

I)

Am urmarit tutorialul urmator pe care l-am adaptat pentru obiectul meu:
https://www.youtube.com/watch?v=tSABdIpsBEA&t=964s&ab_channel=BlueInversion

Pentru încălțăminte am început cu modelarea tălpii. Am pornit de la un plan pe care l-am modificat astfel încât să fie în concordanță cu imaginea de referință. Am adăugat o serie de muchii (*loop cut*) pe care le-am scalat și translatat. Pentru a obține un rezultat cât mai apropiat de realitate a fost folosit *subdivision surface modifier*. După realizarea formei, am selectat muchiile laterale, apoi am aplicat funcția *extrude* pentru a crea „înălțimea” tălpii, la obiectul obținut utilizând și metoda *crease*.

În continuare, am construit bombeul pantofului: un plan care a fost „tăiat” cu *loop cut*, vârfurile fiind translatare, iar muchiile și fețele scalate și rotite. Au fost folosiți *subdivision surface modifier* și *solidify modifier*, cel din urmă adăugând grosime, adâncime obiectului. Următorul pas a fost modelarea fețelor laterale (din față și din spate). Pentru acestea am utilizat alte două plane distincte, procesul fiind similar cu cel de mai sus.

Pentru limbă am pornit de la un plan căruia i-am crescut numărul de muchii. Apoi, am aplicat transformările obișnuite (translații, rotații și scalări), având grija ca limba să se afle între fețele laterale, respectiv sub bombeul pantofului. De asemenea, au fost folosiți *solidify modifier* și *subdivision surface modifier*. Pentru capută am „extins” o față laterală, găurile fiind realizate cu ajutorul unui add-on - *LoopTools* - cu metoda *circle*.
[<https://docs.blender.org/manual/en/latest/addons/mesh/looptools.html>]

Catarama a fost creată printr-o metodă similară, un aspect diferit fiind dat de utilizarea *mirror modifier* care oglindește rețeaua poligonală (*mesh*) de-a lungul unei axe - în cazul acesta X. Un detaliu subtil este dat de amplasarea unui cilindru și a două paralelipede.

Mai departe, pentru modelarea capselor și țințelor am folosit un cilindru, respectiv un con. Acestea au fost puțin modificate față de primitivele oferite de program și multiplicare. Pentru amplasarea lor pe suprafața pantofului am activat modul *snap during transform*, *snap to face*, *snap with median* (pentru a folosi originea - *origin point*), bifând și funcția *allow rotation to target* (pentru a potrivi orientarea feței). Un aspect important este dat de poziția cursorului 3D care trebuie situat la baza pieselor metalice, originea fiind și ea setată în punctul cursorului.

Rama pantofului a fost construită cu ajutorul unei curbe Bézier - definită de un poligon de control care este memorat și determină geometria curbei. Un detaliu de final este partea laterală a tălpii care a fost realizată cu *array modifier*, *fit type: fit curve*, pentru a multiplica obiectul (un paraleliped) de-a lungul unei curbe. Am adăugat *curve modifier*, iar la obiect am selectat curba dorită. Întrucât talpa nu are o înălțime constantă aspectul era unul neplăcut și am utilizat *lattice modifier* pentru a deforma obiectul de bază conform formei grilajului (*lattice object*).

II)

FBX este un format de fisier folosit pentru a transfera cu usurinta continutul digital dintr-o aplicatie in alta, de exemplu, din Blender in Unity. Formatul a fost dezvoltat de Autodesk care furnizeaza un SDK C++ care poate citi, scrie sau converti fisiere fbx. Acestea se pot reprezenta pe disc atat in binar, cat si sub forma ASCII. Formatul ASCII este un document sub forma de arbore cu identificatori clar denumiti. [<https://en.wikipedia.org/wiki/FBX>]

Practic, documentul este o lista de noduri. Fiecare nod are un identificator de tip (denumirea clasei), un tuplu cu proprietatile asociate (elementele sunt de obicei tipuri de date primitive: float, int, string), o lista care contine nodurile in acelasi format, in mod recursiv. O prezentare mai detalita se gaseste la urmatorul link: <https://code.blender.org/2013/08/fbx-binary-file-format-specification/> .

Un format mai modern este glTF si este de preferat mai ales in cadrul platformelor care il accepta. Salveaza timp deoarece transfera mai bine proprietatile materialelor. De exemplu, pentru utilizarea formatului pentru un transfer in timp real catre o aplicatie client, mai ales pentru cele AR, FBX poate fi mai incet decat un format de transmisie optimizat precum glTF. (<https://www.threkit.com/blog/when-should-you-use-fbx-3d-file-format>) O prezentare pe larg a formatului glTF se poate accesa la urmatorul link: https://docs.blender.org/manual/en/2.80/addons/io_scene_gltf2.html .

III)

Scriptul si fisierul .blend sunt in arhiva.

IV)

2.5 Particle System

Particulele sunt o multime de obiecte emise din obiectele formate din rețele poligonale, de obicei de ordinul miilor. Fiecare particulă poate fi un punct de lumină sau o altă rețea, să fie legată sau dinamică. Acestea pot reacționa la multe influențe diferite și forțe, având noțiunea de durată de viață. Particulele dinamice pot reprezenta focul, fumul, ceața sau alte lucruri precum praful sau vrăji (magic spells).

Pentru realizarea părului, genelor și sprâncenelor am ales următoarea abordare. Particulele de tip păr sunt un subset de particule obișnuite. Sistemele de păr formează curbe care pot reprezenta părul, blana, iarba sau peri (de exemplu, de origine animală). Particulele pot fi randate drept curbe de păr care urmează drumul particulei. Aceste curbe de păr pot fi manipulate în 3D prin pieptănare, adăugare, tăiere, mișcare sau relaxare.

Orice obiect poate purta multe astfel de sisteme. Sistemele pentru păr pot avea până la 10000 de copii pentru fiecare particulă (copiii se comportă mai mult sau mai puțin ca părinții lor). Singurele limitări sunt dotările calculatorului (RAM, CPU, GPU) și răbdarea. [<https://docs.blender.org/manual/en/latest/physics/particles/index.html>]

Astfel, am creat grupuri pentru vârfuri (*vertex groups*), selectând fețele corespunzătoare oaselor parietale, temporale, occipital și frontal. După aceea, am folosit proprietățile pentru particule, ajustând diferiți parametri pentru a obține un rezultat apropiat de cel dorit. Am editat curbele create în modul de editare particule, cât și densitatea lor în modul weight paint, atribuind „greutate” vârfurilor din obiect pictând cu pensule de greutate. În figura de mai jos se poate observa o parte din stadiile prin care am trecut.

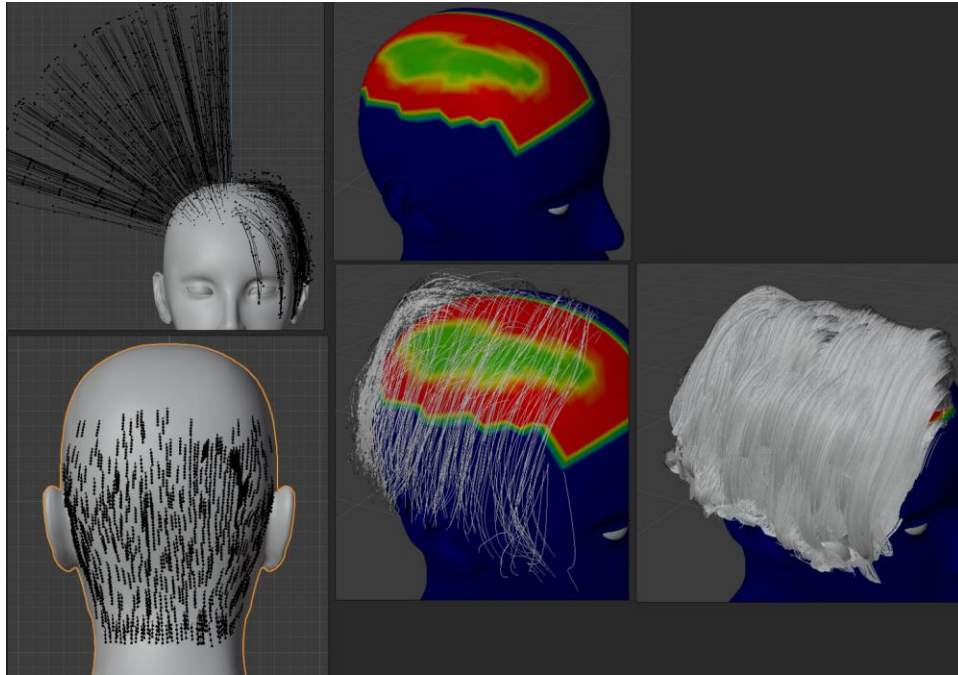


Figura par: Etape păr - vedere particule edit (prima coloană); weight painting (particule fără copii, particule părinți)

Pentru sprâncene am creat un nou vertex group și am pictat zona în modul weight paint. Nu este nevoie de o precizie foarte mare, deoarece se pot face modificări ulterior. În continuare, am adăugat un sistem de particule de păr separat pentru noul grup. Am șters o parte din cele care erau în zone necorespunzătoare și am dat formă sprâncenelor în modul particle edit, ajustând lungimea și curbele potrivite firelor de păr. Procesul se poate vedea în figura următoare.

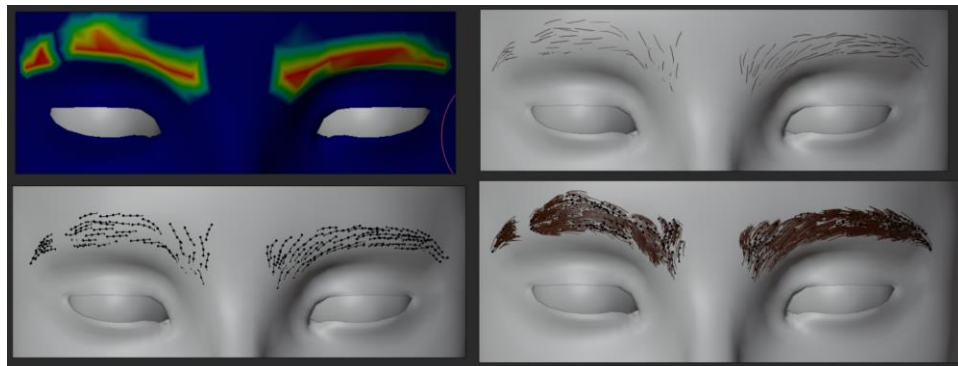


Figura sprancene: Dezvoltarea sprâncenelor

2.6 UV Mapping

Procesul de modelare și sculptare prezentat anterior oferă personajului forma tridimensională, însă nu este suficient. Orice model are nevoie de informații despre culoare și materiale, astfel încât personajul să nu fie doar o siluetă seacă, lipsită de viață. Aceste detalii sunt adăugate printr-un proces numit texturare, unde imaginile (texturile) sunt aplicate pe suprafața personajului. Cu toate acestea, calculatorul nu știe cum să aplice aceste imagini 2D peste o formă 3D, fiind necesară reprezentarea poligoanelor într-un spațiu bidimensional potrivit imaginilor.

UV mapping - sau UV unwrapping - este procesul care face reprezentarea amintită mai sus. Suprafața pe care poligoanele sunt transpuse (mapped) se numește UV map. U și V reprezintă coordonatele axelor unui plan 2D, dat fiind faptul că X, Y și Z sunt folosite pentru un spațiu 3D. Fiecare poligon, muchie și vârf din model sunt reprezentate pe UV map, iar locația acestora de pe map este păstrată în coordonatele UV ale vârfurilor din model. [<https://docs.blender.org/manual/en/latest/editors/uv/introduction.html>]

Există mai multe metode de mapping, cum ar fi diferite tipuri de proiecții. Totuși, pentru modele complexe precum personajele de jocuri sau animații, este recomandat ca unwrapping-ul să fie realizat manual. Acest proces poate fi comparat cu jupuirea pielii unui animal - mai întâi aceasta este tăiată, după care poate fi aplatizată prin întindere, netezire. [T. Terävä, Workflows for Creating 3D Game Characters, 2017] În contextul graficii, acea tăiere este dată de marcarea unor muchii - seams. În figura de mai jos muchiile relevante - seams - sunt evidențiate cu roșu, iar în dreapta se observă UV map.

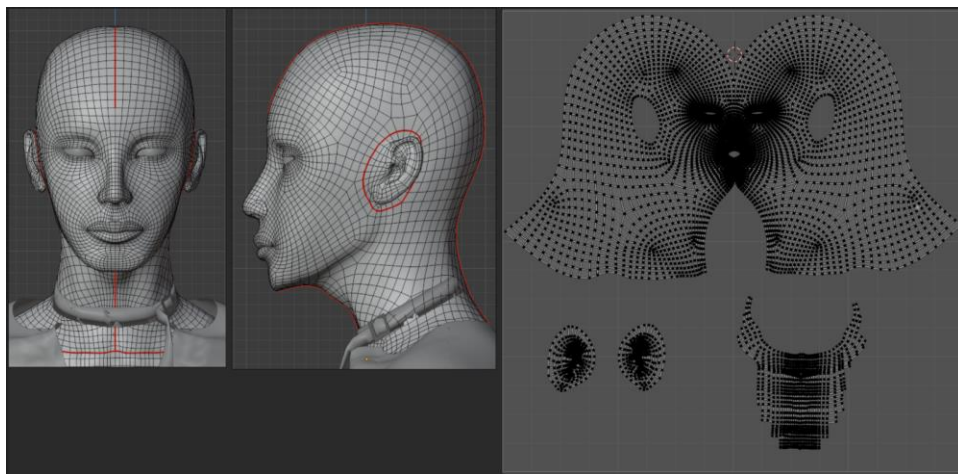


Figura uv mapping: UV mapping

2.7 Texturing

Odată ce fiecare obiect a fost UV mapped, acesta este gata de texturare. Imaginile aplicate pe model pot fi create în diferite moduri - pot fi pictate manual în programe de editare a imaginilor sau direct pe suprafața modelului 3D în cele de specialitate. Pentru majoritatea obiectelor care compun personajul am optat pentru utilizarea software-ului Substance Painter [<https://www.substance3d.com/products/substance-painter/>], întrucât conține toate ustensilele necesare pentru optimizarea dezvoltării texturilor, oferind totodată și o experiență plăcută utilizatorilor.

Sunt mai multe tipuri de texture maps care pot fi folosite, fiecare având un anumit impact asupra obiectului tridimensional. Color map definește culoarea suprafeței modelului - mai exact, reflecția difuză a luminii de pe suprafață. Normal map influențează felul în care obiectul este redat, putând crea iluzia unor detalii care de fapt nu există în geometria acestuia. De asemenea, modifică direcția normalelor de la suprafața modelului 3D, lucru ce permite transferul de detalii de la un obiect cu o geometrie mai densă, către unul cu un număr redus de poligoane - proces numit baking. Prin folosirea baked normal maps un obiect low poly poate părea aproape la fel de detaliat ca unul high poly și necesită o putere computațională aproape nesemnificativă pentru a afișa, a randa modelul. [T. Terävä, Workflows for Creating 3D Game Characters, 2017]

Height maps sunt texturi alb-negru care reprezintă diferența de înălțime de pe suprafața modelului prin diferite intensități de culoare. Acestea indică felul în care vârfurile obiectului sunt deplasate, deci modifică forma modelului. [T. Terävä, Workflows for Creating 3D Game Characters, 2017] Ambient occlusion map imită felul în care lumina ambientală afectează suprafața modelelor 3D. Aceasta creează un aspect de „umbre”, evidențiind zonele din obiect în care lumina ambientală ajunge mai puțin pe suprafață, deci se obține un efect mai verosimil.

Roughness map descrie neregularitățile de pe suprafața obiectului care cauzează difuzia luminii. Lumina reflectată va varia aleator în funcție de rugozitatea suprafeței. Roughness map schimbă direcția luminii, însă intensitatea rămâne constantă. Metallic map este folosită pentru a defini zonele din model care au însușiri similare metalului, luciu reprezentativ (înfățișate cu alb), separându-le de zonele dielectrice (înfățișate cu negru). [<https://academy.substance3d.com/courses/the-pbr-guide-part-2>]

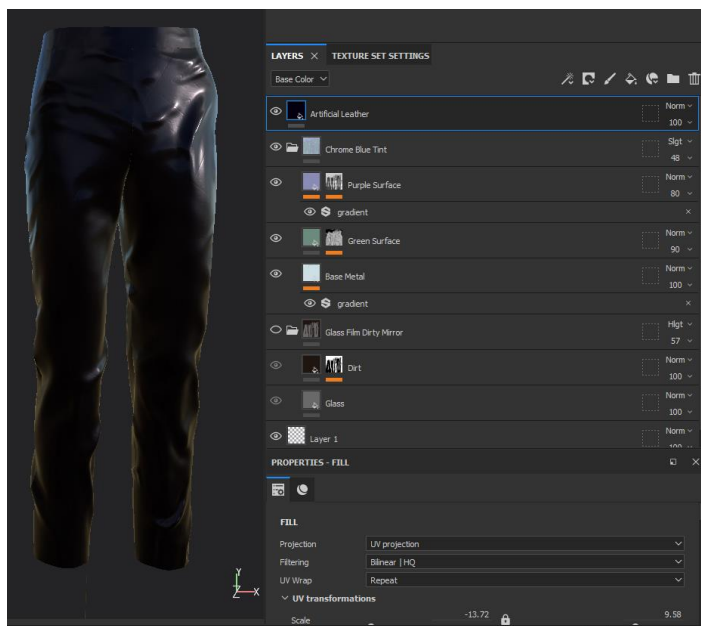


Figura subst text: Texturarea în Substance Painter

În figura de mai sus se observă modul de lucru din Substance Painter: vizualizarea obiectului în 3D și a modificărilor în timp real, cât și partea artistică formată din utilizarea diferitelor layere cu opacități sau moduri de suprapunere distincte. În realizarea ochilor am optat pentru o abordare procedurală, în care am folosit și editat diferiți parametri în cadrul modului shading oferit de Blender. [https://docs.blender.org/manual/en/latest/render/shader_nodes/index.html] Node setup-ul final se poate observa în figura următoare:

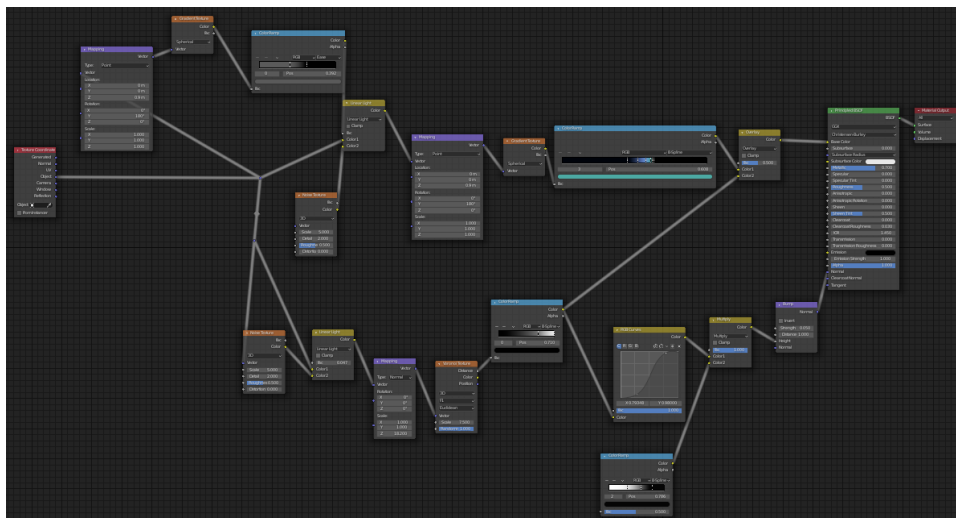


Figura node editor: Texturare procedurală a ochilor