

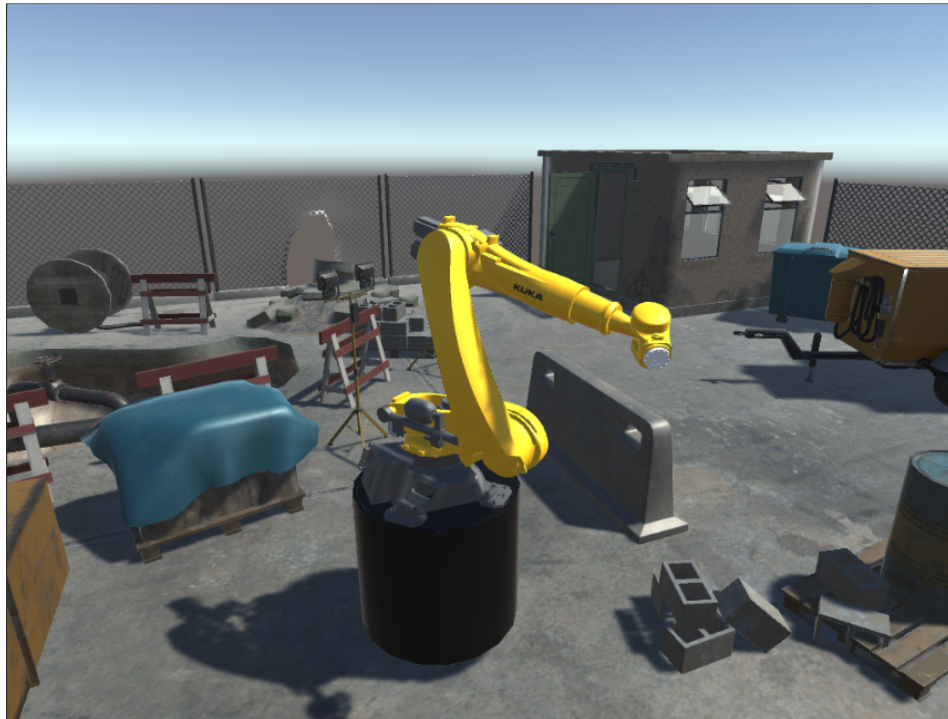
Simulacija robotske ruke u Unity-u

Dominik Brdar

Motivacija

Ovaj Unity projekt napravio sam u sklopu vježbe na predmetu Računalna animacija, Fakultet elektrotehnike i računarstva, sveučilište u Zagrebu.

Osnovna motivacija je bila napraviti simulaciju stvarnog robota KUKA koji se koristi u proizvodnji raznih naprava. Cilj vježbe je predstavljanje robota u virtualnoj sceni i implementacija upravljanja kinematičkim lancem robotske ruke uz poštovanje fizčkih ograničenja iz stvarnog svijeta.



Opis projekta

Organizacija u Unity projektu se sastoji od asset-a (modela objekata sa geometijskom i grafičkom logikom), virtualne scene koja se sastoji od terena, osvjetljenja, kamere za predstavljanje pogleda u 3D scenu i objekata među kojima se mogu odvijati interakcije te konačno programskih skripti pisanih za Unity okruženje u programskom jeziku C# koje služe za obradu događaja kao što su ulazne upute od korisnika (upravljanje tijekom izvođenja simulacije) i interakcija među objektima koje ovise o njihovom trenutnom stanju, položaju i drugim zadanim pravilima, na primjer, simulacija fizike.

Opis realizacije

Interaktivni simulacijski sustav korisniku omogućuje upravljanje robotskom rukom pomoću tipkovnice, i okretanje pogleda u scenu pomoću miša. Pogled u scenu je ostvaren na način da je točka gledišta fiksirana na robota, a očiste se pomicanjem miša okreće oko gledišta uz stalnu udaljenost.

Upravljanje robotskom rukom se zasniva na konceptu upravljanja kinematičkim lancem direktnom kinematikom. Pritiskom na numeričke tipke tipkovnice (1, 2, 3, 4, 5) aktiviraju se određeni zglobovi robota kojim se zatim upravlja (rotacijski zglobovi se zakreću oko svojih osi rotacije) tipkama strelica. Prilikom svakog iscrtavanja određuje se stanje navedenih tipki te se odabrani zglobovi pomiču zadanom brzinom u odabranom smjeru. Opisana funkcionalnost ostvarena je u skripti `ControlJoint1DOF`.

Objekti udova organizirani su u stablo, tako da se zadane transformacije zglobova prenose djeci tog zgloba. Također, implementirao sam funkciju kojoj se može predati rječnik <zglob, kut zakretanja> koja animira robotsku ruku za zadane parametre. Budući da je glavna motivacija bila simulacija stvarnog sustava, napisao sam posebnu skriptu koja se može dodijeliti svakom zglobov posebno (`ControlJoint1DOF`). Na taj način možemo svakom zglobov posebno zadati osi rotacije, ograničenja kao što su maksimalni i minimalni kut rotacije ili duljina pomaka za translacijske zglobove, brzinu i maksimalno opterećenje. Maksimalno opterećenje određuje maksimalnu masu objekta kojeg je moguće pomaknuti s robotskom rukom, što se provjerava prilikom detekcije sudara.

Osvrt na rezultat

Ostvario sam modularno rješenje direktne kinematike kinematičkih lanaca, koje sam primijenio za simuliranje stvarnog robota KUKA, poštivajući tehničkih obilježja. Za referencu sam koristio podatke iz dokumenta:

https://www.kuka.com/-/media/kuka-downloads/imported/6b77eecacfe542d3b736af377562ecaa/0000189663_en.pdf?rev=11595cfd397c4c2a921a9975a8f9efa6&hash=3A6343E4A4D7CBFFDCE75B9F02D7A04E

Prilikom izrade vježbe bilo je potrebno obratiti pozornost na graf scene, koordinatne sustave objekata (na primjer, oko koje osi se rotira zglob i je li pravilno centriran, je li pravilno pozicioniran u koordinatnom sustavu roditelja). Korisnik u Unity editoru može učitati bilo koje modele objekata, postaviti ih u strukturu stabla i pozicionirati u 3D virtualnoj sceni (pomoću *scene graph* sučelja i prikaza virtualne scene). Svakom zglobov može dodati skriptu za upravljanje tim zglobov `ControlJoint1DOF`. Putem *Inspector* sučelja u Unity editoru, moguće je jednostavno zadavanje ograničenja svakom zglobov

posebno bez intervencija u kodu. Zatim dodajemo skriptu `DirectKinematic` roditeljskom objektu robota i povučemo reference na objekte zglobova. Iako unity engine sam obrađuje kolizije potrebno je postaviti obujmice i dobro testirati dolaze li različiti načini detekcije sudara u konflikte te ispraviti takve probleme. Podržano je interaktivno upravljanje robotom tijekom simulacije kao i animiranje unaprijed zadanih parametara (funkcija `rotate` u skripti `ControlJoint1DOF`). Dio sustava koji simulira fiziku je napravljen na jednostavan način, ali vizualno daje dobre rezultate. Upravo u tome postoji najviše prostora za unapređenje, umjesto provjere mase objekta kojeg ruka može ili ne može pomaknuti bilo bi bolje rafinirati sa silama, no to je daleko složenije i izvan dosega kompleksnosti ove vježbe. Također, u ostvarenom sustavu možemo jednostavnije dodati funkcionalnost inverzne kinematike, što je u planu za budućnost ovog projekta kao i dodavanje grafičkog sučelja za unos parametara animacije.

Upute za postavljanje projekta

Čitava mapa projekta zapakirana je u .zip datoteku i potrebno ju je otpakirati.

Korištena verzija unity editora: 2019.4.17f1

Pomoću Unity Hub aplikacije dodajte projekt (odaberite mapu koju ste otpakirali).

U mapi Assets se nalaze skripte koje sam ja pisao za ostvarivanje funkcionalnosti.

Za izradu projekta koristio sam pakete asseta preuzete sa Unity Assets store-a:

Prespective Robot Pack:

<https://assetstore.unity.com/packages/3d/props/industrial/perspective-robot-pack-161711>

Standard Assets:

<https://assetstore.unity.com/packages/essentials/asset-packs/standard-assets-for-unity-2018-4-32351>

Station_v3:

<https://assetstore.unity.com/packages/3d/environments/industrial/station-v2-104479>

License agreement

[Standard Unity Asset Store EULA](#)

Pakete je potrebno preuzeti i uključiti u projekt pomoću Unity Editor, Package Managera