

Razvoj filtra za obogatitje resničnosti

Miha Malenšek, Blaž Pridgar, izr. prof. dr. Narvika Bovcon

Univerza v Ljubljani, Fakulteta za računalništvo in informatiko, Večna pot 113, 1000 Ljubljana

E-pošta:

mmalen98@gmail.com

bp4769@student.uni-lj.si

narvika.bovcon@fri.uni-lj.si

Development of an Augmented Reality (AR) Experience

***Abstract.** This article describes the development of an Augmented Reality filter, or "lens", designed for use on the social network app Snapchat. We started the project at the beginning of the COVID-19 pandemic, and since we did not find any relevant filters online that would warn of the dangers of spreading the virus, we opted for a two-part thematic filter that would remind users in a fun way to use a protective mask and keep the safety distance in public. On the front phone camera with the help of face tracking algorithms the lens projects a 3D protective mask over the user's nose and mouth, which can also be removed and then raised again by raising the eyebrows. On the rear phone camera our mascot named "Safey" appears in front of the user and keeps the safety distance from the user, even if one approaches him. We created all the 3D models in Blender and developed the lens in Lens Studio, the dedicated platform for creating lenses for Snapchat.*

1 Uvod

Članek opisuje razvoj filtra za obogatitje resničnosti, ki je namenjen uporabi na družbenem omrežju Snapchat. V članku predstavimo izhodišča projekta, razvoj ideje, uporabljene tehnologije, implementacijo filtra in načrtovanje oblikovalske rešitve.

Projekt je bil zasnovan ob začetku pandemije novega korona virusa: ker na spletu ob začetku projekta nismo zasledili nobenih aktualnih filtrov, ki bi opozarjali na nevarnosti širjenja virusa, smo se odločili, da izdelamo tematski filter, sestavljen iz dveh delov, ki bo uporabnike na zabaven način opomnil na upoštevanje varnostnih ukrepov. Prvi del filtra opozarja na uporabo zaščitne maske, drugi pa na upoštevanje varnostne razdalje v javnosti. Dvodelni filtri so tudi sicer zelo popularni, saj se lahko en del uporablja na sprednji telefonski kameri, drugi pa na zadnji [1]. Izhajali smo iz ideje o AR izkušnji, ki bi simulirala nameščanje varnostne maske na obraz, kar se odlično poda uporabi sprednje (selfie) telefonske kamere, želeli pa smo ustvariti tudi zabavno maskoto, ki bi filter naredila še bolj všečen. Ob razvijanju filtra smo se oprli predvsem na naše dosedanje izkušnje [2] ter na primere filtrov, ki ljudi osveščajo o boleznih, npr. o virusu HIV [3].

Končni cilj projekta je bil javna objava filtra in s tem osveščanje ljudi glede varnosti med pandemijo ter seveda pridobivanje izkušenj za prihodnje projekte.

2 Razvoj AR

Obogatena resničnost obstaja že kar nekaj let. Enega izmed prvih uporabnih AR sistemov je razvil Louis Rosenberg leta 1992 za uporabo pri Vojnem letalstvu Združenih držav Amerike (USAF). Komercialna uporaba tehnologije je sledila nekaj let kasneje, s prvo AR predstavo *Dancing in Cyberspace* Julie Martin leta 1994. Z razvojem interneta in z internetom povezanih tehnologij se je dalje razvijala tudi tehnologija obogatene resničnosti. Leta 2009 se je za uporabo na spletnih brskalnikih pojavilo orodje ARToolkit, ki je izključno namenjeno razvijanju AR izkušenj. [4] Septembra 2015 je Snapchat kot prvo družabno omrežje z obogateno resničnostjo razvil AR izkušnjo v realnem času z uporabo telefonske kamere. Razvoj tehnologije se je tako samo pospešil, kmalu sta se trgu pridružila tudi Facebook in Instagram. Decembra 2017 je Snapchat razvil prosto dostopen program Lens Studio, ki je namenjen izdelavi Snapchat AR filtrov (AR filter je v angleškem izvirniku poimenovan "lens", torej leča). Temu je leto kasneje sledil Facebook s programom Spark AR. [5] Področju pa se je pridružil tudi Google s splošnim orodjem WebAR za razvoj AR aplikacij. V okviru svojega širokega raziskovalnega projekta, ki snuje nove modele komunikacije skozi povezavo novih tehnologij z različnimi umetniškimi mediji, je Google eno izmed zbirk inovativnih eksperimentov posvetil eksperimentom z obogateno resničnostjo. [6]

3 Lens Studio

Lens Studio [7] je zastojnski program, sprva razvit kot interno orodje za izdelavo filtrov na socialnem omrežju Snapchat, ki pa je postal javno dostopen. Uporabniku omogoča, da sam razvije filter za Snapchat in ga tudi javno objavi. Uporabniško narejeni filtri so dostopni vsem.

Program ima redne posodobitve z novimi funkcionalnostmi in je dobro vzdrževan z zelo aktivno skupnostjo, ki se s tem ukvarja.¹ Prav tako ima

¹ Tudi avtor projekta Miha Malenšek je član programa Official Lens Creator pri podjetju Snap, zaradi česar ima dostop do določenih novih funkcionalnosti aplikacije Snapchat in programa za ustvarjanje filtrov Lens Studio. Poleg testiranja in razvoja izboljšav sodeluje v različnih tematskih projektih, ki jih izvaja Snapchat, npr. pri izdelavi lokacijske AR izkušnje na Brandenburških vratih v Berlinu.

podroben in dobro razdelan API (application program interface), ki je dostopen na spletni strani.

Delo v programu temelji na slojih (ang. layer) in virtualnih kamerah, kar omogoča uporabniku, da ima aktivnih več 3D ali 2D elementov brez prekrivanj, ki pa jih morajo uporabniki ustvariti v drugih, zunanjih aplikacijah. Obnašanje elementov filtra lahko uporabniki animirajo z že obstoječimi funkcijami ali pa sami napišejo skripto v programskem jeziku JavaScript. Za preprostejše filtre je dovolj, da je uporabnik splošno računalniško pismen, za kompleksnejše interakcije, kot so zaznavanje dotika, potega ali drugih gest s prsti, spremljanje uporabnikove lokacije v virtualnem 3D prostoru, pa je potrebno znanje programiranja.

Programiranje v Lens Studiu temelji na dogodkih (ang. event-based programming), čemur se odlično poda JavaScript. V programu se definirajo vhodi, ki so elementi filtra. To so lahko 3D modeli, 2D slike v 2D prostoru (ang. screen-space) ali v 3D prostoru (ang. world-space), texture, virtualne kamere, itd. Delovanje teh elementov je ločeno glede na prostor, v katerem se nahajajo. 3D elementov ne moremo naslikati na 2D ekransko ploskev, ta se poslužuje 2D elementov – slik. Tem lahko dinamično spreminjamo velikost in položaj. 2D elemente pa lahko uporabimo tudi v prostoru, kjer se obnašajo enako kot 3D elementi. V API-ju za Lens Studio so definirane funkcije in dogodki, ki so uporabniku na voljo v tem okolju in jih lahko uporabi za kompleksnejše delovanje svojega filtra. Ti dogodki so lahko odpiranje ust, oči, ali, za kar smo se odločili tudi mi, dvig obrvi. Ko se dogodek zgodi, lahko znotraj skripte definiramo primeren odziv.

Program, ki vodi delovanje filtra, lahko uporabniki pišejo kar v Lens Studiu, saj od verzije 3.0 naprej vključuje interaktiven urejevalnik, ki ima dostop do knjižnice tako osnovnih JavaScript funkcij kot tudi funkcij, po meri napisanih za Lens Studio.

4 Oblikovanje in izdelava 3D modelov

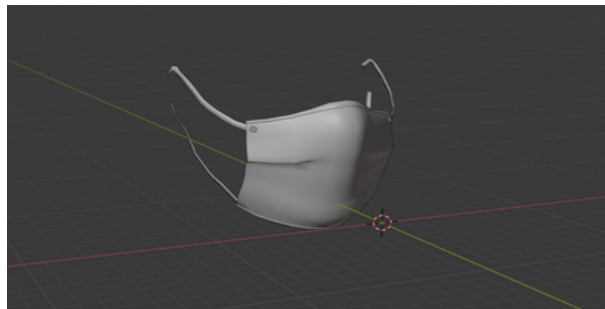
Za izdelavo virtualnih modelov smo izbrali že dobro znano orodje Blender; v članku opozorimo le na posebnosti, na katere moramo biti pozorni pri ustvarjanju modelov za AR filtre, saj predstavljajo nekaj omejitev, ki modeliranje nekoliko otežijo.

Zaradi prostorske omejitve izvoženega filtra (4 MB) moramo biti pazljivi pri velikosti vseh modelov, ki jih pri filtru uporabljamo. Lens Studio je preventivno postavil tudi zgornjo mejo za število trikotnikov na vsakem modelu (65,536), kar prepreči prezahtevnost prikazovanja modelov ob uporabi filtra. Zaradi omenjenih omejitev morajo biti modeli sestavljeni iz enostavnejših oblik z manj detajli, kar včasih pomeni manj realistične prikaze modelov predmetov, za katere bi sicer želeli, da bi bili videti karseda realistični. Postavljena je tudi zgornja meja za ločljivost tekstur, ki jih nanašamo na modele (2048 x 2048), kar pa ne predstavlja večjih težav, še posebej, če texture ustvarimo sami.

Za doseganje lepših, bolj realističnih, vendar še vedno ne preveč kompleksnih modelov ima Blender na

voljo nekaj funkcij, ki ustvarjanje precej olajšajo. Najpomembnejša je funkcija *Smooth Shade*, ki enostavnim modelom z malo ploskvami pogladi robove, kar nam omogoči, da tudi z malim številom ploskev dobimo lepe in gladke oblike.

Za naš filter je bilo potrebno ustvariti dva modela in sicer zaščitno masko (Slika 1) in maskoto »Safey« (Slika 2).



Slika 1: Model zaščitne maske (brez texture)



Slika 2: »Safey« v akciji

V nasprotju z modeliranjem maskote, ki je fiktivni lik, sama izdelava maske ni zahtevala veliko testiranja, saj smo poustvarili predmet iz realnosti, na katerega videz smo se lahko naslonili in ga skušali čim bolj vestno ponoviti. Vendar pa je bilo samo ustvarjanje, zaradi posebne oblike in texture maske, nekoliko zahtevnejše. Poiskati smo morali pravo razmerje med gladkostjo modela in realnostjo gub na maski, ki se pojavijo ob uporabi dejanske maske. Nasproten pristop smo morali ubrati pri modeliranju maskote, saj smo želeli čim bolj enostavno in prikupno maskoto, ki bi bila všeč uporabnikom, hkrati pa bi že sama po sebi spominjala ne ukrepe za zaščito pred korona virusom.

Zato smo se odločili za okroglo obliko (to je namreč oblika mila in milnih mehurčkov), ki lebdi na več milnih mehurčkih – ti naj bi uporabnika spomnili na pomembnost pravilne higiene in umivanja rok. Dodatka klobuka in nekaj preprostih obraznih potez sta maskoto ravno dovolj počlovečila in ji dodala osebnost, da je uporabnika pritegnila in ustrezno nagovorila tudi na čustveni ravni. »Safey-ja« smo izdelali v več različicah z različnimi razmerji velikosti trupa, mehurčkov ter klobuka, z različnimi teksturami, pozicijo klobuka in celo dodatkom nog. Za pomoč pri izbiri modela smo se obrnili na nekaj sošolcev in prijateljev, ki so podali konstruktivno kritiko vseh modelov z vidika bodočih uporabnikov, na koncu smo se odločili za obliko, ki jo prikazuje slika 2.

5 Implementacija filtra v programu Lens Studio

Filter sestavljata dva dela, ki sta povezana prvi s sprednjo kamero in drugi z zadnjo kamero. Na sprednji, t. i. "selfie" kameri se pojavi 3D model kirurške maske, projiciran na uporabnikov obraz. Ob dvigu obrvi se 3D model maske skrči po osi Y, ki je v Lens Studiu globalna vertikalna (ang. world up) os. Projiciranje maske in njeno premikanje v 3D prostoru glede na uporabnikov obraz je avtomatsko implementirano v programu Lens Studio s funkcionalnostjo *head-tracking*. Dodatne interakcije in obnašanje ter videz maske smo implementirali sami.

Na zadnji kameri se pojavi 3D maskota Safey, ki uporabnika opazuje na primerni varnostni razdalji. Če se mu uporabnik preveč približa, Safey odskoči in se umika tako dolgo, dokler ni zopet na primerni varnostni razdalji.

Prikazovanje elementov je vezano na kamere telefona – to ločimo z uporabo dogodka iz API knjižnice Lens Studia, ki zazna, katera kamera je aktivna in kdaj pride do menjave. Temu primerno, glede na aktivno kamero, omogočimo ali onemogočimo željene elemente.

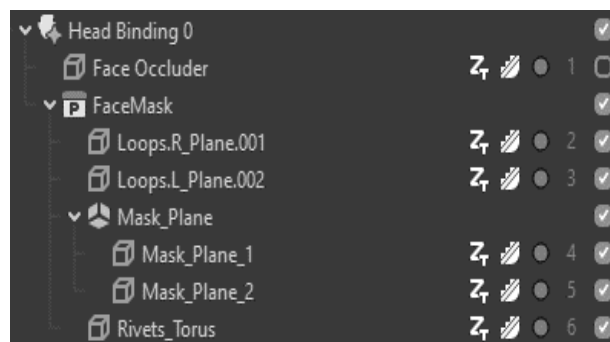
5.1 Implementacija maske

Kot je razvidno na sliki 3 je 3D model maske, imenovan FaceMask, sin objekta Head Binding 0, ta ga projicira uporabniku na obraz in vodi rotacijo in lokacijo 3D modela v prostoru. Sam 3D model maske je razdeljen na več delov – Loops.R_Plane.001, Loops.L_Plane.002, Mask_Plane, in Rivets_Torus. Razloga za to delitev sta: 1) množica različnih materialov, ki so uporabljeni za teksturo 3D modela, 2) prikazovanje maske s pomočjo segmentacije.

Nekateri deli 3D modela maske, kot so trakovi, ki masko držijo na obrazu, namreč ne smejo biti vidni iz vseh zornih kotov, saj jih ponekod prekriva obraz. To smo dosegli s spremljanjem rotacije obraza. Ko se uporabnikov obraz obrne po osi Y za določen kot, se za enega od trakov zamenja njegov sloj (ang. layer). Ker prikazovanje elementov v Lens Studiu temelji na slojih, smo tako dosegli učinek segmentacije obraza, saj menjava sloja 3D elementa omogoča, da ga prikazuje

maskirana virtualna kamera. Ta kamera uporablja že v naprej obstoječo segmentacijo obraza uporabnika, torej prikazuje samo to, kar je vidno zunaj obraza.

Odstranjevanje in ponovno pojavitev zaščitne maske smo animirali z uporabo knjižnice funkcij *Tween* v Lens Studiu. Ta funkcionalnost temelji na dogodku dviga obrvi. S preprostim binarnim števcem v programu spremljamo dvig obrvi uporabnika, da se 3D model maske pravilno odstrani in ponovno pojavi. Animacija je preprosta, vendar zaradi hitrosti in preprostosti giba tega ne izda. Model se skrči po osi Y in zavrti po osi X. Transformacije se dogajajo v lokalnem prostoru modela, saj bi v globalnem lahko prišlo do napak, zaradi gibanja obraza uporabnika. Hkrati s transformacijami pa se tudi poveča transparentnost maske, dokler ne postane popolnoma nevidna.



Slika 3: Prikaz slojev za 3D model maske

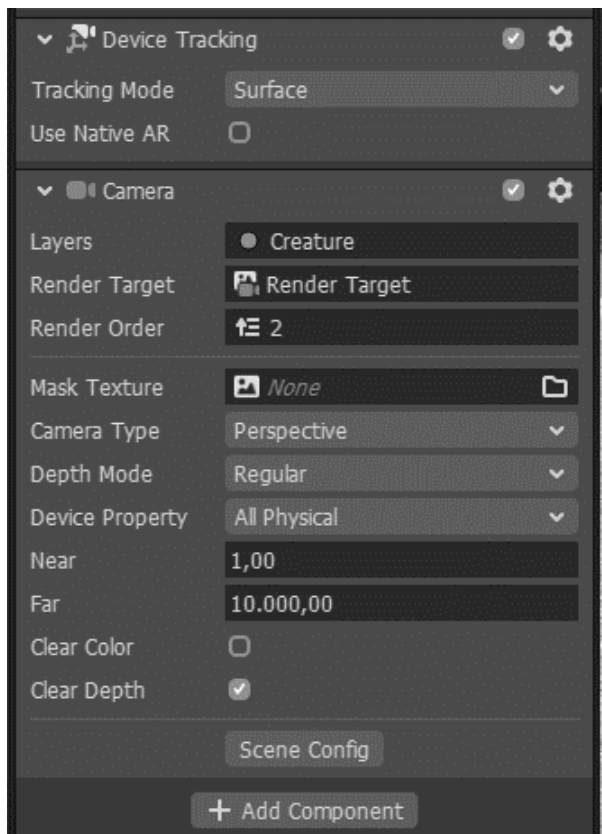


Slika 4: Maska v uporabi

5.2 Implementacija 3D maskote Safey

Implementacija 3D maskote Safey temelji na uporabi funkcionalnosti *Surface Tracking* v Lens Studiu. Ta zahteva svojo virtualno kamero in sceno, kar pa pomeni

tudi svoj sloj. Iz slike 5 je razvidno, da je 3D model maskote na sloju "Creature", virtualna kamera pa ima komponento "Device Tracking" nastavljeno na "surface" oz. tla. Z uporabo te komponente lahko med delovanjem spremljamo uporabnikovo lokacijo v 3D prostoru, katerega izhodišče je tam, kjer je uporabnik začel z uporabo filtra.



Slika 5: Prikaz komponente *Surface Tracking* za maskoto

```
//GIBANJE
var updateEvent = script.createEvent("updateEvent"); //Omogoča klic funkcije ob začetku novega framesa
updateEvent.bind(function(eventData){
  var sTransform = script.safey.getTransform(); //Transform 3D lika Safey
  var laTransform = script.lookat.getTransform(); //Transform komponente "Look At" - vedno usmerjena proti kameri
  var laRot = laTransform.getWorldRotation(); //rotacijski vektor "Look At"
  var sRot = sTransform.getWorldRotation(); //rotacijski vektor 3D lika Safey
  var laPos = laTransform.getWorldPosition(); //pozicijski vektor "Look At"
  var sPos = sTransform.getWorldPosition(); //pozicijski vektor 3D lika Safey
  var camPos = script.creatureCamera.getTransform().getWorldPosition(); //pozicijski vektor virtualne kamere

  //Nastavi rotacijo 3D lika Safey tako, da je orientiran proti kameri
  sTransform.setWorldRotation(new quat(sRot.x, sRot.y, laRot.z, sRot.w));
  //Razdalja med likom in virtualno kamero
  var distance = calculateDistance(sPos.x, camPos.x, sPos.z, camPos.z);

  //Razdalja v Z in X osi
  var toCamZ = sPos.z - camPos.z;
  var toCamX = sPos.x - camPos.x;
  //normirana razdalja Z in X osi, glede na celotno razdaljo
  toCamZ = toCamZ/distance;
  toCamX = toCamX/distance;

  //če je razdalja manjša od 250 enot, se izračuna nov položaj lika, ki se uporabi za premik s Tweenom
  if(distance < 250) {
    var newPos = new vec3(sPos.x + toCamX*200, sPos.y, sPos.z + toCamZ*200);
    global.tweenManager.setValue(script.creatureTween, "move", newPos);
    global.tweenManager.startTween(script.creatureTween, "move");
  }
});
```

Slika 6: Koda, ki upravlja razdaljo maskote od uporabnika

Sledenje lokaciji omogoča preprosto izračunavanje aktualne razdalje med uporabnikom in projekcijo 3D modela v prostor, kar posledično omogoča tudi spreminjanje te razdalje. Tako smo lahko implementirali

varnostno razdaljo, ki jo vzdržuje 3D model maskote do uporabnika.

6 Sklep

V prispevku predstavimo postopek razvoja tematskega filtra za bogatenje resničnosti v programu Lens Studio. Zadovoljni smo z izdelavo 3D modelov zaščitne maske in maskote Safey, saj smo kljub omejitvam ustvarili prepoznavne modele z malo ploskvami, a kljub temu dovolj detajli. Večino dela je predstavljal razvoj filtra, kjer smo implementirali projiciranje maske na uporabnikov obraz in odmikanje maskote od uporabnika. Za to smo uporabili že v Lens Studio implementirano detekcijo obraza in vezanje filtra na lokacijo, vseeno pa je bilo potrebno še veliko dela za popraviljanje malenkosti, kot so razdalja, na katero se maskota umakne, boljše prilaganje maske na obraz, pospravljanje in ponovna uporaba maske ... Filter smo objavili na Snapchatu za širšo uporabo in dobili precej pozitivnih odzivov. Menimo, da smo dosegli zastavljeni cilj širjenja ozaveščenosti glede nevarnosti virusa in upoštevanja varnostnih ukrepov.

Od začetka pisanja članka smo zasledili nekaj podobnih filtrov drugih razvijalcev, omeniti velja poziv Združenih narodov digitalnim ustvarjalcem k razvoju AR izkušenj za osveščanje populacije o pandemiji, katerega rezultat je bil mdr. filter »Snap Safe«, ki uporabnike opozarja na vzdrževanje varnostne razdalje. [8]

Do našega filtra lahko dostopate na povezavi: <https://www.snapchat.com/unlock/?type=SNAPCODE&uuid=ef1f64fcb331497c942edf0e6bb33453&metadata=01>

Literatura

- [1] Space miner filter, https://www.reddit.com/r/SnapLenses/comments/ihx96d/space_miner/
- [2] Snap filtri Mihe Malenška, <https://lensstudio.snapchat.com/creator/w75o4pLLh2EUQAYIjB94A>
- [3] (RED), <https://www.red.org/retorial/2015/11/30/24-hours-only-get-snapchat-red-filters-for-world-aids-raise-3-million>
- [4] "The Lengthy History of Augmented Reality", Huffington Post, http://images.huffingtonpost.com/2016-05-13-1463155843-8474094-AR_history_timeline.jpg
- [5] The brief history of social media AR filters, <https://www.industry.com/blog/2019/10/11/the-brief-history-of-social-media-ar-filters#:~:text=Just%20a%20couple%20of%20hours,expanded%20on%20Instagram%20as%20well>
- [6] Experiments with Google, AR Experiments, <https://experiments.withgoogle.com/collection/ar>
- [7] Lens Studio, <https://lensstudio.snapchat.com/>
- [8] Snap Safe, <https://wearesocial.com/blog/2020/04/snap-safe-our-answer-to-the-uns-call-to-creatives>