

UPRAVLJANJE ELEMENTIMA KORISNIČKOG SUČELJA VIRTUALNE STVARNOSTI BEZ FIZIČKOG KONTAKTA S UNOSNIM UREĐAJEM U OKVIRU APLIKACIJE ZA UČENJE

CONTROLLING THE USER INTERFACE OF VIRTUAL REALITY APPLICATION FOR LEARNING USING CONTACTLESS INPUT DEVICE

Nikola Poljanec, Mihael Kukec

Stručni rad

Sažetak: U radu je opisana izrada prototipne aplikacije za edukativne svrhe korištenjem beskontaktnog unosnog uređaja, prilikom čega je opisan sam uređaj, prototip aplikacije i način na koji radi. Prototip aplikacije simulira jednostavno sastavljanje računala u okviru sustava virtualne stvarnosti. Beskontaktni unosni uređaj koji je pri tome korišten je „Leap Motion“ koji prenosi prirodne pokrete ruke u virtualni svijet tako da korisnik može upravljati elementima korisničkog sučelja na intuitivan način, korištenjem ruku i prstiju. Proces sastavljanja računala u okviru prototipne aplikacije pojednostavljen je tako da se ugrađuju samo najvažnije komponente. U radu se opisuju korišteni alati i prikazuje kako se „Leap Motion“ uređaj koristi i integrira u program za izradu igara i aplikacija. Kako bi prototip aplikacije funkcionirao u navedenoj primjeni, izmijenjeno je i doručeno postojeće korisničko sučelje i kôd za integraciju uređaja. U procesu izrade rada istražene su mogućnosti uređaja, način postavljanja i instalacije i sama ograničenja uređaja i takve tehnologije.

Ključne riječi: detekcija pokreta, virtualna stvarnost, sustavi za učenje

Professional paper

Abstract: The work describes the development of prototype applications for educational purposes using contactless input device. The prototype application simulates a simple assembly of computer systems within the virtual reality system. Contactless input device used is the "Leap Motion" device which conveys the natural movements of the hand in the virtual world so that users can manage the user interface elements in an intuitive way. The process of assembling computer within prototype applications is simplified, it includes only the most important components. The paper describes the tools used and shows how the "Leap Motion" device is used and integrated into the application. To make the prototype application functional for the specified purpose, we have modified and upgraded existing user interface and program code for integration of device. In the process of making the work we have studied the capabilities of the device, together with basic limitations of such devices and technologies.

Keywords: motion detection, virtual reality, learning systems

1. UVOD

Aplikacijski prototip koji će biti opisan u ovom radu prikazuje i demonstrira osnove interakcije u virtualno stvorenom svijetu prenošenjem prirodnog pokreta ruke u računalo. Pri tome se koristi uređaj za praćenje pokreta naziva „Leap Motion“. Tehnologija praćenja pokreta donedavno je bila skupa i neprecizna, no pojavom „Leap Motion“ uređaja na tržištu cijena se smanjuje na razinu cijene standardnih unosnih uređaja te je tako postala dostupna i zanimljiva širem skupu krajnjih korisnika, što nam je omogućilo demonstriranje i ispitivanje mogućnosti te načina korištenja i implementiranja programskih sučelja uređaja na konkretnom primjeru primjene.

Razvijena prototipna aplikacija uz samo ispitivanje načina rada tehnologije praćenja pokreta ruke i prstiju ima i drugu važnu ulogu. Njome se prikazuju mogućnosti razvoja suvremenih virtualnih svjetova u području učenja

i podučavanja, posebno na području sklapanja većih sustava ili proizvoda od manjih komponenata. Sam prototip je izrađen na primjeru sklapanja stolnog računala pa aplikacija ovog tipa može poslužiti za edukaciju ljudi koji nemaju iskustva sa sklopovljem računala i njihovim sklapanjem. Ovakav tip aplikacije moguće je primijeniti u edukaciji (npr. u osnovnim školama), za učenje osnova o sklopovlju računala, o računalnim komponentama i postupku sastavljanja računala. Iako učenje u virtualnom okruženju ne može zamijeniti učenje u stvarnom svijetu, virtualno okruženje nam pruža mogućnost bržeg i čak zabavnijeg učenja koje može biti prošireno i dalje od samog procesa koji se pokušava simulirati.

Tehnologija uređaja „Leap Motion“ razmjerno je nova i unatoč određenim nedostacima ima široki spektar primjene. Zanimljivost upotrebe takve tehnologije dolazi od same ideje da računalo možemo kontrolirati rukama, što je promjena u odnosu na dosadašnju praksu korištenja

računala na klasičan način – mišem i tipkovnicom. Takav način upravljanja može biti koristan mnogim djelatnostima i strukama i otvoriti nove mogućnosti interakcije koja je intuitivna.

U narednim odlomcima opisani su glavni postupci rada, alati i programi korišteni u izradi rada, detalji programskog koda, kao i sam opis aplikacije, te se uz to objašnjava i prikazuje i sam uređaj „Leap Motion“ i tehnologija na kojoj radi.

2. PROGRAMSKI ALATI KORIŠTENI ZA RAZVOJ APLIKACIJE

Alati i programi korišteni u ovom radu su „Blender“ (verzija 2.73a), „Unity 4“ (Unity 3D platforma za razvoj interaktivnih 3D sadržaja verzije 4.6.2f1), program Paint.net (verzija 4.0.5) i razvojna okolina naziva „Monodevelop“ (verzija 4.0.1).

2.1. Programski alat „Blender“

„Blender“ je besplatan alata otvorenog kôda za stvaranje 3D računalne grafike. Koristi se za izradu animiranih filmova, vizualnih efekata, 3D virtualnih objekata, interaktivnih 3D aplikacija i video igara. [1]

„Blender“ u sebi uključuje module za 3D modeliranje, teksturiranje, uređivanje rasterske grafike, simulaciju fluida i dima, simulaciju čestica, simulaciju mekih tijela, oblikovanje, tj. skulpturiranje, animiranje, praćenje kamere, renderiranje te video uređivanje i komponiranje [1].

2.2. Okolina za stvaranje računalnih 3D igara „Unity 3D“

Okolina za stvaranje računalnih igara, „Unity game engine“ ili skraćeno „Unity“, višepatformski je softver za stvaranje igara, koji razvija tvrtka „Unity Technologies“. Alat uključuje platformu za razvoj igara i razvojnu okolinu (engl. Integrated development environment - IDE). Iako početno zamišljen kao alat za isključivo Mac OS sustave, proširio se na veći broj različitih platformi i postao primarni programski alat za razvoj (engl. software development kit - SDK) na Nintendo Wii U konzoli. [2]

Najprepoznatiji je zbog mogućnosti izrade igara ili aplikacije na više platformi paralelno (npr. mobilni uređaji, mrežni preglednici, osobna računala i konzole) [2].

Neke od glavnih značajki su uključena baza resursa, uključena podrška za tehnologije Direct3D, OpenGL i OpenGLES. Unutar samog programa postoje opcije za kompresiju teksture i postavke rezolucije za svaku platformu koju podržava. Materijali imaju podršku za „bump mapping“, „reflection mapping“, „parallax mapping“, sjena (engl. Shadowmaps), „render-to-texture“ (projiciranje slike na teksturu), dok od ostalih efekata podržava „screen space ambient occlusion“ (SSAO), dinamičke sjene i mape i „full-screen post processing“ efekte. [2,3]

2.3. Razvojni alat „Mono Develop“

Platforma za razvoj interaktivne 3D grafike i računalnih igara „Unity“ dolazi uz svoj program za uređivanje programskog koda pod nazivom „Mono Develop“. Sami podržani programski jezici izgrađeni su na temelju Mono implementacije „.NET Frameworka“. Programeri mogu upotrebljavati programski jezik „UnityScript“ (prilagođena verzija ECMAScript sintakse), programski jezik C# ili programski jezik Boo koji je orijentiran na sintaksu programskog jezika Python [4].

3. ULAZNI UREĐAJ ZA DETEKCIJU POKRETA

Prvi uređaj koji je koristio tehnologiju pokreta bila je konzola NintendoWii koja se primarno koristila u računalnim igrama. Tu tehnologiju kasnije je unaprijedila tvrtka „Microsoft“ razvivši sustav naziva „Kinect“. Kinect je za razliku od praćenja kontrolera, kojeg korisnik drži u ruci, pratio kretanje cijelog tijela. Nova generacija uređaja stvorena je najavom „Leap Motion“ uređaja. Radi na sličnom principu kao i Microsoftov Kinect, samo što je fokusiran isključivo na ruke [5-7].



Slika 1. Uređaj „Leap Motion“

„Leap Motion“ je maleni uređaj koji se spaja pomoću USB kabla u računalo. Sam uređaj je dimenzija 8 cm x 3 cm x 1.2 cm (Slika 1). Sadrži 3 kamere i infracrvena senzora koji detektiraju pokret s preciznošću od 1/100 mm te ima kut pogleda od 150 stupnjeva. Tako preciznom detekcijom pokreta moguće je stvoriti precizne slike i crteže na ekranu. Polje interakcije od 150 stupnjeva omogućava korisniku slobodno pomicanje ruku u 3D prostoru, što znači da „Leap Motion“ aplikacije pružaju mogućnost da se (virtualni) objekti pomiču i da se njima manipulira baš kao u pravome svijetu. [5,6]

3.1. Instalacija i korištenje

Proces instalacije je jednostavan. Nakon priključenja uređaja na računalo putem USB kabla, uređaj treba postaviti ispred tipkovnice (prijenosna računala) ili između ekrana i tipkovnice (stolna računala). Nakon toga potrebno je preuzeti i instalirati softver sa službene stranice. Softver će korisnika upoznati s načinom na koji

uređaj radi i kako prati svih 10 prstiju u virtualnom prostoru iznad uređaja. Isto tako, s instalacijom pogonskih programa uređaja dolazi i program za kupovinu ostalih aplikacija za korištenje s „Leap Motion-om“.



Slika 2. Korištenje „Leap Motion“ uređaja unutar platforme za razvoj igara „Unity“

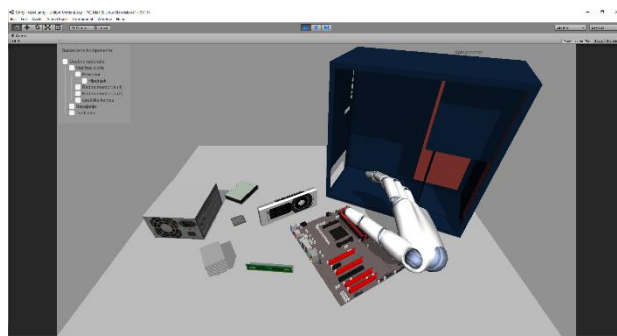
Za najbolje iskustvo prilikom korištenja uređaja preporučena visina interakcije s uređajem je 20 do 30 cm u slučaju kada je uređaj položen na stol, tj. na vodoravnu površinu, što je prikazano na Slici 2. Uređaj se može koristiti sa zaslonom pričvršćenim na glavu korisnika (engl. Head mounted display - HMD), kao što je „Oculus Rift“, i u tome slučaju uređaj se postavlja na sam HMD tako da gleda prema rukama korisnika, tj. prema naprijed.

3.2. Ograničenja uređaja „Leap Motion“

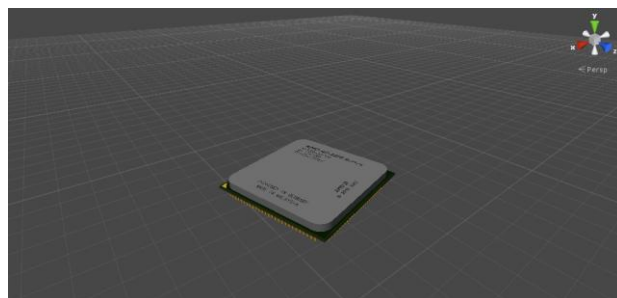
Glavni izvor ograničenja uređaja je nužnost optičke vidljivosti objekta koji se očitava, tj. za kojeg se pokušava detektirati pozicija u prostoru. Ukoliko se rukama izađe izvan polja dosega uređaja, gubi se vidljivost, tj. uređaj „ne vidi“ ruke te prestaje njihovo praćenje. Isto tako, ukoliko se uređaj koristi u visini između 5 i 15 cm značajno se smanjuju širina i dubina polja u kojem uređaj „vidi“ ruke, a kako kamere uređaja gledaju u samo jednom smjeru (okomito na prednju površinu uređaja) sve što se nalazi iza kamere postaje zaklonjeno, tj. uređaju nije vidljivo. Kako je potrebno osigurati optičku vidljivost, uređaj isto tako ne vidi objekte koji su zaklonjeni nekim drugim objektom. Primjerice, u slučaju kada korisnik skupi (stisne) šaku, uređaj „Leap Motion“ najvjerojatnije neće to moći vidjeti i ispravno očitati. Razlog tome je to što su u skupljenoj šaci prsti djelomično zaklonjeni i skriveni pa ih uređaj ne može ispravno detektirati. Za takve slučajeve postoje modeli predviđanja o tome što korisnik radi i poze u kojoj bi šaka mogla biti. Unatoč tome, neželjena ponašanja mogu se dogoditi te rezultirati pogrešnim očitajima.

4. OSTVARENI PROTOTIP APLIKACIJE

Ovdje će se opisati sama aplikacija, način na koji funkcionira i kako je koristiti. Aplikacija se može kontrolirati uređajem „Leap Motion“ ili pokazivačem miša. Prozor aplikacije sastoji se od 3 glavna dijela: glavnog prozora, panela za prikaz sastavljenih dijelova i panela za prikaz detalja. Glavni prozor (virtualni stol) prikazuje radnu površinu sastavljanja računala, što je prikazano na Slici 3.



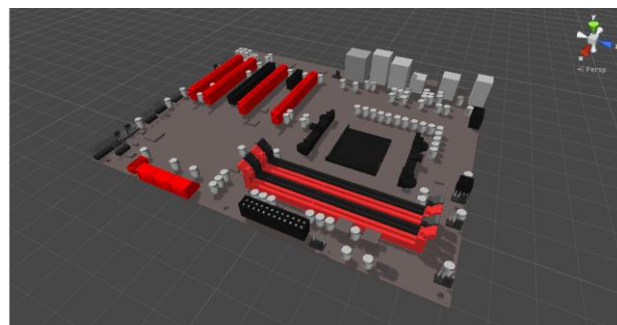
Slika 3. Glavni prozor aplikacije



Slika 4. Model procesora prikazan u okviru platforme „Unity“

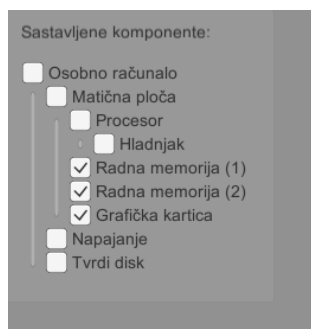


Slika 5. Model komponente radne memorije prikazane u okviru platforme „Unity“



Slika 6. Model matične ploče prikazane u okviru platforme „Unity“

Virtualno računalo sastavlja se od dostupnih komponenti, od kojih su neke prikazane na slikama 4, 5 i 6. Sastavljanje virtualnog računala u okviru izrađene aplikacije odvija se po redu, tj. hijerarhiji koja se prikazuje u gornjem lijevom kutu aplikacije (Slika 3), a detaljnije je vidljiva na Slici 7.



Slika 7. Prozor koji prikazuje koje komponente su sastavljene

Aplikacija implementira pravila za ispravno sastavljanje komponenti po hijerarhijskom redoslijedu pa tako, primjerice, prije nego matična ploča može biti ugrađena unutar kućišta potrebno je u nju ugraditi sve potrebne dijelove.

Kako bi se sastavilo virtualno računalo, najprije treba provjeriti je li uređaj pravilno spojen na računalo te je li ga pogonski program prepoznao, a nakon toga može početi sastavljanje. Komponenta se može uhvatiti gestom skupljene šake (engl. grab) ili gestom štipanja (engl. pinch). Da bi se tako odabrana komponenta ugradila na predviđeno mjesto, sve što treba je postaviti je iznad odgovarajućeg utora (engl. slot) i pričekati trenutak kako bi je aplikacija čvrsto pozicionirala na zadano mjesto. Prilikom prolaska, tj. prenošenja komponente iznad svakog utora, ukoliko je komponenta u dometu utora, na sučelju aplikacije iscrtat će se žuta linija od komponente do utora, čime se simbolizira da je komponentu moguće ugraditi u pojedini utor. Nakon uspješne ugradnje na hijerarhijskom prikazu, prikazanom na Slici 7, komponenta koja je ugrađena bit će označena.

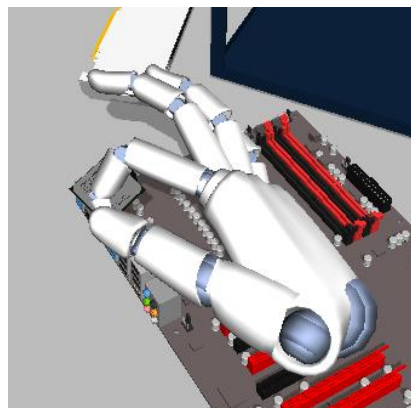
5. IMPLEMENTACIJSKI DETALJI

Sastavljeni prototip aplikacije, tj. scena, uključuje resurse sa paketa „Unity 3D platform integration“ u kojemu se nalaze resursi kao što su grafika za virtualne ruke, pripremljeni „Leap Motion“ kontroler (virtualni uređaj) i nekoliko scena koje služe kao primjeri. [5,6]

5.1. Upravljanje virtualnom rukom

Implementacija virtualne ruke i integracija s platformom „Unity“ funkcionira tako što postoji virtualni uređaj (engl. controller) koji odgovara stvarnome uređaju. Pomaci i podaci virtualnog uređaja mogu se po potrebi skalirati i time se dobiva razlika između stvarnog i virtualnog pomaka. U protivnome pomak odgovara stvarnome pomaku. Integraciju uređaja „Leap Motion“ i platforme „Unity“ olakšavaju gotovi izrađeni i dostupni resursi i programske skripte. Tako se novi virtualni uređaj

koji prenosi podatke s uređaja „Leap Motion“ integrira korištenjem i pozivanjem razreda iz „Leap Motion“ dinamički povezane knjižnice koda (engl. Dynamic-link library – DLL) [5, 6].



Slika 8. Hvatanje predmeta virtualnom rukom

Dostupni su i unaprijed pripremljeni skripte i objekti koji predstavljaju ruke. Tako pripremljene objekte odvlači se na odgovarajuće mjesto u parametrima skripte naziva „Hand Controller“. Virtualne ruke implementirane su kroz dva segmenta: grafika ruke i fizika ruke. Nakon što se podese oba segmenta odabirom grafike i fizike, sustav je spreman za korištenje podataka dobivenih s uređaja, koji se pak u virtualnom svijetu reprezentiraju grafičkim prikazom virtualnih ruku (Slika 8).

5.2. Ulazna točka aplikacije i inicijalizacija

Ulazna točka u implementiranu aplikaciju, tj. glavna skripta aplikacije, naziva se „GameControl“ skripta te je zadužena za upravljanje svim ostalim i može se nazvati „upraviteljem“ aplikacije. Prikazuje sastavljene objekte na ekranu, upravlja sučeljem za prikazivanje više detalja o komponentama te ispituje i prati koja komponenta je sastavljena i jesu li sastavljene grupe komponentata (procesor, matična ploča i PC kućište).

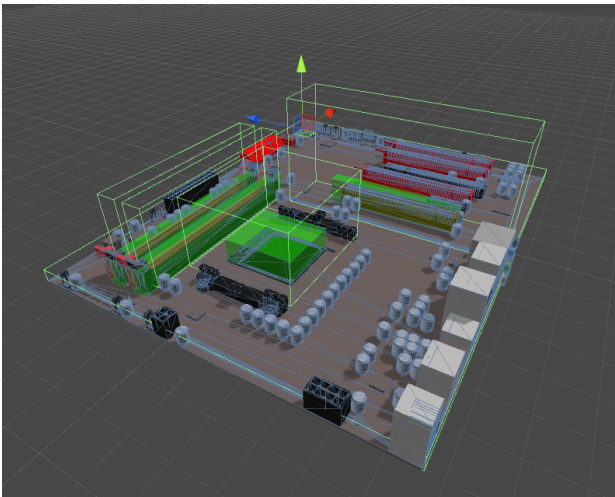
```
void Start () {
    Instance = this;
    komponenteField = GameObject.FindGameObjectsWithTag("Komponenta");
    pcField = GameObject.FindGameObjectsWithTag("PC");
    holderField = GameObject.FindGameObjectsWithTag("Holder");
    MTBComponentsField = GameObject.Find("MTB");
    CPUComponent =
    GetComponentInChildren<PullObject>();
    GameObject.Find("CPU").GetComponentInChildren<PullObject>();
    caseComponentField = GameObject.Find("PC_case");
    GetComponentInChildren<PullObject>();
    leapController = GameObject.FindObjectOfType<HandController>();
    foreach (GameObject o in komponenteField) {
        if (o.name == "MTB") {
            MTB = o;
            Destroy(o.rigidbody, 0.5f);
            break;
        }
    }
    foreach (GameObject o in holderField) {
        if (o.name == "mtb_holder") {
            MTB_target = o;
            break;
        }
    }
    foreach (GameObject o in pcField) {
        if (o.name == "PC_case") {
            PC_case = o;
            break;
        }
    }
    rightImage.CrossFadeAlpha(0f, 1f, true);
    ShowHelpBox("null");
}
```

Slika 9. Programski kôd ulazne točke u aplikaciju, metode „Start“

Metoda „Start“, prikazana na Slici 9, poziva se prije iscertavanja tako da je pogodna za inicijalizacijski programski kôd. Prvo se učitavaju sve komponente i odgovarajući utori u polja kako bi im se moglo pristupiti kasnije. Učitavanje u polje odvija se tako što se pretražuju svi objekti u sceni te se dohvaćaju prema odgovarajućoj oznaci (engl. „tag“). Nakon toga se kroz petlju iz liste izvlače glavni objekti u varijable i na kraju se postavljaju podaci u panel za prikaz detalja prikazan na Slici 7.

5.3. Komponente virtualnog računala

Komponente virtualnog računala od kojih su neke prikazane na slikama 4, 5 i 6 nose karakteristike te imaju mogućnosti iscertavanja, što omogućava karakteristika naziva „Mesh Renderer“. Nadalje, sve komponente koriste mogućnosti detekcije sudara s drugima, što je implementirano karakteristikom naziva „Mesh Collider – Rigidbody“. Karakteristika naziva „Selected“ omogućava interakciju s virtualnim komponentama.



Slika 10. Matična ploča na kojoj su vidljivi prošireni rubovi (engl. Collider) koji služe kao okidači

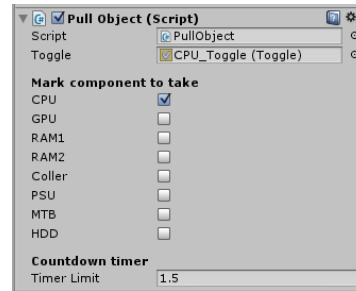
Karakteristika „Mesh Collider“ omogućava fiziku objekta, čime objekti u virtualnom prostoru dobivaju fizikalne karakteristike, tj. ponašanje kakvo se očekuje u stvarnom, fizičkom svijetu. Tom karakteristikom komponentama se definira prostor oko njih koji omogućava fizičku interakciju s drugim objektima ili taj prostor može služiti kao okidač određenih akcija (Slika 10). Moguće je detektirati sve objekte koji dodiruju, ulaze, ostaju unutar ili izlaze izvan granica proširenih rubova drugih objekata.

Karakteristika naziva „Rigidbody“ komponentama omogućava djelovanje gravitacije, tj. možemo reći da daje masu objektu. Boja objekta određuje se karakteristikom naziva „Material“. Ona ujedno određuje metodu sjenčanja koji se koristi na objektu, moguću teksturu (ili više njih) te se pomoću nje podešavaju svojstva svakog virtualnog materijala primijenjenog na 3D objekt.

5.4. Sastavljanje komponenata

Skripta naziva „PullObject“ upravlja prihvaćanjem određene komponente na određeni utor. Time je osigurano

ispravno spajanje komponenata virtualnog računala te se ne može dogoditi da se komponenta postavi na neodgovarajući utor. Tako, primjerice, utor za procesor može prihvatiti samo komponentu procesora (Slika 11).



Slika 11. Skripta „PullObject“ u inspektor prozoru – utor uzima samo CPU komponentu

```
void OnTriggerStay(Collider c)
{
    if (c.tag == "Komponenta" && !_assembled)
    {
        Debug.DrawLine(c.transform.position, transform.position,
        Color.blue);
        lr.SetPosition(0, transform.position);
        lr.SetPosition(1, c.transform.position);

        curObjectType = TypeOfObject(c.gameObject);

        if (curObjectType == takingComponentType)
        {
            timer += Time.deltaTime;
            //Debug.Log("timer: " + timer);

            if (timer >= timerLimit)
            {
                SnapObject(c.gameObject);
                if (curObjectType == Helper.Components.CPU)
                {
                    GameObject bla =
                    c.transform.Find("Coller_socket").gameObject;
                    bla.GetComponent<BoxCollider>().enabled = true;
                    bla.GetComponent<PullObject>().enabled = true;

                    GameObject bla1 = bla.transform.GetChild(0).gameObject;
                    bla1.SetActive(true);
                }
            }
        }
    }
}
```

Slika 12. Programski kôd koji upravlja interakcijom objekta

Programski kôd koji upravlja interakcijom objekta, prikazan na Slici 12, detektira koliziju između objekata – okidača koji predstavlja utor i same računalne komponente koju pokušavamo priključiti na utor. Ukoliko se detektira ulazak objekta unutar okidača, prvo se provjerava je li objekt tipa „Komponenta“ te se nakon toga provjerava je li objekt kompatibilan s utorom. Ukoliko je i to zadovoljeno, nakon 1.5 sekundi (postavka „Timer Limit“ vidljiva na dnu Slike 11) komponenta se postavlja na utor i označava kao sastavljena. [8]

5.5. Implementacija upravljanja korisničkog sučelja mišem

Zbog potreba ispitivanja rada aplikacije kada nije dostupan uređaj „Leap Motion“, implementirano je upravljanje korisničkim sučeljem skriptom za rad s mišem. Skripta naziva „MouseControl“ pruža alternativu sklapanja računala s uređajem „Leap Motion“. Ona sadrži programski kôd koji pruža manipulaciju komponenti s pokazivačem miša. [8]

Skripta kojom se reagira na događaje i akcije pokrenute mišem prikazana je na Slici 13. U metodi naziva „OnMouseDown“ nalazi se programski kôd koji omogućava akciju ekvivalentu uzimanju u ruku,

omogućava hvatanje komponente te njeno odnošenje i pozicioniranje iznad utora. Kako je manipulacija pokazivačem miša dvodimenzionalni način kontrole, kotačić miša koristi se za promjenu visine objekta. Metoda „OnMouseDown“ poziva se kada korisnik pusti tipku miša.

```

void OnMouseDown() {
    glowObject.renderer.material.color =
        new Color(defColor.r, defColor.g, defColor.b, 0.275f);
}

void OnMouseExit() {
    glowObject.renderer.material.color =
        new Color(defColor.r, defColor.g, defColor.b, 0f);
}

void OnMouseDown() {
    height += Input.GetAxis ("Mouse ScrollWheel");
    if (height < 5.9f) height = 5.9f;
    else if (height > 9f) height = 9f;

    Vector3 mousePos = MouseControl.mouse3DPosition;
    selectable = false;
    Screen.showCursor = false;
    if (rigidbody != null) {
        rigidbody.isKinematic = true;
        transform.position = new Vector3(mousePos.x, height, mousePos.z);
    }

    GameController3.Instance.SetCurentObject(gameObject.name);
    GameController3.Instance.ShowHelpBox(gameObject.name);
}

void OnMouseDown() {
    selectable = true;
    height = 6.4f;
    Screen.showCursor = true;

    GameController3.Instance.SetCurentObject("");
    GameController3.Instance.ShowHelpBox("");

    if (gameObject.rigidbody != null) {
        rigidbody.isKinematic = false;
        rigidbody.AddForce(Vector3.up);
    }
}

```

Slika 13. Programski kôd metoda koje se aktiviraju akcijama miša

6. ZAKLJUČAK

Ovim radom prikazuje se kako jednostavna aplikacija koristi mogućnosti nove tehnologije. Uređaj „Leap Motion“ omogućuje korisniku upravljanje računalom na intuitivan način, svojim rukama, čime se postiže veći stupanj uranjanja u virtualni svijet. Oponašanjem stvarnog svijeta pokušava se smanjiti jaz u interakciji između čovjeka i računala, čineći tako cijeli proces učinkovitijim i ugodnijim za korisnika.

Potrebno je istaknuti kako primjena ovakvog načina rada sa korisničkim sučeljem računala ima puno širu primjenu nego je prikazana u ovome radu. Moguće ga je primijeniti na interaktivne izloge u trgovini, manipuliranja objektima u softverima za izradu 3D modela, prezentiranja ne konferencijama, igranje igara, pomaganja ljudima s otežanom pokretljivošću itd.

Izrađena prototipna aplikacija, prikazana ovim radom, pruža mogućnost korisniku da svojim rukama manipulira komponente računala u tri dimenzije. Iako se najbolje iskustvo dobiva sklapanjem pravog računala, virtualno računalo možemo predočiti i priuštiti sklapanje različitih komponenti u različite sustave koji nam inače zbog svoje cijene ne bi bili dostupni. Nadalje, moguće je stvoriti i ostvariti simulacijsku okolinu u kojoj akcije nemaju svoje ishode u stvarnom svijetu, primjerice, virtualne komponente računala nije moguće (trajno) oštetiti, što onome koji uči i eksperimentira daje dodatnu slobodu. Uz to, unutar virtualnog okruženja korisnik je oslobođen straha od ozljede ili kvara i ima mogućnost slobodnog eksperimentiranja. Ostale primjene u edukaciji mogu biti

od najjednostavnijeg učenja do prikazivanja interaktivnog učenja povijesti, matematike, fizike ili ostalih predmeta.

7. REFERENCE

- [1] Blain, J. M: The Complete Guide to Blender Graphics, Second Edition: Computer Modeling and Animation, A K Peters/CRC Press, Boca Raton Florida, 2014
- [2] <http://unity3d.com/unity>, (Accessed: 23.06.2015.)
- [3] Goldstone, W.: Unity 3.x Game Development Essentials, Packt Publishing, Birmingham, 2011
- [4] Okita, A.: Learning C# Programming with Unity 3D, Routledge, Oxford, 2014
- [5] Sanders, B.: Mastering Leap Motion, Packt Publishing, Birmingham, 2014
- [6] Spiegelmock, M.: Leap Motion Development Essentials, Packt Publishing, Birmingham, 2013.
- [7] <https://www.leapmotion.com/product>, (Accessed: 15.7.2015.)
- [8] <http://docs.unity3d.com/ScriptReference/>, (Accessed: 11.9.2015)

Authors` contacts:

Nikola Poljanec

Student Međimurskog veleučilišta u Čakovcu
B. J. Jelačića 22a
40000 Čakovec

Mihael Kukec, PhD

Međimursko veleučilište u Čakovcu
B. J. Jelačića 22a
40000 Čakovec