

Grafikus Rendszerek Projektmunka

Mechanikus Billentyűzet modellezés és animáció

Készítette: Pataki Dávid Ferenc

Debrecen, 2024

Tartalomjegyzék

Élképzelés	3
Blender.....	4
Modellek és Textúrák.....	4
Billentyűzet.....	4
Billentyű (Keycap).....	4
Billentyű Textúra	5
Switch.....	6
PCB.....	7
PCB Textúra.....	8
Billentyűzetház.....	9
Súly	10
Hátlap.....	10
Szivacsok	10
Egyéb modellek a jelenetben	10
Asztal	11
Asztalalátét.....	11
Képernyő.....	11
Képernyő Tartó	11
Képernyő szöveg.....	11
3rd party modellek.....	12
Bögre pozicionálása	12
Kábel	12
Scripting	13
SetMesh.py	13
AssignAnimation.py	13
AssignTextAnimation.py	13
Animáció	15
Render + Végző lépések.....	17
Irodalomjegyzék.....	18
Források:.....	18

Élképzés

Modellezés előtt el kellett döntenem, hogy mi is legyen az animáció tárgya.

Organikus élőlényt nem szerettem volna modellezni, mert ahhoz, hogy az jól is nézzen ki, több művészeti tehetség kell, mint ami nekem van. Ezért valamiféle precízebb tárgyra kellett fókuszálnom, amelynek méretei pontosan vannak dokumentálva. Ekkor jött az ötlet, ha már nincs annyi pénzem, hogy vegyek egy mechanikus billentyűzetet, akkor miért ne modelleznék le egyet.

A projektem főpontja egy mechanikus billentyűzet összerakása és annak „használata”.

A billentyűzet maga a [Polaris V1](#) modellen alapul, amelynek az összerakását és alkatrészeinek kinézetét a honlapon lévő dokumentációból és az alábbi videók alapján modelleztem le.

- [Polaris Build Stream By Taeha Types](#)
- [Timelapse : Polaris Mechanical Keyboard Build with Novelkeys Creams by TaeKeyboards](#)

Modell nem tökéletes másolata a képeken látható billentyűzetnek, csavarok például teljesen ki vannak hagyva a modellezésből, de az összerakott billentyűzet eléggé hasonlít a valódihoz.

Amennyiben a videó nem fut le, a [YouTube](#)-on is elérhető (gyengébb minőségben).

Egész projekt, egyéb képek és .docx dokumentáció elérhető [Github](#)-on.



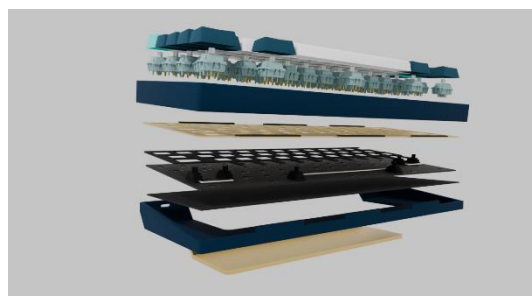
1 referencia



2 saját modell



3 referencia



4 saját modell

Blender

Dokumentáció és képek főbbként a teljes projekt elkészítése után készült, mert elég nehéz úgy dokumentálni valamit, hogy még te se vagy biztos a végeredményben. Ezért valamennyi részletet lehet elhagyni, de próbálom minél pontosabban dokumentálni azt.

Auto-Smooth szinte az összes objektumon van különböző él beállításokkal ezt nem fogom mindenhol dokumentálni.

Ahol egy egyszerű szín van textúrának megadva azt se fogom részletesen dokumentálni.

Modellek és Textúrák

Csak a modellek és textúrázással kapcsolatos információk. Amennyiben egy objektum pl. csontozva van animálás miatt az [Animáció](#) fejlécben lesz részletesen leírva.

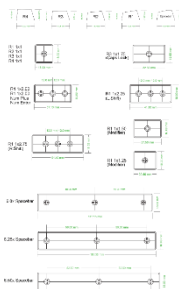
Modellezés során próbáltam minél pontosabban betartani a valós méretezéseket, ezt nagyrészt meg is valósítottam, de párszor elcsúsznak a dolgok.

Főbb elemeknél figyeltem a modell topológiájára, a N-gon csökkentésére és az oldalak normál irányára.

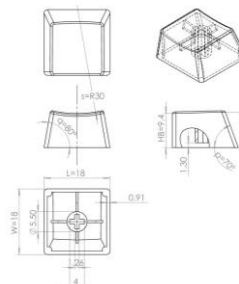
Bíllentyűzet

Billentyű (Keycap)

A billentyűket a Cherry Profil alapján készítettem.



5 Billentyü referencia 1



6 Billentyű referencia 2

A modell egy egyszerű kocka objektummal kezdődött, aminek sarkait a megadott kép alapján pozicionáltam.

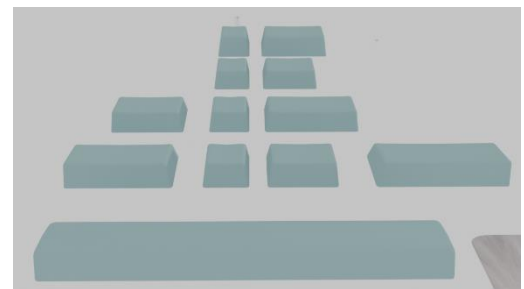
Solidify módosítóval adtam neki vastagságot (1.2mm), majd az alját levágtam.

Billentyű tetején található völgy eléréséhez, 3 részre osztottam a modellt *Loop Cut* segítségével, majd

Proportional editing-el lentebb toltam 0.5mm el.

Ezután az oldalsó és felső oldal éleit elsimítottam 1.3mm-el.

Az elején törölt alsó oldal helyett a megfelelő éleket összekötöttem.



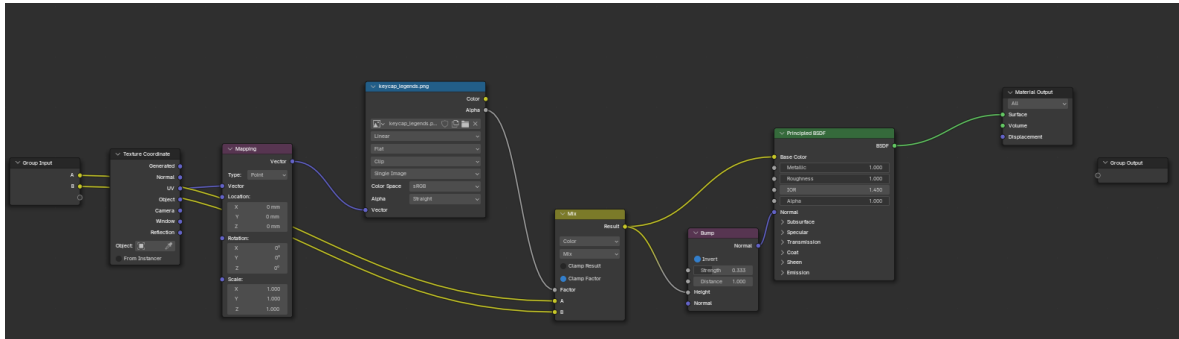
7 Billentyű modell textúra nélkül

Legvégül egy 60° Auto smooth-l lekerekítettem az éleket.

Alján található „törzs” egy egyszerű henger objektum, aminek a két oldalapjára egy-egy téglalap segítségével kivágtam a megfelelő „+” csatlakozást, majd a kapott objektumot Selection to Cursor és Snap To Edge beállításokkal megfelelő helyre igazítottam.

Billentyű Textúra

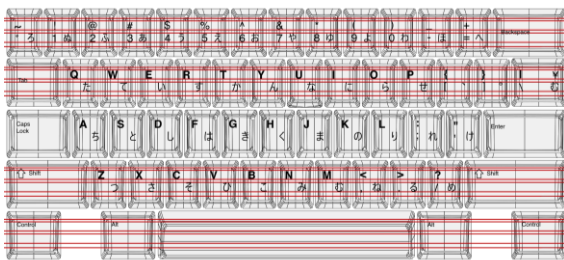
A projektben található 3 különböző szín eléréséhez készítettem egy generikus, képen látható, textúra csoportot. Ennek két szín bemenete van, amellyel lehet változtatni a billentyű háttérszínét és szöveg színét.



8 Billentyű generikus textúra

A generikus textúrához [Paint.net](https://www.paint.net/)-ben létrehoztam egy saját textúrát. A billentyűk körvonalát UV map Project from view gombbal kiszedtem, főleg részeket kihúztam a projekcióból majd, manuálisan ráírtam a megfelelő angol és japán hiragana betűket, azért japán mert betölti az elpazarolt helyet és érdekesebbé/egyedibbé teszi a modellt.

Ezután a képből kinyert színekből egy Mix node-al és egy Bump node-al „ráégetem” a betűket a billentyűkre.



9 Billentyű felirat paint.net-ben
(piros vonalak az egyenesség betartásához van)



10 Billentyű modellek rendezve és textúrázva

Switch

Switch, magyar nevén kapcsoló, lényege lenyomás esetén a PCB-n lévő áramkör bezárása, ezzel kinyerve a megnyomott gomb „betűjét”.

A modellt 2 külön részben modelleztem, teteje és az alja.

Teteje egy kocka amire Mirror módosítóval csak a felét modellezem.

A kiinduló objektum egy kocka, aminek a tetejét bentebb húztam.

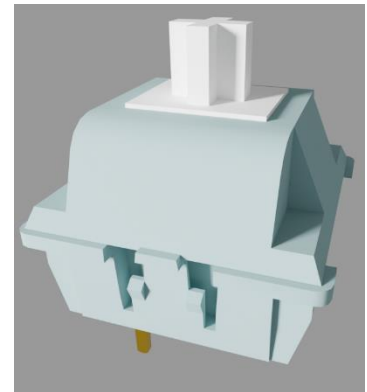
Ezután Loop Cut-al elvágtam alul és az oszlopok helyén, a Loop Cut pontjait Snap To és Edge Slide-al egyenesbe és megfelelő helyre pozícionáltam.

A vágatok eléréséhez megfelelő oldalakat kivágtam és Duplicate Vertices, Snap to és Create Face-el létrehoztam a köztes oldalakat.

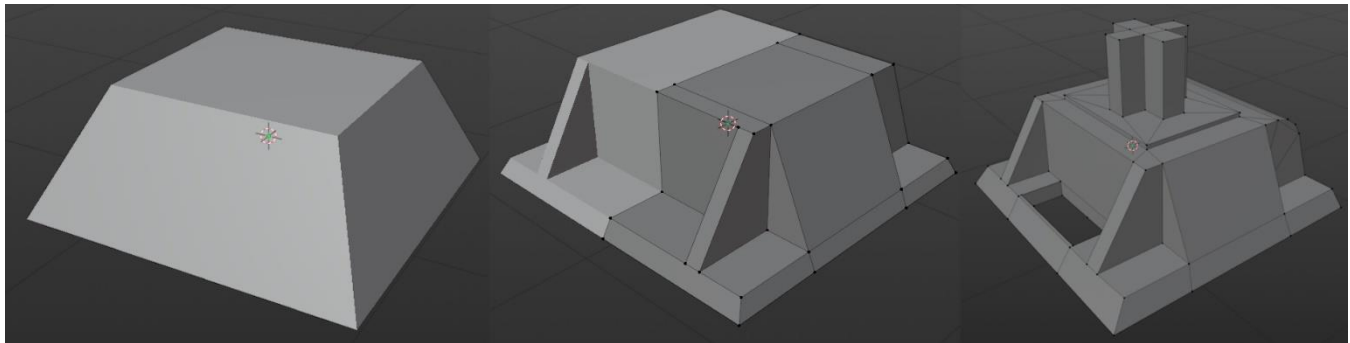
A megfelelő sarkát lekerekítettem. A tükör módosítót elfogadtattam.

Tetején az illeszkedési pont az a Billentyű (Keycap)-ból kivágott „+” -ből készült, csak most kivágás helyett, az inverzét húztam és kötöttem össze.

Ezután manuálisan javítottam a topológián. Az egyik oldalán lévő kivágás később jött a modellezés során, amely egy Insert Face után létrejött oldal kitörlése és azután a modellben lévő lyukak kijavításából állt.



11 Switch modell



12 Switch felső részének modellezése

Az alja egy picikét komplikáltabb.

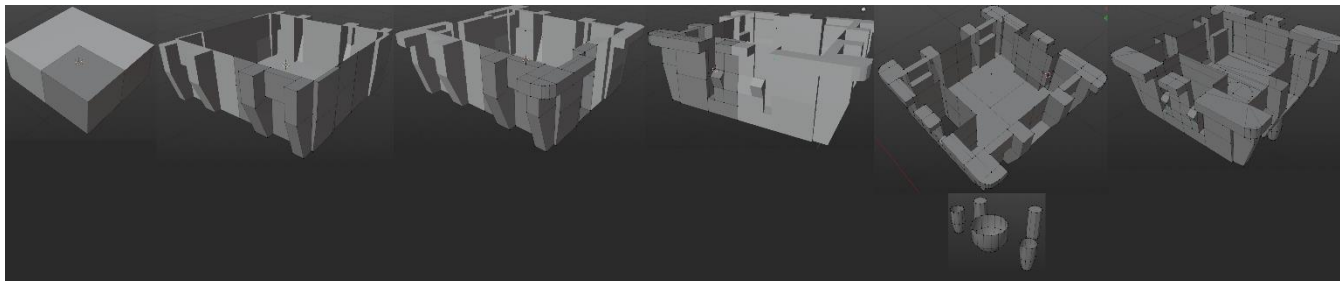
Itt inkább egy sima kockából kifele építettem, kockát Mirror módosítóval letüköröztem 3 fele, hogy könnyebb legyen a modellezés.

Ezután a kiállásokat kezdtem el modellezni, ahol Loop Cut, Subdivide, Extrude, Edge Slide és Move műveletekkel az adott oldalakon kihoztam a geometriát. A tetején lévő platformhoz a kiálló oszlopok megfelelő oldalakat felosztottam és kintebb húztam, majd a sarkára adtam egy kerekítést.

Oldalán lévő kampók szintúgy ilyen módon készült, csak ott még volt oldal, amit bentebb Extrude-oltam és a fölösleges oldalakat töröltem.

Ezután alkalmaztam a módosítót és az egyik oldalán lévő mélyedést kibővítettem, alsó lapját töröltem. Az alján lévő kapcsolódások sima henger/téglalap, aminek az alsó oldala összébb van húzva.

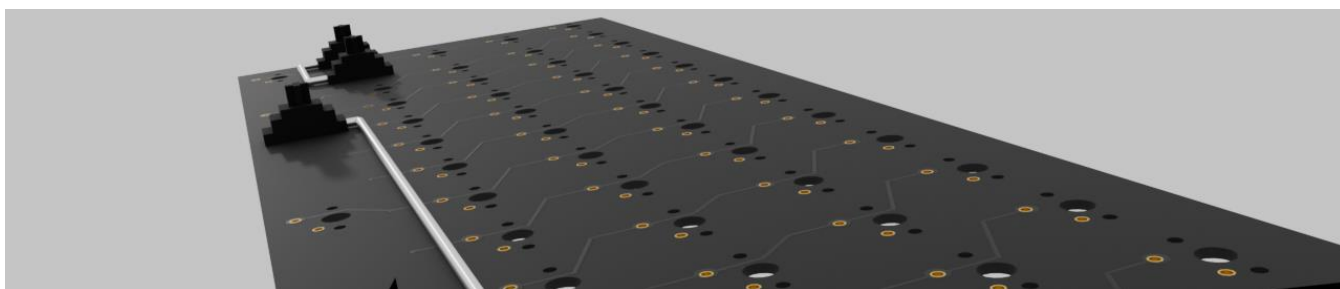
Ahogy a képen is látható a jelenlegi felállás topológiája borzalom, de egy könnyen módosítható modell, szóval ezt manuálisan javítottam Subdivide-al, megfelelő pontok összekötögetésével és a Ctrl+T és Alt+J billentyű kombinációkkal. Eközben a modell arányait folyamatosan javítottam.



13 Switch alsó részének modellezése

Ezután a két oldalt összeraktam és egy objektumként összefűztem. Textúrája 3 egyszerű műanyag és arany szín.

PCB



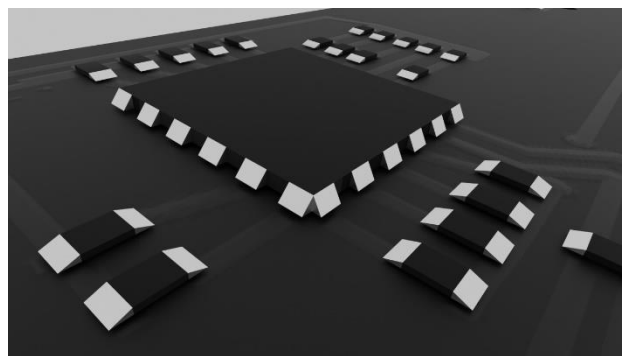
Referencia kép nincs.

Egyszerű téglatest, lyukak eléréséhet a Switch objektumnak levágtam az alját és kiegészítettem úgy, hogy az érintkezési pontok kör alakúak legyenek. Ezután azokat a sorbarendezett billentyűk alá helyeztem és egy objektumként kezelve Boolean módosító segítségével kivágtam a téglatestből. Az N-Gon csökkentése érdekében Ctrl+T majd Alt+J billentyűkombinációkkal N-Gon-ból, háromszöget, majd négyszöget csináltam.

Ezután az érintkezési pontokhoz Insert Face-el extra részletet adtam a későbbi textúrázás miatt.

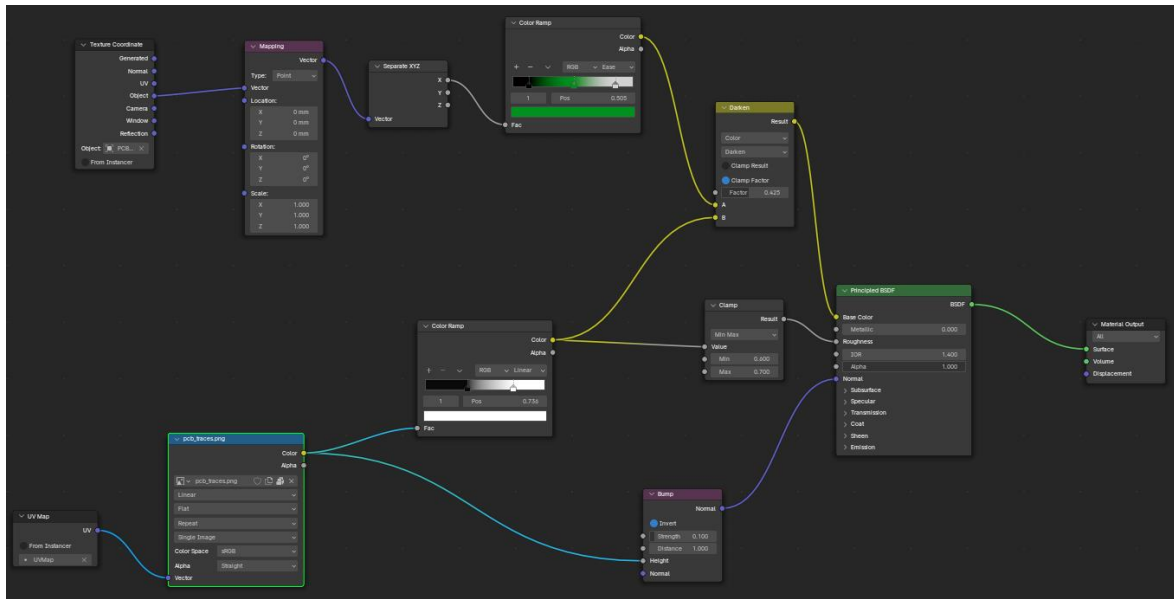
A PCB felszínére készítettem egyszerű „ellenálót” (resistor) és minden lyuk mellé be duplikáltam azt. Ez csak egy téglatest, aminek két oldalán van egy-egy háromszög. Létezik pár más chip is a modellen, ez mind ugyanaz a koncepció csak több duplikált háromszöggel, csatlakozási ponttal. *Csak távolról kell jól kinéznie.*

Type-C csatlakozó egy téglatest, amelynek két oldalsó oldala 50% le van kerekítve, Insert Face-el és Extend-el kivágtam a közepét. Végül a közepére helyeztem még egy téglatestet.



14 Chippek a PCB-n

PCB Textúra



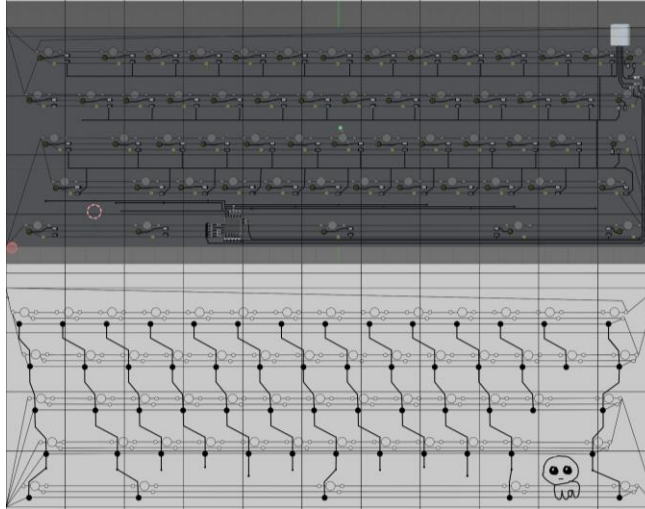
Ahogy a képen lehet látni a textúra két nagy részre van osztva, a felső, amely animáció szempontjából fontos és az alsó, amely általános kinézet.

Felső részben létezik egy téglalap („PCB color control”) objektum, amely mozgatásával változik a modell alapszíne. Ezzel egy egyenletes színátmenetet tudunk változtatni Fehér, Zöld és Fekete alaplap szín között.

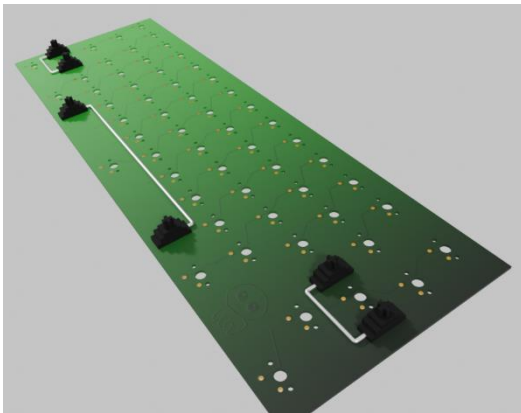
Alsó részben megint egy saját textúrát alkalmaztam, amelyben a PCB-re rárajzoltam egy áramkört, majd Bump node-al, Color Ramp-el és PCB-alapszínével egy Mix Color/Darken Node-al hamis mélységet, sötétebb színt és egy kis csillogást adtam az áramkörnek.

Az áramkör, vonalai nem valóságű, nem vagyok alaplap tervező se elektronikus, úgy rajzoltam rá, hogy jól nézzen ki.

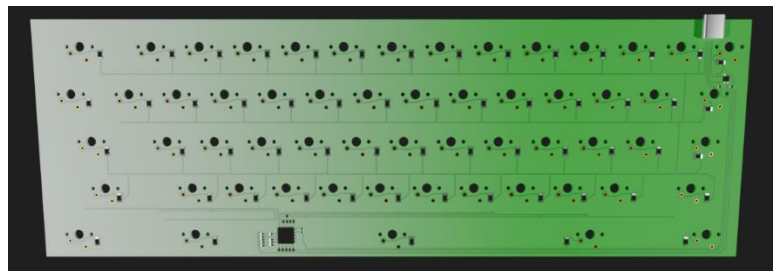
Még létezik három szín a modellen, egy egyszerű fekete a chippeknek, egy ezüstös a chippek csatlakozóinak és a Type-C csatlakónak és egy arany szín a Switch-ek csatlakozási pontjainál.



15 PCB Trace paint.net-ben



16 Kész PCB eleje



17 Kész PCB hátulja

Billentyűzetház

Teteje egy üres téglalap, amit megvastagítottam lefele és alkalmaztam rajta egy Solidify módosítót.

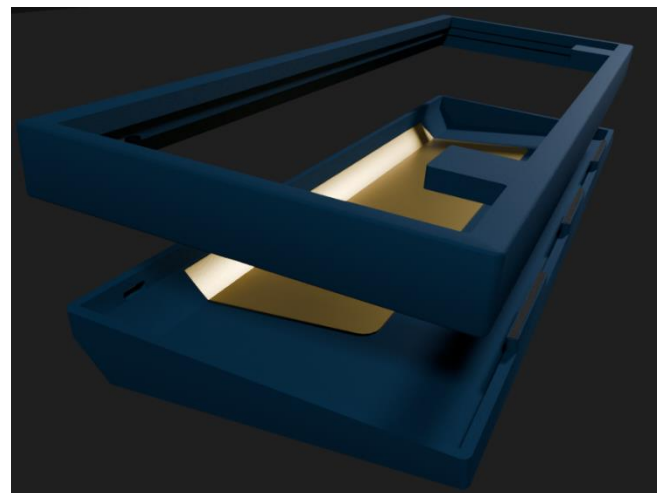
Belső lépcsős geometriához kettő Loop Cut-ot alkalmaztam, egyiket Edge Slide-al behúztam a másikba, majd fel skáláztam.

Kiálló lapokat a megfelelő oldal felszeletelésével és Extrude-olásával értem el

A hátulján lévő kivágás pedig egy Boolean módosító egy kerekített kockával.

Végül megfelelő oldalakat lekerekítettem.

Alja hasonló módon készült, itt egy téglalap oldalait Extrude-oltam a test körvonalára és arra alkalmaztam a Solidify-t.



18 Billentyűzház modell

Textúrája egy Csiszolt alumínium-nak kezdődött el, de végül inkább valami matt textúra lett. Ezt egy Musgrave Texture elnyújtásával értem el amit Bump map-ként használok.

Súly

A súly alapból az alsó ház része volt, ahol, amikor még nem alkalmaztam a Solidify módosítót, Subdivide-okkal kivágtam a megfelelő formát sarkát lekerekítettem. Módosító alkalmazása után szeparáltam a két objektumot egymástól és alkalmaztam rajta a módosítót.

Textúrája egyszerű arany szín.

Hátlap

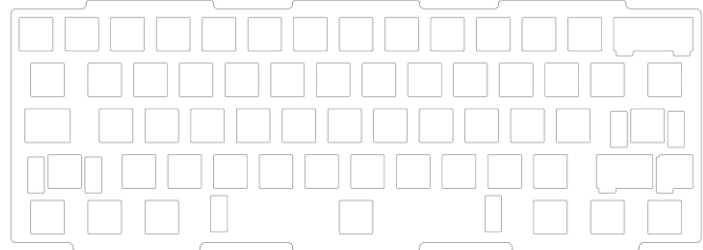
Az eredeti billentyűzet [honlapján](#) található egy körvonal, ez megkönnyítette ennek az objektumnak a modellezését.

Ugyanazt a trükköt alkalmaztam, mint a [PCB](#) létrehozásánál, csak most egyszerű kockákat (aminek sarkai le voltak kerekítve) vágtam ki a téglatestből.

Miután ez meg volt, manuálisan javítanom kellett a kivágásokon, mert a Boolean módosító itt nem mindenhol kötötte össze a téglatest tetejét az aljával.

Az oldalán lévő kivágásokat az oldalak Subdivide-olásával majd a megfelelő lapok Extrude-olásával értem el.

Textúrája sima réz/arany szín, az oldalán lévő szürkesség pedig szivacs, amiről a következő fejlécbe beszélek.



19 Hátlap referencia

Szivacsok

A jelenben két hangtompító „szivacs” van jelen. Ezeknek a lényege a való világban a hang akusztika változtatására van.

Az alsó az egy sima téglalap.

Felső pedig igazából a [Hátlap](#) amelyből Knife Tool-al levágtam egy kis részt.

Textúrája egy sűrű zaj, amellyel sötétítem a szürke alapot fekete pontokkal, sajnos ez a részlet a végső Render-en nem jön át, mert a zajtalanításnak pont ez a lényege.

Egyéb modellek a jelenetben

Modellek, amelyek nem konkrétan része a billentyűzetnek.

Asztal

Téglatest, amely alsó lapja le van skálázva, négy sarka lekerekítve.

A láb az egy henger, amelyet Loop Cut és Skálázással tetején és alján változtattam. Mirror módosítóval az asztal négy sarkára pozicionáltam.

Asztalon látható textúra fehér fát ábrázolna, amelyet egy Noise Textúra megnyújtásával és az ebből kinyert adat szín módosításával és a Bump Node hamis magasság adásával értem el.

Asztalalátét

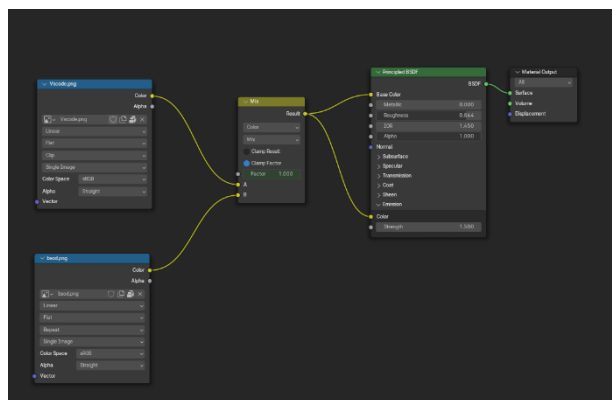
Téglalap, amelynek hosszabb oldala Subdivide-olva van későbbi animáció miatt. Insert Face-el adtam keretet neki, majd Extrude-oltam lefele egy kis mélység eléréséért. Végül sarkait és éleit lekerekítettem.

Textúraként felhasználtam, a netről szedett, [Fabric 039](#)-et, majd a modell tetejét UV Unwrap-eltem és [Reddit](#)-ről szedett háttérképet használok fel kinézetre.

Képernyő

Téglatest, amely Insert face és Extrude kombinációval létrehoztam a kijelzőt. Ezután adtam alul neki egy kis vastagságot és az egész objektumot Proportional Editing-el, két oldalán, elgörbítettem.

Textúrája kívül egy sima fekete műanyag. A képernyő pedig kettő kép, ami között Mix Node-al tudok majd az animálás során váltani. Adtam neki egy Emission beállítást, hogy majd az animáció végén megvilágítsa a jelenetet.



20 Kijelző textúra

Képernyő Tartó

Én is ilyet használok való életben, az alapján modelleztem.

Egy sima henger, aminek az alját Loop Cut-tal megvastagítottam, hátulján lévő pár pontot töröltem és összekötöttem, hogy egyenes legyen majd vastagítással létrehoztam a talpakat. Csak azt a rész modelleztem le, ami látszik is.

Képernyő szöveg

Ez az objektum teljesen videó alapján készült mert Geometry Node használatán kívül más lehetőséget nem találtam.

Videó: [Typewriter Effect Animation with Geometry Nodes | Blender Tutorial](#)

Valamiért ilyen megoldással más Módosítók már nem működnek szóval nem tudtam Shrink Wrap-elni a monitorhoz, ezért próbáltam minél pontosabban beforgatni azt, hogy ez ne legyen észrevehető a végső animációban.

3rd party modellek

- [Coffee Mug By JamesDmodels](#)
- [Logitech G603 by Juxius](#)

Miért használtam mások modeljeit?

~~Mert se időm, se kedvem nincs meg többet modellezgetni.~~

Mert a felhasznált modellek csak háttér elemekként szolgálnak és nem része a jelenet fő hangsúlyának.

Bögre pozicionálása

A bögre fölé beduplikáltam pár [Switch](#) objektumot és adtam nekik Active Rigid Body fizikát Box Collision Shape-el, a bögrét magát pedig beállítottam Passive Active Body-ra Mesh Collision Shape-re. Mind két objektumon beállítottam a Collision Sensitivity-t 1mm-re. Miután belesett az összes Switch, elforgattam a bögrét, hogy az asztalon legyen, végül kiégettem az animációt és Ctrl+A Apply Visual Transform-al kiszedtem az utolsó képkockát, ezután már törölhettem a fizikát minden objektumról. Ötlet a [r/MechanicalKeyboards/](#) subredditől jött (eredeti postot törölték).

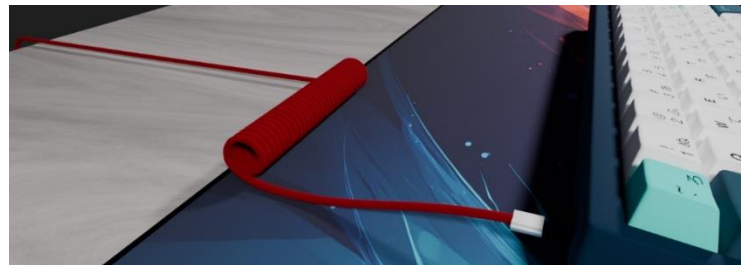


21 Referencia

Kábel

Sajnos ez az objektum nem jelenik meg a végső animációban, mert nem tudtam rendesen leanimálni, hogy jól is nézzen ki az animáció többi részével.

Beépített Blender kiegészítések közül bekapcsoltam az Add Curve: Curve Tools és az Add Curve: Extra Objects-et. Ezzel a két kiegészítővel létre lehet hozni egy Logaritmikus görbét, ami spirálzik valamilyen irányba. Ezután manuálisan meghosszabbítottam a két végét és görbe beállításoknál adtam neki Bevel-lel vastagságot. Csatlakozó a végén az a [PCB](#) csatlakozó leskálázva. És adtam neki egy Follow Path kényszerítést, aminek a Offset factor-át beállítottam 1-re. Ezzel elérve, hogy akárhogy mozgatjuk a görbe végét, a csatlakozó mindig a görbe végén marad.



22 Kábel modell

Textúrája megegyezik a [Asztalalátéttel](#) csak nem adtam neki háttérképet, hanem csak egy egyszerű színt és a Fabric textúrát felhasználtam a Bump Node mellett még egy Mix Color/Darken Node-al is, amivel adtam neki plusz textúrát.

Scripting

Első kettő kód inkább a modellezés megkönnyítésére van, utolsó viszont dinamikusan használható az animáció végén lévő gépelés szekció megváltoztatására.

SetMesh.py

Egyszerű script, amellyel (megfelelő kommentek kiszedése esetén), egy bizonyos collection vagy kiválasztott objektumok mesh adatát újra- vagy szétválaszthatjuk.

Polygon szám csökkentése érdekében X: -1101 mm Y: 87 mm Z: 3 mm környékén található az eredeti objektumok, miután elhelyeztem a [Switch](#)-eket megfelelő helyükre ezt a scriptet alkalmazva, 35625 csúcsot megspóroltam a jelenetből. Ha Blender-ben lehetne lehetőség *Instance*-eket (Objektum, aminek ugyanaz a mesh adata, mint egy másinak) különböző módon textúrázni, akkor a [Billentyűk](#)-en is ezt használtam volna, megspórolva 9918 csúcsot, de sajnos erre nincs lehetőség, ezért van a scriptben olyan sor, amely egyedivé teszi a Mesh-t.

AssignAnimation.py

A ~235 képkockánál látható Switch-ek gyors megjelenítése sorban különböző időközönként.

A Switch Collection objektumait sorbarendezelem X és Y koordináták alapján és 1 képkockás eltéréssel, beállítom, hogy látható legyen-e a végső Render-ben.

AssignTextAnimation.py

A legkomplexebb script a projektben.

Lényege, hogy a megadott paraméterek alapján a képernyőn megjelenített szöveg szinkronizálva és animálva legyen a billentyű gombok lenyomásával.

Működése:

Paraméterek (majdnem) tetszés szerint állíthatóak. Miután megszerezzük a megfelelő értékeket a Blender objektumokból, a képernyőn látható szöveget beállítjuk és a megjelenített hosszúságát 0-ra állítjuk, hogy még ne látszódjon semmi.

Ezután az előző animációkat töröljük a billentyűkről és mindegyikre rárakunk egy üres animációt.

A megjelenített szövegen manuálisan kell kitörölnünk az animációt mert maga az animáció nem az objektumon, hanem a *Geometry Node* egyik értékén van és erre nem találtam erre megfelelő metódust. Szóval kiválasztjuk a „ConsoleText” objektumot és a képernyő alján található Timeline-ban kitöröljük a megfelelő keyframeket.

A projectbe létezik 3 darab egyszerű kocka animációkkal rajta, teljes gomb lenyomás, csak lenyomás, csak felengedés.

Ezután minden betűn végig megyünk, ha speciális (csak négyet raktam be, de egyszerűen kiegészíthető) vagy nagy betű akkor a Bal Shift gombot is animáljuk.

Lépések:

- ha speciális karakter vagy nagy betű akkor shift le + karakter lenyom + shift fel sorozatot animálunk
- megfelelő animációt lemásoljuk és offsetAction-al minden képkockáját eltoljuk az adott képkockára
- az animált objektumra appendAction-nel hozzáfűzzük az animációt
ezt úgy érjük el, hogy a hozzáadott animáció összes képkockájának az adatait egyesével hozzá adjuk a másik objektum animációjához
- ezután a megjelenített szöveg látható hosszúságát megnöveljük egyel és képkockát rendelünk hozzá

Az appendAction függvény erősen, de nem teljesen, [ChatGPT](#) generált, mert valahogy én vagyok az első, aki ilyen problémába ütközne és mindenki a Non Linear Animation panellel akarja ezt megoldani, amihez még jobban nem értek.

Az eredeti generált kód itt látható.

```
1 import bpy
2
3 def combine_actions(action1_name, action2_name, new_action_name):
4     # Retrieve the two actions
5     action1 = bpy.data.actions.get(action1_name)
6     action2 = bpy.data.actions.get(action2_name)
7
8     if not action1 or not action2:
9         print("One or both actions not found")
10        return
11
12    # Create a new action
13    new_action = bpy.data.actions.new(name=new_action_name)
14
15    # Combine the F-Curves from both actions into the new action
16    for fcurve in action1.fcurves:
17        new_fcurve = new_action.fcurves.new(data_path=fcurve.data_path, index=fcurve.array_index)
18        new_fcurve.keyframe_points.add(count=len(fcurve.keyframe_points))
19        for i, keyframe in enumerate(fcurve.keyframe_points):
20            new_fcurve.keyframe_points[i].co = keyframe.co
21            new_fcurve.keyframe_points[i].interpolation = keyframe.interpolation
22
23    for fcurve in action2.fcurves:
24        new_fcurve = new_action.fcurves.find(data_path=fcurve.data_path, index=fcurve.array_index)
25        if not new_fcurve:
26            new_fcurve = new_action.fcurves.new(data_path=fcurve.data_path, index=fcurve.array_index)
27            new_fcurve.keyframe_points.add(count=len(fcurve.keyframe_points))
28            for i, keyframe in enumerate(fcurve.keyframe_points):
29                new_fcurve.keyframe_points[-len(fcurve.keyframe_points) + i].co = keyframe.co
30                new_fcurve.keyframe_points[-len(fcurve.keyframe_points) + i].interpolation = keyframe.interpolation
31
32    # Optionally assign the new action to the active object
33    if bpy.context.object:
34        bpy.context.object.animation_data.action = new_action
35
36    print(f"Combined actions '{action1_name}' and '{action2_name}' into new action '{new_action_name}'")
37
38    # Example usage
39    combine_actions('Action1', 'Action2', 'CombinedAction')
40
```

23 prompt: "Bpy combining two actions into one without NLA"

Ahogy látható a második ciklus az nagyon hasonlít az enyémre, de a kód többi része az saját gondolataimból és az [Irodalomjegyzék](#)-ben megjelölt oldalakról jött.

Animáció

Minden objektumot próbáltam csak akkor megjeleníteni a Show in Renders opcióval amikor már láthatónak kéne lennie.

A Képkocka az Keyframe-t jelentené, a Insert Location pedig Insert Location and Rotation-t.

Animáció egy Asztalalátét legöngyölésével kezdődik.

Magát az asztalalátétet legöngyölését egy görbével értem el, ami egy Logaritmikus befelé gördülő görbe („Deskmate Control”), aminek egyik vége meg van nyújtva. Ezután magára az alátétre raktam egy Curve módosítót (nem ajánlom, változtatja a modell méretét és kínlódás beállítani). Miután ez megvan, a görbe mozgásával elérjük, hogy a modell kövesse a görbe görbületét.

A kamerát úgy animáljuk, hogy rárakunk egy Limit distance kényszerítést, miután már nem akarjuk tovább követni azt az animálás közbe, berakunk egy Insert Keyframe by Visual Location-t és a kényszerítésnek beállítjuk az Influence opcióját 0-ra. Ezután a kamerát Insert Location képkockákkal mozgatjuk a megfelelő pozícióba.

Következő animáció a PCB körüli forgást.

Maga a PCB lehozása egy sima Location Keyframe.

A kamera itt egy Empty („Camera Control”) objektumnak a gyermeke, ami inheritálja a szülő transzformációit, ez a PCB közepére van pozícionálva. Ennek forgatásával a kamera az objektum a PCB körül forog.

A színváltozást egy másik Empty („PCB Color Control”) objektummal érem el, amely az X koordinátája alapján és a texturázás szabályai szerint változtatja a PCB színét.

Végül a PCB-t lehelyezzük az asztalra és a kamerát is megfelelően pozícionáljuk.

Ezután a Szivacsot és a Hátlapot animáljuk.

A szivacsnak adtam egy csontozást, (Curve módosítóval akartam elsőnek, de ahogy írtam ez az opció rémálom), manuálisan adtam neki egy görbületet és ahogy Insert Location mozdulat közbe mozog lefele, a csontokat is egyenesbe rakom.

A Hátlapot megfelelő helyre mozgatjuk és ugyanazt a trükköt használjuk mint a PCB kamera körüli forgatásnál, azaz létrehozunk egy gyerek/szülő kapcsolatot egy Empty („Backplate Control”) objektummal amit a hátlap sarkára pozícionálunk és ezt forgatjuk és mozgatjuk.

Maga a belső felépítés utolsó lépése a Switchek berakása.

Itt egy darab Switchet manuálisan Insert Location-nel leanimálunk, hogy a nézőnek egy gondolata a következő gyors sorozatban történő eseményekről.

Ezután a AssignAnimation.py Script lefuttatásával a Show in Renders opciót kapcsolgatom. Ezzel egy gyors, kényeztető animációt hozunk létre ahogy megjelennek az objektumok.

Ideje belerakni a Billentyűzetházba.

Itt maga a ház mozgását egyszerű Insert Location képkockák sorba rakásával érem el.

Maga a belső szerkezet megint a Kamera/Hátlap animációs trükkön alapul, Empty („Assembled PCB Control”) objektum, amihez hozzá van rendelve minden és ezt mozgatjuk és forgatjuk.

A szivacs mozgása szintúgy egy már használt technika, ugyanaz, mint a másik szivacsnál, csak most a csontozás az objektum közepéről indul és kifelé megy.

Ház alját és majd később az egész billentyűzet, billentyűk nélküli, mozgatásához ismételten hozzá rendelek egy Empty objektumot. („Case Bottom Control” és „Keyb Without Keys Control”)

Már csak a [Billentyűket](#) kell berakni a megfelelő helyükre.

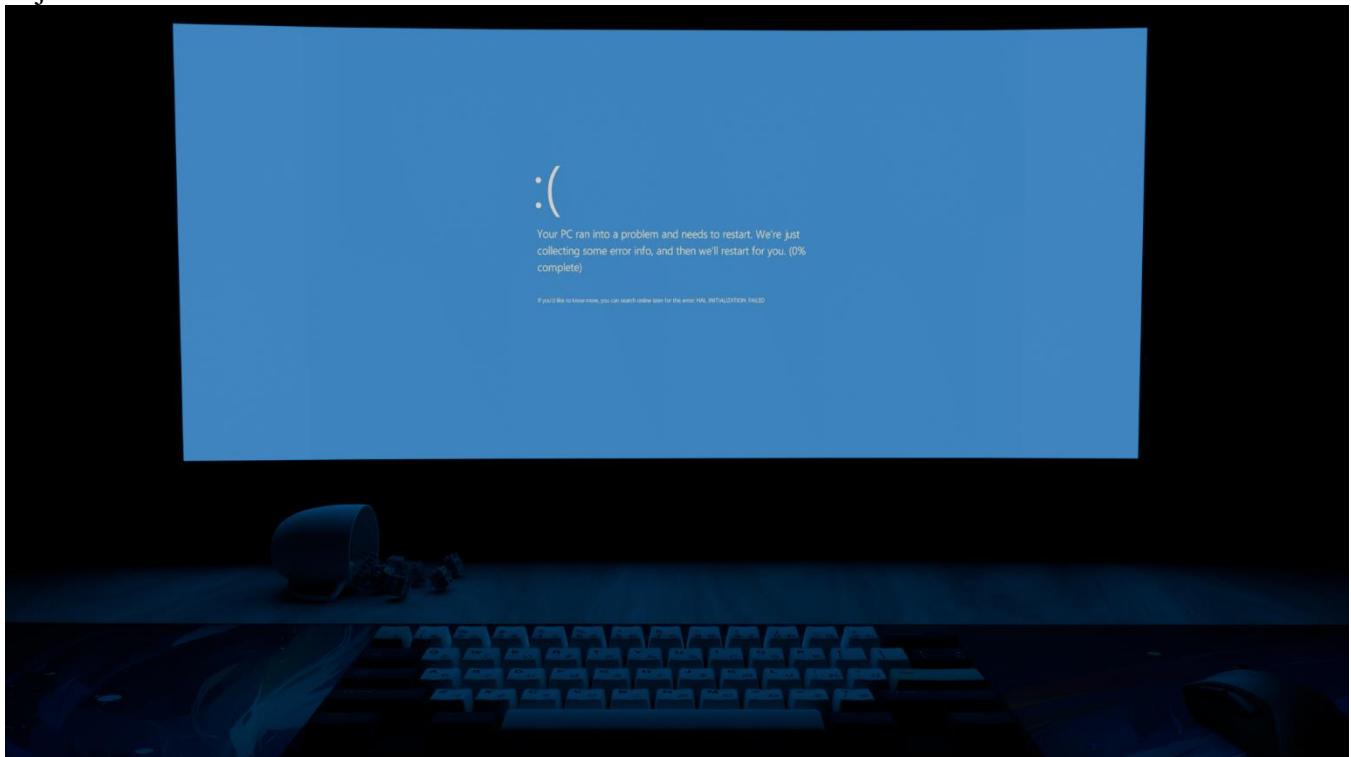
Itt létezik egy síkfelület („Wave animation”) a jelenet fölött, amely *Subdivide*-olva van a következő lépések miatt. Ezen az objektumon van egy *Wave* módosító, ami csak egyszer fut le és ad egy hullámváz effektet. Ezután a billentyűket hozzárendelem ehhez a felülethez *Vertex* módban, ezzel azt érjük el, hogy nem az egész objektumhoz és annak mozgásához rendeli hozzá a gyerekeit, hanem a mesh pontjaihoz és annak helyéhez. Ezzel elértük, hogy magán a Billentyűken legyen ez a hullámváz animáció, animáció közben *Insert Position*-nel berakjuk a megfelelő magasságra.

Ezután a kamerát fentebb emeljük és közbe beállítjuk, hogy lehessen látni a végső renderben az Egeret, monitort és a tartóját.

Utolsó előtti animáció a Kiírtatás, ezt ahogy a [AssignTextAnimation.py](#) fejrészben leírtam, teljesen kód készíti. Eközben az egész jelenet háttérszínének az erőssége folyamatosan csökken a következő „probléma” miatt...

Kékhálál.

A [Képernyő](#) Textúráján lévő *Mix node* értékét átállítom, hogy csak a második kép jelenjen meg, ami egy Windows-os kékhálál képernyő kép. A szöveget eltüntettem a renderből és a jelenet fényeit el sötétítem teljesen.



Render + Végő lépések

A 3 Render motort párhuzamosan használtam.

- „Workspace”: Animáció és annak sebessége ellenőrzésére.
- „EVEE”: Fényezés + Textúrák változása animálás közben.
- „Cycles”: Pillanatképek textúra ellenőrzésére vagy egész animáció tesztelése nagyon alacsony mintavételi rátával és felbontással

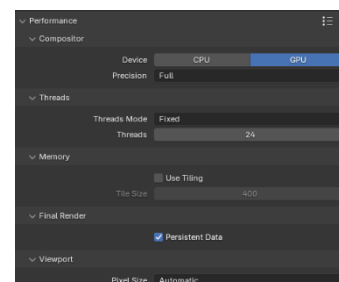
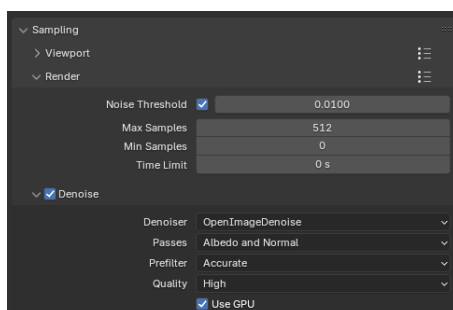
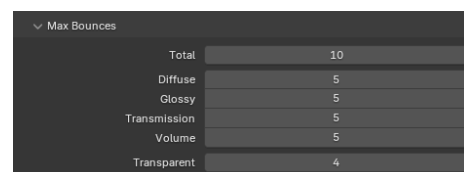
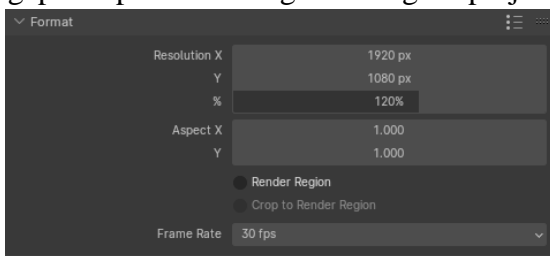
Végő videót nem én Rendereltem ki, mert szegény i5-3470 + GTX 1050 kombinációmnak, a képen látható, képkockát 3 perc 11 másodpercbe tellett megcsinálni, vannak könnyebb jelentek, ahol 1 perc 9 másodpercbe telik ki-Renderelni.

Viszont szeretném másra is használni a gépem, mint egy szoba melegítő 13-41 órán keresztül.

Igaz, lentebb vehettem volna a minőséget vagy Eevee-t is használhattam volna, de annyi időt töltöttem a projektbe, hogy szeretném is látni a végleges maximum minőségi eredményt.



Szóval megkértem középiskolás barátomat, hogy Renderelje ki nekem. Az ő (i7-8700K + RTX 3060) gépe 40 perc alatt megette az egész projektet, az alábbi beállításokkal.



Irodalomjegyzék

- <https://docs.blender.org/api/current/index.html>
- <https://docs.blender.org/manual/en/latest/>
- <https://devtalk.blender.org/>
- <https://blenderartists.org/>
- <https://blender.stackexchange.com/>

Források:

- <https://ai03.com/projects/polaris/>
- <https://www.turbosquid.com/>: 3D modellek
- <https://3dtextures.me/>: szövet textúra
- <https://chatgpt.com/>: kód generálás