

## **RIGIFY INOM PRODUKTION**

Ett gångbart produktionsverktyg för mindre aktörer och företag?

## **RIGIFY IN PRODUCTION**

A valid tool for production for smaller actors and companies?

Examensarbete inom huvudområdet Medier, estetik och berättande

Grundnivå 30 högskolepoäng

Vårtermin 2022

Johan Axdorph

Henrik Olsson

Handledare: Torbjörn Svensson

Examinator: Lars Vipsjö

# Sammanfattning

Detta arbete har haft som syfte att undersöka huruvida Blenders (1994) tilläggsverktyg Rigify kan användas för att frigöra arbetskraft inom spelindustrin, då det är ett verktyg för riggning av 3D-karaktärer. Forskare har tidigare gjort försök till att skapa algoritmer som kan hjälpa till inom 3D-animation genom att ha skapat självgenererade riggar (Baran & Popovic 2007; Pan et al 2009; Bharaj et al 2012), samt riggar som också skinns, eller knyts, direkt till karaktären (Le & Deng 2014). Undersökningen i detta arbete bestod utav en 3D-karaktär som riggas med hjälp av Rigify, som därefter utvärderades av anställda inom spelindustrin. Deltagare fick sedan dela sina åsikter om karaktärens rigg under en intervju. Resultatet visade på att Rigify kan skapa avancerade riggar, dock med mindre brister i genereringen. Riggen upplevdes som överflödigt med de många funktioner som lades till. Rigify framstår som ett bra verktyg för mindre nystartade spelstudior, fritidsutvecklare, samt yrkesmässiga egenutvecklare.

**Nyckelord:** Blender, Animation, Spelutveckling, Riggning, Rigify

# Innehållsförteckning

<b>1</b>	<b>Introduktion</b>	<b>1</b>
<b>2</b>	<b>Bakgrund</b>	<b>2</b>
2.1	Relaterad forskning	5
2.2	Manuell riggningsteknik	7
2.2.1	Rignings-schema	7
2.2.2	Planera funktioner	7
2.2.3	Konstruera skelettet	7
2.2.4	Implementera funktioner	8
2.2.5	Kontrollera riggen / Orientering	8
2.2.6	Skinning	8
2.2.7	Avslutande testning	8
2.2.8	Animera	8
2.3	Rigging med Rigify	10
2.3.1	Planering	10
2.3.2	Eventuell Meta-rigg & Eventuella Benbyten	10
2.3.3	Placering av Ben/joints	11
2.3.4	Generering av Kontroller & Constraints	11
2.3.5	Eventuell Omgenerering	11
2.3.6	Skinning	11
2.3.7	Avslutande testning	11
2.3.8	Animera	11
<b>3</b>	<b>Problemformulering</b>	<b>13</b>
3.1	Metodbeskrivning	14
<b>4</b>	<b>Projektbeskrivning</b>	<b>16</b>
4.1	Artefaktens riggningsprocess	16
4.2	Pilotundersökningen	19
5.1	Intervjuer	20
5.1.1	Intervju 1	20
5.1.2	Intervju 2	20
5.1.3	Intervju 3	21
5.1.4	Intervju 4	21
5.2	Analys	22
5.3	Slutsatser	23
<b>6</b>	<b>Avslutande diskussion</b>	<b>24</b>
6.1	Sammanfattning	24
6.2	Diskussion	24
6.3	Framtida arbete	26

# 1 Introduktion

Riggning är en del av den grafiska processen där ett skelett skapas åt en 3D-modell av, allra oftast, en animatör. Skelettet skinnas, eller binds, därefter till modellen och gör det möjligt att kunna skapa animationer. Problemet med riggningen är att den ofta är tidskrävande, och kan kräva mycket resurser som blir kostsamma för företag.

Därmed torde det vara av intresse för företag och studior som använder sig av 3D-animation att finna arbetssätt som kan förenkla och förkorta riggningsprocessen. Detta särskilt för mindre företag, frilansarbetare och eget företagande då de ofta saknar de resurser som större företag har i fråga om både kapital och arbetskraft.

Under vår undersökning kommer vi att testa om huruvida Rigify i Blender (1994) kan spara tid och resurser för spelföretag. Rigify är ett inbyggt tilläggswerktyg, även kallat add-on på engelska, som redan finns med i den grafiska mjukvaran Blender (1994) när den laddas ned.

Undersökningens deltagare kommer vara yrkesutövande animatörer som är bekanta med att animera i Blender (1994). Rigger kommer att skapas med hjälp av Rigify, som deltagarna sedan kommer att få utvärdera. Efter att deltagarna har fått tid att undersöka rigger kommer det genomföras enskilda intervjuer. Där kommer undersökningsdeltagarna kunna dela med sig av sina utvärderingar och åsikter av både Rigify som verktyg och rigger i sig.

Större företag inom spelindustrin har ofta egna verktyg som är särskilt anpassade för deras arbetsprocess, och har därmed ingen användning av Rigify. Rigger får kritik av undersökningsdeltagarna för att vara alltför invecklad än nödvändigt, vilket gör att många delar och funktioner hos rigger blir överflödiga. Karaktären kan utföra grundliga rörelser och poser, men alltjämt finns det funktioner som inte kommer till användning. Detta påverkar även hur mycket plats rigger tar i en spelmotor, vilket stora spelföretag ofta måste undvika för att hålla produktionsdelarna optimerade.

Slutsatser angående Rigify är att verktyget framförallt gynnar mindre nystartade spelstudior med begränsade resurser, frilansare, fritidsutvecklare samt yrkesmässiga egenutvecklare.

## 2 Bakgrund

När det kommer till produktionsarbetet av spel använder sig spelindustrin i dagsläget av termen *pipeline* för att beskriva arbetsupplägget för vad som behövs produceras, samt i vilken ordning. Under utvecklingen av ett spel finns det många olika områden som behöver planeras som programmering, grafik eller design. Detta för att utvecklingen ska vara som mest praktiskt effektiv, samt finansiellt hållbar. För företaget handlar det oftast om att kunna skapa spel med god kvalitet, samt att hålla den satta budgeten.

Att spel ofta är kostsamt att skapa och producera är ingen överraskning för de som är minsta insatta i spelbranschen. Med alla olika avdelningar som arbetar på spelet en längre tid resulterar det hela i en kostsam produktion. Att kunna förkorta utvecklingstiden, och därmed spara företaget pengar, torde vara av intresse för de allra flesta spelstudior och bolag förutsatt att produktkvaliteten förblir åtminstone densamma som innan.

Ur själva spelutvecklingens synvinkel har det senaste årtiondet har på många sätt kännetecknats av att många avancerade och kraftfulla mjukvaror och verktyg blivit tillgängliga för allmänheten, såsom exempelvis branschledande spelmotorer och grafiska mjukvaror. Detta genom att antingen bli kostnadsfria, anta ett slags prenumerations-modell eller ta ut en viss procentsats av eventuella framtida inkomster. Många av de mjukvaror som tidigare endast varit tillgängliga för stora professionella och etablerade studior med mycket resurser, finns nu att ladda ned och kan användas på vanliga persondatorer i hemmet. Därtill har flertalet andra kompetitiva mjukvaror utvecklats vid sidan av de redan etablerade, som allra oftast kan användas av vem som helst antingen till en låg kostnad eller helt kostnadsfritt.

En sådan kostnadsfri mjukvara är Blender (1994), som har under senare år sakta men säkert blivit arbetsmjukvaran för ett stigande antal spelutvecklare och grafiker. Blenders källkod har varit öppen för allmänheten sedan 2002, samtidigt som The Blender Foundation grundades, men det var först när version 2.8 av mjukvaran släpptes i juli 2019 som många spelutvecklare, bolag och frilansgrafiker på allvar började titta på Blender som ett möjligt gångbart alternativ för grafisk produktion. Dessförinnan ansågs Blender hos många vara för primitivt, för bristfälligt och sakna för många av de funktioner och verktyg som andra grafiska mjukvaror redan hade. Därför fanns det, utöver det kostnadsfria skälet, ingen godtagbar anledning att gå över till Blender för redan etablerade studior. På grund av Blenders öppna källkod har det utvecklats egna tilläggsverktyg av både Blenderutvecklarna och privata användare och entusiaster. Tilläggsverktygen har allra oftast som syfte att underlätta, eller effektivisera, en särskild del av en grafisk arbetsuppgift eller arbetsområde vid produktion i Blender. Efter att version 2.8 släpptes började allt fler verktyg att utvecklas och släppas med tiden, varav många därefter uppdateras kontinuerligt i takt med att nya Blender-versioner släpps.

Dessa tilläggsverktyg, eller add-ons, som skapas användarbasen ger Blender möjligheten att utvecklas som mjukvara i många olika riktningar genom användarnas egna engagemang och idéer. Tilläggsverktygen gör även att varje enskild användare kan till stor del skraddarsy sin egna Blender-mjukvaruversion till sina egna krav och behov, vilket således gör Blender än mer attraktivt att arbeta med som mjukvara i helhet.

Det är här vi kommer till tilläggsverktyget för riggning kallat Rigify, vars yrkesmässiga tillämpningsmöjligheter inom riggningsprocessen detta examensarbete har ämnat att

utforska. Rigify skapades för att förenkla och förkorta riggningsprocessen av 3D-karaktärer, så att mer arbetsresurser och tid skulle kunna ägnas åt annat än riggning, samt även göra riggning tillgängligt för de som är oerfarna inom området. Detta ämnar Rigify att göra genom en automatisk generering av en färdigställd rigg, som skapas utifrån en så kallad meta-rigg, med alla dess delar och funktioner, samtidigt som det är helt kostnadsfritt att använda.

Riggning är ett skede under det grafiska arbetet där man skapar ett skelett för en 3D-modell för att kunna få den att röra sig och animeras. Rigger, eller skelettet, binds till modellen för att sedan skapa animationer genom det som kallas skinning.

Rigger består oftast utav joints, eller bones beroende på vilken mjukvara som används, constraints, kontroller och IK- och FK-funktioner. Olika riggningsätt och arbetsflöden kan skilja sig mellan mjukvaror, men i allmänhet så liknar riggningsprocesserna varandra. Den riggningsprocess som kommer att användas i denna studie är en som kan användas i den grafiska mjukvaran Blender (1994). Själva rigger består utav bones, som fungerar som skelettben. De placeras på olika delar på modellen där den planeras att kunna röra sig. Dessa länkas därefter ihop på samma vis som ett människoskelett, förutom att i 3D-mjukvaran blir det tänkta skelettet klassificerat i en hierarki så att benen vet vilka som ska följa vilka, samt vilka som är överordnade vilka.

Constraints är en funktion som gör så att ett bens, eller objekts, egenskaper och värden överförs till ett annat ben eller objekt. Ett exempel på dessa värden är att överföra ett bens rotations-eller positionsvärden. Kontroller används för att styra, eller överföra värden, till ben eller objekt man lagt en constraint på. Kontroller har oftast olika geometriska former för att enklare kunna användas och styras vid animeringsarbete. "IK" står för *inverse kinematics* och "FK" för *forward kinematics*. IK-funktionen gör så att benet längst ner i skelettets hierarki styr de övre benen, så att om exempelvis karaktärens högra fot lyfts upp så följer resten av benet med den rörelsen. FK-funktionen verkar åt motsatt håll från IK-funktionen och gör att benet överst i hierarkin kontrollerar de andra understående benen i samma benlänk. Ett exempel på detta är en människoarm (Doody, 2016), som kan styras från axeln överst hela vägen ut till handen. Funktionerna kan användas till alla kroppsdelar eller objekt, men används oftast beroende på vilken funktion objektet eller kroppsdel är tänkt att ha och hur den har planerats att animeras.

Detta arbete kommer att fokusera på den grafiska delen av spelproduktion, och närmare på vad en animatör idag har för verktyg till hands för att korta ner arbetsprocessen, och därmed utvecklingstiden. Den grafiska arbetsprocessen, eller pipeline, av en spelproduktion har oftast många mindre arbetsfält till inriktningen av spelutveckling processen, och är ofta uppdelad inom olika inriktningar. Dessa arbetsfält och inriktningar kan skilja i namn och arbetssätt från studio till studio i en spelproduktions genomgående skapandeprocess. I allmänhet går det dock i de flesta fall att dela in dem enligt följande: från början ritas 2D-artisterna koncept för karaktärer eller miljön, 3D-artisterna modellerar därefter konceptet i en 3D-mjukvara, som då sedan kan riggas och animeras vid behov av en animatör (Hristov & Kinaneva, 2021).

I denna genomgående arbetsprocess är det enkelt hänt att flaskhalsar uppstår när en inriktning inte har hunnit färdigt med sin arbetsuppgift. Följden kan då bli att övriga inriktningar och arbetsfält måste vänta tills det arbetet har blivit färdigställt och godkänt för

implementering i spelet. Genom att en animatör då automatiskt kan generera en tillfredsställande rigg, som sedan kan animeras utan att lägga tiden att skapa hela riggen från grunden, kan mycket arbetstid och resurser således möjligen kunna sparas av både enskilda utvecklare i den nämnda arbetsprocessen och studior i helhet.

En av slutsatserna från undersökningen blev att Rigify är som bäst lämpat, och användbart, för i allmänhet främst nystartade studior med jämförelsevis få resurser vad gäller både kapital, arbetskraft och effektiva arbetstimmar. Utifrån detta är det därför rimligt att även dra slutsatsen att detsamma gäller för frilansare och enskilda spelutvecklare som arbetar utanför etablerade studior, samt fritidsutvecklare och hobbyister.

För sådana resursmässigt begränsade och småskaliga spelutvecklare och studior kan Rigify anses vara ett mycket användbart verktyg att tillämpa inom sin grafiska produktion. Flera skäl finns hos Rigify för att kunna argumentera för detta, och antagligen förblir det främsta skälet fortsatt det faktum att både Rigify och Blender är, trots deras jämförelsevis höga kvalitet, helt kostnadsfria att använda för alla utvecklare, bolag och studior oavsett inkomst, produktion och ekonomisk omsättning. Detta skäl innebär i sig ett möjligt, och i många fall avsevärt, kostnadssparande för de som väljer att använda Blender, där Rigify ingår kostnadsfritt som tilläggsverktyg från att mjukvaran öppnas.

## 2.1 Relaterad forskning

Baran och Popovic (2007) arbetade med att försöka skapa en automatiskt genererad rigg för 3D-modeller. Deras motivation för studien är att även ett barn enkelt ska kunna trycka på en knapp för att få sin 3D-karaktär riggad, och skinnad, för att få se deras skapelse röra sig. Vad som representerar en god rigg är att den följer proportioner, benorientering, storlek på karaktären och att namnbeteckningar såsom "fötter" borde tilldelas benen nära botten på karaktären. Deras algoritm, vid namn Pinocchio, testades på 16 humanoida 3D-karaktärer varav inga hade undersökts på förhand. Pinocchio hade där framgång i att skapa riggar till 13 av de 16 riggarna. Två problem som Baran och Popovic stötte på var att om karaktärens lemmar var för smala kunde riggen skapas med avbrutna länkar mellan ben, vilket försämrade prestandan. Det andra problemet var att Pinocchio felplacerade sina knän och armbågar, och inte kunde hitta ett svar på hur problemet kunde lösas på ett gediget sätt. Slutsatsen talar för att deras algoritm var till stor del lyckad, och att Pinocchio kan skapa automatiska riggar till karaktärer med relativt lite ansträngning från användaren. Baran och Popovic redogör att Pinocchio inte var kapabel att hantera riggning av händer eller ansikte, men att det skulle vara möjligt med fortsatt utveckling.

Pan et al. (2009) förespråkar om att riggning och skinning är de mest tidskrävande och besvärliga arbetsaspekterna för en karaktärsanimatör. Det har gjorts flera studier som har försökt att skapa ett slags automatiskt genererade riggar, vilka Pan et al. anser varit för dyra och opålitliga för att vara värda finansiering. Deras studie låg i att skapa en algoritm, som namngavs 3D-Silhouette. Algoritmen fungerar så att under steg 1 görs en genomgående sökning efter vertiser över objektet, därefter i steg 2 letar algoritmen efter delar av vertiser, eller punkter, som ska länkas samman. Resultatet visar att 3D-Silhouette är en användbar algoritm för automatisk riggning av karaktärer med varierande former. Nackdelar till 3D-Silhouette är att även om den kan hantera de flesta karaktärer, har den ännu svårt att förstå sig på asymmetriska former vilket kan få skelettet att hamna någon annanstans än i mitten av karaktären, vilket allra oftast ej är önskvärt. Såvida de planerade animationerna inte kräver att riggen befinner sig i mitten av karaktären, ska den i stort sett ändå alltid finnas där för att få ut bästa resultat från både skinningen och animeringen. För fortsatt utveckling av automatiskt genererade riggning ska 3D-Silhouette därför upplösas, och tas över av en ny algoritm. Denna ska kunna använda en mall för att göra bättre positioner av ben i mitten av karaktärer. Pan et al. överväger att antingen ha en färdiggjord helkroppsrigg där alla ben redan är utsatta, eller använda segment av kroppsriggen där man söker efter benet i mitten av segmentet, exempelvis som armbågen på en arm eller ett knä på ett ben.

Bharaj et al. (2012) såg även att riggningsprocessen var allt för tidskrävande, vilket ledde till försök att skapa en autogenererad rigg som även kunde skapa riggar för vad de kallar en multikomponent 3D-modell. Detta menas då vara en 3D-modell eller karaktär har exempelvis fler än två armar eller andra egenskaper som avviker från humanoid kroppsform. Bharaj et al. menar att det finns gott om multikomponenta 3D-modeller/karaktärer online, och siktar på att skapa en algoritm för att kunna hantera dessa 3D-karaktärer och ge ett passande skelett för animation. Algoritmens funktion har liknande uppgift som Pan et al. (2009) framtidsplaner för deras algoritm 3D-Silhouette, genom att använda färdigkonstruerade riggar, samt en funktion inom algoritmen som placerar ut benen så att riggen hamnar rätt i modellen.

Arbetet visade på goda resultat för algoritmens generering av riggar till multikomponenta 3D-karaktärer som klarade av olika slags animationer, från enkla gånganimationer till mer



komplexa kampsportsrörelser. Bharaj et al. (2012) redogör att deras algoritm är kompatibel med program som är industristandard, och enkla att använda för både yrkesutövande animatörer och nybörjare inom animation.

Algoritmens brister uppstår med placeringen av riggen i modellen, vilket kan leda till orealistiska rörelser under animering. Modellen/karaktern behöver också vara poserad i en liknande pose som algoritmens riggmallar för att fungera korrekt.

Le och Deng (2014) gjorde sin algoritm för att skapa möjligheten att få fram en rigg som redan är skinnad och klar för animation. Le och Deng ansåg att deras algoritm är så pass anpassad att den lätt kan användas i de flesta 3D-mjukvaror eller spelmotorer, med målet att kunna sänka produktionskostnaderna inom exempelvis film och spelindustrin. Le och Deng talar för att deras algoritm till skillnad från tidigare försök kommer vara mer exakt när det kommer till benpositionering och *constraints* för bentransformationen, vilket i deras egna ord räknas som en mer robust rigg. Utvärderingen gjordes genom att jämföra deras algoritm med tre andra autogenererade riggningsalgoritmer.

Resultatet för algoritmerna visar att Les och Dings algoritm överlag fick bäst resultat. Undersökningen hade bara en annan algoritm som fick ett lika bra resultat på två av ett flertal tester. Trots att Les och Dings algoritm hade bättre resultat överlag tog de senare även upp bristerna med den. Algoritmen har låg beräkningseffektivitet, lite exempeldata och begränsad uppskattad kraft för LBS-skinningfunktion.

Sedan forskarnas försök till att skapa en algoritm som kan generera en rigg för 3D-karakterer, har nu även både Blender (1994) och Autodesk Maya (1998) lanserat egna program och verktygstillägg som kan generera riggar med ett enkelt knapptryck. Det finns inte någon forskning kring utvecklingen av programmen, men efter att ha praktiskt använt båda ser vi likheter till Blenders och Autodesk Mayas lösningar. Exempelvis genereras skelettet utefter redan skapade rigg-mallar i respektive mjukvara. Därefter kan användaren justera placeringen på de enskilda benen dit de var tänkta att vara innan man godkänner riggen i programmet.

För utvecklare av industrimjukvaror för spel- och filmproduktion är det klart gynnsamt att kunna erbjuda en funktion som kan dra ned kostnaden för kundernas produktionstid. Det går att anta att utvecklingen av mjukvarorna för de industriledande programmen har haft längre utvecklingstid och finansiering, däremot finns det inte någon forskning kring vad deras programvarors styrkor och brister är.

## 2.2 Manuell riggnings teknik

Nedan följer en beskrivning på hur en arbetsprocess för manuell riggning kan gå till. Rigningsprocesser och arbetsflöden kan på en del sätt skilja sig mellan både olika mjukvaror, och olika arbetsplatser. I allmänhet liknar de dock varandra ofta så pass mycket att en animatör som är van vid att rigga enligt en viss process i en mjukvara, eller på en viss studio, ofta inte behöver någon större inläringstid för att kunna arbeta med riggningsprocessen inom en annan mjukvara, eller på en annan arbetsplats.

Detta då även fast olika arbetsplatser kan ha olika processer, så har de allra flesta flera grundläggande delar gemensamt, såsom exempelvis att ben/joints, IK/FK och kontroller sätts ut och implementeras, beroende på planerad funktion för riggen.

De delarna kan dock måhända utföras på ett skiljande sätt från varandra som kan kräva viss inläring från en nyanställd animatör, men grundfunktionerna förblir ofta liknande. I detta avsnitt finns nedan, utöver den beskrivna processen, även ett flödesschema över den manuella riggningsprocessen.

### 2.2.1 Rignings-schema

Processen att göra en rigg har olika steg som behöver tas innan arbetet kan starta. Först behöver ett rignings-schema göras där man ritat ut hur skelettet ska se ut i karaktären. Denna process görs oftast efter att karaktären är klar att börja riggas. Under denna sektion ser man över om karaktären ska ha ett symmetriskt skelett eller om de finns avvikelser till karaktären. Avvikelser kan exempelvis vara olika storlek på kroppsdelar som armar eller ben. Karaktären kanske har ett par vingar som ska kunna flaxa eller en svans som behöver kontrolleras.

Dessa saker är vanligtvis planerade från början av modelleringen av karaktären och som regel brukar animatörer sträva efter att ha full kontroll över alla delar av kroppen. Skulle karaktären ha en svans exempelvis men inte kunna röra den framstår den som onaturlig vilket ger skapar *uncanny valley*, vilket är namnet på fenomenet som uppstår när virtuella karaktärer görs fotorealistiska men upplevs obekväma för betraktare (Macdorman et al. 2009).

### 2.2.2 Planera funktioner

Samtidigt att ett rignings schema framställts övervägs även vilka funktioner riggen ska kunna utföra. Att använda IK och FK i armar och ben är vanligt förekommande, förutom hos enklare riggar som inte behöver utföra komplicerade rörelser då kan detta ses över. Igen om karaktären har en svans eller vingar kan det behövas ett passande sätt att kunna kontrollera dessa kroppsdelar. Planeringen blir då var constraints och kontroller placeras samt vilka delar de styr över.

### 2.2.3 Konstruera skelettet

När rignings-schemat och funktioner planeras kan konstruktionen av riggen börja. Om planeringen innan varit väl uttänkt kan arbetet på skelettet bli effektivt, eftersom skelettet redan är ritat och behöver bara konstrueras igen i mjukvara.

### **2.2.4 Implementera funktioner**

Vad som följer därefter är att lägga till de planerade funktioner med IK/FK, constraints, kontroller, drivers samt bygga en hierarki av dessa funktioner så att riggen vet vilka funktioner som dikterar över andra.

### **2.2.5 Kontrollera riggen / Orientering**

Vid detta stadiet finns det nu en färdig rigg med alla funktioner som kontrollerar skelettet. Under implementeringen behöver även dessa funktioner testas så att de uppför sig som tänkt. Om något inte fungerar som planerat är detta sista chansen att korrigera problem, överse att orienteringen av skelettet stämmer och att riggen matchar den planerade produkten.

### **2.2.6 Skinning**

När riggen är klar är sista momentet att skinna skelettet till karaktären. Tills att karaktären är skinnad till riggen kommer inte karaktären att följa rörelserna av riggen. Skinningen är alltså momentet där 3D karaktären får informationen om hur mycket den ska påverkas av specifika ben i riggen. Benen i riggen knyts med andra ord till 3D-modellens motsvarande delar på så sätt att karaktären rör sig som planerat när den ska animeras. När exempelvis en vänsterhand på en mänsklig karaktär ska animeras och röra på sig, ska det vänstra handbenet i riggen vara det som animeras. När det handbenet animeras så ska endast vänsterhanden på 3D-modellen röra på sig, och inget annat. Ifall något mer och oförutsett rör på sig vid den animationen, såsom exempelvis 3D-modellens högra ben, får man gå tillbaka igen i skinningmomentet och åtgärda detta.

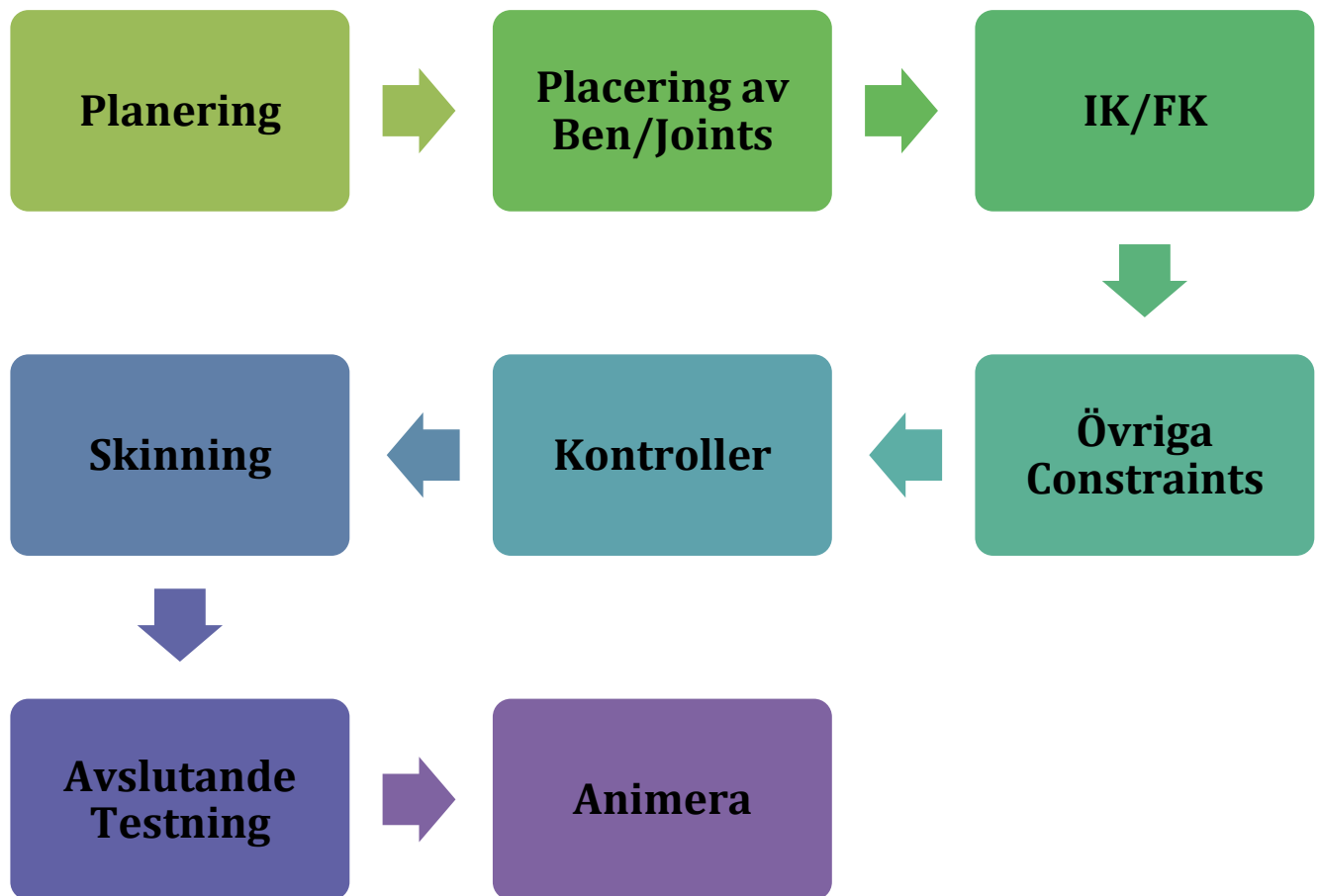
### **2.2.7 Avslutande testning**

Detta stadiet är det sista i själva riggningsprocessen. Här testas riggen på nytt för att se till att den beter sig såsom planerat. Skillnaden i detta riggtest gentemot de tidigare är att man här framför allt undersöker hur väl skinningmomentet har gått. Här animeras karaktären i en mängd överdrivna och onaturliga poseringar och ställningar, samt får sina ben och lemmar förvridna i sådana rörelser som karaktären möjligtvis aldrig var planerad att kunna göra från början.

Anledningen till detta är att man vill testa och kontrollera hur väl riggens alla ben, leder och delar har fästs vid själva 3D-modellen av karaktären. Man tittar då främst på hur modellen deformeras och förvrids vid extrema rörelser och poseringar, och så att varje ditsatt ben i riggen påverkar rätt del av själva modellen. Om man märker att det finns ben i riggen som har en inverkan på delar av modellen som de inte ska ha, så går man tillbaka till skinningstadiet och rättar till dessa. Om modellens deformationer däremot ser tillfredsställande ut, så är riggen färdig och numera redo för att påbörja animationsarbetet med.

### **2.2.8 Animera**

Detta stadiet är inte en egentlig del av själva riggningsprocessen, utan skrivs endast ut som ett steg i processen för att markera att själva animationsarbetet kan påbörjas. I detta stadiet ska riggen vara fullkomligt färdigställd, testad och haft alla eventuella brister och uppkomna problem åtgärdade. Med andra ord är riggen nu helt redo för att börja animeras av en animatör.



**Figur 1:** Exempel på flödesschema över den manuella riggningsprocessen i sin helhet.

## 2.3 Riggning med Rigify

Likt föregående stycke följer här nedan en beskrivning på hur en arbetsprocess för riggning kan gå till, men i detta fall med användning av Rigify istället för helt manuell riggning.

Även här går det att se att flera stadier och delar från den beskrivna manuella riggningsprocessen återkommer i riggningsprocessen med Rigify, då vissa grundläggande delar alltid återfinns oavsett riggningsprocess. I denna del kommer inte tidigare steg att repeteras utan hänvisar tillbaka till de manuella riggningsteknikerna i 2,2.

### 2.3.1 Planering

Samma process som beskrivs i manuell riggning 2,2.

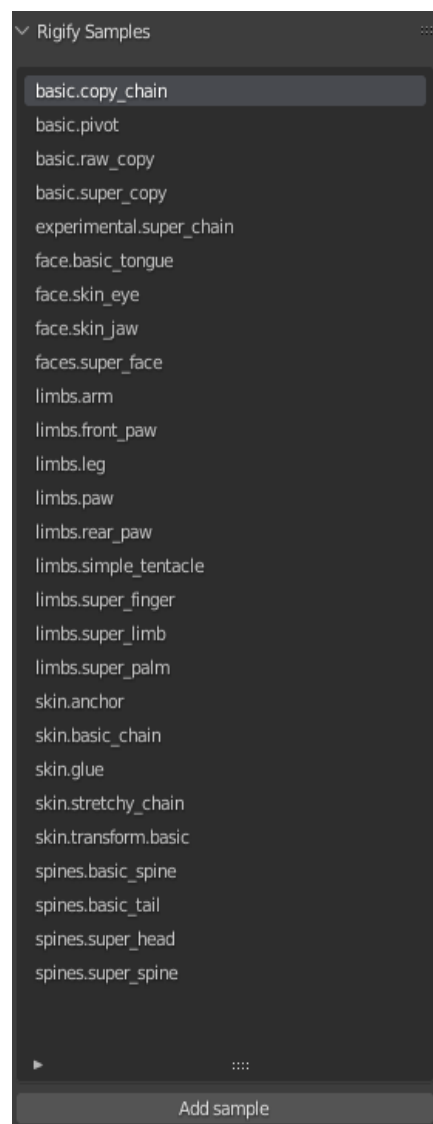
### 2.3.2 Eventuell Meta-rigg & Eventuella Benbyten

Efter planeringsfasen kommer det till att välja ut en passande *meta-rigg*, vilket är namnet på de förproducerade skelett-armaturerna som går att välja mellan. När lämpligast meta-rigg valts så kommer Blender (1994) att generera skelettet. Skelettet som genereras består av ett stort antal ben och kan behövas korrigeras med att ta bort ben som inte anses nödvändiga för karaktären.

Med Rigify kan användaren enkelt byta ut ben eller leder av ben, leder av ben är exempelvis alla ben som behövs för att röra en arm. Rigify har ett antal olika meta-riggs-leder som sen kan läggas till och ersätta den ursprungliga benleden, dessa kallas *samples*.

Det krävs dock viss förkunskap om hur Rigify och *samples*-benen fungerar för att resultatet ska bli som planerat om dessa ska användas. De olika *samples*-benen och lederna har olika funktioner, kan vara vitt skilda från varandra och används till olika ändamål i en rigg. För att nå uppåt önskat resultat krävs därför även ett visst mått av förkunskap, förplanering och möjligt experimenterande för att ta fram rätt *sample* för ändamålet i riggen när den sedan genereras.

Värt att nämna är även att det är fullt möjligt att bygga ihop en helt egen rigg endast med hjälp av *samples*. Om ingen utav de förproducerade metariggarna i Rigify passar till 3D-modellen som ska riggas är det inte nödvändigt att använda en sådan för att bygga en rigg i Rigify.



**Figur 2:** Rigifys lista med leder och ben, så kallade *samples*.

### **2.3.3 Placering av Ben/joints**

När alla ben har genererats för riggen kommer nästa del i arbetet att placera dem så att de passar in på karaktären. Rigify är ett bra verktyg för riggning men det kan inte förutse hur karaktären ser ut och var då ben ska placeras, motsvarande från tidigare försök till att skapa automatiserad riggning (Baran & Popovic 2007; Pan et al 2009; Bharaj et al 2012; Le & Deng 2014).

Nästa steg blir då att korrigera placeringen av ben så att de passar in på karaktären.

### **2.3.4 Generering av Kontroller & Constraints**

Vid detta steget har nu karaktären fått sitt skelett på plats på det sätt som planeringen tänkt. Därefter godkänns riggen och Rigify genererar riggens kontroller och constraints som skapar en hierarki av skelettet. Detta gör framförallt att riggens ben vet vilket ben som den ska följa, med hjälp av kontroller som styr ben och leder samt constraints som bestämmer på vilket sätt benet kan röra sig.

### **2.3.5 Eventuell Omgenerering**

Sista steget innan skinningprocessen är att testa kontroller på ett sådant sätt att skelettet fungerar som planerat. Har det inte förändrats något med riggen är det stor chans att den fungerar. Om det har ersatts något ben eller led är det viktigt att se så att den fungerar med varandra även när kontrollerna genererats, annars kan man behöva gå tillbaka och undersöka fel för att prova att generera nya kontroller och constraints.

### **2.3.6 Skinning**

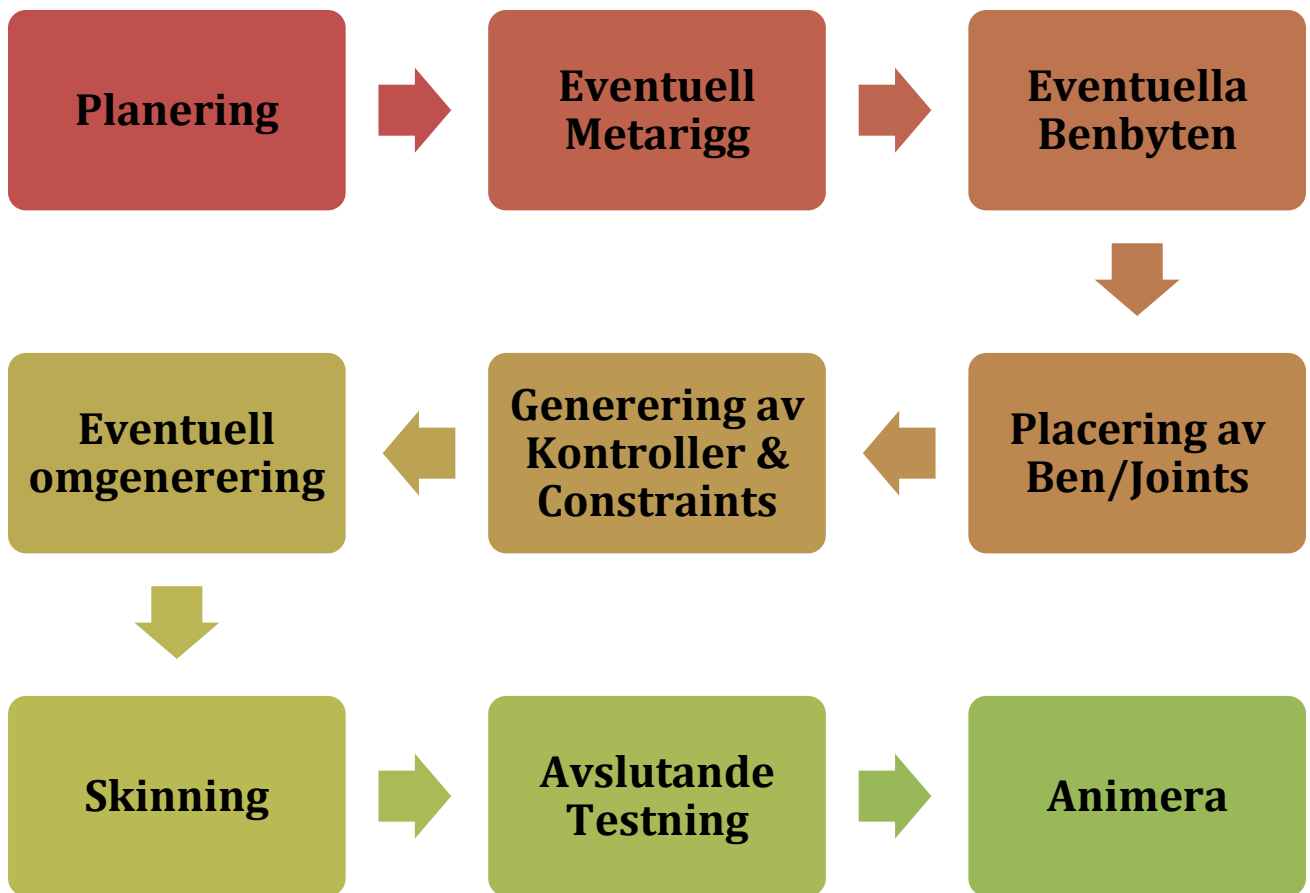
Då Rigify varken har kapaciteten eller funktionen att kunna utföra skinning måste detta göras för hand på samma vis som finns beskrivet i manuell riggning 2,2.

### **2.3.7 Avslutande testning**

Samma process som beskrivs i manuell riggning 2,2.

### **2.3.8 Animera**

Samma process som beskrivs i manuell riggning 2,2.



**Figur 3:** Exempel på flödesschema över riggningsprocessen i sin helhet med Rigify.

### 3 Problemformulering

Animatörerna jobbar idag med att skapa användarvänliga riggar för att sedan kunna skapa animationer för spelprojekt. Under tiden spel- och filmindustrin har utvecklats har även verktygen i 3D-mjukvaran till spel- och filmproduktion utvecklats. Flera försök har gjorts för att kunna automatiskt generera riggar till 3D-karaktärer, då några av dessa algoritmer idag är gamla, eller kan vara svåra att få tag på, har det i denna studie beslutats om att testa den autogenererade riggen som kan tas fram i Blender (1994), vilket är en av de ledande grafiska 3D-mjukvarorna inom industrin idag.

Med tanke på arbetsprocessens utformning blir frågeställningen: Kan Rigify spara arbetstid och resurser i riggningsprocessen i jämförelse med manuell riggning?

Anledningen till att använda Blender (1994) för experimentet är främst för det används mer frekvent idag runt om i spelindustrin, och konkurrerar med industriledande Autodesk Maya (1998). Således till varför arbetet inte valt att använda sig av Autodesk Maya autoriggning är för att programmets automatiskt genererade riggar är mer fokuserade till humanoida former. Blender är en offentlig förmånsorganisation och strävar efter att vara det bästa 3D-programmet på marknaden utan kostnad. Deras vision är att alla ska kunna ha tillgänglighet till ett 3D-program som uppmuntrar individen att visa sin kreativa sida. Licensen Blender har är till för att individen ska kunna använda Blender till vad som helst denne vill skapa. För frilansartister, eller för de som vill yrka inom 3D-grafik, är Blender (1994) ett alternativ som både, är ekonomiskt gångbart då att programmet är gratis att använda, samt att det går att tjäna pengar på att använda mjukvaran utan att behöva betala en summa till Blender. Autodesk Maya i jämförelse har både en hög månadsavgift, samt att man behöver köpa en licens för att få använda mjukvaran i sitt arbete i kommersiella syften, vilket många privatpersoner inte har kapital för.

Blenders (1994) förinbyggda riggnings-tilläggsprogram Rigify har olika alternativ mellan riggar av människa, katt, hund, haj, fågel eller häst. För denna undersökning har vi valt att inte använda oss av en humanoid varelse, utan valt modellen av en katt. Eftersom Rigify har en kattdjursrigg redan anser vi att detta blir mest rättvist för utvärderingen då den automatiska kattdjursriggen bör möta en vis kvalitet som Blender förlitar sig kunna användas av användare.

Målet som forskare strävat efter har varit att sänka produktionstiden för riggning med hjälp av algoritmer (Pan et al. 2009; Le & Dang 2014). Deras arbete har skapat tillgångar för vidareutveckling av goda verktyg för framtidens industri dock inte utan brister i deras algoritmer. Ingen av forskarna har sett att deras algoritm är tillförlitlig nog eller ekonomisk hållbar för företags användning. Utöver detta har ett industriledande företag som Blender (1994) utvecklat Rigify som motsvarighet för riggningsalgoritmerna. Detta är då en algoritm som haft goda finansiella tillgångar och flera uppdateringar sedan november 2019 när Rigify lanserades. Om Rigify då har vad som idag krävs för att kunna skapa användningsbara riggar för användare av Blender, kan detta möjligtvis vara ett bra verktyg för företag att sänka sin produktionskostnad och inte behöva lägga ytterligare arbetstid till riggningsprocessen.



### 3.1 Metodbeskrivning

Undersökningen skedde så att det skapas en rigg till ett fyrfota kattdjur, där riggen skapades av Blenders egna autogenererade rigg-program *Rigify*. Då forskningen ämnade att undersöka ifall *Rigify* kan spara arbetstid och resurser hos en mindre studio, samt vara tillfredsställande funktionsduglig för sitt produktionsändamål, var det viktigt att riggen fungerar i spelmotorer. Detta även för att spelmotorsfunktionalitet anses vara både grundläggande och nödvändigt för alla animationsriggar som ska användas i spelproduktion.

I undersökningen fick deltagarna utvärdera riggens funktionsduglighet och användbarhet i Blender (1994) för spelproduktion. Undersökningsdeltagarna bestod av yrkesutövande animatörer från olika studior i Skövde. Dessa animatörer behövde ha åtminstone viss erfarenhet med animation i Blender för att utvärderingen skulle anses värdefull för resultatet. Om det skulle ha blivit brist på animatörer som var tänkta att ställa upp som deltagare, till en gräns som ej kunde bortses från eller var acceptabel, kunde även animatörer som endast hade erfarenhet av andra animationsmjukvaror komma till att bli tillfrågade om att delta. Denna eventuella brist på erfarenhet inom Blender borde då ha tagits med i resultatet och utvärderingen, då det kunde ligga till grund för olika svar, resultat och utvärderingar från deltagarna.

Kattdjursriggen presenterades som en karaktär i ett spel, och gav därmed deltagarna ett perspektiv av vad som bör krävas eller förväntas av riggen. Deltagarna fick tillgång till riggen för att kunna utvärdera den genom att använda den. De gavs inga instruktioner kring vad som förväntades göras, eller skapas, med riggen i animationsväg. Ett videoklipp av kattringen med en gånganimation skapades som en arbetsreferens för deltagarna att använda, men fanns endast med som ett möjligt verktyg att använda och var inte på några sätt ett krav att förhålla sig till på något sätt. Då kvaliteten på riggens animationer och poseringar som skapades av deltagarna inte var av betydelse för varken undersökningen eller resultatet, fick deltagarna själva bestämma vad som skulle skapas eller hur riggen skulle användas. Deltagarna kunde skapa en animation, poseringar eller bara leka runt med riggen utan något mål. Det som var viktigt för undersökningen var att riggen användes tillräckligt länge, och på ett sådant sätt, att en utvärdering av den kunde ske. Där fick deltagarna själva avgöra om tillräckligt med tid hade getts innan intervjun, eller om mer tid behövdes för att få en grundlig uppfattning om riggen och dess kvalitet och egenskaper. Efter testningen av riggen fick sedan deltagarna bedöma vad som var bra och dåligt med den, genom en kortare kvalitativ semistrukturerad intervju såsom beskrivs av Östbye et.al. (2004).

Detta upplägg blev för undersökningens ändamål, i vår mening, det mest gångbara, då studien inte ämnade utröna ren mätbara data, men ändå krävde viss grundläggande struktur av intervjuernas upplägg, samt frågorna som skulle ställas. Antalet undersökningsdeltagare blev troligtvis inte tillräckligt i antal för att mer kvantitativa metoder skulle kunna ha gett värdefulla resultat och data att analysera tillförlitligt. Antalet yrkesutövande animatörer från mindre studior i Sverige är tämligen begränsat. Att hinna tillfråga en stor mängd animatörer i allmänhet hade varit vanskligt tidsmässigt, och även att hitta yrkesutövande animatörer som både har erfarenhet av att animera i Blender, samt skulle vara villiga att delta i undersökningen, riskerade att göra det möjliga urvalet av deltagare ännu mindre. Med det begränsade antalet animatörer vi hade tillgång till som undersökningsdeltagare, föreföll det

sig mest fördelaktigt med ett kvalitativt upplägg på intervjuerna, då det gav en möjlighet för strukturerade teman och frågor, men även stor flexibilitet för spontana uppföljningsfrågor som närmar sig samtalsintervju. Ett bekymmer med detta upplägg blev att den data som skulle samlas in kunde bli svår att analysera för att utröna värdefulla slutsatser.

Till följd av det valda forskningsområdet och vald grupp av deltagare, blev det vanskligt att utföra fältobservationer, eller observera deltagarna alls, medan de använde riggen. Detta främst då deltagarna befann sig på olika arbetsplatser under dagen, till vilka tillträde för utomstående ej ges, samt att deltagarna befann sig på olika platser i sina hem efter arbetstid. Det var heller inte rimligt att på ett sådant aktivt sätt inkräkta på deltagarnas fritid, istället för att låta deltagarna själva bestämma när, var och hur de ville undersöka själva riggen i sig. Därför bestämdes en tidpunkt för enskilda intervjuer med deltagarna för att utvärdera riggen och få ta del av deltagarnas åsikter och upplevelser av riggen, efter det att deltagarna hade fått en överenskommen tidsperiod att använda och utvärdera riggen på egen hand. Även fast observation med aktivt lyssnande som beskrivs av Östbye et.al (2004) hade kunnat fungera som ett bra komplement till intervjun, var det i vår mening inte nödvändigt för att besvara frågeställningen. Detta för att frågeställningen förlitade sig på utvärdering av deltagarnas observationer av riggen. Med andra ord var varken frågeställningen, eller resultatet, beroende av deltagarens omedelbara reaktioner.

För att analysera data sågs deltagarnas svar över angående om riggen framstod som positiv eller hade märkbara brister. Med tanke på att deltagarna kunde ha olika mängd erfarenhet inom industrin samt erfarenhet av riggar i Blender (1994), analyserades allas svar med målet att hitta någon typ av generell överblick om hur riggen kan vara till användning.

Data jämfördes med frågeställningen huruvida Rigify är ett bra alternativ för att spara produktionstid, samt resultera i svar om det kan vara ett nyttigt verktyg för företag. Slutligen var målet att kunna få en övergripande blick för vilka användningsområden Rigify är lämpligt för ur produktionssynvinkel.

## 4 Projektbeskrivning

### 4.1 Artefaktens riggningsprocess

Under planeringsstadiet, efter att artefakten hade beslutats till att bestå av en rigg gjord i Blender (1994) tillsammans med Rigify, behövdes även ett beslut om vilket slags karaktär som skulle riggas, samt vara föremål för artefakten och undersökningen. Att använda en mänsklig, eller humanoid, karaktärer diskuterades, men ansågs vara för enkelspårigt och även till viss del överflödigt. Detta för att det inom spelvärlden är till en överväldigande del vanligast att ha mänskliga, humanoida eller människoliknande karaktärer som både spelbara och icke-spelbara karaktärer. Detta är inte konstigt då vi som människor lättare känner igen oss i mänskliga, eller människoliknande, karaktärer än i exempelvis olika djur. Det finns därför redan mycket material att hämta för både forskning och allmänt kunskapssökande när det kommer till att rigga humanoida karaktärer.

Lika mycket material finns dock inte för icke-humanoida karaktärer, och ansågs av oss vara en jämförelsevis oberörd och utforskad del av kunskapsmaterialet för riggning, vilket gjorde att beslutet togs att använda en rigg och 3D-modell av ett djur från vår egen värld. Djuret som valdes för artefakten blev då en katt, då alla människor i världen vet vad en vanlig huskatt är, samt hur den kan se ut och röra på sig, men den är också fyrfota och odiskutabelt icke-humanoid.

Kattdjur är en av de förproducerade metariggarna i Rigify som går att använda utöver människa, hund, haj och fågeldjur. Eftersom undersökningen ser till hur Rigifys kvalitet är i jämförelse till manuell riggning är det därför rimligt att använda en av dessa förproducerade metarigggar. Det var då även passande för undersökningsresultatets skull att denna kattmetarigg skulle vara det enda i skelett-väg som skulle sättas ut som en rigg. Inga ben eller joints, utöver de som redan finns med metariggen när man skapar den från menyn, fick alltså tas bort, läggas till eller ändras på. På så sätt kan alla återskapa den rigg som vi använder till undersökningen om man har Blender-mjukvaran och Rigify aktiverat däri.

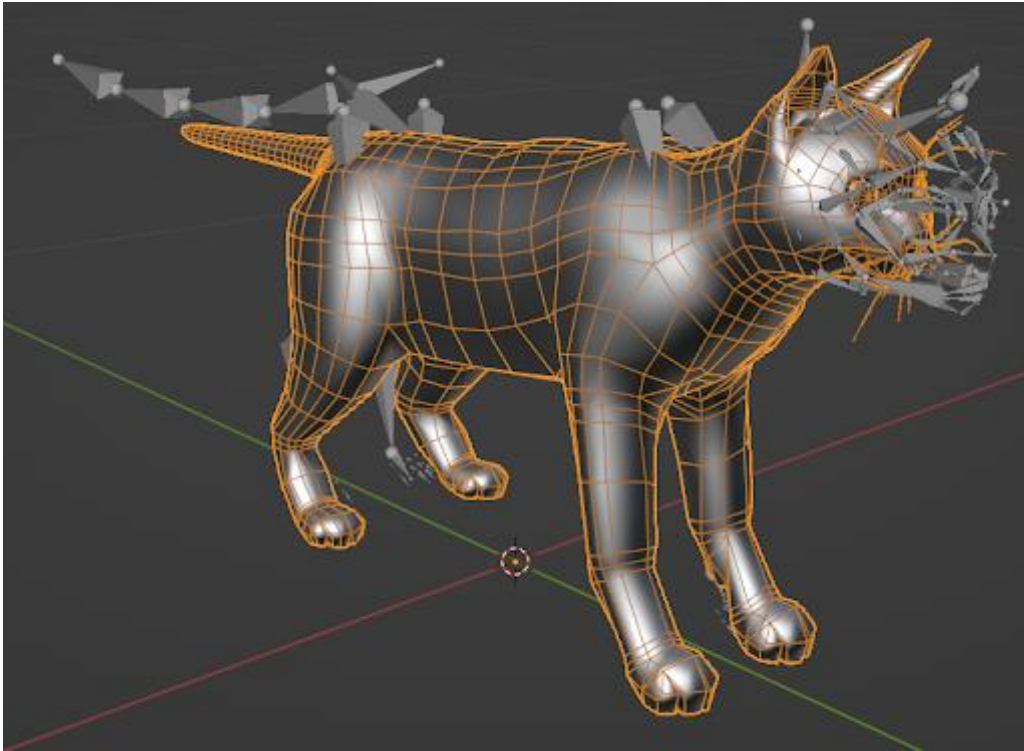
Första kattmodellen som valdes ut var gjord av en Schlossbauer (2018). Problemet med denna var att modellen hade en plattform som katten stod på som behövdes tas bort och därmed korrigeras så att modellen kunde riggas. Vidare upptäcktes det att modellen inte kunde öppna munnen, detta är inte nödvändigt för en gång animation men för att hela riggen skulle kunna fungera som tänkt. Slutligen krävdes det för mycket tid att korrigeras modellen vilket gjorde att det söktes efter en ny.



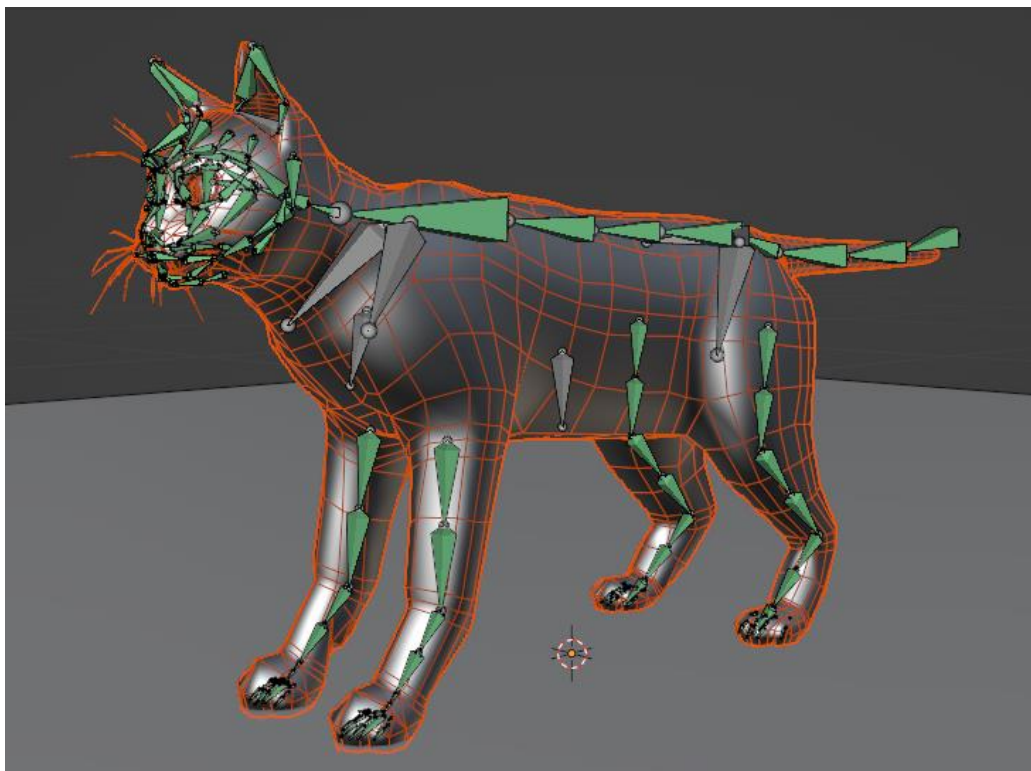
**Figur 4:** Första 3D-modellen som användes (Schlossbauer, 2018).  
Denna ersattes sedan av den aktuella modellen.

Kattmodellen som är aktuell är gjord av Moore (2018). Modellen var optimerad för animation och kom med en befintlig rigg samt animationer. Detta gjorde att valet av modellen blev obestridd när den redan gjorts en rigg. För undersökningen togs den befintliga riggen bort då det är Rigify rigg som ska undersökas, animationerna som följde med spelades in som referensmaterial för deltagare.

När modellen var på sin plats behövde Rigify-riggen implementeras och korrigeras efter modellen. Eftersom Rigify riggen konstrueras efter en mall behövdes arbete läggas på att få meta-riggen att passa modellen. Utmaningar uppstod av att meta-riggen ansågs för avancerad för kattmodellen trots att den var optimerad för riggning. Kattens ansikte hade framförallt fler ben än nödvändigt dock skulle inga ändringar göras på meta-riggen. Katt-anatomin blev även en utmaning för placeringen av ben i Blender (1994) då kattens skuldror är betydligt högre placerade från ryggraden än hos en människa. När justeringar gjorts för meta-riggen kunde därefter skinning-processen börja. Därefter var den klar för våran undersökning.



**Figur 5:** Aktuella modellen med en ny meta-rigg implementerad



**Figur 6:** Aktuella modellen med korrigerad meta-rigg utplacerad utefter modellen.

## 4.2 Pilotundersökningen

Innan undersökningen kan utföras ska en pilotstudie utföras för att försäkra om att undersökningen är optimerad för att kunna svara på frågeställningen. För denna pilotundersökning kommer deltagaren få tillgång till riggen samt en fråga om hur lång tid deltagaren uppskattar att deras företag skulle lägga på att konstruera en liknande rigg. Varför deltagaren redan får en fråga innan intervjun är för att frågan kan behöva extra tid att besvara och evaluera. Katt-karaktern kommer att presenteras som en karaktär i ett spel som utvecklas av ett spelföretag. Deltagaren kommer inte att få veta att riggen är ifrån Rigify utan ska introduceras som att den var manuellt konstruerad. Eftersom deltagarna ska bestå av folk som jobbar inom spelindustrin, finns det risk att vetskap om att riggen kommer från Rigify kan göra svaren partiska.

Deltagaren kommer att få några dagar att evaluera riggen, dessutom vill vi att deltagarna ska göra ungefär lika mycket samt att deltagarna får en omfattande bild av riggen. Därför får deltagaren en uppgift att skapa en gång animation till katten. Detta ger deltagaren ett tydligt mål och ramverk för uppgiften. Även om animationerna inte kommer att utvärderas kan det användas för deltagare att upplysa om fel eller brister av karaktären.

När uppgiften har gjorts och några dagar passerat bokar vi tid för en intervju, här kommer vi att fullfölja frågan om hur lång tid deltagaren uppskattade att riggen skulle ta att konstruera. Därefter gå vi över till frågor om hur länge deltagaren jobbat inom industrin samt om riggen har egenskaperna till att passa för en karaktär i ett spel. Intervjuns fokus är på deltagarnas utvärdering av riggen och förekommande intryck från användandet av den.

## 5 Undersökning & Resultat

Efter en utvärdering av pilot-undersökningen valdes de frågor som gav mest fördelaktiga resultat till den slutgiltiga undersökningen. Kontrollfrågor gav mer fokus på att ge bild av upplevelsen deltagaren hade med Blender (1994), samt Rigify. Därefter utfördes undersökningen på ytterligare deltagare som valt att ställa upp, samt deltagaren från pilot-undersökningen. Deltagaren som redan hade varit med i den tidigare pilot-undersökningen ombads omvärdera karaktären, då undersökningens upplägg skulle bli omarbetat och frågorna uppdaterade. Samtalen spelades även in för att kunna hålla intervjuerna mer flytande, och undvika avbrott för att ta anteckningar. Tillsammans blev det fyra deltagare i den kvalitativa studien.

### 5.1 Intervjuer

Nedan följer de 4 enskilda deltagarnas utvärderingar genom intervjuerna.

#### 5.1.1 Intervju 1

Första deltagaren jobbade som ensam animatör av totalt 8 anställda. Dock hade studion ytterligare en anställd som kunde assistera, även fast denne inte hade samma yrkesroll. Deltagaren hade tidigare erfarenhet av Blender (1994) och visste vad Rigify var och används till, men hade aldrig använt verktyget. Gällande riggen så upplevdes den alltför avancerad, samt att deltagaren saknade kontroll. På frågan om karaktären skulle vara godkänd på deras studio svarade denne ja, dock skulle den antagligen ta upp onödigt stor plats i spelmotorn. Andra nackdelar var att kontrollen för huvudet gömdes när man poserade karaktären, då IK-leden för benen inte gick längst med hela benet, vilket gjorde det svårt att posera. Deltagaren lade till egna funktioner för att få kattens tass att böjas när den träffar marken "för att det skulle vara lättare att använda". Vidare togs frågan upp om vilket som tog mest tid, att använda Rigify utan vidare erfarenhet och behöva ändra i riggen, eller att konstruera en själv.

#### 5.1.2 Intervju 2

Deltagare var anställd som *art director* på samma studio som den första deltagaren och jobbar framför allt med 3D-grafik och hjälpte till med riggning på deras studio. Liksom som den första intervjun hade deltagaren erfarenhet av Blender, men hade endast hört talas om Rigify förut. Huruvida Rigify var ett tänkbart verktyg för deras studio i dagsläget var svårt att säga. Deltagaren ville veta mer om hur lång tid det skulle ta att använda Rigify, då studion söker efter att ha en så snabb pipeline som möjligt. Karaktären upplevdes att ha ett enkelt upplägg i sin rigg, och blev därför enkel att animera med. Kontrollen för framtassarna var svåra att komma åt, så de hade behövt varit större. Utöver detta ansåg deltagaren att riggen var imponerande "om jag animerade och hade fått detta levererat till mig så hade jag varit skitglad".

### 5.1.3 Intervju 3

Nästa deltagare jobbade på samma studio som deltagarna från intervju 1 och 2. Deltagaren hade yrkesrollen som 3D-grafiker och hade erfarenhet av Blender, men hade dock inte använt Rigify. Då deras studio i nuläget använder sig av enklare karaktärer, ansågs de inte behöva Rigify. Deltagaren påstod att "av det jag sett av Rigify så finns det mycket man måste fixa för att passa våra *low poly*-karaktärer". Rigify kan nog passa många, förutsatt att det går att anpassa riggen som genereras. Karaktären ansågs avancerad och hade ett flertal onödiga ben. För studios som gör detaljerade animationer, eller strävar efter att skapa *cartoon*-spel, skulle Rigify säkert vara ett användbart verktyg. Riggen ansågs dock vara alltför överflödigt och invecklad för att få ett godkännande att använda inom studion. Det fanns alltför många funktioner i riggen som inte kändes nödvändiga, samt att kroppsdelar kunde expanderas på onaturliga sätt och huvudet rörde sig på ett konstigt sätt. Deltagaren menade på att om man ska animera något verkligt som en katt, så har man en viss förståelse för hur en katt kan röra sig.

### 5.1.4 Intervju 4

Sista Deltagaren jobbade på en större studio med totalt 36 anställda, varav 3 jobbade som animatörer. Deltagaren hade erfarenhet av Blender (1994) och hade använt Rigify i privat bruk. Studion använder Maya (1998) och ansåg inte behovet fanns att byta till Blender, då Rigify inte heller var nödvändigt för dem. Deltagaren menade på att skapa egna riggar resulterar i att animatören får mer kontroll under produktionen. Riggen var för avancerad överlag, och deltagaren hade också problem med att IK lederna inte gick genom hela benen. Kontrollen för huvudet var för liten och försvann under tiden karaktären poserades. Deltagaren gick vidare med att säga att "riggar ska tjäna animatören", och ansåg att denna riggen hade många väsentliga brister som gör det svårt att ha kontroll för animation. Även om riggen i sig skulle bli godkänd för produktion på deltagarens studio menade denne att det är animationerna som blir godkända, inte riggen.

Vidare anser deltagaren att Rigify är ett verktyg som borde användas av de med övergripande kunskap om riggning. Detta menade deltagaren kunde bli en nybörjarfälla genom att använda Rigify. Först och främst finns risken att utvecklaren inte får grundliga kunskaper om riggning, och kan därmed inte felsöka och åtgärda de problem och konflikter som kan uppstå när Rigify används. Den andra risken är att utvecklaren endast använder sig utav Rigifys förtillverkade meta-riggar, såsom exempelvis katt-riggen, till all produktion och inte efterforskar kring hur man kan bygga en helt egen rigg från grunden. Kunna skraddarsy och anpassa riggen för sina egna behov med hjälp av Rigify blir även det en följd av bekvämligheten att endast använda de förtillverkade meta-riggarna. Den tredje risken är att utvecklaren, till följd av sin då brist på någorlunda djupgående kunskap inom riggning i helhet, blir låst till, och därmed beroende av, att använda Rigify till all grafisk produktion som kräver en rigg.



## 5.2 Analys

Den största problematiken som framkom under alla intervjuer var att riggen var alltför avancerad. Exempelvis hade kattens ansikte ett högt antal ben, som samtliga deltagare inte ansåg nödvändiga eller fyllde någon funktion alls. Det flertal funktioner, såsom kontrollerna och constraintsen i riggen, blev en positiv aspekt från deltagaren i intervju 2. Dock ansågs de överflödiga av deltagare i intervju 3 för en enklare 3D-modell, som exempelvis den katt-modell de hade blivit tilldelade till undersökningen. Deltagare förespråkade att riggen skulle kunna vara godkänd inom deras egen studio och arbetsplats, men var alltför onödigt invecklad. Eftersom en större rigg tar på prestanda i spelet, så är det gynnsamt att ha en enklare rigg som fortfarande kan göra de rörelser som planerats. De animatörer som intervjuats för undersökningen ansåg också att känslan av kontroll var en viktig aspekt för deras arbete. Detta syftar till att ha kontroll över riggen, känna till dess funktioner på djupet, samt kunna hantera riggen på ett enkelt sätt.

Frågan togs upp vilken, eller vilka, som skulle kunna använda Rigify som ett tänkbart verktyg för produktion. Svaren bestod till stor del att nystartade spelstudior mest skulle gynnas av att använda Rigify till en början. Senare ansåg deltagaren från intervju 4 att skapa eller införskaffa sig egna verktyg kunde vara en bättre investering. Det visade sig under intervjun att större företag som främst använder sig utav den grafiska mjukvaran Maya (1998) har personal som skapat egna algoritmer och kodade skript för sina egna Maya-mjukvaror, så att programmet blir skraddarsytt till deras arbetsprocess och behov. Därav uppstår frågan om Rigify kunde vara gynnsamt för yrkesmässiga egenutvecklare, med andra ord individer som utvecklar egna spel. Dessa egenutvecklare behöver trots allt arbeta med alla delar utav spelutvecklingsprocessen, vilket gör att Rigify kan vara ett användbart verktyg till att lätta på det grafiska arbetet vid behov av en rigg.

Omfattande brister för riggen var hur IK-segmentet för benen inte sträckte sig över hela kattens ben, utan slutade innan skulderbladet. Detta gör så att om hela benet rör sig när tasskontrollen flyttas, så står skulderbladet still och skapar en avvikande deformation. Vanligtvis vill en animatör att hela benen följer med rörelsen för att inte behöva kontra-animera med delar av benet. Kontroller, såsom för huvudet och tassarna, fick kritik för att vara små, vilket resulterade i att de kunde vara svåra att hitta under tiden animatören poserade katten. Vidare anmärkningar på att riggen kändes för invecklad grundades i att deltagare 1 ansåg att många av kontrollerna som Rigify genererat inte hade någon till synes inverkan på karaktären. Deltagaren från intervju 1 tog sig friheten att lägga till egna funktioner till riggen för att få den mer hanterbar att arbeta med. Vidare diskuterades det angående om en individ har erfarenhet av Rigify, och därmed kan använda sig utav *samples*, så kan riggen göras enklare. Dock blev det en fråga från intervju 2 huruvida det tar mindre tid att generera en rigg i följd med att ersätta och ta bort den från meta-riggen, eller om det tar mindre tid att skapa riggen manuellt från grunden. Eftersom det inte var möjligt att räkna ut en uppskattning om hur lång tid det skulle ta att göra en rigg med Rigify, i jämförelse med manuell riggning på en industrinivå, kunde svaret på denna fråga endast spekuleras om. Arbetets fokus fick inrikta sig till möjligheterna hos karaktärens rigg, vilket till stor del fick positiva omdömen, även fast det finns utrymme till förbättringar.

Ett fåtal deltagare ansåg att de saknade djup förståelse för riggning som ämne och hade därmed svårt att göra en djupare utforskning av Rigify-riggen, utan att behöva bekanta sig med Rigify. Frågan om Rigify skulle gynna spelstudios blev en större fråga om resurserna att faktiskt ta tid att lära sig Rigify. Utöver det så rättfärdigar inte den slutliga kvaliteten på riggar med att vara värt för studios att byta till att använda Rigify. Detta då deltagarna lyfte fram fördelar med att göra riggar manuellt, det ger animatören mer kontroll och behöver inte i många fall vara så invecklad som Rigifys meta-rigg.

### 5.3 Slutsatser

Slutsatser som kan dras från denna undersökning är att meta-riggen som genereras av Rigify är alltför avancerad. Det kan förstås gynna spelstudior som använder sig av karaktärer med högre upplösning. Dock upplevs den nuvarande karaktären ha för låg upplösning och därmed blir riggen överflödig. För de som har erfarenhet av Rigify, och har möjligheten att ändra om med programmets *samples*, finns möjligheten att skapa enklare och mer anpassade riggar. Detta förutsatt att de har erfarenhet och kunskap av Rigify och hur det kan användas. Karaktärens rigg fick trots allt goda betyg genom att förbise brister i genereringen av riggen, samt de constraints och kontroller som skapade komplikationer för animatörerna.

Resurserna som krävs att lära ut Rigify när det kommer till arbetstid samt att jämföra detta till studios nuvarande metoder spekulerade deltagarna kring att vara högst kostsamt. Eftersom större studios har egna verktyg som är anpassade för deras verksamhet samt har liknande funktioner som Rigify. Därför blir Rigify från ett större studio-perspektiv inte lockande att använda i sitt nuvarande läge.

Rigify anses vara bäst lämpad för nystartade spelföretag utifrån deltagarnas åsikter. Även utvecklare som jobbar på egen hand kan ha användning av Rigify, då spelutveckling består av olika delar såsom exempelvis programmering, ljud/musik, berättelse samt design. I denna situation skulle Rigify gynna utvecklare genom att ta bort en stor del av den grafiska delen till spelproduktion.

## 6 Avslutande diskussion

### 6.1 Sammanfattning

Rapporten undersöker hur kapabel Rigify kan vara för professionellt arbete inom spelindustrin. Rigify är ett tilläggswerktyg, eller add-on, i Blender (1994) som innehåller meta-rigggar som fungerar som mallar av riggnings-skelett. Dessa kan genereras automatiskt för att snabbt kunna rigga karaktärer. Då Rigify är ett verktyg som alla kan använda genom att ladda ner Blender gratis, ställer detta arbete Rigify på prov om det kan lätta på arbetsresurser för animatörer inom industrin. Ett fyrbent djur valdes ut och riggades med hjälp av Rigify, vilket gjorde så att endast korrigering av ben och skinning av riggningsprocessens delar behövdes göras. För undersökningen bestod deltagarna av anställda inom spelindustrin, framförallt från studios som använder Blender samt deltagare som har erfarenhet att animera i mjukvaran. Efter att deltagarna mottagit karaktären och fått chans att utvärdera den planerades en intervju. Under intervjun fick deltagaren dela sina åsikter om karaktären, således dess styrkor och brister.

Resultatet visar på att riggen har en del brister när det kommer till funktioner av ben samt tillgängligheten av kontroller. Deltagare var dock till stor del positiva till hur karaktären har många funktioner fastän de också upplevdes överflödigt. Det blev en omfattande synpunkt att riggen var allt för avancerad, det vill säga mer än nödvändigt och att manuell riggning skapar mer kontroll hos animatören. Vidare var frågan om hur lång tid det skulle ta att faktiskt kunna bruka Rigify på en studio. Större företag använder sig ofta utav egna verktyg som är anpassade för deras studio och behov, och har i det fallet ingen egentlig användning av Rigify. Däremot kan det vara ett bra verktyg för nystartade studios, samt för att avlasta arbetet för eget företagande.

### 6.2 Diskussion

Utöver det nämnda ekonomiska skälet går det även att argumentera att Rigify möjliggör för alla spelutvecklare, oavsett inriktning eller yrkesroll, att jämförelsevis snabbt och enkelt kunna skapa en fullt funktionell och tämligen utförlig och invecklad rigg som tidigare arbeten strävat efter (Baran & Popovic 2007; Pan et al 2009; Bharaj et al 2012; Le & Deng 2014). Visserligen behöver den som väljer att använda Rigify göra åtminstone viss efterforskning för att veta hur verktyget ska användas, vad det kan göra och vad riggen den skapar kan göra och består av. Detta gäller förstås alla spelutvecklare som inte har använt Rigify tidigare, men i synnerhet nybörjare och andra utvecklare som endast besitter antingen endast ytlig, eller ingen, tidigare grundkunskap om vad riggning är och dess arbetsprocess. Trots detta gäller fortfarande argumentet för Rigify, då det oavsett tidigare kunskapsnivå om riggning som ämne, ändå kräver mindre tid och ansträngning att lära sig än att lära sig rigga manuellt från grunden (Baran & Popovic 2007; Pan et al 2009).

I frågan om Rigify är tillrådligt att införa och användas på större studios med jämförelsevis mycket resurser i form av kapital, arbetskraft och effektiva arbetstimmar blir den dragna slutsatsen utifrån undersökningen och efterforskningen att större studios troligtvis inte gagnas av att använda Rigify i sin grafiska produktion. Även fast många likheter finns att dra om utvecklingsprocessen i allmänhet mellan stora studios och små, finns det också betydande

skillnader som gör att Rigify inte kan rekommenderas i dagsläget för stora studior och bolag. Detta betonades särskilt av en av undersökningens deltagare som arbetar på en, med Skövde-mått mätt, större studio. På denna studio arbetar idag drygt 40 personer, varav 3st är anställda animatörer. Denne deltagare är en av dessa animatörer, och menade på att en så pass stor omställning, som hade krävts för de att gå över till både Blender och Rigify i dagsläget, inte kan rättfärdigas varken finansiellt eller kvalitetsmässigt när det kommer till själva produktionen. Delvis skulle studior av denna, och ännu större, storlek i mycket större omfattning behöva ändra och flytta fram sin produktionsfas tidsmässigt, då deras animatörer skulle behöva lägga satt arbetstid på att efterforska, lära sig och experimentera med Rigify och alla dess beståndsdelar, innan de kan använda det till riktig produktion. Därtill behöver studion även veta huruvida Rigify fungerar väl i spelmotorn som spelet utvecklas i. Många av de riktigt stora studiorna, använder egenutvecklade spelmotorer som förmodligen saknar dokumentation och undersökning kring möjligheten till implementering av Rigify-rigg, då varken Blender eller Rigify är särskilt vanliga att arbeta med på så pass stora studior. Dessutom måste de animatörer som ska använda Rigify även utbildas och lära sig producera i Blender, om de inte redan är erfarna inom mjukvaran sedan tidigare. Att börja använda Rigify skulle för stora studior därmed kräva en omställningsperiod som skulle vara svår att tidsuppskatta. Omställningen skulle bli än mer omfattande, kostsam och tidskrävande om studion inte redan använder Blender i sin produktion, utan även måste ställa om från sin befintliga grafiska mjukvara. Studiorna kommer även att behöva räkna in de redan spenderade kostnaderna för sin nuvarande grafiska mjukvara, samt alla eventuella egenutvecklade tilläggswerktyg till denna mjukvara, för att ens kunna börja undersöka om en sådan omställning är aktuell och berättigad på ens ett tidigt idé-stadie.

Däremot är det tydligt att både Blender och Rigify har utvecklats avsevärt under de senaste åren, och har stor potential i framtiden genom sina ständiga uppdateringar. Sålunda går det inte att utesluta att denna slutsats kan vara fel, eller till och med ovidkommande, om exempelvis ett årtionde härifrån.

Vad gäller själva undersökningen och de kvalitativa intervjuerna med deltagarna, skapade de ett kunskapsmässigt ovärderligt underlag för de slutsatser som har kunnat dras utifrån frågeställningen. Genom att undersökningen vände sig särskilt till de som arbetar på en studio inom spelutvecklarbranschen, samt även har viss erfarenhet av att animera i Blender sedan tidigare, kunde viktig insikt insamlas kring vad både studior och enskilda professionella spelgrafiker behöver att ett riggningsverktyg som Rigify ska kunna producera och utföra. Flera av synvinklarna och åsikterna från deltagarna kom utifrån att de både fick utvärdera Rigify som riggningsverktyg, samt att de fick utvärdera en av verktygets förproducerade riggar, vilket då var katt-riggen. Att alla 4 fick använda just en sådan förproducerad rigg, samt att de allihopa fick exakt samma rigg, gjorde att Rigify kunde utvärderas på egen grund och att variabler som skulle bli tämligen svåra att arbeta omkring kunde strykas bort från början. Hade vi ändrat om i katt-riggen med egna lösningar, tillägg och ben- och ledbyten av Rigifys samples hade vi varit tvungna att ta dessa i beaktande till resultaten, samt även förklara vad som har ändrats av oss i katt-riggen för både undersökningsdeltagarna och för läsaren av denna undersökning.

Samtidigt går det inte att förneka att en av undersökningens brister var att en sådan rigg som den medkommande katt-riggen var även vid snabb anblick, alldeles för onödigt invecklad och överflödigt för 3D-modellen av katten och dess grafiska upplösning. Detta var till nackdel för undersökningen då kattmodellen fick röra sig på sätt och vis som var olämpliga för modellen,

och gjorde att flertalet deformationer såg kvalitetsmässigt usla ut vid användning av riggen. Detta påtalades av samtliga deltagare och var ett väntat svar som onekligen påverkade både deltagarnas åsikter om Rigify, och i förlängningen de dragna slutsatserna.

En annan brist i undersökningen handlade om själv undersökningsdeltagarna i sig. Problemet fanns inte hos deltagarna, eller deras kunskap eller yrkesroll, i sig på något sätt. Istället uppenbarade sig ett problem att 3 av de 4 deltagarna arbetade som olika slags grafiker på en och samma studio. Denna studio har endast drygt 12 anställda spelutvecklare, så även fast flera svar och åsikter blev olika mellan de 3 deltagarna, var det en hel del av svaren som blev antingen likadana eller liknande varandras, då de alla tillämpade sina egna synvinklar på utvecklingsprocessen av ett och samma spel. Till detta ska även tilläggas att en till brist ligger i att endast anställda från 2 olika medelstora studior deltog i undersökningen, och att båda dessa studior ligger i Skövde, som trots en stor mängd spelstudior i förhållande till sitt invånarantal ändå räknas som en mindre tätort sett till Sverige och Norden i stort.

Till denna brist kan även räknas att en jämförelse i arbetstid mellan att använda Rigify och att utföra manuell riggning från grunden saknades, då det ej var möjligt att mäta eller uppskatta med tillförlitliga resultat i denna undersökning. Slutsatsen till den frågan blev då endast spekulerande åsikter och synvinklar från både författarna och deltagarna, och kan därför ej räknas som faktabaserat besvarad.

### 6.3 Framtida arbete

Ett bättre, mer värdefullt och tillämpningsbart resultat hade säkerligen kunnat erhållas om deltagarantalet hade varit högre, och om mångfalden mellan deltagarna vad gäller främst arbetsplats och produktion hade varit större. Med dessa brister i själva undersökningen i åtanke går det därför att dra slutsatsen att detta arbete är en rimlig och välmenad början till ett jämförelsevis utforskat område av spelutveckling, men att resultatet och slutsatserna kan komma att bli annorlunda beroende på hur många yrkesutövande spelgrafiker som deltar, samt vilka arbetsplatser de bedriver sitt yrke på.

För vidare arbete kan en ny undersökning utföras allt eftersom Blender ständigt uppdateras, med mån om att Rigify gör detsamma. Meta-riggen som har använts under denna undersökning är av en katt dock finns det fler typer av meta-riggar såsom människa, hund, fågel eller haj. Det finns i nuläget två olika meta-riggar för människan, en enklare och en mer invecklad. Undersökningar kan även genomföras där riggen byggs utav *samples*-leder, såsom meta-riggsmallarna är konstruerade, för att skapa en enklare rigg som är mer lockande för spelstudior. Viktigt inför vidare undersökning är att kunna estimerat tiden det tar att faktiskt generera och modifiera meta-riggen, så att det går att jämföra med arbetstiden som krävs för manuell riggning från grunden.

## Referenser

Autodesk (1998) Autodesk Maya (2019) [Computer Program]. (7 december 2019)  
Tillgängligt på internet: <https://www.autodesk.se/products/maya/overview?term=1-YEAR&tab=subscription>

Baran, I. & Popovic, J. (2007) Automatic Rigging and animation of 3D Characters, *ACM Transactions on Graphics*, vol 26(3), pp.72.

Bharaj, G., Thormählen, T., Seidel, H.-P. And Theobalt, C. (2012). Automatically Rigging Multi-component Characters. *Computer Graphics Forum*, 31(2). pp.755-764. DOI: 10.1111/j.1467-8659.2012.03034.x

Blender (Version 3.0) (1994) [Datorprogram] Blender Foundation. Tillgängligt på internet: <https://www.blender.org/download>

Evan Doody (2016). Advanced Animation Techniques: FK&IK.  
<http://wulverblade.com/advanced-animation-techniques-fk-ik/> [2022-02-08]

Hristov, G. & Kinaneva, D. (2021) A Workflow for Developing Game Assets for Video Games, *3rd International Congress on Human computer interaction, Optimization and Robotic Applications*,

Le, B.H. and Deng, Z. (2014). Robust and accurate skeletal rigging from mesh sequences. *ACM Transactions on Graphics*, 33(4), pp.1-10.

MacDorman, K.F., Green, R.D., Ho, C., Koch, C.T. (2009) Too real for comfort? Uncanny responses to computer generated faces. *Computers in Human Behavior*. 25, ss. 695-710. doi:10.1016/j.chb.2008.12.026

Moore, J. (2018). Sketchfab. [https://sketchfab.com/3d-models/animated-game-ready-calico-cat-ado19702f6e24101a68c01c8f52f7671?fbclid=IwAR3UbsEgjdW9saHJZFo5\\_DsBcElnj7z8xCVOduVBSE3tf-5SFHB\\_hPn3YhM](https://sketchfab.com/3d-models/animated-game-ready-calico-cat-ado19702f6e24101a68c01c8f52f7671?fbclid=IwAR3UbsEgjdW9saHJZFo5_DsBcElnj7z8xCVOduVBSE3tf-5SFHB_hPn3YhM) [2022-02-02]

Pan, J., Yang, X., Xie, X., Willis, P. and Zhang, J.J. (2009) Automatic rigging for animation characters with 3D silhouette. *Computer Animation and Virtual Worlds*, 20(2-3),pp.121-13.

Schlossbauer (2018). Cat. <https://www.thingiverse.com/thing:3317627/files> [2022-01-31]

Östbye, H., Knapskog, K., Helland, K. & Larsen, L.O. (2004) *Metodbok för medievetskap*. Malmö, Sverige: Liber AB. ISBN 91470735