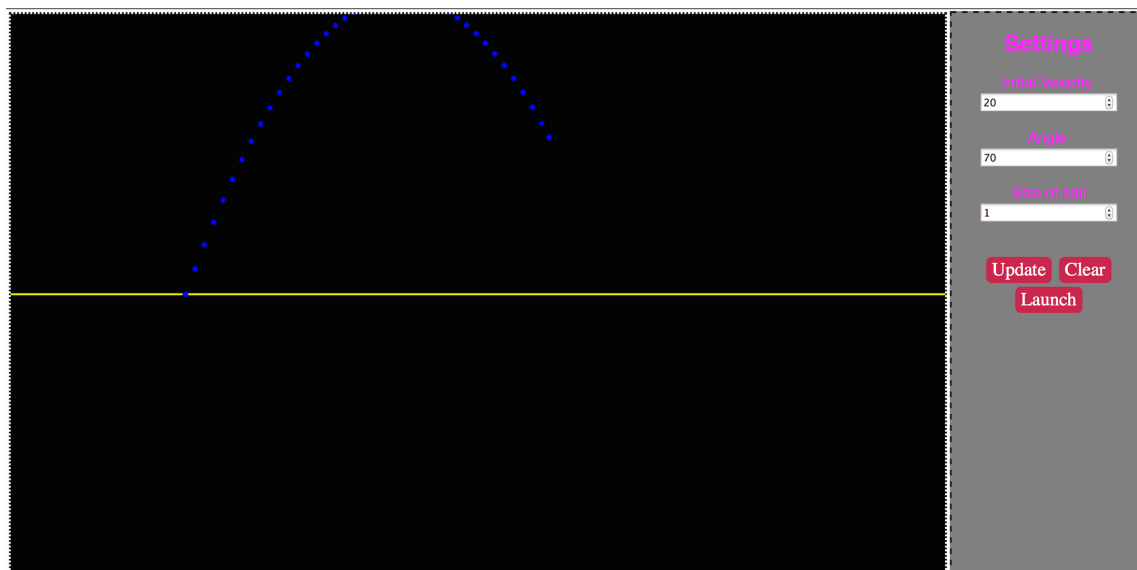
Figur 2.4: Jämförelse mellan Eulers stegmetod och Runge-Kutta metoden i MATLAB. Tidssteg = 0.1, massa = 0.14, radie = 0.03, starthastighet = 50, utgångsvinkel = 45°

2.2.3 WebGL

När de matematiska modellerna var klara i MATLAB gick gruppen vidare med att implementera simuleringen för Eulers stegmetod i WebGL. Steglängden baseras på hur lång tid det tar att rendera två bilder efter varandra. Den matematiska modellen delades upp i ett antal element där varje element implementerades i en egen funktion. Scenmiljö och användargränssnitt designades och programmerades. Gruppen ville åstadkomma en simpel och tydlig design. WebGL kompletteras med biblioteket Three.js². Three.js är ett bibliotek som underlättar bl.a. ljussättning, rendering och kamera.

Det är möjligt att simulera flera kastbanor i samma scen. Kastbanorna ritas ut så att man kan kolla på dem i efterhand i scenen. Användaren kan även välja att ta bort tidigare kastbanor med knappen *Clear*. Flera parametrar kan ändras vid simuleringen, utgångshastighet, utgångsvinkel, objektets radie, objektets material, vindens vinkel och vindens hastighet. En tidig version av implementationen kan ses i figur 2.5.

²<http://threejs.org>



Figur 2.5: Första versionen av implementationen i WebGL

Projektgruppen fokuserade sedan på att få en mer visuellt tilltalande simulering. En 3D simulering togs fram med bättre ljus, texturer och position för scenkameran.

Under simuleringen användes ett par konstanter. Varje material har en densitet och en stötkoefficient. Densiteten användes för att räkna ut massan och stötkoefficienten användes vid kollisioners uträkning. Konstanterna för material, tyngdacceleration och luftens densitet finns samlad i tabell 2.1 och 2.2.

Material	Densitet (kg/m^3)	Stötkoefficient	Radie (cm)
Golfboll	1184	0.858	2,1
Tennisboll	400	0.712	3,5
Biljardboll	1700	0.804	2,7
Träboll	690	0.603	3,0
Stålboll	7820	0.597	3,0
Glasboll	2500	0.658	3,0

Tabell 2.1: Tabell över material och dess tillhörande värden

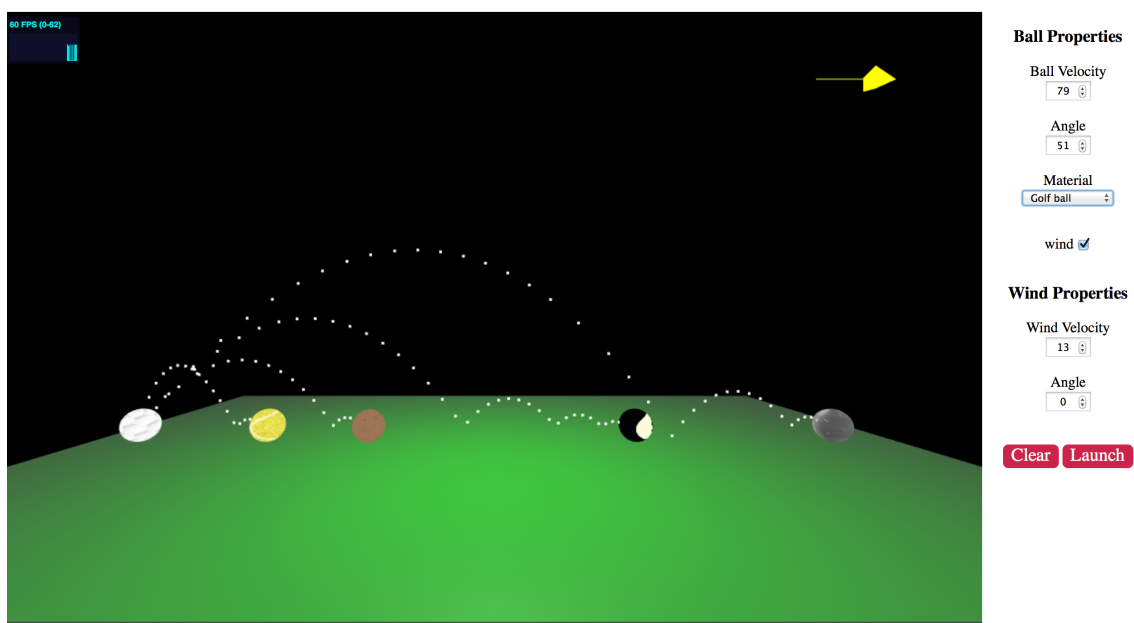
Konstanter	Värde	Enhet
Tyngdacceleration	9.82	(m/s^2)
Luftens densitet	1.2754	(kg/m^3)

Tabell 2.2: Konstanter

Kapitel 3

Resultat

Resultatet blev en webbsida, se figur 3.1 nedan, som innehåller applikationen där man kan testa att simulera en kastbana.



Figur 3.1: Slutversionen av implementationen. Pilen uppe i högra hörnet, visar vindens riktning.

Användaren har möjlighet att ändra följande parametrar som påverkar kastbanan.

- Startvinkel boll: Användaren kan sätta alla vinklar på enhetscirkeln som är ett heltal i grader.
- Starthastighet boll: Hastigheten ska vara positiv och den kan maximalt vara 500 m/s .
- Material: Materialen användaren kan välja mellan finns i tabell 2.2
- Vind: Användaren kan bestämma vindens vinkel och hastighet. Det finns inga begränsningar till hastigheten.

Materialet hos kulan bestäms genom en rullgardins lista. När materialet ändras så uppdateras texturen i simuleringen. Utöver det kan användaren välja att klicka i en kryssruta om vinden ska tas med i visualiseringen eller inte. Om användaren klickar i vind kryssrutan så kan även vindensparametrar ändras. Det vill säga vindens vinkel och hastighet.

Beroende på vilket material som väljs så sätts en fördefinierad radie för bollen. Anledningen till detta är för att golfbollen, tennisbollen och biljardbollen har bestämda mått från olika organisationer. Om de skulle vara större eller mindre skulle det innebära att de inte är den typen av boll längre.

3.1 Diskussion

Projektgruppen valde att använda WebGL. WebGL valdes på grund av att det är väldokumenterat och går fort att komma igång med. Gruppen har även diskuterat OpenGL och Java3D som tänkbara utvecklingsspråk. Men på grund av att OpenGL tar längre tid att få igång och att Java3D är dåligt uppdaterat och ett döende språk valdes de bort. Projektgruppen ville minimera upplärningstiden för utvecklingsmiljön. Three.js visade sig vara ett mycket användbart bibliotek och sparade projektgruppen mycket tid. Tid som sedan har lagts på att utöka den fysiska modellen.

Att simulera och modellera en kastbana var en bra idé. Eftersom MATLAB kunde användas för att snabbt komma igång med modellen samt simuleringen. En kastbana är också bra att modellera då man hela tiden kan lägga till fler externa krafter allt eftersom. På så sätt är det lätt att gå vidare med modellen. Implementation stötte dock på problem i WebGL och visade sig under vissa förhållanden vara felaktig. Kastbanorna blev orealistiska för vinklar mellan 90 och 270 grader. Det blev även ett problem när fler funktioner skulle implementeras, vissa funktioner fungerade dåligt med varandra och fick skrivas om. Projektgruppen fick därefter lägga ner ett par dagar på att fixa buggar i simuleringen. Största anledningen till detta var att implementationen i MATLAB inte kunde översättas direkt till WebGL, projektgruppen fick därefter modellen vi tog fram i MATLAB. Implementationen i WebGL följer dock självklart den matematiska modellen.

Den fysiska modellen består av ett flertal konstanta krafter. Luftmotståndet förväntas vara konstant oavsett hur högt kastbanan åker. Detta är orimligt då luften får en lägre densitet desto längre upp man kommer i atmosfären. Vindens styrka och riktning har satts till ett konstant värde. Vilket även det är orealistiskt för verkliga väderförhållande. Tänkbara förbättringar är att ta fram en luftmotstånds modell som beror på höjden av kastbanan. Dynamisk vind som ändrar riktning och styrka under simuleringen. Den matematiska modellen kan även utökas med en tredjedimension, rotation för Magnuseffekt¹ samt en mer realistisk kollision som beror på impulser istället för att sätta en invers hastighet.

¹Magnuseffekt - Ett fysikaliskt fenomen som uppträder för roterande kroppar

3.2 Slutsats

Det finns fortfarande funktioner att finslipa och ytterligare externa krafter att lägga till modellen. Trots det är projektgruppen nöjd med slutresultatet. Simuleringen av kastbanan fungerar som tänkt och användargränssnittet är lätthanterligt.

Litteraturförteckning

- [1] Halliday D, Resnick R, Walker J. Fundamentals of Physics. John Wiley and Sons, Inc; 2011.
- [2] Young HD, Freedman RA. Sears and Zemansky's university physics. vol. 1. Pearson education; 2006.
- [3] Education P. Projectile Motion with Air Resistance; 2003. <http://wps.aw.com/wps/media/objects/877/898586/topics/topic01.pdf>. Access: Mars 10, 2015. Website.
- [4] Cull T. If a ball thrown at the speed of wind, is there wind resistance?; 2003. <http://madsci.org/posts/archives/Mar2003/1046720933.Ph.r.html>. Access: Mars 10, 2015. Website.