

Fyrverkerisimulering med Blenders partikelsystem

MÅRTEN BORGSTRÖM
och JOHAN CRONSIOE



**KTH Datavetenskap
och kommunikation**

Fyrverkerisimulering med Blenders partikelsystem

MÅRTEN BORGSTRÖM
o c h J O H A N C R O N S I O E

Examensarbete i datalogi om 15 högskolepoäng
vid Programmet för datateknik
Kungliga Tekniska Högskolan år 2011
Handledare på CSC var Lars Kjelldahl
Examinator var Mads Dam

URL: [www.csc.kth.se/utbildning/kandidatexjobb/datateknik/2011/
borgstrom_marten_OCH_cronsioe_johan_K11040.pdf](http://www.csc.kth.se/utbildning/kandidatexjobb/datateknik/2011/borgstrom_marten_OCH_cronsioe_johan_K11040.pdf)

Kungliga tekniska högskolan
Skolan för datavetenskap och kommunikation

KTH CSC
100 44 Stockholm

URL: www.kth.se/csc

Sammanfattning

Denna rapport syftar till att undersöka lämpligheten av 3D-modelleringsverktyget Blender och dess partikelsystem för att simulera fyrverkerier. Dokumentet inleder med en beskrivning av Partikelsystem i allmänt och sedan mer specifikt partikelsystem i Blender. Under avsnittet Simuleringar beskrivs två fyrverkerisimuleringar och partikelsystemen som använts. Till varje fyrverkeri hör även analyser som söker förklara vad som var viktigt ur en teknisk och en estetiskt synvinkel och vilka brister som upptäckts. Sist i rapporten diskuteras förslag på förbättringar som kunnat tillämpas, avsaknad funktionalitet i Blender samt övriga slutsatser om fyrverkerisimulering med Blenders partikelsystem.

Abstract

This report is written in order to evaluate how suitable the 3D modeling software Blender and its particle system is for simulating fireworks. The document begins with background on particle systems in general and continues with particle systems in Blender in particular. The section Simuleringar describes two simulations and the particle systems used in these. The results are then followed by analyses that seek to explain what was important from a technical and an aesthetic point of view and what shortcomings that have been identified. Finally, the report discusses suggestions for improvement that could have been applied, lack of functionality in Blender and other findings concerning firework simulation with Blenders particle systems.

Innehåll

Innehåll	iv
1 Inledning	5
1.1 Bakgrund om partikelsystem	5
1.2 Fyrverkerisimulering med partikelsystem	6
1.3 Vårt verktyg — Blender	6
1.3.1 Inställningar	6
2 Simuleringar	11
2.1 Fyrverkeriraket	11
2.1.1 Tillvägagångssätt	11
2.1.2 Inställningar i Blender	13
2.1.3 Analys	14
2.2 Fyrverkerifontän	15
2.2.1 Partikelsystemen	15
2.2.2 Inställningar i Blender	16
2.2.3 Analys	17
3 Diskussion	19
3.1 Förbättringar	19
3.2 Brister	20
3.3 Slutsatser	21
Litteraturförteckning	23

Fördelning av arbete

Större delen av denna rapport har skrivits tillsammans av de båda författarna. De simuleringar som har gjorts är uppdelade så att Johan har arbetat med och skrivit större delen av kapitlet gällande fyrverkeriraketen medan Mårten på samma vis har jobbat med fyrverkerifontänen. Både analyser, teoretiskt arbete och inhämtande av information har gjorts gemensamt och är även det som har tagit den största delen av vår tid. Då ingen av oss hade någon tidigare erfarenhet inom arbete med 3D-modellering har vi på var sitt håll utforskat området för att sedan lära av varandra. Arbetet är utfört under handledning av Lars Kjelldahl.

Nyckelord

Blender	3D-modelleringsprogram för att skapa visuella effekter.
Källa	Skapar partiklarna i ett partikelsystem, bestämmer partiklarnas mängd och attribut.
Reaktor	Typ av källa som påverkas av ett annat partikelsystem. Kan exempelvis startas som reaktion på att en partikel kolliderar med något eller dör.
Rendering	Att rendera betyder att den 3D-värld man byggt upp översätts till en 2D-bild eller animation som kan visas på en datorskärm.

Kapitel 1

Inledning

1.1 Bakgrund om partikelsystem

Partikelsystem är en relativt gammal teknik för att skapa visuella effekter. Det användes exempelvis i filmen *Star Trek II : The Wrath of Khan*, 1982 för att animera en explosion och eldhavet som följde. Systemet är speciellt lämpat för att skapa just dessa typer av effekter där objekten ej har en definierad gränslinje eller kan sägas ha en kaosartad natur.[1] Numera har användningsområdet utökats till att även skapa vätskor och solida ting.

Konceptet är enkelt, en källa skapar partiklar som var och en har attribut som påverkar deras rörelse och utseende. För att partiklarna ska vara unika slumpas källan vissa av deras värden så att de exempelvis har en högre hastighet eller längre livslängd. När en källa har skickat iväg alla sina partiklar dör den och detta kan aktivera andra källor. Källor bestämmer inte bara attributen för varje partikel utan även antalet partiklar och med vilken intensitet de skapas. Egenskaperna kan enkelt utökas, exempelvis kan källorna programmeras att skapa nya källor, de kan dessutom själva ha en hastighet och påverkas av yttre faktorer.

Det unika med partikelsystem är att man använder det mest primitiva, en partikel, för att skapa objekt och beteenden med hög komplexitet. Som nämnts kan en partikel inneha många attribut och genom att för varje steg i animationen iterera över alla partiklar, plocka ut attributen, beräkna deras nya värden, uppdatera partiklarna och sedan kunna rita upp det nya tillståndet.

Att det finns så många möjligheter för att styra partiklarna och att skapa sofistikerade beteenden leder också till svårigheter. De effekter som används inom ett område passar sällan i andra, därför är det viktigt att veta just vilka effekter man bör inrikta sig på. När man väl vet vilka egenskaper partikelsystemet ska använda är nästa steg att justera inställningarnas värden. Ofta ger välgrundade antaganden ett bra avstamp, men för att frambringa en verklighetstrogen känsla krävs finkalibrering av dessa inställningar. Till detta följer att systemet gång på gång måste renderas och analyseras, det sistnämnda gärna av flera personer då uppfattningen om vad som ser bra ut är subjektivt.

1.2 Fyrverkerisimulering med partikelsystem

Fyrverkerier är en typisk tillämpning av partikelsystem. Komplexiteten kan variera från ett enda partikelsystem till att bestå av en hel hierarki av partikelsystem. Exempelvis skulle ett mer avancerat fyrverkeri kunna delas upp i följande delar: När raketerna först skjuts upp används ett partikelsystem för att visualisera raketens svans. Då den har nått sin högsta punkt slutar det första systemet och låter nästa ta över, själva explosionen, vars partiklar också kan ha en svans på samma sätt som raketerna. I detta fall används tre olika system men man kan tänkas behöva utöka med ännu fler om man exempelvis vill använda olika färger i explosionen.

Partikelsystem är mycket skalbara vilket man till exempel ser när ett färdigt fyrverkeri enkelt kan utökas till flera genom att låta ett nytt partikelsystem skapa ytterligare fyrverkerikällor. På detta sätt kan man snabbt sätta ihop en fyrverkeri-uppvisning utan att behöva modifiera det ursprungliga fyrverkeriets system.

1.3 Vårt verktyg — Blender

I projektet skapas simuleringar av fyrverkerier med hjälp av ett färdigt partikelsystem i programmet Blender. Detta för att inte ägna större delen av projektets tid på själva implementationen av partikelsystemet. Blender är ett program för 3D-modellering och animering där partikelsystem används för att skapa specialeffekter. Beslutet att använda just Blender beror på att det är baserat på öppen källkod och därmed kan användas utan att behöva köpas.[2]

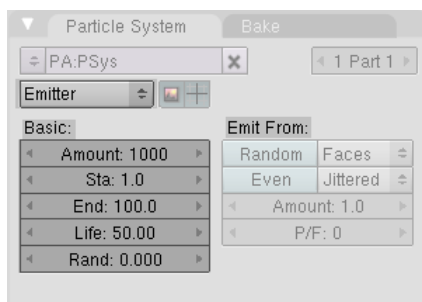
Det finns två typer av källor i Blender, den ena typen är då ett objekt agerar källa, det enda kravet är att den är en "mesh" (polygonyta), det vill säga att den endast består av hörn som binds ihop med enkla polygoner. Den andra typen är när man låter partiklar agera källor, på så sätt kan man skapa sammankopplade partikelsystem som visar sig mycket användbart vid många typer av fyrverkerier.

1.3.1 Inställningar

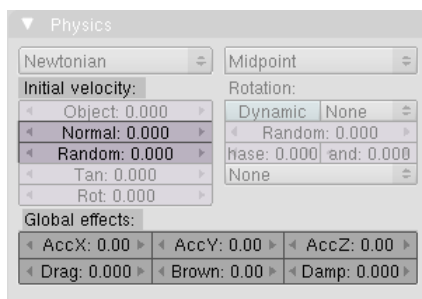
De inställningar som avgör hur ett partikelsystem ser ut i slutändan kan i Blender kategoriseras i två delar, partikelsystem och material. Inställningar för partikelsystem gäller hela systemet, såsom egenskaper hos källan och fysik, medan material är de egenskaper som ger partiklarna deras utseende. Denna separation är dock inte absolut och här ges en överblick av de inställningar för partikelsystem i Blender som används för att uppnå resultatet i rapporten.

Partikelsystem

Den första delen av inställningar kan ses i Figur 1.1 och är grundläggande egenskaper hos källan. Den första kolumnen innehåller inställningar för hur många partiklar som ska avges, mellan vilka tider och hur länge dessa lever. Tiderna anges i bildrutor vilket innebär 1/25 sekund i våra animeringar. Inställning "Rand" kan användas



Figur 1.1: Blenders grundläggande inställningar för partikelsystem av typen emitter(källa).



Figur 1.2: Blenders inställningar för fysik i ett partikelsystem av typen emitter(källa).

för att ge en varierande livstid åt partiklarna, den kan anta värden mellan 0 och 1 och anger hur stor del av livstiden som partiklarna kan dö. Den andra kolumnen avgör om partiklarna ska avges från ytan, hörnen eller volymen hos källan samt hur partiklarna fördelas över dessa.

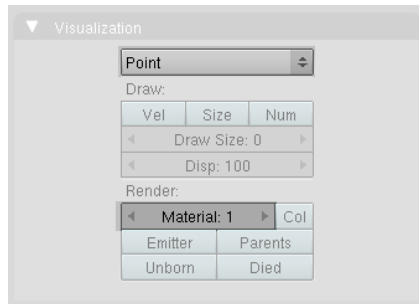
Här kan även typen ändras från källa till reaktor vilket ger inställningar för vilka partiklar som systemet ska reagera på och vad som utlöser reaktionen. En reaktor kan utlösas på olika sätt, de metoder vi använder är “Near” vilket betyder att en partikel agerar som källa och “Death” dvs. partikelsystemet utlöses där partikeln dör.[4]

Nästa sektion av inställningarna hanterar rörelser hos partiklarna i partikelsystemet och kan ses i Figur 1.2. Här kan man välja hur stor hastighet partiklarna ges i olika riktningar. I projektets används tre av dessa, “Normal”, normalen till polygonytan och “Random” där riktningen är slumpmässigt vald. Den tredje varianten syns inte i figuren och är endast för partikelsystem av typen reaktorer och baseras på färdriktningen hos ursprungspartikeln.

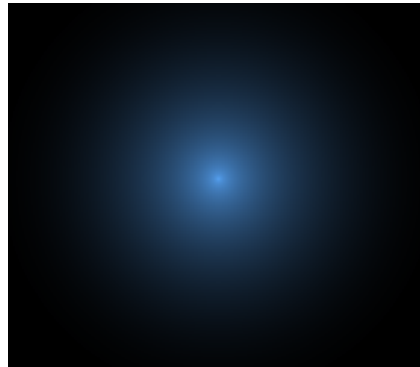
Med “Newtonian” som fysikmotor får man även tillgång till ett antal globala effekter som kan ses i den nedre delen av Figur 1.2. De tre första av dessa (AccX, AccY, AccZ) används för att simulera globala krafter såsom gravitationskraft. “Drag” och “Damp” är båda krafter som bromsar partiklarnas hastighet men betendet är olika, “Drag” är till för att simulera luftmotstånd och är proportionell mot hastigheten hos partikeln medan “Damp” innebär en ren deceleration. “Brown” är en ständigt varierande slumpmässig kraft och används för att simulera slumpmässig rörelse på grund av luftmotstånd.[5]

Sektionen “Visualization” som ses i Figur 1.3 innehåller några av inställningarna för utseende. Den översta inställningen är partikeltypen och avgör därmed hur den

Figur 1.3: Visualization-panelen i Blender som styr hur en partikel representeras i renderingen.



Figur 1.4: En partikel som använder sig av ett material av typen Halo.



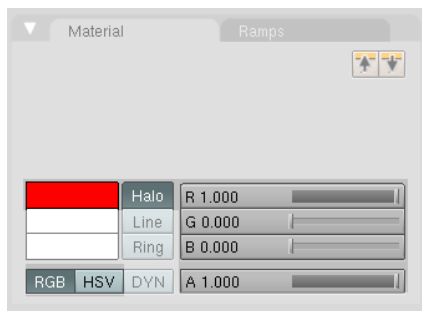
ritas, de varianter som används av simuleringarna är “Point” vilken är vanliga partiklar och “Line” där den ritas som en tunn linje i partikelns färdriktning. Den mest centrala inställningen är dock den för material, där länkas partikelsystem med ett material vilket i sin tur påverkar resten av utseendet. Inställningarna under “Draw” styr utseendet i Blender men har ingen effekt på renderingen.[6]

Material

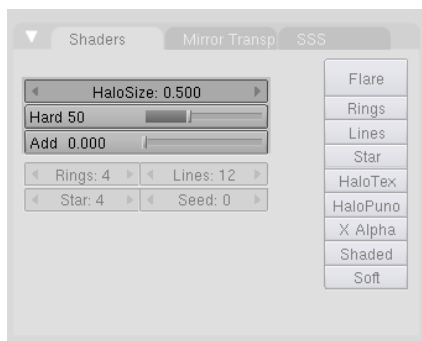
För att styra vilka färger, mönster och synlighet ett objekt har använder man sig av materialinställningarna. När ett material appliceras på partiklar bör de vara av typen “Halo”, det vill säga en gloria som kan ses i Figur 1.4. Här finns även inställningar för “Render Pipeline” som beskriver hur materialet ska behandlas av renderingsalgoritmen men dessa påverkar inte glorior och därmed inte partikelsystemen i den här rapporten.[7]

Om “Halo” väljs ser nästa sektion, “Material”, ut som i Figur 1.5. Här justeras färg och genomskinlighet lätt med hjälp av RGB-reglage men kan alternativt anpassas efter Hue, Saturation och Value.

Shaders-inställningarna som kan ses i Figur 1.6 kontrollerar till störst del partikelns form och hur utseendet blir om flera partiklar befinner sig i samma område. “HaloSize” bestämmer glorians storlek medan “Hard” styr hur mjuk övergången från glorians färg till bakgrunden är. “Add” avgör hur mycket färgerna på gloriorna adderas när de överlappar, det vill säga hur mycket ljusare resultatet blir om flera partiklar överlappar. Resten av inställningarna är extra effekter till glorian men



Figur 1.5: Färginställningar hos ett material som använder sig av glorior.



Figur 1.6: Ytterligare inställningar för utseendet hos glorior.

ingen av dessa används i detta projekt.[9]

Normalt har glorior endast en färg som gradvis smälter in med bakgrunden och för att få fler färger använder man texturer. En textur i Blender är i princip en gråskalig 2D-bild som används för att blanda två stycken lager. I rapporten används detta för att ge färgskiftningar till partiklarna.

Kapitel 2

Simuleringar

Rapporten består av modelleringar av två fyrverkerier som ska skapas med Blender och sedan analyseras. För att lyckas med detta ska vi studera bild- och filmsekvenser på fyrverkerier i syfte att lära oss om deras uppbyggnad och fysikaliska egenskaper.

Det första fyrverkeriet, en raket, kommer att vara ett sådant som har egenskaperna att explosionens partiklar lyser starka och har en röksvans efter sig. Det andra fyrverkeriet står fast på marken, en så kallad fontän, som istället för att explodera, sprutar ut flammor och gnistor i olika färger. Resultaten från de båda fyrverkerierna kommer att presenteras i form av bilder och även länkar till de renderade filmerna. Dessa analyseras sedan utifrån ett tekniskt och ett estetiskt perspektiv för att på så vis finna vad som kan förbättras.

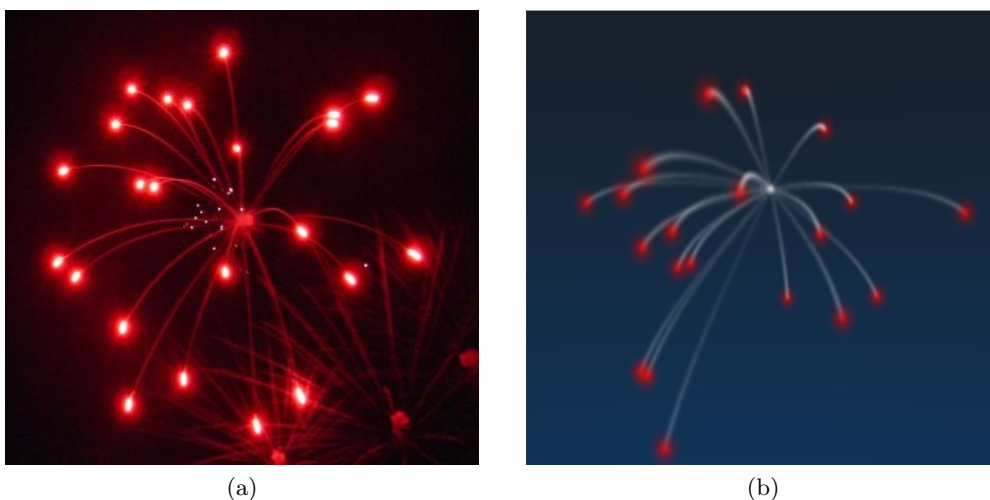
Effekter såsom avancerad ljussättning och rörelseoskärpa ligger inte inom projektets omfattning och vi ämnar inte spendera extra tid med detta för att förbättra resultatet av simuleringen. Rapporten kommer att lägga störst fokus på det som Blenders partikelsystem involverar och vad det kan göra för simuleringarna. På detta sätt söker vi finna vilka attribut inom partikelsystem som lämpar sig för vissa typer av fyrverkerier och hur olika tekniker bäst används.

När man animerar fyrverkerier är en viktig del av upplevelsen att höra de ljud som ett riktigt fyrverkeri ger ifrån sig. Även detta är något som inte tas upp i denna rapport men för vidare utveckling är det ett utmärkt tillägg. Att endast lyssna till ljudet av ett fyrverkeri kan vara till hjälp när man vill få en bättre uppfattning av dess tajming.

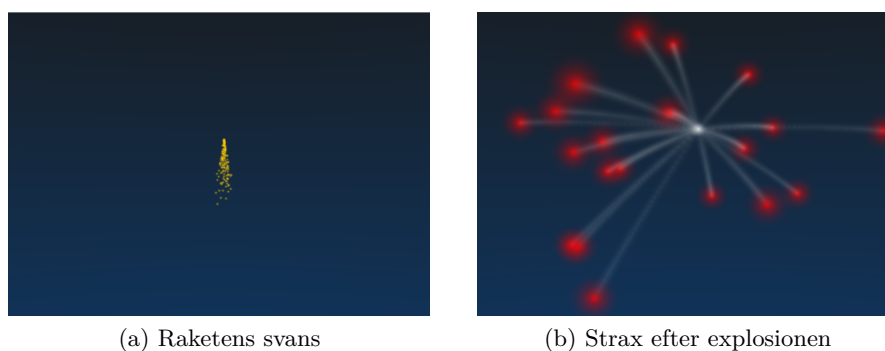
2.1 Fyrverkeriraket

2.1.1 Tillvägagångssätt

Fyrverkeriet består av fyra stycken olika partikelsystem, vardera med en specifik uppgift. Det första systemet är det som skickar iväg raketerna och kan även konfigureras så att det sänder iväg två eller flera raketer. Själva raketerna är inte synliga men verkar som katalysatorer för svansen och explosionen som följer. Den påverkas av gravitationen och dess utfallsvinkel är mellan 80-90 grader vilket ger den en ka-



Figur 2.1: Jämförelse av modellbilden och det simulerade resultatet



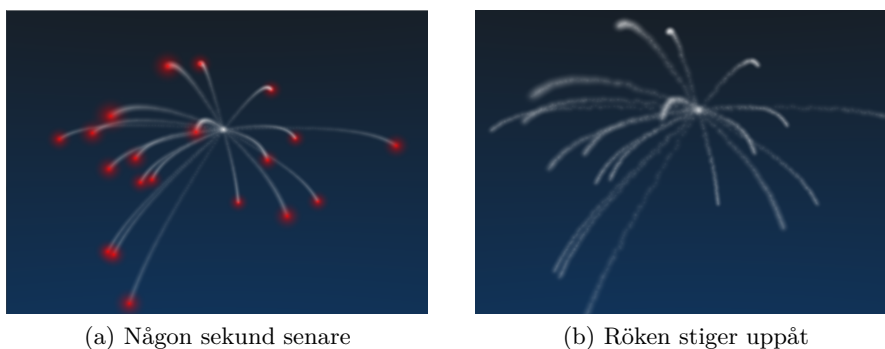
Figur 2.2: De första två stegen i simulationen.

rakteristisk båge. Dess hastighet är satt på så sätt att när raketen brunnit ut har den förlorat nästan all sin hastighet och är då högt upp i luften.

Svansen, i Figur 2.2a, består av en reaktor som utlöses runt om raketerna och är synlig från det att raketerna startar tills att de exploderar. Gnistorna har en hög utgångshastighet motsatt raketens riktning samt en viss hastighet åt sidan så att det bildas en konform. De påverkas svagt av gravitationen men dör snabbt, med något varierade livslängder. Svansfärgen är av en mörkare orange och svansens livslängd är inställd så att gnistorna blir färre med tiden.

Raketerna flyger ca 2-3 sekunder innan de slutligen dör och utlöser explosionen, Figur 2.2b, som lyser starkt vitt. Explosionen sänder ut ett 20-tal partiklar i hög fart, åt olika håll, som bromsas av luftmotståndet.

Samtidigt sänder dessa partiklar ut rök genom ännu ett partikelsystem som bildar linjer efter partiklarna. Explosionens partiklar försvinner när de tappat fart och röken dröjer sig kvar länge medan den långsamt stiger uppåt, se Figur 2.3b.



(a) Någon sekund senare

(b) Röken stiger uppåt

Figur 2.3: Steg 3 och 4 i simulationen.

Viktigt för detta fyrverkeri är att komponenterna har en verklighetstrogen hastighet och att krafterna påverkar på ett naturtroget sätt. Kameran måste vara placerad så att man ser hela fyrverkeriet från start till slut, dock från marken vilket gör att den positioneras en bit bort. För att ta detta ett steg längre kan åskådarens vy animeras så att denne följer fyrverkeriet med blicken, vilket har gjorts i videosekvenserna.¹

2.1.2 Inställningar i Blender

Partikelsystem	Raket	Svans	Explosion	Rök
Amount	1	2000	20	2000
Life	39	4	13	60
Rand	0	0,5	0	0
Normal	14	0	0	0
Random	1	1,7	21	0
AccZ	-9,8	0	-9,8	0,2
Brown	0	0	0	0,4
Damp	0	0	0,18	0
Typ	Emitter	Reactor	Reactor	Reactor
Beror av	-	Raket	Raket	Explosion
Reagerar på	-	Near	Death	Near

¹ Animationen kan hittas på följande youtube-länk: <http://www.youtube.com/watch?v=aOym38tlhFw>

Amount	Svansen och rökens höga värden sattes efter kraven att svansen ska vara intensiv och livlig medan röken bör vara tjock. Explosionens partikelmängd härleddes direkt ifrån antalet partiklar i originalbilden.
Life	Raketens livslängd anpassades efter dess hastighet och inbromsning så att den dör när dess fart kraftigt minskat. Svansens gnistor lever en kort stund och påverkas även av "Rand" för att tunna ut svansen i änden. Explosionens livstid lämpades efter partiklarnas hastighet, dragningskraften och luftmotståndets påverkan för att uppnå önskad effekt. Röken ligger egentligen kvar mycket längre men behöver inte beskådas mer för att uppvisa sitt beteende.
Random	Hjälper raketten att få en svag utfallsvinkel åt ett slumpmässigt håll och tillsammans med "Normal" får raketten en hög utgångshastighet. Svansens låga värde ger den en något konartad form och dess livslängd kompenserar för den ökade hastigheten. För explosionen krävs ett högt värde som gör att partiklarna kastas åt alla möjliga håll så snabbt att de når full längd på ett ögonblick.
AccZ	Använder gravitationskonstanten för att ha en fast punkt att utgå från när hastigheterna sätts. Svansen påverkas marginellt och utslöts för att inte behöva anpassa dess livslängd. Röken kan man justera så att den påverkas av luftdensitet och vindar men för enkelhetens skull fick den istället en svag acceleration uppåt.
Brown	Ger röken en luddig form genom att låta dess partiklar röra sig åt slumpmässiga riktningar.
Damp	Bromsar in explosionens partiklar för att simulera luftmotstånd.

2.1.3 Analys

I detta första försök har mycket av förarbetet bestått i att finna vad som karakteriserar ett fyrverkeri och de viktiga delarna har plockats ut för att analyseras.

Uppskjutet

De viktiga fysikaliska aspekterna som här spelar roll är följande: acceleration, hastighet, luftmotstånd och livslängd. En rakets massa är relativt liten den kraft som den skapar vilket gör att den på nolltid når sin maximala hastighet. För ett riktigt fyrverkeri syns svansen en kort stund innan raketten har lyft vilket beror på trögheten men i vårt fall har denna effekt varit svår att simulera, något som lett till att raketten direkt vid födseln har en hög hastighet. Hastigheten måste uppfylla kraven att på kort tid ta raketten högt upp samtidigt som luftmotståndet inte ska hinna stoppa rakten helt innan den exploderar, detta har balanserats väl i simuleringen.

Vad gäller de visuella effekterna är det endast svansen som syns då raketten i sig inte visas. Att gömma raketten helt var ett avvägande mellan detaljnivå och komplexitet eftersom att få modellen att bete sig naturligt är svårt. Svansens roll är

att visa åt vilket håll raketen rör sig, hur högt upp den är och när man kan förvänta sig explosionen. Tack vare att dess partiklar skjuts ut motsatt raketens riktning, med stadig hastighet, uppfylls de två första kraven. Det tredje är svårare då explosionen sker direkt efter att raketen dör och svansen ger därför inte någon vink om att explosionen är annalkande. Detta är ett problem som löses med IPO-kurvor vilket diskuteras under avsnittet Förbättringar. Sist bör poängteras att raketsvansen i detta exempel ej lämnar efter sig rök liksom explosionens partiklar, men endast i fall med fler än en raket finns det ljuskällor kvar för att lysa upp denna rök och effekten implementerades ej på grund av detta.

Explosionen

Det är hos denna del som det finns mest utrymme för förbättringar. Partiklarna skjuts för det mesta ut på ett trovärdigt sätt men det finns undantag. För att partiklarna ska få sin karakteristiska båge påverkas de av en kraft nedåt för att simulera gravitationspåverkan. Detta fungerar väl för partiklar som skjuts ut åt alla håll förutom nedåt då nedåtpartiklarna istället rör sig orealistiskt långt, detta eftersom att "Damp" användes istället för "Drag", vilket enkelt kan åtgärdas.

Explosionspartiklarna saknar den vita kärnan som skulle kunna återskapas på flera sätt, exempelvis genom att göra en kopia av partikelsystemet med samma "Seed" (slumpfrö) men låta färgen vara vit eller manipulera materialet så att partikelns mitt är vit.

När partiklarna dör upphör de att synas omedelbart vilket man, liksom raketens svans, hellre ser att de tonar bort, också något som IPO-kurvor kan hjälpa till med.

Sist är det röken som, även då den rör sig realistiskt, fortfarande saknar en viktig fysikalisk egenskap, att reflektera ljus. Som synes i originalet färgas röken röd från partiklarnas ljus men i vårt exempel är den vit. Att lösa detta problem är svårare då man måste tilldela en ljuskälla till varje partikel från explosionen och låta röken reflektera tillbaka rött ljus. På detta vis växlar rökens färg till grått när ljuskällorna dör men problemet går återigen också att komma till rätta med genom att använda IPO-kurvor.

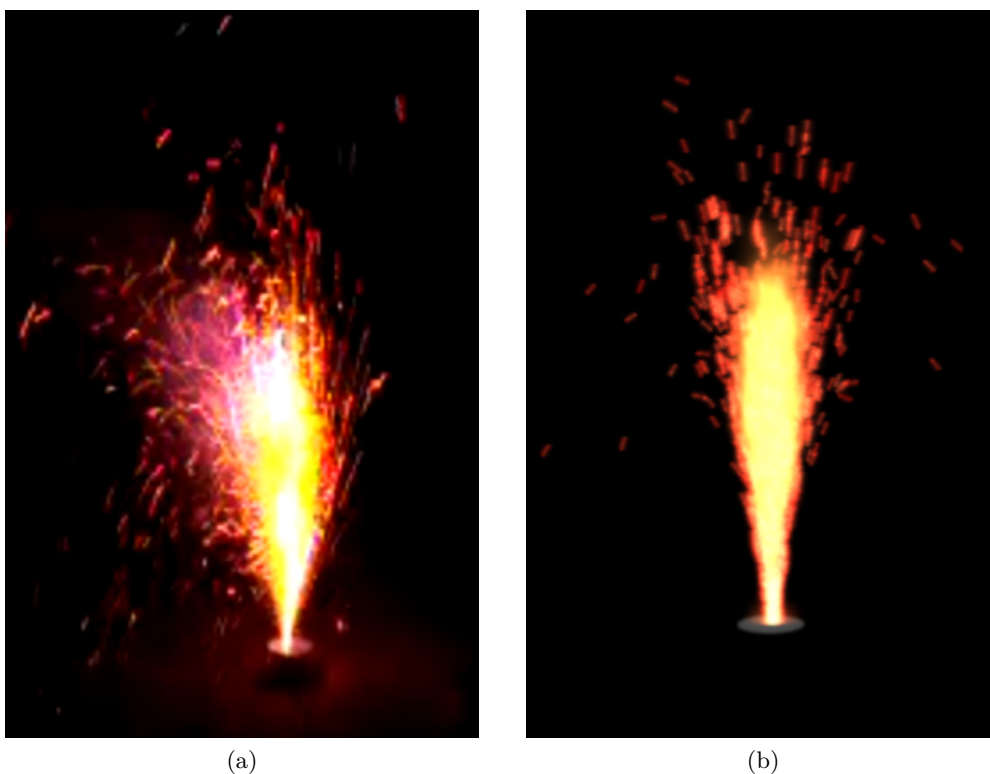
2.2 Fyrverkerifontän

Den andra fyrverkerisimuleringen är av en så kallad fontän, ett fyrverkeri som står på marken och skjuter upp färgade gnistor och flammor.[10] I Figur 2.4a syns den fontän som användes som utgångspunkt och i Figur 2.4b visas simuleringen av denna.²

2.2.1 Partikelsystemen

I detta fyrverkeri kan två stycken typer av partiklar identifieras, det första är de som utgör en intensiv ljusgul flamma i centrum och skiftar till orange i kanterna.

²Animationen kan hittas på följande youtube-länk: <http://www.youtube.com/watch?v=9aTYquXvebk>

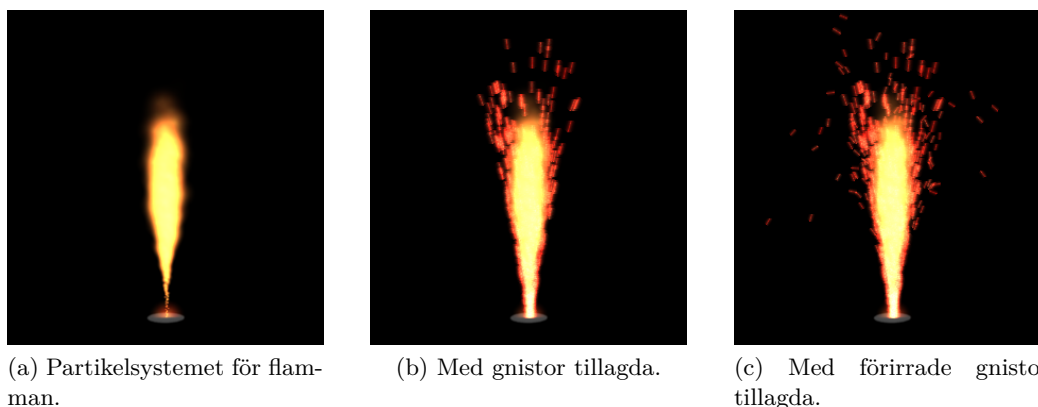


Figur 2.4: Bilderna visar två fyrverkerifontäner, den vänstra är en bild som har använts som modell och den högra visar det simulerade resultatet.

Den andra typen är de röda gnistorna som skjuts ut från fontänen. Partiklarna skjuts alla uppåt från en och samma källa. De röda partiklarna har dock generellt två olika beteenden, de allra flesta åker med en hög fart uppåt och brinner ut fort men det finns även några av dessa partiklar som åker runt i omgivningen, så kallade förrirade partiklar. För att få bättre kontroll över de olika varianterna delades de upp i två olika partikelsystem med olika fysik men med nästan identiska partiklar. I Figur 2.5 visas partikelsystemen stegvis tillagda i samma ögonblick av animationen.

2.2.2 Inställningar i Blender

Inställningarna i denna sektion beskrivs i detalj i avsnitt 1.3.1 Inställningar. Denna animationen är 10 sekunder lång och består av 250 bildrutor.



Figur 2.5: Bilden visar simuleringen av fontänen med de olika partikelsystemen stegvis tillagda.

Partikelsystem	Flamma	Gnistor	Förrirade gnistor
Amount	5000	20000	5000
Life	13	18	30
Rand (Life)	0.1	1	1
Normal	5	6	6
Random	0	1	1
AccZ	4.96	-5.00	-8.15
Brown	12	0	15
Drag	0.2	0	0
Visualization	Point	Line	Line
LineLength	-	0.15	0.10

Material	Ljusgul flamma	Röda gnistor
Alpha	0.8 - 0.0 (Animated)	0.8
HaloSize	0.018 - 0.227 (Animated)	0.12
Hard	45	45
Add	0.646	0.857

2.2.3 Analys

Flamma

I mitten av fontänen ses en flamma vilket förutom en snabbare rörelse är detsamma som att simulera eld, en typisk tillämpning av partikelsystem.[1] Fysikaliskt består eld av heta gaser som accelererar uppåt samtidigt som den expanderar då de har lägre densitet än de omgivande gaserna, elden är ljus i mitten och lite mörkare i kanterna [11], dessa egenskaper syns tydligt i modellfyrverkeriet.

Normal och AccZ	Flamman är starkt riktad och förflyttar sig snabbt, den ges en hastighet och acceleration riktad uppåt.
Brown	För att flamman ska ha det karaktäristiskt fladdrandet ges den en liten slumpmässig rörelse.
Drag	För att sätta en gräns för hur mycket elden kan accelerera läggs lite luftmotstånd till.
Alpha	Flamman är initialt lite genomskinlig men för att den inte bara ska försvinna i intet animeras partiklarnas alpha-kanal med hjälp av så kallade IPO-kurvor så att de blir gradvis mer genomskinlig när deras livstid börjar nå sitt slut.
HaloSize	För att simulera att gaserna expanderar och ge flamman rätt form så animerades partiklarnas storlek. I början är partiklarna tätt packade och expanderar men efter ett tag når de en balans mellan viljan att expandera och avkylningen från luften.

Gnistor

De röda gnistorna i fontänen är brinnande färgpulver som skjuts upp med en hög hastighet.[10] Ett brinnande pulver borde gå perfekt att simulera med partikelsystem men på grund av den höga hastigheten så kan man i modellbilden se hur partiklarna upplevs avlånga, ett fenomen som kallas rörelseoskärpa. Detta uppstår eftersom när en bild tas med kamera så exponeras sensorn under en viss tid och föremål som rör sig blir otydliga, bilder och filmer med denna effekt upplevs som verklighetstrogn då detta liknar hur våra ögon uppfattar saker. Renderade bilder är perfekta avbildningar av en tidpunkt och rörelseoskärpa måste därmed läggas till i efterhand om bilderna skall upplevas som verklighetstrogn.[12]

I vår simulering valde vi att inte använda rörelseoskärpa men för att denna simulering ska vara någorlunda lik modellen behövdes det att partiklarna inte endast ser ut som prickar, vi valde därför att använda partiklar av typen "Line" och på så sätt få ett utseende som mer liknar modellbilden.

Life	De förrirade partiklarna har en något längre livstid då de ska ha tid att separeras från de andra partiklarna.
Normal och Random	Partiklarna ges en hastighet uppåt samt en slumpmässig hastighet och fördelar sig då i en kon uppåt. Båda typerna av gnistor börjar med samma hastighet och beter sig liknande till en början.
Brown	För att de förrirade partiklarna ska vara just förrirade ges de en slumpmässig rörelse.
LineLength	De partiklar som åker rakt upp har en högre hastighet och ges därför en något längre längd än de förrirade partiklarna.

Kapitel 3

Diskussion

3.1 Förbättringar

Vi valde att i simuleringarna endast använda oss av så kallade glorianer för att visualisera partiklarnas utseende vilket visade sig ha en del begränsningar då dessa är gjorda för att vara enkla och snabba att rendera. Detta innebär inte att det inte går att göra men det kräver att man använder andra typer av partiklar än glorianer. I Blenders synvinkel är glorianer i princip endast 2D-bilder som skalas beroende på avstånd och kan endast påverka varandra, till exempel färgförändring med hjälp av "Add" som tas upp i 1.3.1 Settings. Glorianer kan alltså varken påverkas av eller påverka ljussättningen i 3D-världen.[14]

Partiklar som ljuskällor

En lösning på problemet att glorianer inte kan avge ljus skulle vara att använda objekt som partiklar istället, detta möjliggör användning av lampor som partiklar och på så sätt kan de ge ifrån sig ljus. Lampor syns dock inte och man skulle då få göra partikelsystemen i par med samma inställningar och samma slumpfrö så att varje partikel egentligen består av ett par av partiklar, en glorian och en lampa. Det krävs dock enorm datorkraft för att simulera allt ljus som avges från partiklarna men det skulle kunna användas för att få en realistisk belysning av marken i fyrverkerifontänen. Det skulle även kunna användas tillsammans med billboards som beskrivs nedan för att låta ljuset färga röken på ett realistiskt sätt då röken endast syns så länge det finns partiklar i närheten som lyser upp den.

Billboards

Billboards är en typ av partiklar som kan göra allting som glorianer kan och mycket därtill, de löser ett av problemen med glorianer då de dessutom kan påverkas av ljus. Detta skulle kunna användas i fyrverkeriraketen för att göra så att röken ändrade färg på grund av ljuset från de andra partiklarna.[14] Billboards är dock ansedda

som ett av de mer svårbegripliga områdena i Blender och var därför inte aktuella i rapporten.[15]

Animationer med IPO(InterPOLation system)-kurvor

I Blender kan i princip alla egenskaper av objekt ändras med tiden. Detta görs med hjälp av så kallade IPO-kurvor som styr egenskaperna med hjälp av interpolerade kurvor.[13] I fyrverkeriraketen har IPO-kurvor använts för att animera kameran så att den följer en raket genom att gradvis justera sin vinkel för varje bildruta och i fyrverkerifontänen används det för att partiklarna ska försvinna på ett snyggt sätt genom att animera genomskinlighet för partiklarna.

Tekniken har endast använts i dessa fall då den har tätt sig relativt avancerad och svår att använda. Dock har den gång på gång visat sig nödvändig, speciellt i de fall då partiklars färger och synlighet behövt justeras under vissa perioder och är alltså oundgänglig om man ska skapa ett snyggt fyrverkeri.

Bakgrund och Miljö

Intrycket av ett fyrverkeri beror också på i vilken miljö det befinner sig i och detta gäller även simulerade fyrverkerier. I en av fyrverkerirenderingarna används en stjärnhimmel som bakgrund och detta ger två fördelar. Ett, man ser att fyrverkeriet rör sig relativt de fasta stjärnorna vilket, i motsats till homogena bakgrunder, gör att det inte ser ut att stå stilla. Två, det ger en mer realistisk och vackrare upplevelse som kan förbättras ännu mer genom att låta fyrverkeriet explodera över exempelvis vatten för fler visuella effekter.

Fysik

I exemplet fyrverkerifontän ses fontänen skjuta spikrakt uppåt till skillnad mot originalet som är något lutande på grund av vinden. Blender kan simulera många olika krafter varav vind är en sådan och skulle alltså kunna användas i syfte att efterlikna originalet.

3.2 Brister

Det finns relativt få brister i Blender, de allra flesta problem som stöttes på hade kunnat lösas med de mer avancerade verktygen som finns i Blender och diskuterades i Förbättringar ovan. En del begränsningar som inte hade någon lösning hittades dock under projektet. Till en början användes den senaste versionen av Blender, 2.56a, men ersattes snart av den äldre versionen 2.49b. Detta på grund av att 2.56a fortfarande är under uppbyggnad med ett nytt användargränssnitt och nya funktioner men saknar i skrivande stund stöd för reaktorer vilket var nödvändigt för att kunna simulera mer avancerade fyrverkerier. Bland de nya funktionerna i 2.56a finns dock stöd för röksimulation vilket skulle ha förbättrat fyrverkeriernas rökeffekt avsevärt.

3.3 Slutsatser

Att skapa fyrverkerier med partikelsystem i Blender är enkelt, att göra dem snygga är svårare. Resultatet beror till stor del av hur mycket tid man lägger ned och självklart vilken erfarenhet man har sedan tidigare. Även då rapporten använder många av de tekniker som Blender erbjuder är det fortfarande mycket som är utforskat. När fyrverkerierna modellerades användes främst de simplare inställningarna som var nära relaterade till partikelsystem medan de mer avancerade funktionerna endast utvärderades teoretiskt i diskussionen. De förbättringar som har identifierats har dock stor potential och vi tror att om de implementeras på ett bra sätt kan resultatet bli mycket bättre. Målet med denna rapport var bland annat att finna eventuella brister i Blenders partikelsystem, något som visade sig vara en utmaning. Bristerna ligger inte i avsaknaden av funktioner utan snarare i svårigheten att hitta och använda dem. Blender är ett mycket kraftfullt program som man relativt snabbt kommer igång med men är svårt att bemästra.

Litteraturförteckning

- [1] *W. Reeves, Particle systems – a technique for modelling a class of fuzzy objects, Transaction on Graphics.* April 1983 <http://portal.acm.org/citation.cfm?id=357320>, Hämtad 13 april 2011.
- [2] *Blender*, [http://en.wikipedia.org/wiki/Blender_\(software\)](http://en.wikipedia.org/wiki/Blender_(software)), Redigerad 12:27 den 6 juni 2011, Hämtad 13 april 2011.
- [3] *Types of Particle systems*, http://wiki.blender.org/index.php/Doc:Manual/Physics/Particles#Types_of_Particle_systems, Redigerad 10:50 den 20 juli 2010, Hämtad 13 april 2011.
- [4] *Creating a Particle System*, http://wiki.blender.org/index.php/Doc:Manual/Physics/Particles#Creating_a_Particle_System, Redigerad 10:50 den 20 juli 2010, Hämtad 13 april 2011.
- [5] *Newtonian Physics*, <http://www.blender.org/development/release-logs/blender-246/particles/newtonian-physics/>, Skapad maj 2008, Hämtad 13 april 2011.
- [6] *Visualization*, <http://www.blender.org/development/release-logs/blender-246/particles/visualization/>, Skapad maj 2008, Hämtad 13 april 2011.
- [7] *Links and Pipeline Options*, http://wiki.blender.org/index.php/Doc:Manual/Materials/Options#Links_and_Pipeline_Options, Redigerad 17:31 den 25 september 2009, Hämtad 13 april 2011.
- [8] *Material Options*, http://wiki.blender.org/index.php/Doc:Manual/Materials/Options#Material_Options, Redigerad 17:31 den 25 september 2009, Hämtad 13 april 2011.
- [9] *Halo Materials*, http://wiki.blender.org/index.php/Doc:Manual/Materials/Halos#Halo_Materials, Redigerad 13:26 den 4 september 2010, Hämtad 13 april 2011.
- [10] *Fyrverkeridefinitioner*, <http://www.sveafyrverkerier.com/se/information/definition.asp>, Hämtad 13 april 2011.

- [11] *Noob to Pro/Making Fire*, http://en.wikibooks.org/wiki/Blender_3D:_Noob_to_Pro/Making_Fire, Redigerad 13:18 den 29 mars 2010, Hämtad 13 april 2011.
- [12] *Motion Blur*, http://en.wikipedia.org/wiki/Motion_blur, Redigerad 05:42 den 7 april 2011, Hämtad 13 april 2011.
- [13] *IPO Curves*, <http://wiki.blender.org/index.php/Doc:Manual/Animation/Editors/Ipo/Curves>, Redigerad 12:06 den 19 januari 2011, Hämtad 13 april 2011.
- [14] *Visualisation*, <http://wiki.blender.org/index.php/Doc:Manual/Physics/Particles/Visualisation>, Redigerad 18:46 den 27 september 2009, Hämtad 13 april 2011.
- [15] *Billboards*, http://en.wikibooks.org/wiki/Blender_3D:_Noob_to_Pro/Billboard_Animation, Redigerad 17:37 den 25 februari 2011, Hämtad 14 april, 2011.

