

Računalna grafika

Laboratorijska vježba 3

Jakov Dorontić

Zagreb, siječanj 2021.

Sadržaj

1. Uvod.....	2
2. Shader Graph	3
2.1. Efekt obasjavanja	4
2.2. Efekt štita	5
2.3. Efekt vode	6
2.4. Efekt portala.....	7
2.5. Efekt rastapanja	8
3. Literatura.....	9
Dodatak A: Pokretanje programa	10

1. Uvod

Cilj ove laboratorijske vježbe je upoznati se i izraditi jednostavne procesore za sjenčanja (eng. *shader*) unutar pokretača *Unity*. U sklopu laboratorijske vježbe, fokus će biti na najnovijim alatima koje pokretač nudi za izradu naprednih 3D i 2D igara.

Unity se trenutno nalazi u tranziciji iz fiksnog cjevovoda za prikazivanje u onaj programabilni (eng *Scriptable Render Pipeline*). Drugim riječima, do sada su opcije za modificiranje cjevovoda za generiranje slikovnog okvira bile veoma limitirane. Od sada *Unity* nudi opciju razvojnim timovima da modificiraju cjevovod u većoj mjeri nego prije i na način prikladan za njihov projekt. Postoje dvije vrste programabilnih cjevovoda:

- **Universal Rendering Pipeline URP** – vrsta programabilnog cjevovoda koji je namijenjen za najopćenitije primjene te je fokus stavljen na dobre performanse sustava
- **High Definition Render Pipeline HDPR** – vrsta programabilnog cjevovoda u kojem se naglasak stavlja na najmodernije i najnaprednije načine prikaza virtualnih svjetova

Razvoj projekta pomoću pokretača sastoji se od korištenja već implementiranih funkcionalnosti i modifikacije njihovih parametara, te proširivanja bazičnih funkcionalnosti pomoću komponenata. Vlastite komponente se razvijaju pomoću jezika C#.

2. Shader Graph

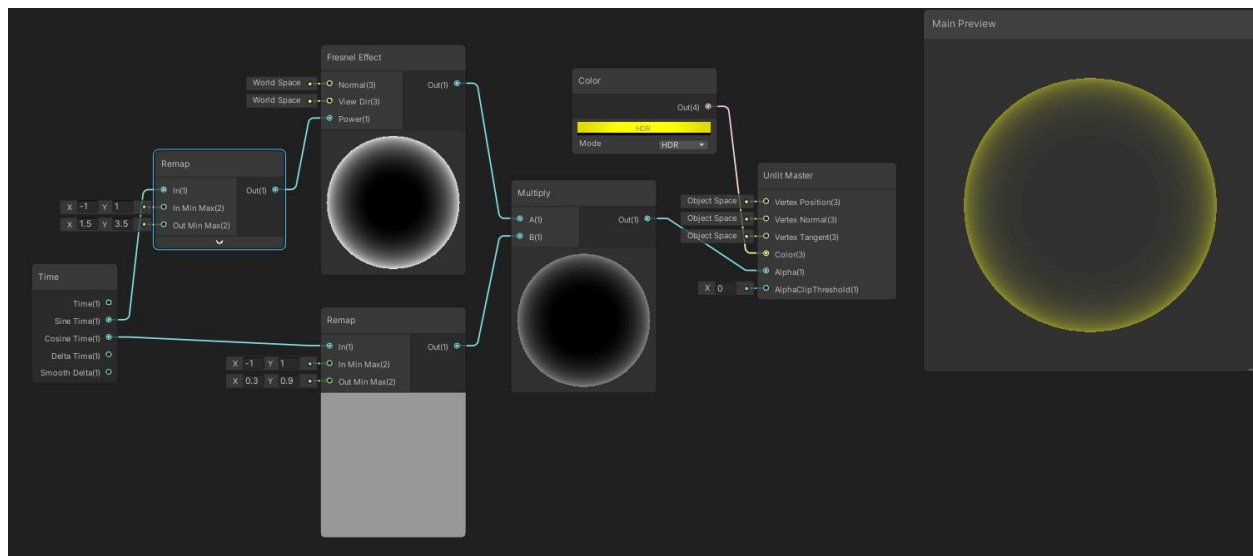
Shader Graph je vizualni jezik za kreiranje procesora za sjenčanje. Izrada procesora za sjenčanje za različite efekte koji nisu direktno implementirani je izrazito zahtjevan posao jer zahtijeva samostalno pisanje vlastitih skripti za svaki procesor sjenčanja. *Shader Graph* je alat koji nastoji izradu procesora za sjenčanje za raznorazne specijalne efekte znatno pojednostaviti. Izrada jednog procesora za sjenčanje pomoću ovog alata sastoji se od odabira prikladnih čvorova i veza između tih istih čvorova. Čvorovi su, esencijalno, već implementirane funkcije koje obavljaju određenu zadaću i imaju dobro definirano izlazno i ulazno sučelje. Pokretač *Unity* u dokumentaciji pobliže opisuje svaki čvor, njegove ulaze i izlaze. Veze između čvorova predstavljaju spajanje izlaza određenih čvorova na ulaze drugih čvorova.

Za 3D prikazivač *Unity* nudi opciju kreiranja neosvijetljenog (eng. *Unlit*) grafa, PBR (eng. *Physically Based Rendering*) grafa i VFX (eng. *Visual effects*) grafa. U ovoj laboratorijskoj vježbi korist će se neosvijetljeni graf. Glavni čvor neosvijetljenog grafa ima ulaze za poziciju, normalu i tangentu vrha, te ulaze za boju i alfa vrijednost.

Osim čvorova i veza, treći važan element *Shader Grapha* alata su parametri. Naime, svi procesori za sjenčanje koji se izrađuju pomoću ovog alata su namijenjeni kao baza za izradu materijala. Drugim riječima, procesor za sjenčanje je baza za izradu materijala, dok se razlika u materijalima postiže mijenjanjem parametara samog procesora. Za svaki graf je moguće definirati skup parametara koje će biti moguće mijenjati prilikom izrade materijala. U sklopu ovog projekta *Shader Graph* će se iskoristiti za izradu procesora za sjenčanje koji će generirati efekt obasjavanja, efekt štita, efekt vode, efekt portala i efekt rastapanja.

2.1. Efekt obasjavanja

Slika 2.1.1 prikazuje implementaciju efekta obasjavanja pomoću *Shader Grapha*. Kreiranje efekta kreće od čvora za generiranje Fresnelovog efekta na čiji se ulaz dodaje sinusoidna promjena vremena koja se kreće u intervalu od $[-1, 1]$ što se pomoću čvora rampe skalira na interval $[1.5, 3.5]$. Ovime postizemo postepeno debljanje pa postepeno stanjivanje rubova Fresnelovog efekta kako vrijeme prolazi. Dodatno, uzimamo kosinusoidnu promjenu vremena koju s intervala $[-1, 1]$ skaliramo na interval $[0.3, 0.9]$ te takvu promjenu množimo sa ranije opisanim (promjenjivim) Fresnelovim efektom. Izlaz množenja povezujemo na alfa ulaz glavnog čvora. Također, potrebno je dodati čvor za boju kako bi efekt bio realističniji i postaviti glavni čvor u transparentni način rada kako bi mu površina bila prozirna na dijelovima gdje je alfa jednaka nuli.

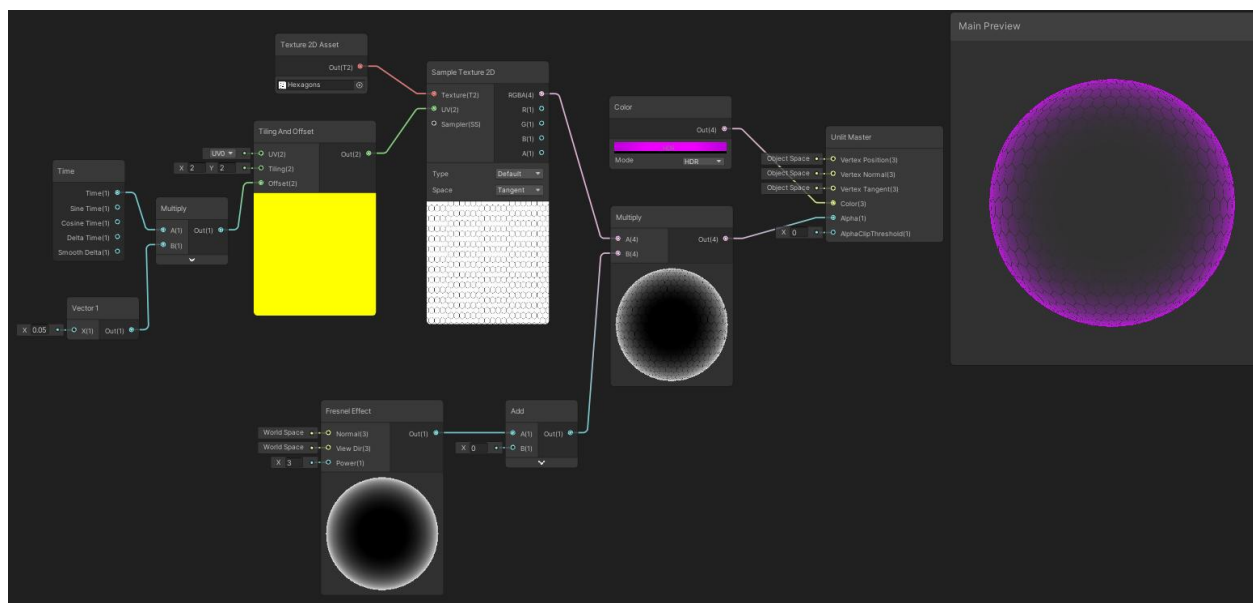


Slika 2.1.1: Efekt obasjavanja

2.2. Efekt štita

Slika 2.2.1 prikazuje implementaciju efekta štita pomoću *Shader Grapha*.

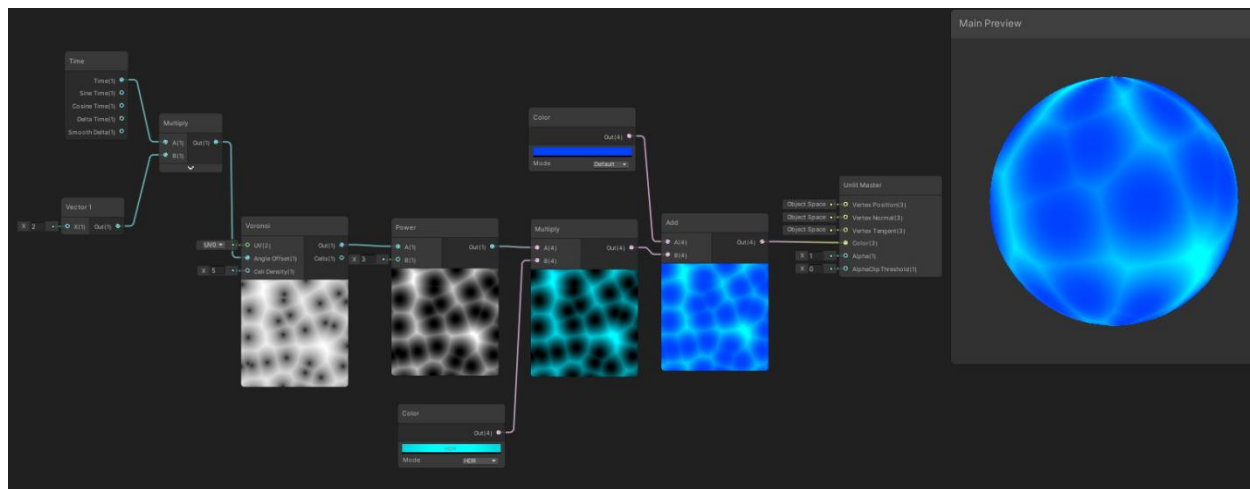
Za razliku od efekta svijetljenja, efekt portala koristi teksturu koja na bijeloj pozadini ima crnom bojom nacrtane heksagone. Kao i ranije, koristi se vrijeme kako bi se animirala promjena teksture u vremenu. Tekstura se neprestano pomiče u smjeru negativne x i y koordinatne osi (dole lijevo) pomoću čvora za replikaciju i pomak. Takva animirana tekstura se množi sa Fresnelovim efektom. Rezultat množenja povezuje se na alfa ulaz glavnoga čvora. Također, potrebno je dodati čvor za boju kako bi efekt bio realističniji i postaviti glavni čvor u transparentni način rada kako bi mu površina bila prozirna na dijelovima gdje je alfa jednaka nuli.



Slika 2.2.1: Efekt štita

2.3. Efekt vode

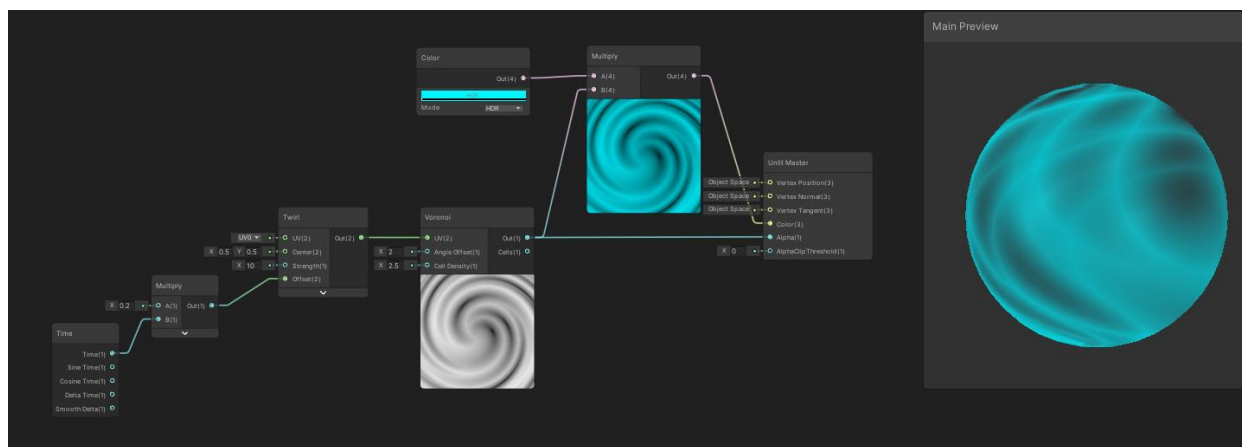
Slika 2.3.1 prikazuje implementaciju efekta vode pomoću *Shader Grapha*. Kreiranje efekta kreće od čvora koji generira Voronoiev dijagram. Na ulaz *Angle Offset* spajamo izlaz čvora za kontinuirano vrijeme. Time dobivamo pomak ćelija Voronoieva dijagrama. Drugi važan parametar je *Cell Density* odnosno gustoća ćelija Voronoieva dijagrama. Taj parametar će određivati koliko će biti gusti odsjaji na površini vode. Izlaz iz Voronoi čvora nadalje potenciramo željenom potencijom, što je vrijednost potencije veća to su odsjaji tanji, a ćelije deblje. Nakon potenciranja slijedi bojanje, prvo se boja odsjaj vode tako da se izlaz čvora za potenciranje množi sa željenom bojom odsjaja. Zatim se izlaz množenja zbraja sa željenom bojom kako bi dobili boju vode bez odsjaja. Konačno, rezultat svega navedenoga povezuje se na komponentu boje glavnoga čvora.



Slika 2.3.1: Efekt vode

2.4. Efekt portala

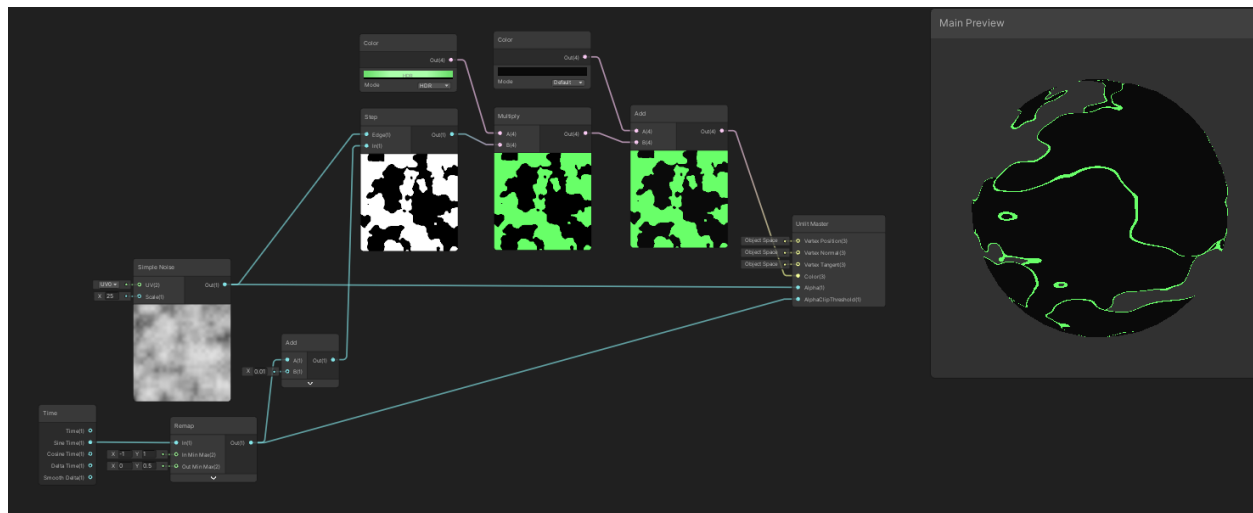
Slika 2.4.1 prikazuje implementaciju efekta portala pomoću *Shader Grapha*. Kao i u prijašnjem efektu, kreiranje efekta portala kreće od čvora koji generira Voronoiev dijagram. Ovoga puta na ulaz UV koordinata Voronoieva dijagrama stavlja se efekt vrtnje (eng. *Twirl*) čime se ćelije i praznine nejednoliko razmažu. Dodatno, efektu vrtnje postavlja se pomak ovisan o vremenu čime se dobiva animacija vrtnje. Rezultat svega do sada navedenoga povezuje se na alfa vrijednost glavnoga čvora. Također, potrebno je dodati čvor za boju kako bi efekt bio realističniji i postaviti glavni čvor u transparentni način rada kako bi mu površina bila prozirna na dijelovima gdje je alfa jednaka nuli.



Slika 2.1.1: *Efekt portala*

2.5. Efekt rastapanja

Slika 2.5.1 prikazuje implementaciju efekta rastapanja pomoću *Shader Grapha*. Kreiranje efekta kreće od čvora za generiranje jednostavnoga šuma čiji izlaz odmah povezujemo na ulaz za alfa vrijednost glavnoga čvora, ovime dobivamo mapu za daljnju manipulaciju alfa vrijednostima. Kako bi efekt postepeno nestajao na nekim svojim dijelovima, koristi se prag za alfa odsijecanje (eng. *alpha clip threshold*). Ideja je uzeti sinusoidno vrijeme i skalirati ga sa intervala $[-1, 1]$ na interval $[0, 0.5]$ i tu vrijednost skalacije povezati na prag za alfa odsijecanje glavnoga čvora. Na taj način uklanjaju se svi pikseli sa alfa koeficijentom manjim od ranije dobivene skalacije. Dodatno, kako bi dobili obrub rastapanja u drugoj boji potrebno je ranije dobiveno skalirano sinusoidno vrijeme posmaknuti za konstantni iznos dodajući mu neku malenu vrijednost (0.01). Nad dobivenim rezultatom posmaka i nad čvorom za šum potrebno je provesti step funkciju. Dobivenu step funkciju potrebno je pomnožiti sa željenom bojom obruba te taj rezultat množenja povezati na ulaz za boju glavnoga čvora.



Slika 2.5.1: Efekt rastapanja

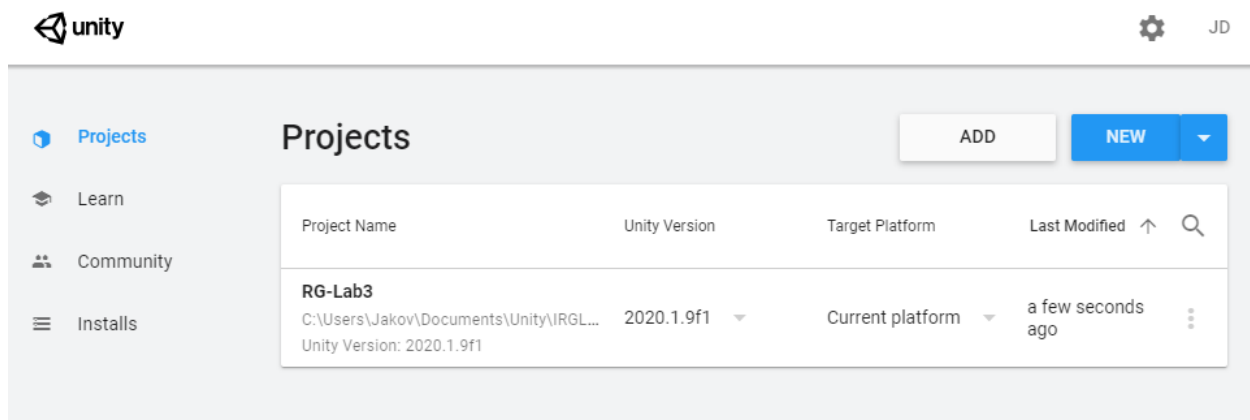
3. Literatura

- [1] <https://unity.com/> - službena stranica pokretača Unity [27.12.2020]
- [2] <https://docs.unity3d.com/2020.1/Documentation/Manual/UnityManual.html> - službena dokumentacija pokretača Unity za razvoj projekta [27.12.2020]
- [3] <https://docs.unity3d.com/Packages/com.unity.shadergraph@8.2/manual/Shader-Graph-Samples.html> - službena dokumentacija paketa Shader Graph za pokretač Unity [27.12.2020]

Dodatak A: Pokretanje programa

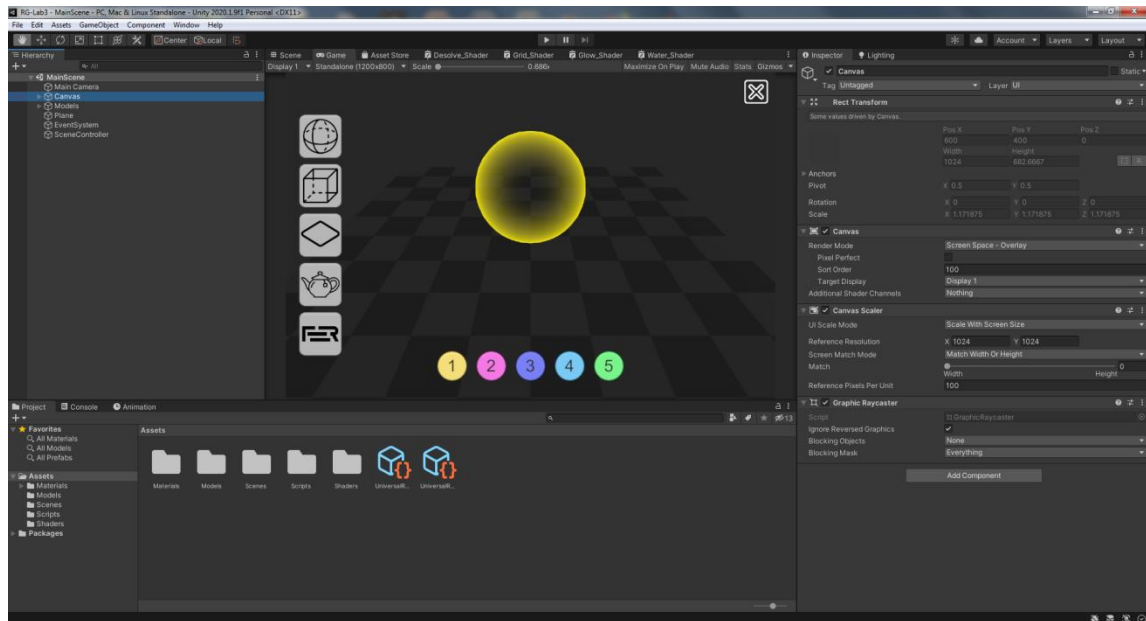
Za brzo pokretanje programa pozicionirajte se u direktorij *Export* (unutar glavnog direktorija *RG-Lab3*) i pokrenite datoteku *RG-Lab3.exe* za što će vam trebati Windows operativni sustav.

Za pregled i mijenjanje programa preuzmite i instalirajte pokretač *Unity* čija je verzija 2020.1.9f1. Zatim preuzmite izvorne datoteke projekta s GitHub-a (direktorij *RG-Lab3*) i prekopirajte ga na željenu poziciju na računalu. Nakon što su se sve datoteke prekopirale, pokrenite Unity i uvezite projekt tako da odaberete ranije preuzet direktorij. *Slika A.1* prikazuje pravilno postavljen projekt.



Slika A.1: Pravilno postavljen projekt

Pritiskom na karticu "RG-Lab3" pokreće se Unity i učitavaju se podatci. Nakon što su svi podatci učitani projekt je spreman za korištenje kako je prikazano na *Slika A.2*. Unutar glavne mape *Assets* nalaze se sve komponente potrebne projektu (materijali, modeli, scene, izvršni kodovi i procesori za sjenčanje). Pritiskom na ikonicu *Play* unutar pokretača *Unity*, projekt se pokreće zajedno sa svim svojim komponentama.



Slika A.2: Učitani projekt