SVEUČILIŠTE U ZAGREBU
**FAKULTET ELEKTROTEHNIKE I RAČUNARSTVA**

Računalna animacija

# IGRA S ROBOTSKOM HVATALJKOM

Antonija Engler

3. laboratorijska vježba

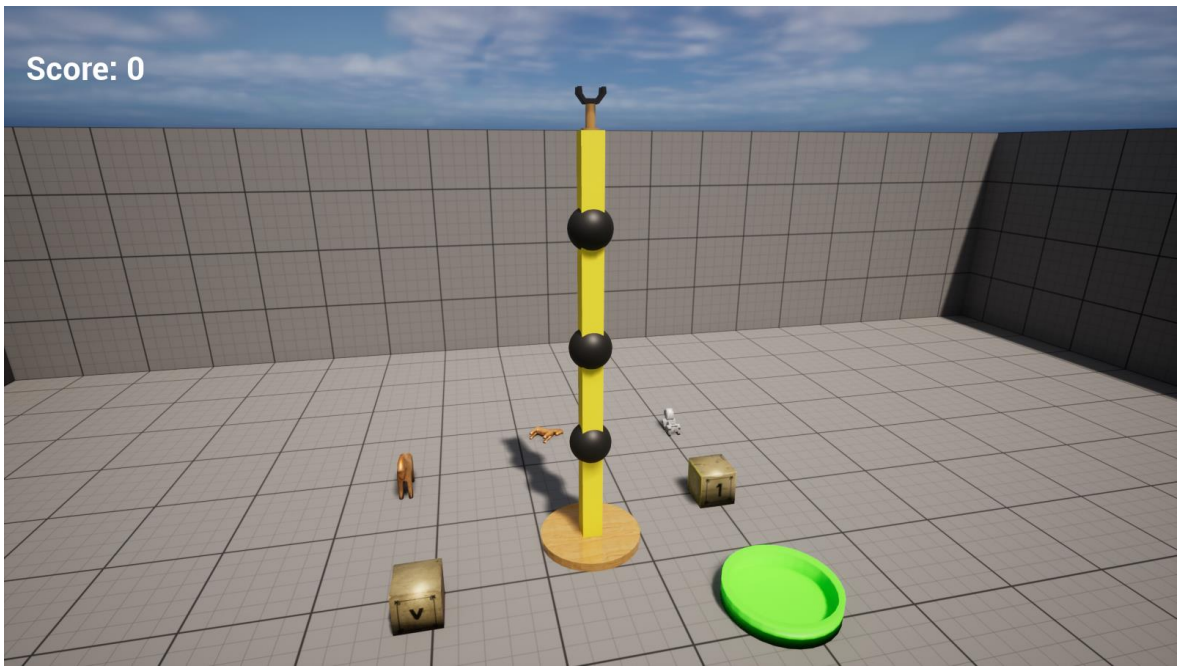
Zagreb, siječanj 2023.

## Sadržaj

# 1. Opis projekta

*Claw Game* je igra koja implementira inverznu kinematiku na zabavan način. Cilj je pomoću robotske ruke sakupiti sve igračke u što kraćem vremenu.

Početna scena sastoji se od dva drvena konjića, kocke s brojevima, kocke sa slovima te jedne zebre (slika 1.). Svaka igračka se „primi“ tako da se vrh robotske ruke približi željenoj igrački. Nakon što je igračka uhvaćena, potrebno ju je odnijeti do zelene posude i staviti je unutra. Ne može se držati više od jedne igračke u isto vrijeme.




Slika 1. Početna scena

Inverzna kinematika je postupak u kojem se zadaje ciljna pozicija manipulatora (u ovom slučaju vrha robotske ruke) te se na temelju nje izračunavaju promjene kutova u zglobovima. Nasuprot tome, u direktnoj kinematici se početno zadaju parametri u zglobovima pomoću kojih se, uz hijerarhijska pravila, računa položaj krajnjeg vrha[1].

Modeli igračka su preuzeti pomoću aplikacije Bridge, dok su modeli robotske ruke i zelene posude napravljeni u aplikaciji Blender. Funkcionalnost igre napravljena je u softveru Unreal Engine.

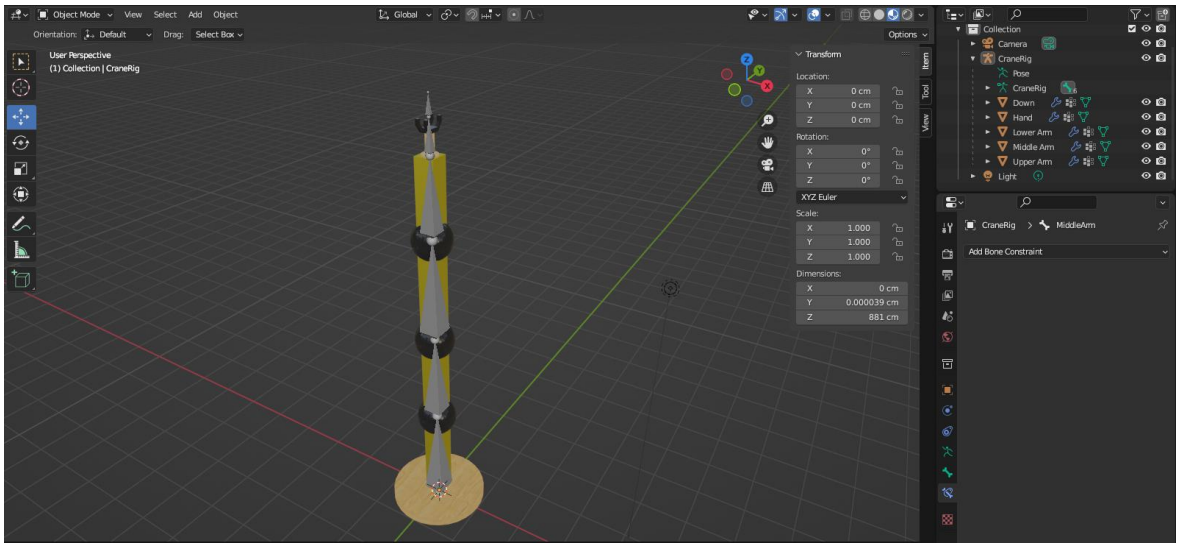
# 2. Implementacija

## 2.1. Blender

Blender je besplatan alat otvorenog koda koji se koristi u području 3D računalne grafike, a pruža mogućnosti animiranja, modeliranja, simuliranja, praćenja kretanja, stvaranja vizualnih efekata i interaktivnog sadržaja [2].

Model robotske ruke (slika 2.) napravljen je pomoću manipulacije tri osnovna tijela: cilindra (baza i dio vrha ruke), kocke (žuti segmenti i „štipaljke“) i kugle (crni zglobovi). Nakon toga, dodan je materijal na odgovarajuća mjesta.

Kostur ruke dodan je pomoću opcije „*Armature*“ te su kosti hijerarhijski povezane. Kako bi alat znao koji dio modela mora pomaknuti kada se pomakne određena kost, kostur i model su povezani pomoću opcije „*Parent*“ → „*Armature Deform*“ → „*With Automatic Weights*“. Na vrhu kostura je dodana još jedna kost koja nije pričvršćena na nijednu točku (engl. *vertex*) modela te služi kao manipulator lanca inverzne kinematike.



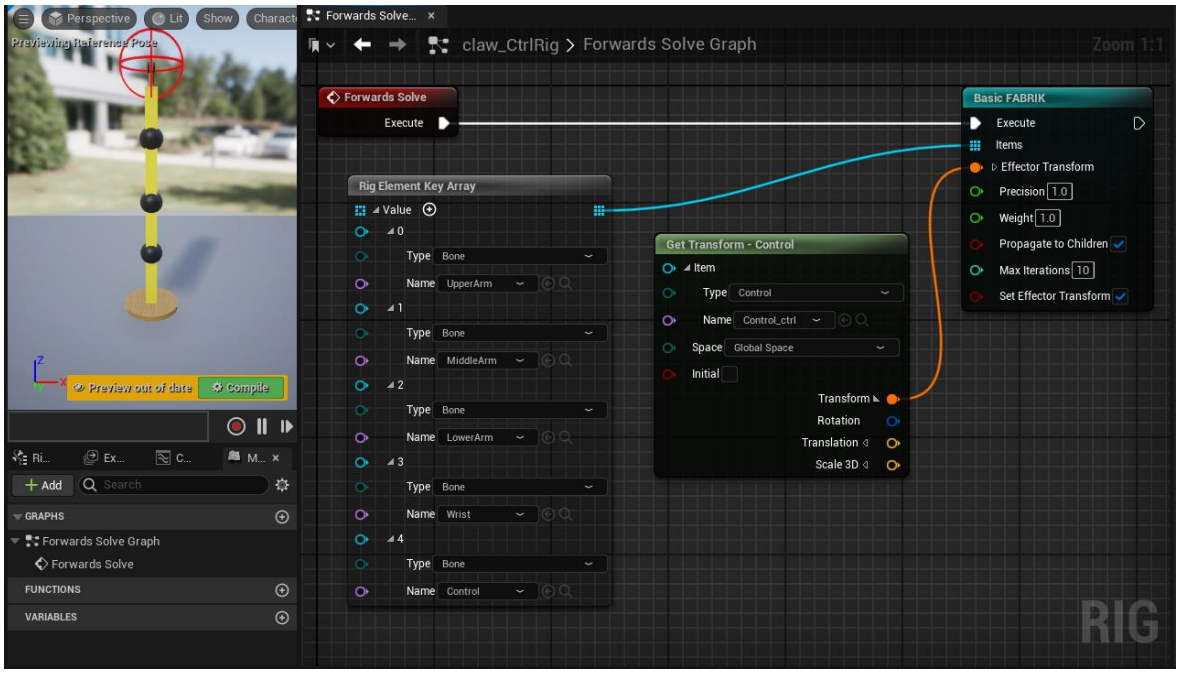

Slika 2. Model i kostur robotske ruke

## 2.2. Unreal Engine

Unreal Engine je softver koji se koristi u području računalnih igara, simulacija, arhitekture, dizajna, televizijske industrije i virtualne stvarnosti[3]. On pruža opciju programiranja u jeziku C++ ili grafički pomoću sustava *Blueprints*.

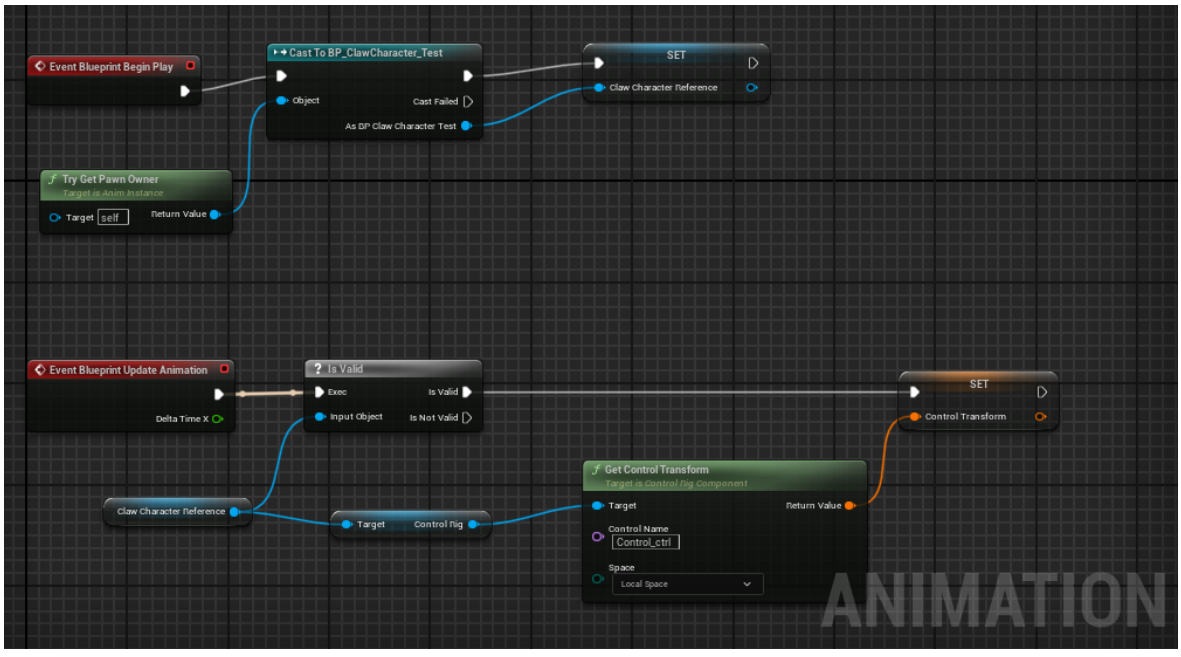
U ovom je projektu inverzna kinematika definirana unutar funkcije FABRIK (Forward and Backward Reaching Inverse Kinematics). FABRIK je iterativan heuristički algoritam koji na temelju predodređenih pravila dovodi lanac iz početnog u ciljni položaj. Iako FABRIK ne mora uvijek doći do konačnog rješenja, njegova brzina rješavanja problema poželjna je u ljudskim i neljudskim animacijama. Algoritam je primjenu primarno našao u razvoju računalnih igara i računalnoj grafici, no koristi se i u robotici i predviđanju ljudskih pokreta[4].

Čvoru (engl. *node*) FABRIK može se pristupiti unutar strukture „*Control Rig*“ koja služi za animaciju i namještanje raznih ograničenja i pravila kretanja objekta**Pogreška! Izvor reference nije pronađen.**. Postavke za robotsku ruku mogu se vidjeti na slici 3. Čvor FABRIK prima parametre „*Items*“ koji je lista svih kosti na koje će algoritam djelovati i „*Effector Transform*“ koji je transformacija kosti manipulatora (dodatna kost na vrhu ruke). Cijeli postupak se pokreće na događaj „*Forward Solve*“. Dodatno, za kost manipulatora koja pokreće lanac potrebno je definirati kontrolu (engl. *control*) čiju transformaciju uzima FABRIK.




Slika 3. Postavke čvora FABRIK

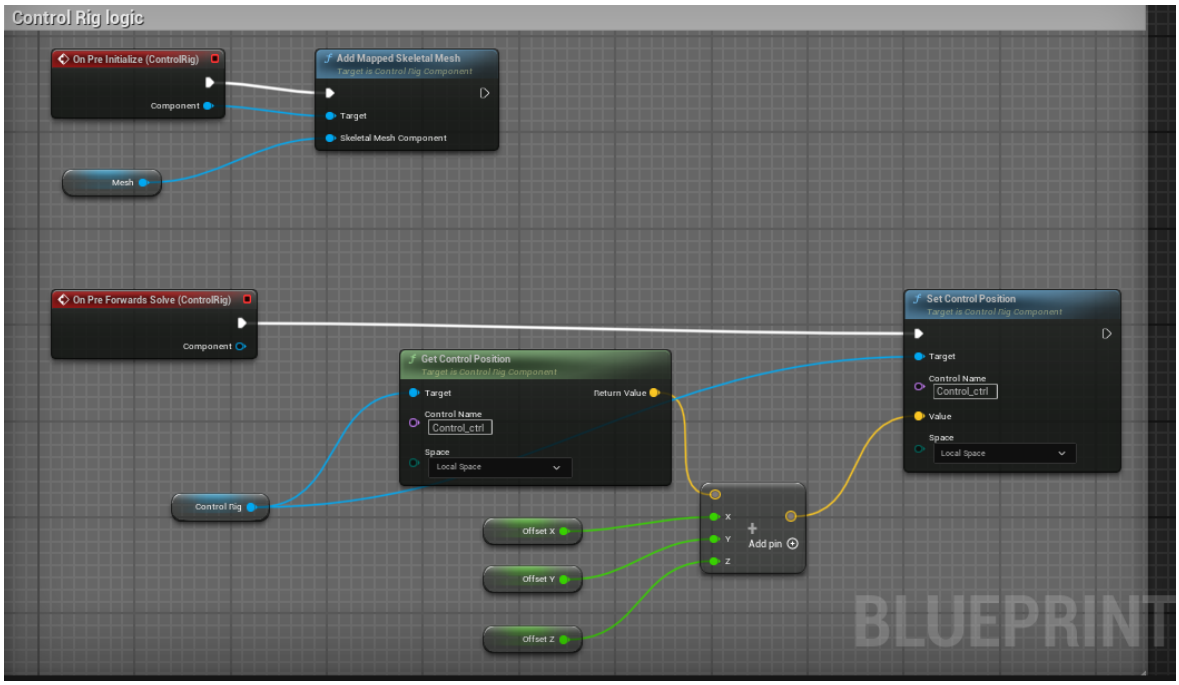
U strukturi „*Animation Blueprint*“ povezuje se „*Control Rig*“ s animacijom te se uzima transformacija manipulatora (slika 4.) koja će se dinamički mijenjati tijekom igre i dojavljivati „*Control Rig*“-u kako bi mogao osvježiti položaj robotske ruke.




Slika 4. Graf događaja u klasi `BP_ClawAnimation_Test`

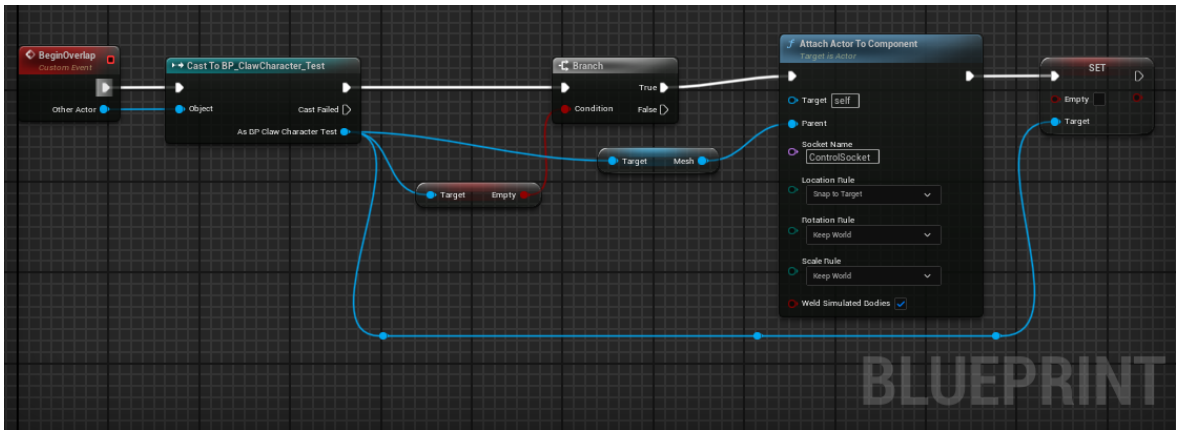
Funkcionalnost igrača, odnosno robotske ruke, definirana je u klasi `BP_ClawCharacter_Test`. Kada igrač stisne tipke „w“, „s“, „a“, „d“, „q“ i „e“ vrhu robotske ruke se dodaje ili oduzima određena konstanta koja služi kao brzina kretanja vrha.

Tipke „w“ i „s“ služe za micanje naprijed – natrag, „a“ i „d“ za micanje lijevo – desno, a „q“ i „e“ za micanje gore – dolje. Prije pokretanja animacije potrebno je inicijalizirati „*Control Rig*“ tako da se poveže s modelom i kosturom nad kojim treba djelovati. Tijekom pokretanja animacije postavlja se pozicija kontrole (manipulatora) na onu koju je igrač zadao te se tako ažurira animacija.



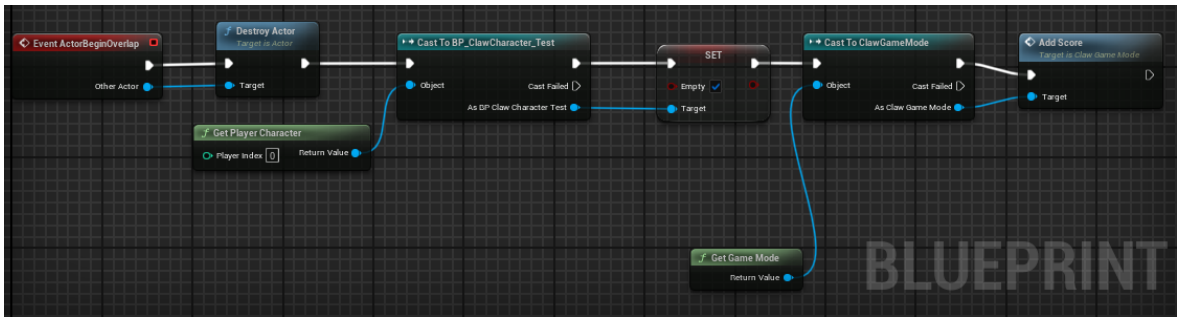

Slika 5. Funkcionalnost kretanja robotske ruke

Hvatanje igrača implementirano je pomoću detekcije kolizije. Definirana je klasa `BP_Toy` koju nasljeđuju sve igračke, a koja ima definiranu funkciju `BeginOverlap` (slika 6.). Ta funkcija provjerava je li vrh robotske ruke prazan i ako je, dodaje objekt igračke na određenu poziciju u kosturu ruke (engl. *socket*). Svaka igračka ima svoj omeđujući volumen (engl. *Bounding Box*) pomoću kojeg se detektira kolizija s drugim objektom. Kada se to dogodi, poziva se funkcija `BeginOverlap` nadklase.



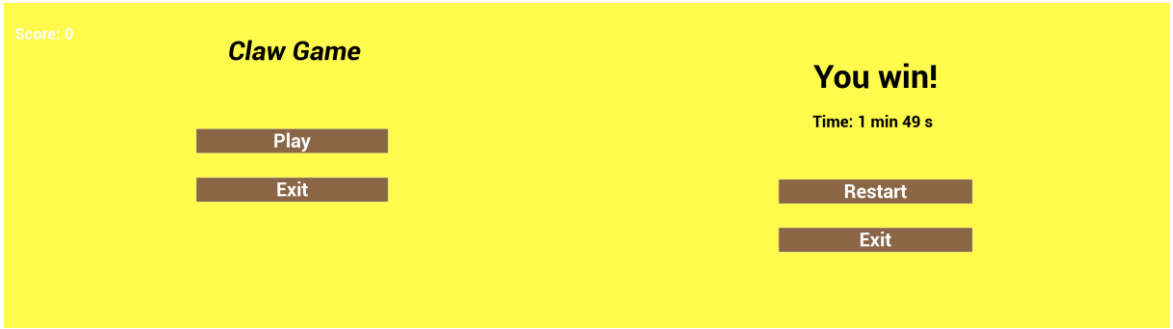

Slika 6. Funkcija `BeginOverlap` u klasi `BP_Toy`

Kako bi došao do kraja igre, korisnik mora staviti igračke u zeleni spremnik. Ta funkcionalnost također je implementirana pomoću detekcije sudara. Kada igračka uđe u omeđujući volumen spremnika, uništi se instanca u sceni te se unutar funkcije `Add Score` (slika 7.) doda bod igraču, što se prikazuje u gornjem lijevom kutu.




Slika 7. Funkcija `Add Score` unutar klase `BP_Finish`

Također se provjerava  je li igrač prebacio sve igračke. Ako jest, prikazuje se konačni zaslon (slika 8. desno) s vremenom koje je bilo potrebno igraču da završi igru. Sličan ekran (slika 8. lijevo) pojavljuje se na početku igre.




Slika 8. Prikaz početka i kraja igre

# 3. Upute za pokretanje

Projekt je rađen u alatu Unreal Engine 5.1. Igra se može pokrenuti ako se preuzme mapa *Build* (*RA_labos*/*labos3*) te se pokrene datoteka ClawGame.exe.

Kontrole:

- „w“ / „s“ – naprijed, natrag
- „a“ / „d“ – lijevo, desno
- „q“ / „e“ – gore, dolje
- pomicanje miša – rotacija oko robotske ruke

# Sažetak

U ovom projektu korištena je inverzna kinematika kako bi se dobila simulacija kretanja robotske ruke. Ta simulacija je uklopljena u igru u kojoj je cilj postaviti sve igračke u zelenu posudu. Korisnik može pomicati robotsku ruku pomoću tipkovnice i miša. Za ostvarenje rješenja korišten je Unreal Engine 5.1 i Blender 3.4.1.

# Summary

In this project, inverse kinematics was used to simulate the movement of robotic arm. This simulation is embedded in the game in a way that the goal is to put all toys in a green vessel. The user can move robotic arm using keyboard and mouse. Unreal Engine 5.1 and Blender 3.4.1 were used to create the solution.

# Literatura


[1] Željka Mihajlović, *Direktna i inverzna kinematika*, http://www.zemris.fer.hr/predmeti/ra/predavanja/4_kinemat.pdf; pristupljeno 17. siječnja 2023.

[2] https://www.blender.org/about/; pristupljeno 17. siječnja 2023.

[3] https://www.unrealengine.com/en-US/unreal-engine-5; pristupljeno 17. siječnja 2023

[4] Maurice Lamb, Seunghun Lee, Erik Billing, Dan Högberg, and James Yang, *Forward and Backward Reaching Inverse Kinematics (FABRIK) Solver for DHM: A Pilot Study*, https://pubs.lib.uiowa.edu/dhm/article/31772/galley/140227/view/; pristupljeno 17. siječnja 2023.

[5] https://docs.unrealengine.com/5.0/en-US/control-rig-in-unreal-engine/; pristupljeno 17. siječnja 2023.