

SVEUČILIŠTE JOSIPA JURJA STROSSMAYERA U OSIJEKU

FAKULTET ELEKTROTEHNIKE, RAČUNARSTVA I

INFORMACIJSKIH TEHNOLOGIJA OSIJEK

Sveučilišni prijediplomski studij Računarstvo

INTERAKCIJA ČOVJEKA I RAČUNALA U VIDEOIGRAMA

Završni rad

Tomislav Celić

Osijek, 2024.

**FERIT**FAKULTET ELEKTROTEHNIKE, RAČUNARSTVA
I INFORMACIJSKIH TEHNOLOGIJA OSIJEK**Obrazac Z1P: Obrazac za ocjenu završnog rada na sveučilišnom prijediplomskom studiju****Ocjena završnog rada na sveučilišnom prijediplomskom studiju**

Ime i prezime pristupnika:	Tomislav Celić
Studij, smjer:	Sveučilišni prijediplomski studij Računarstvo
Mat. br. pristupnika, god.	R4627, 27.07.2021.
JMBAG:	0165090491
Mentor:	doc. dr. sc. Bruno Zorić
Sumentor:	
Sumentor iz tvrtke:	
Naslov završnog rada:	Interakcija čovjeka i računala u videoigrama
Znanstvena grana završnog rada:	Programsko inženjerstvo (zn. polje računarstvo)
Zadatak završnog rada:	U teorijskom dijelu rada potrebno je opisati pojam interakcije čovjeka i računala te njegove glavne značajke. Posebnu pozornost posvetiti mehanizmima interakcije koji se pojavljuju i primjenjuju u videoigrama te prikazati primjere iz prakse i njihova ograničenja. U praktičnom dijelu rada potrebno je ostvariti programsko rješenje u obliku videoigre koja rabi nekoliko različitih mehanizama za interakciju prilikom igranja te je nužno rješenje prikladno vrednovati. (Tema rezervirana za: Tomislav Celić)
Datum prijedloga ocjene završnog rada od strane mentora:	17.09.2024.
Prijedlog ocjene završnog rada od strane mentora:	Izvrstan (5)
Datum potvrde ocjene završnog rada od strane Odbora:	27.09.2024.
Ocjena završnog rada nakon obrane:	Izvrstan (5)
Datum potvrde mentora o predaji konačne verzije završnog rada čime je pristupnik završio sveučilišni prijediplomski studij:	28.09.2024.

**FERIT**FAKULTET ELEKTROTEHNIKE, RAČUNARSTVA
I INFORMACIJSKIH TEHNOLOGIJA **OSIJEK****IZJAVA O IZVORNOSTI RADA**

Osijek, 28.09.2024.

Ime i prezime Pristupnika:

Tomislav Celić

Studij:

Sveučilišni prijediplomski studij Računarstvo

Mat. br. Pristupnika, godina upisa:

R4627, 27.07.2021.

Turnitin podudaranje [%]:

2

Ovom izjavom izjavljujem da je rad pod nazivom: **Interakcija čovjeka i računala u videoigrama**

izrađen pod vodstvom mentora doc. dr. sc. Bruno Zorić

i sumentora

moj vlastiti rad i prema mom najboljem znanju ne sadrži prethodno objavljene ili neobjavljene pisane materijale drugih osoba, osim onih koji su izričito priznati navođenjem literature i drugih izvora informacija.

Izjavljujem da je intelektualni sadržaj navedenog rada proizvod mog vlastitog rada, osim u onom dijelu za koji mi je bila potrebna pomoć mentora, sumentora i drugih osoba, a što je izričito navedeno u radu.

Potpis pristupnika:

SADRŽAJ

1.	UVOD	1
1.1.	Zadatak završnog rada	1
2.	INTERAKCIJA ČOVJEKA I RAČUNALA	3
2.1.	Normanov ciklus izvršenja i evaluacije	4
2.2.	Okvir interakcije	5
2.3.	Sučelje	7
2.4.	Pristupačnost	8
2.5.	Interakcija čovjeka i računala u videoigrama	9
2.5.1.	Procesi za razliku od rezultata	9
2.5.2.	Definiranje ciljeva za razliku od uvoza cilja	9
2.5.3.	Par alternativa za razliku od puno alternativa	10
2.5.4.	Dosljednost za razliku od varijacije	11
2.5.5.	Micanje/strukturiranje ograničenja za razliku od namjernog dodavanja ograničenja 12	
2.5.6.	Funkcionalnost za razliku od raspoložena	13
2.5.7.	Izgled ishoda za razliku od izgleda svijeta	13
2.5.8.	Organizacija za razliku od pojedinca	14
2.5.9.	Forma slijedi funkciju za razliku od funkcija slijedi formu	14
2.5.10.	Standardni ulazni uređaji za razliku od različitih ulaznih uređaja	14
2.6.	Postojeća rješenja	16
2.6.1.	Sony EyeToy	16
2.6.2.	Xbox Kinect	17
2.6.3.	Wii Remote	18
2.6.4.	Xbox Adaptivni Kontroler	18
3.	DIZAJN I IZRADA VIDEOIGRE ZA USPOREDBU METODA UNOSA	20
3.1.	Odabir žanra i osnovne funkcionalnosti	20
3.2.	Korišteni alati i tehnologije	20
3.2.1.	Unity	20
3.2.2.	OpenCV plus Unity	21
3.3.	Prikaz osnovnih dijelova videoigre	21
3.3.1.	Prepreke i uvjeti neuspjeha	21
3.3.2.	Bonusi i rezultat igrača	22
3.4.	Sustav interakcije	24
3.4.1.	Dizajn sustava za odabir interakcije	24

3.4.2.	Interakcija tipkovnicom/Xbox kontrolerom.....	28
3.4.3.	Detekcija lica.....	28
3.4.4.	Interakcija detekcijom pozicije lica.....	29
3.4.5.	Interakcija detekcijom pokreta lica	30
3.4.6.	Interakcija detekcijom boje	31
3.5.	Vrednovanje igre i metode unosa	34
3.5.1.	Pitanja.....	34
3.5.2.	Rezultati upitnika i interpretacija	34
4.	ZAKLJUČAK	39
	LITERATURA.....	40
	SAŽETAK.....	44
	ABSTRACT.....	45
	PRILOZI.....	46

1. UVOD

U današnje vrijeme računala su svakodnevica, te su neophodna u poslovnim i privatnim svrhama. Čovjek preko računala ima uvid u virtualni svijet, pa zbog toga drugačije pristupa korištenju računala. Interakcija čovjeka i računala (engl. *Human-computer interaction*, HCI) je znanstvena disciplina koja proučava kako čovjek koristi računala, te kako dizajn istih utječe na lakoću korištenja. Znati kako ljudi pristupaju računalima, bitno je dizajnerima, razvojnim inženjerima i drugima kojima je cilj dizajnirati i stvoriti uspješne računalne sustave. Također, HCI je važna disciplina u pogledu pristupačnosti, jer ona također, istražuje kako ljudi s određenim invaliditetom interagiraju sa računalom.

Videoigre su računalni softver pa zbog toga imaju koristi od istraživanja interakcije čovjeka i računala. HCI nije ista u videoigrama kao kod ostalog softvera pa je iznimno važno razumjeti razlike, kako bi igraču bilo moguće ponuditi zabavna iskustva. Jedan od pristupa koji je bitan kod videoigara je proučavanje kako sučelje (ulaz i izlaz iz računala) utječe na igrača. Alternativnim metodama unosa moguće je igraču nuditi nova, zabavna iskustva, ali moguće je i igračima koji imaju fizičke invalidnosti nuditi opciju da igraju igre koje su prethodno bili spriječeni igrati. Zbog toga, već postoje uređaji stvoreni u svrhu nuđenja alternativnih metoda unosa. Cilj ovog rada je istražiti kako HCI utječe na videoigre, te vrednovati nekoliko alternativnih metoda unosa kroz izradu videoigre. Postoji puno različitih alternativnih metoda unosa, a samo neke od njih su korisne i zabavne igračima.

U drugom poglavlju pojašnjeno je što je HCI te su prikazani model i okvir za opis interakcije, a zatim je objašnjen pojam pristupačnosti. Nakon toga, prikazane su razlike videoigara i produktivnog/poslovnog softvera te su uz pomoć razlika objašnjeni daljnji pojmovi HCI-a. Na kraju drugog poglavlja prikazane su već postojeće tehnologije. U trećem poglavlju opisana je izrađena videoigra, čija je svrha nuditi alternativne metode unosa, a za kraj trećeg poglavlja je napravljeno testiranje izrađene videoigre na potencijalnim korisnicima. U četvrtom poglavlju napravljen je zaključak istraživanja, izrade videoigre kao i ideje za poboljšanja u budućim istraživanjima.

1.1. Zadatak završnog rada

U teorijskom dijelu rada potrebno je opisati pojam interakcije čovjeka i računala te njegove glavne značajke. Posebnu pozornost treba posvetiti mehanizmima interakcije koji se pojavljuju i primjenjuju u videoigrama te prikazati primjere iz prakse i njihova ograničenja. U praktičnom

dijelu rada potrebno je ostvariti programsko rješenje u obliku videoigre koja rabi nekoliko različitih mehanizama za interakciju prilikom igranja te je nužno rješenje prikladno vrednovati.

2. INTERAKCIJA ČOVJEKA I RAČUNALA

Interakcija čovjeka i računala znanstvena je disciplina koja proučava čovjeka, računalo i njihovu interakciju. Istraživači koji se bave ovom disciplinom pokušavaju naći nove, bolje načine interakcije čovjeka i računala. Disciplina je nastala 1980-tih, kada su osobna računala dobivala na popularnosti u široj populaciji [1]. Kako šira populacija, za razliku od tehničara koji su prolazili posebne obuke, nije dobivala obuku za korištenje računala [2], softver na računalima morao je biti intuitivan i smislen kako bi uspješno bio korišten. S vremenom, računala su poprimila više oblika, a čovjek ih je sve češće susretao. Danas postojanje pametnih telefona, pametnih televizora, softverskih sučelja u automobilima i sl. znači da veliki broj ljudi u svakodnevnicu susreće računala u različitim oblicima [3], te ih koristi na različite načine [4].

Glavni razlog postojanja HCI-a je taj što se ljudi i računala fundamentalno razlikuju. Računalo i čovjek moraju komunicirati [5] kako bi se interakcija mogla uspješno odviti. Problem je što računala razumiju strojni kod, dok ljudi pričaju ljudskim jezicima. Problem koji nastoji riješiti interakcija čovjeka i računala jest dizajniranje tehnologije s ciljem da bude korisna što širem spektru ljudi. Različiti ljudi imaju različite potrebe, sposobnosti i mogućnosti, te je potrebno sukladno dizajnirati računalni sustav prema njima.

Prve naznake HCI-a nastaju početkom 20. stoljeća, kada je industrijalizacija dovela do istraživanja produktivnosti radnika [5]. Tokom drugog svjetskog rata postojala je dodatna potreba za istraživanjem ergonomije, odnosno inženjeringa ljudskog faktora. Prvotno, istraživači su proučavali fizički dizajn sustava, te su kognitivne sposobnosti kao faktor proučavane kasnije. Isprva, disciplina se zvala „interakcija čovjeka i stroja“ [5], ali kasnije zbog kompleksnosti računala, pa tako i drugih strojeva, nastaju discipline poput interakcije čovjeka i računala, interakcije čovjeka i robota te slične.

Potrebno je razumjeti da se HCI ne bavi samo interakcijom jednog čovjeka i jednog računala. Čovjek u kontekstu discipline predstavlja pojedinca, ali i skupinu ljudi. Također, računalo predstavlja jedno računalo, u svim svojim oblicima (mobilna, prijenosna, stolna te druga), ali i skupine računala koje tvore jedan ili više računalnih sustava. Također, moguće je da čovjek preko računala ima interakciju s drugim ljudima [5] (društvene mreže, videoigre s više igrača i sl.), što i dalje spada unutar discipline.

Istraživanje interakcije čovjeka i računala je multidisciplinarno, a neke od znanstvenih grana/discipline koje imaju ulogu u istraživanju su [5]:

- Psihologija – psihička ograničenja čovjeka i njegove kognitivne mogućnosti
- Ergonomija – fizička ograničenja čovjeka
- Računarstvo – istraživanje novih algoritama i pristupa
- Programsko inženjerstvo – istraživanje potencijalnih implementacija koje rješavaju probleme koje su otkrile druge discipline
- Grafički dizajn – istraživanje novih načina prikazivanja sučelja

2.1. Normanov ciklus izvršenja i evaluacije

Najčešće korišten model za prikazivanje interakcije čovjeka i računala (iako se isti koristi za prikaz interakcije čovjeka s drugim stvarima) je *Normanov ciklus izvršenja i evaluacije*. U svojoj knjizi „*Design of everyday things*“, Donald Norman tretira interakciju kao ciklus izvršenja i evaluacije [6], gdje dodatno dijeli izvršenje i evaluaciju na:

- Izvršenje:
 - o Postavljanje cilja – čovjek želi ostvariti neki cilj, poput kontaktiranja poslovnog partnera.
 - o Formiranje namjere – čovjek namjerava kontaktirati poslovnog partnera elektroničkom poštom.
 - o Specifikacija niza akcija – čovjek određuje konkretne radnje koje mora poduzeti, mora unijeti naslov, primatelja te tijelo poruke i u konačnici stišće tipku za slanje.
 - o Izvršenje akcije – čovjek prema prethodnim koracima vrši radnje koje je odredio, unosi naslov, primatelja, tijelo te pritišće tipku za slanje.
- Evaluacija:
 - o Percepcija stanja sustava – nakon što je čovjek izvršio radnju, pokušava evaluirati svoje prethodne postupke, kao i postupke sustava.
 - o Interpretacija stanja sustava – nakon što je čovjek percipirao sustav, interpretira dobivene informacije, u slučaju elektroničke pošte ukoliko je dobio poruku da je pošta uspješno poslana ili ako se pošta nalazi u izborniku za poslanu poštu zaključuje da je pošta poslana. U suprotnom razmišlja o mogućim uzrocima. Uzroci mogu biti s njegove strane (nije dobro koristio program) ili sa strane servisa elektroničke pošte (serveri nisu prenijeli poruku).
 - o Evaluacija stanja sustava s obzirom na ciljeve i namjere – čovjek uspoređuje trenutnu situaciju s očekivanim rezultatom. Ukoliko je rezultat jednak prvobitno postignutom cilju, čovjek je završio svoju interakciju. Ukoliko je rezultati drugačiji, čovjek

odlučuje o daljnjim postupcima. U slučaju elektroničke pošte, čovjek može odlučiti ponovno probati poslati poštu ili zatražiti pomoć.

Iz perspektive interakcije čovjeka i računala, stvaraju se dva problema [6]: problem izvršavanja i problem evaluacije. Problem izvršavanja najmanji je kada je korisniku jasno što on mora učiniti da bi postigao svoj cilj. Kada se čovjek želi prijaviti u svoje računala, prikazano mu je polje za unos zaporke kao i tekst koji mu govori što da radi, te je tada problem izvršavanja najmanji. Problem izvršavanja je najveći kada čovjek ne može svoj cilj uspješno razlomiti na korake. Ukoliko korisničko sučelje jasno ne prikazuje svoje mogućnosti, korisniku neće biti jasno što raditi. Također, postoji problem evaluacije, koji je najveći kada korisnik ne može evaluirati svoje postupke. Programi koji se koriste preko konzole znaju obavljati posao bez da korisniku daju povratnu informaciju, te zbog toga dolazi do problema evaluacije. Korisniku nije jasno što se dogodilo, kako se dogodilo te je li se išta dogodilo. Dobar dizajn mora smanjiti oba problema.

2.2. Okvir interakcije

U radu „*Users, Systems and Interfaces: A Unifying Framework for Interaction*“ autori G. D. Abowd i R. Beale predlažu da se Normanov ciklus izvršenja i evaluacije nadogradi u okvir koji bi mogao opisivati interakcije čovjeka i bilo kakvog interaktivnog sustava. Okvir nalaže da postoje dva člana: korisnik (čovjek) i sustav (računalo), te da svaka interakcija ima domenu u kojoj se odvija. Ukoliko korisnik ima interakciju sa softverskim sustavom za manipulaciju slika, onda je domena grafički dizajn. Kako bi korisnik ostvario željeni cilj, mora provoditi ciklus izvršenja i evaluacije na pojedinim zadacima. Zadaci su radnje koje korisnik mora obavljati kako bi manipulirao domenom [7], odnosno ostvario svoj cilj.

Važno je razumjeti da korisnik i sustav nemaju isti način razumijevanja domene. Korisnik i sustav izražavaju koncepte domene drugim jezicima [7], odnosno računalo koristi osnovni jezik (engl. *core language*), dok korisnik koristi jezik zadatka. Osnovni jezik sadržava sustavne atribute [7], odnosno koncepte domene bitne sustavu. Jezik zadatka sadržava psihološke atribute [7], odnosno koncepte domene bitne korisniku. U slučaju domene grafičkog dizajna, osnovni jezik sadrži potrebne matrice, vektore, formule te druge bitne informacije bitne sustavu, dok jezik zadatka sadrži geometrijske likove, boje i oblike bitne korisniku.

Okvir interakcije uvođenjem dva različita jezika, odbacuje ideju Normanovih ciklusa da problem izvršavanja i problem evaluacije dolazi iz loše komunikacije, odnosno zaključuje da prevođenje iz jednog u drugi nije uvijek jednostavno. Okvir interakcije uvodi dva člana [7]: ulaz i izlaz, te tako uz korisnika i sustav, propisuje četiri člana. Svaki član ima svoj jezik, pa tako nastaje još i jezik

ulaza, te jezik izlaza. Jezici ulaza i izlaza se teško mogu preslikavati na domenu kojom se sustav bavi. U slučaju manipulacije slike, jezik ulaza sadrži pokrete miša, te procese pritiskanja različitih gumbova na tipkovnici, dok jezik izlaza sadrži sliku koju korisnik manipulira te dodatne ikone, polja te ostale elemente grafičkog sučelja.

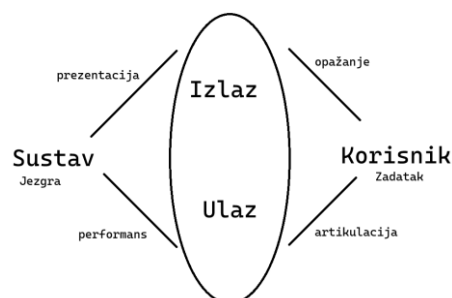
Kako bi interakcija u okviru bila uspješna, moraju se događati 4 koraka prevođenja:

- Artikulacija – korisnik artikulira što želi raditi, odnosno prevodi svoje zamisli iz jezika zadatka u jezik ulaza.
- Performans – ulaz prevodi jezik ulaza u naredbe koje sustav razumije. Sustav se zatim transformira, odnosno manipulira sukladno sa tim naredbama.
- Presentacija – sustav prevodi rezultate u jezik izlaza. Izlaz prezentira rezultate.
- Opažanje – korisnik opaža rezultate na izlazu, te vrši evaluaciju.

Za slučaj manipulacije slikom, koraci prevođenja su sljedeći:

- Artikulacija – korisnik želi zaokružiti nešto što vidi na slici. Koristi ulazni uređaj *miš* tako da ga pomiče i pritišće lijevu tipku
- Performans – ulaz prevodi pritiskanja lijeve tipke miša kao i pomicanje istog u naredbe koje sustav mora odraditi. Sustav se manipulira.
- Presentacija – sustav prevodi svoje trenutno stanje u sliku koja se može prikazati na monitoru.
- Opažanje – korisnik gleda novo prikazanu sliku na monitoru, te vrši evaluaciju. Ukoliko nije dobro nacrtan krug, odnosno ako nije zaokružen element onako kako je on ciljao, on će poduzeti dodatne zadatke.

Prikaz članova okvira, kao i prikaz postupaka prevođenja vidljiv je na slici 2.1.



Slika 2.1. Članovi okvira i postupci prevođenja

Okvir interakcije nalaže da se svaka interakcija vrednuje na osnovu 4 faze prevođenja. Ulaz i izlaz zajedno tvore *sučelje* (još zvan i *međusklop*), a poteškoće između korisnika i sučelja se onda definiraju kao korisničke poteškoće, dok se poteškoće između sustava i sučelja definiraju kao

sustavne poteškoće. Korisnik može imati poteškoće prilikom interpretiranja stanja na izlazu, ali može imati i poteškoća prilikom artikuliranja zadatka. Sustavne poteškoće su zadaća programera, odnosno osobe koja implementira sustav.

2.3. Sučelje

Sučelje, još zvano i *međusklop*, je skup ulaza i izlaza. Sučelje se dijeli na dva podskupa [7], sustavno sučelje i fizičko sučelje. Računala, odnosno sustav, imaju softverski dio, ali i strojnu podršku koja je potrebna. *Spotify*, program za slušanje glazbe, ima sučelje koje se može razlomiti na sljedeće:

- Izlaz:
 - o Sustavno: grafičko sučelje – koristi se grafički prikaz elemenata kao što je tekst, tipke, liste i dr. kako bi se korisniku prezentirao trenutno stanje programa.
 - o Fizičko: video i audio uređaj – koristi se vizualni uređaj, kao i audio uređaj pomoću kojeg „svira“ glazba
- Ulaz:
 - o Fizičko: koristi se uređaj za kontrolu izvođenja glazbe (tipkovnica, miš, tipka na slušalicama i sl.) kako bi korisnik artikulirao
 - o Sustavno: ovisno o informacijama s fizičkih uređaja, izvode se procedure/algoritmi koji transformiraju sustav

Sustavno i fizičko sučelje nalažu ulazno-izlazni uređaji. Postoje slučajevi gdje sustav može nalagati ulazno-izlazne uređaje, poput programa za obradu teksta, koji zahtijevaju pristup izlaznog uređaja koji može prikazati tekst, te ulaznog uređaja koji je efektivan za unošenje tekstualnih znakova (tipkovnica). Međutim često ulazno-izlazne uređaje, pa tako i sučelje, nalaže korisnikovo okruženje. Osoba koja želi slušati glazbu, može je slušati dokle god postoji izlazni uređaj koji može reproducirati glazbu (zvučnici ili slušalice) te ulazni uređaj kojim se može manipulirati stanje sustava. Problem nastaje kada se pokušava dostići što veća populacija slušatelja:

- Slušanje na računalu: postoje korisnici koji slušaju glazbu dok obavljaju poslove na računalu. Zbog njih potrebno je nuditi manipuliranje sustavom tipkovnicom i mišem. Također, sa sustavnog gledišta izlaza, moramo nuditi obavijesti svaki puta kada se promjeni pjesma, jer neki korisnici će igrati igre na računalu dok slušaju glazbu, pa neće moći imati pristup grafičkom sučelju.
- Slušanje dok osoba putuje: korisnici koji slušaju glazbu dok šetaju, voze bicikl i sl. će interaktivirati sa sustavom za glazbu na više potencijalnih načina. Za minimalnu pokrivenost

bitno je osigurati da se glazba može slušati preko zvučnika ili slušalica, te da korisnik može manipulirati sustavom preko ekrana na dodir. Međutim korisnici će također htjeti artikulirati svoje zadatke preko tipki koje se nalaze na slušalicama, a potencijalno će htjeti artikulirati svoje zadatke preko pametnog sata, te opažati promjene na istom.

- Slušanje glazbe u automobilu: korisnik koji želi pustiti određenu pjesmu dok se vozi u automobilu može to napraviti preko dodirnog ekrana, ili tipki koje se nalaze unutar automobila. Problem dolazi kada je taj korisnik ujedno i vozač, te tada korištenje ekrana na dodir ili tipke na dulje vrijeme (npr. što ako korisnik mora unijeti cijelo ime pjesme i autora kako bi pronašao traženu pjesmu) postaje opasno za korisnika. U tom slučaju potrebno je implementirati mogućnost da korisnik artikulira svoj zadatak ljudskim govorom, a da ulaz uspješno prevede ljudski govor u naredbe sustavu.
- Osobe sa invaliditetom: korisnici koji imaju određene invalidnosti potencijalno ne mogu koristiti standardizirane ulazno-izlazne uređaja ili zahtijevaju da se standardizirani uređaji poboljšaju softverskim metodama koje će kompenzirati za njihovu invalidnost. Slijepa osoba ne može vidjeti grafičko sučelje, te za takve korisnike potrebno je osmisliti način na koji trenutno stanje sustava možemo uspješno prezentirati.

2.4. Pristupačnost

Pristupačnost je koncept koji vrednuje koliko pojedini proizvod može koristiti šira populacija [8]. Često je zbog invalidnosti pojedincima otežano korištenje, odnosno interakcija s nekim sustavom. Cilj pristupačnosti je da dizajnere sustava pogura u smjeru dizajniranja sustava koji mogu koristiti svi ljudi.

Ljudi sa invaliditetom ne mogu koristiti sve ulazno-izlazne uređaje:

- Slijepa osoba – ne mogu koristiti izlazne uređaje koji prezentiraju stanje sustava vizualno.
- Gluhe osobe – ne mogu koristiti izlazne uređaje koji prezentiraju stanje sustava audio načinima.
- Osobe invalidnosti ruku – ne mogu koristiti ulazne uređaje prilagođene rukama.

Potrebno je nuditi alternativni uvid u stanje sustava kako bi pristupačnost bila veća:

- Slijepa osoba – nuditi opciju alternativnog teksta za vizualni prikaz.
- Gluhe osobe – nuditi opciju titlova kako bi pretvorili zvučni izlaz u vizualni
- Osobe invalidnosti ruku – nuditi opciju artikuliranja zadatka drugim načinima

Postoje regulatorni i ekonomski [8] razlozi za poboljšanje pristupačnosti. Što je pristupačnost veća, više ljudi je u stanju koristiti određeni sustav, što može biti ekonomski isplativo. Također, sustavi koji se često koriste u javnim institucijama su regulatorno prisiljeni implementirati dodatne načine interakcije.

2.5. Interakcija čovjeka i računala u videoigrama

Videoigre kao softverski proizvodi moraju primijeniti dobre prakse interakcije čovjeka i računala kako bi uspješno bile korištene od strane korisnika. Samim time, postoje prakse koje se jednako mogu primjenjivati i na profesionalni softver kao i na videoigre. Videoigre moraju primjenjivati drugačije prakse prilikom HCI-a [9] zbog načina na koji se razlikuju od produktivnog/poslovnog softvera.

2.5.1. Procesi za razliku od rezultata

Profesionalni/produktivni softver ima svoju svrhu. Njegova svrha je da učini određeni zadatak lakšim, bržim te da smanji šanse za pogreškom. Prilikom dizajniranja ovakvog softvera, cilj je razumjeti kako će on pomoći korisniku da:

- Smanji broj pogrešaka
- Poveća produktivnost (brzinu rada)
- Poveća kvalitetu

Kada bi uzeli primjer *Microsoft PowerPoint* programa, i usporedili ga s prezentiranjem prije tog programa, on prethodno čini na sljedeće načine:

- Natuknice na ekranu smanjuju šansu da slušatelji krivo razumiju predavača.
- Predavač može koristiti alat za provjeru pravopisa kako bi bio siguran da koristi prave riječi te da ih dobro izgovara, nešto što bi trajalo dulje da se predavač priprema za papirnim rječnikom.
- Predavač sada može prikazivati slike, videozapise, zvučne zapise kako bi bolje dočarao svoje predavanje.

Videoigre se razlikuju od profesionalnog/produktivnog softvera, jer njima nije svrha da povećaju produktivnost [9], smanje greške te poboljšaju kvalitetu nečeg. Cilj videoigre je da bude zabavna. Zbog toga nije moguće pristupiti na isti način prilikom proučavanja interakcije, jer prethodno navedena mjerila više nisu važna za kreiranje uspješne videoigre.

2.5.2. Definiranje ciljeva za razliku od uvoza cilja

Microsoft Word ima jasan cilj. Njegova je svrha da korisniku omogući kreiranje i uređivanje tekstualnih dokumenata. To je cilj koji postoji u stvarnom životu van programa, odnosno, program je kreirana jer postoje ljudi koje žele programom dobiti željeni izlaz. Kraće rečeno, to je cilj iz stvarnog života [9], koji je uvezen u program, odnosno softver.

Videoigre se tu razlikuju. U igri *Super Mario* cilj je spašavanje princeze *Peach* od zlog lika *Bowser*-a. To nije cilj koji postoji u stvarnom svijetu, već je nešto što je nametnuto u samoj igri. Zbog toga korisnik softvera ne mora nužno biti motiviran [9], te je cilj da uz pomoć proučavanja interakcije čovjeka i računala, nađemo način na koji ćemo motivirati korisnika na cilj spašavanja princeze. Imamo izbor kako prenijeti igraču da je ovo njegov cilj, a ovo su neke (ne sve) opcije:

- animirana scena (engl. *Cutscene*)
- poruka na ekranu
- monolog glavnog lika, gdje spominje kako treba spasiti princezu

Osim motivacije za velike ciljeve (poput princeze, koja je glavni akt igre), igre također imaju alternativne ciljeve na koje mogu motivirati igrača. Na primjer, u igri *DiRT 3* glavni cilj je postati prvak svijeta u “DiRT” turniru. Igra također motivira igrača da igra utrke koje mu ne pomažu u ostvarenju cilja direktno, tako što mu nudi nagradu u smislu reputacije. Ako igrač igra sporedni sadržaj (svjetska prvenstva), otključati će više reputacije [10] te time može birati između više vrsta automobila kada se vrati na igranje glavnog sadržaja.

Motivacija za ostvarenje cilja je nešto što treba pažljivo odabrati, jer ako je nagrada za igranje sporednog sadržaja prevelika, on više nije sporedan već obavezan, a ako je premala, taj sadržaj na čiji je razvoj potrošen novac i trud ostaje neigran. Također, potrebno je paziti kako prenijeti ovo igraču. Ukoliko igraču predstavimo cilj kao nešto bolje nego što je, uzrokovati ćemo frustraciju kada se igrač razočara. Ako se cilj predstavi kao nešto što nije važno igračevog vremena, a igrač kasnije sazna da je to isto bilo bolje nego što mu je rečeno, opet će se izazvati frustracija.

Motivacija se ne mora nužno ostvariti pozitivnim nagradama. U igrama *Metro*, igrač će umrijeti ako se ne uspije obraniti od čudovišta. Iako igrač ne dobiva nikakvu nagradu, igrač je motiviran ubijati čudovišta zbog penala (smrti) koje će dobiti ako to ne učini. Potrebno je paziti s ovim pristupom, jer ovakva interakcija može proizvesti frustraciju [10] kod igrača.

2.5.3. Par alternativa za razliku od puno alternativa

Profesionalni softver često nema puno alternativa. Ukoliko korisnik želi integrirano razvojno okruženje (engl. *Integrated development enviroment*, IDE) za programski jezik C#, ima opciju

Visual Studio-a od Microsoft-a, te *Rider-a* od JetBrains-a. U *Search and Destroy* podžanru pucačine u prvom licu (engl. *first person shooter*, FPS) postoji izbor igara *Counter-Strike 2* (Valve) i *Rainbow Six Siege* (Ubisoft Montreal). Na prvu, čini se da je izbor u obje vrste softvera jednak, međutim situaciju je potrebno bliže sagledati.

Programski inženjeri koji koriste C# moraju koristiti IDE koji podržava C#, no to nije slučaj kod videoigara. Iako igrač igra *Search and Destroy* pucačine, on može također igrati trkaće igre, strateške igre i druge, kao što može voljeti provoditi svoje slobodno vrijeme gledajući filmove, serije i slično. Videoigre kao proizvod moraju se natjecati za pažnju igrača s ostalim vrstama medija.

Zbog ovog svojstva videoigara, prilikom razvoja videoigre, nije dovoljno proučavati samo kako igrači tog žanra interagiraju s videoigrom [9], već kako i igrači ostalih žanrova imaju odnos s videoigramama. Sam pristup ciljanoj publici je promijenjen, pa tako i načini sakupljanja podataka o njima [9].

2.5.4. Dosljednost za razliku od varijacije

Kod profesionalnog softvera važna je dosljednost. Korisnici ne žele da se rezultati operacije mijenjaju neočekivano [9], jer to smanjuje njihovu vjeru u softver i samim time produktivnost i kvalitetu krajnjeg rezultata. Korisnici također žele da je sučelje dosljedno, odnosno da se ne mijenja naglo i neočekivano jer to utječe na navike i mišićnu memoriju koju su razvili kako bi bili produktivniji u korištenju softvera.

Videoigre se tu razlikuju po tome što velika dosljednost uzrokuje dosadu [9], što uvelike umanjuje uspjeh igre. Igre moraju nuditi jedinstvena iskustva, ali ta iskustva također moraju imati varijacije. Videoigra *Minecraft* (Mojang) jedna je od najuspješnijih videoigara svih vremena, a svoj uspjeh gradi na osnovi proceduralnog generiranja (engl. *procedural generating*). Time igra osigurava da svaki put kada igrač stvori novi svijet unutar igre, taj svijet će biti potpuno drugačiji, te će se igrač morati suočavati s novim izazovima, što produljuje količinu vremena koju igrač može potrošiti u igri bez da mu bude dosadno. Drugim riječima, igra je zabavnija igraču jer se ona sama mijenja.

Još jedna perspektiva na ovu promjenu bi bio softver koji se kontinuirano ažurira. Prilikom razvoja novih ažuriranja za produktivni softver, poput *Adobe Photoshop-a* ključno je istražiti kako korisnici interagiraju s njime, kako nove mogućnosti i unaprjeđenja ne bi uznemirila, odnosno omele korisnike. Za razliku od toga, videoigre koje se kontinuirano ažuriraju (najčešće multiplayer

igre) poput *Fortnite*-a (Epic Games), *Rainbow Six Siege*-a (Ubisoft Montreal) i sl. ciljaju na to da promjene način na koji igrači igraju, kako bi razbili monotoniju. Često to rade na 4 načina:

- Ojačavanje sadržaja (engl. *buff*): cilj je da već postojeći sadržaj u igri postane relativno interesantniji igraču [11]. Ovo se radi kada je sadržaj unutar igre preslab naspram drugog sadržaja [12] (npr. jedan pištolj ima manje metaka od drugog i zato ga nitko ne želi koristiti). Iz perspektive interakcije s videoigrom, treba biti oprezan, jer ne smijemo previše niti premalo ojačati stvari, jer to budi frustraciju kod igrača.
- Oslabljivanje sadržaja (engl. *nerf*): cilj je da već postojeći sadržaj postane manje atraktivan [11] igraču. To se najčešće radi kada veliki broj igrača koristi jedan sadržaj, a ignorira gotovo sve ostalo [12] (npr. jedan lik u igri od njih 10 je uvjerljivo najjači, pa skoro nitko ne igra ostalih 9). Također, zbog načina kako igrač interaktira s videoigrom, ovime možemo probuditi frustraciju [12], jer smo napravili njemu potencijalno omiljeni sadržaj manje zanimljivim.
- Dodavanje novog sadržaja: cilj je igraču ponuditi alternativu na ono što on već koristi, te uvesti nova iskustva igraču. Potrebno je paziti što se dodaje igri, jer sadržaj koji umanjuje vrijednost starog sadržaja može prouzrokovati frustraciju kod igrača koji su povezani s starim sadržajem.
- Promjena sadržaja: ovo je kombinacija prethodna 3 pristupa. Ukoliko postoji sadržaj u igri koji je premalo, ili previše zastupljen [11], može se umjesto oslabljivanja ili ojačanja sadržaja, napraviti promjena. Npr. ukoliko postoji lik koji koristi mač, a napravi se zamjena mač s lukom i strijelom. Ovo mijenja kako igrač koristi istog lika, bez da je to nužno ojačavanje ili oslabljivanje sadržaja (ovisi o tome kako je izvedeno).

2.5.5. Micanje/strukturiranje ograničenja za razliku od namjernog dodavanja ograničenja

Kod poslovnog/produktivnog softvera cilj je da korisnik može ostvariti ono što on želi [9]. Prilikom dizajna sučelja, cilj je osigurati laganu interakciju, odnosno korištenje programa pa će zbog toga program probati riješiti sva moguća ograničenja. Za neke stvari se strukturiraju ograničenja (npr. čarobnjak za instalaciju traži da se prvo unese mjesto instalacije). Jedini slučaj gdje profesionalni/produktivni softver namjerno dodaje ograničenja, je kada ima više vrsti licenci koje se prodaju za različitu cijenu, no to nije područje HCI-a već područje poslovanja.

Videoigre same po sebi postavljaju ograničenja korisniku. Cilj videoigre je da igraču postavi nekakav izazov, odnosno da mu oteža put do cilja. Za primjer ograničenja može se uzeti činjenica

da igrači u *Rainbow Six Siege* (Ubisoft Montreal) ne mogu ponovno stvoriti (engl. *respawn*) tokom runde. Ubisoft Montreal je prilikom testiranja videoigre shvatio da igrači drugačiji interagiraju ako njihova smrt ima dugotrajnije posljedice [13], te da su igrači zbog toga pristupali igri pažljivije i više surađivali sa svojim suigračima.

Također, ograničenja u igrama često služe kao motivacija. Igrač u igri *Grand Theft Auto V* (Rockstar Games) ne može na početku igre pristupiti skupim automobilima poput *Truffade Adder-a* zbog nedostatka novca [14], što ga motivira da prolazi kroz priču te prelaženjem misija i pljački stekne dovoljno novca [15] za kupnju željenog automobila. Ovim načinom, igrač interaktira s igrom na dva načina: jedan je u smislu linearnog prelaženja igre, dok je drugi čista štednja/rad prema otključavanju željenog sadržaja [10].

2.5.6. Funkcionalnost za razliku od raspoloženja

Profesionalni/produktivni softver koristi grafiku i zvuk kako bi dočarao određenu funkcionalnost. Primjerice, animacija diskete može dočarati dojam da se trenutno nešto sprema na disk. Sjena i zvuk mogu služiti kako bi dali dojam da je korisnik pritisnuo na gumb, odnosno da se njegov pritisak tipke miša registrirao.

Dok isto vrijedi i za videoigre, videoigre također koriste zvukove, glazbu, grafiku, vibriranje i slično, kako bi dočarale raspoloženje odnosno ugođaj. S time interakcija između čovjeka i videoigre se mijenja ovisno o ugođaju. Umjetnički dizajn videoigre znatno može utjecati na ugođaj [16], pa tako i ponašanje igrača u videoigri. Kratko rečeno, zvukovi, grafika i ostalo, u produktivnim/poslovnim softverima služi kako bi korisnik dobio povratnu informaciju da se događa ono što on želi, dok te iste stvari videoigrama služe kako bi igra mogla usmjeriti igrača, ili ga staviti u raspoloženje potrebno za igru igre.

U igri *Fortnite Battle Royale* (Epic Games) postoji mehanika koja se zove “oluja” (engl. *storm*). Cilj mehanike je da bilo tko tko završi u oluji prima štetu. Cilj te mehanike je da igra ima sposobnost kontrolirati u kojem djelu prostora igre je igrač. Kako bi prestrašili igrača, Epic Games koristi tamno ljubičastu paletu [17], a svaki igrač koji dopusti da ga oluja sustigne će čuti glasne zvukove oluje, promjenu grafike, zvukove boli od svog lika te sl.

2.5.7. Izgled ishoda za razliku od izgleda svijeta

Produktivni/poslovni softver rijetko ima različite perspektive/točke gledišta, no neki imaju prikaz krajnjeg rješenja. Alat za grafički dizajn ima opciju da se kreirani sadržaj pogleda iz perspektive web preglednika, plakata, mobilnog prikaza i drugih.

Videoigre imaju različite točke gledišta odnosno perspektive. Igre se mogu igrati iz prvog lica, trećeg lica, ptičje perspektive i ostalih. Nerijetko je da ista igra nudi više različitih perspektiva (npr. *Grand Theft Auto V* nudi opciju igranja u prvom ili trećem licu). Izbor perspektive ovisi o žanru videoigre. Primjerice igrači trkaćih igara uglavnom igraju u trećem licu, jer tako ima pogled na cijeli automobil i njegovo okruženje, dok igrači strateških igara igraju iz Božje perspektive (engl. *God's eye perspective*) jer tako imaju pregled svega što se događa. Prikaz nekoliko različitih perspektiva je vidljiv na slici 2.2.



(a)

(b)

(c)

Slika 2.2. a – prikaz u trećem licu (DiRT 3), b – prikaz u prvom licu (Rainbow Six Siege), c – prikaz iz Božje perspektive (Civilization 6)

2.5.8. Organizacija za razliku od pojedinca

Profesionalni/produktivni softver se uglavnom prodaje organizacijama. Samo 13% prihoda od *Microsoft Office-a* došlo je od privatnih korisnika [18]. Igre međutim kupuju uglavnom privatni korisnici [9]. Zbog toga potrebno je očekivati drugačije vrste interakcije sa softverom.

Ukoliko organizacije kupuju softver, žele da njihovi članovi budu produktivni sa softverom, te im nije problem platiti edukaciju svojim članovima. Pošto igre kupuju pojedinci, te svrha igre je zabava, igra mora biti intuitivna korisniku.

2.5.9. Forma slijedi funkciju za razliku od funkcija slijedi formu

Kod profesionalnog/produktivnog softvera sučelja se ne mijenjaju radikalno. Web preglednici u svojoj suštini i dalje imaju traku u kojoj se upisuje web adresa, te horizontalno postavljene kartice. Alati za izradu prezentacija, tablica, tekstnih dokumenata i sl. idalje imaju traku na vrhu, sa više vertikalno poredanih izbornika na vrhu trake.

Korisnici videoigara žele nove tehnologije i nove pristupe [9], iako ti pristupi ne moraju nužno biti bolji. Igrači videoigara računaju na poboljšanje tehnologije, te računaju na to kako bi spriječili monotoniju, odnosno stekli nova iskustva [9].

2.5.10. Standardni ulazni uređaji za razliku od različitih ulaznih uređaja

Većina produktivnog/poslovnog softvera koristi iste ulazne uređaje. Tipkovnica i miš dovoljni su za veliku većinu poslovnog/produktivnog softvera. Česte iznimke mogu biti tablet za crtanje i ekran za dodir. Prednost ovog pristupa jest taj što čovjek vrši interakciju s računalom na način s kojim je već upoznat, te može izoštriti svoje vještine kroz upotrebu različitog softvera. Primjerice, menadžer može izvježbati vještine brzog pisanja tipkovnicom, pa ih onda prenijeti na više programa:

- Brže pisanje mailova
- Brže pisanje poruka u aplikacijama poput *Slack-a*, *Teams-a*
- Brže pisanje dokumentacije u programima za kreiranje tekstnih dokumenata poput *Microsoft Word-a* i *Google Docs*
- Brža obrada tablica u programima poput *Microsoft Excel* i *Google Sheets*
- Brže pretraživanje interneta web tražilicama

Za razliku od poslovnog/produktivnog softvera, videoigre imaju više načina unosa. Razlog tomu je što je doživljaj jedan od bitnih ciljeva, a neke metode unosa nude igraču da se bolje poveže s igrom. Neke od tih metoda su:

- Tipkovnica i miš: nude precizno okretanje u igrama zbog beskonačno puno mogućih pozicija miša. Pogodna metoda unosa kod igranja pucačina u prvom licu (engl. FPS – first person shooter) jer one zahtijevaju precizno okretanje lika zbog nišanja oružja [19].
- Kontroler: udoban, lagan za korištenje te nudi kombinaciju tipki, okidača i palica. Tipke su pogodne za akcije koje su ili “da” ili “ne” (npr. pokupi ovaj element s poda), dok su palice zgodne za akcije koje imaju linearni raspon (npr. dodaj 20% gasa, stisni kočnicu 40%). Konačno palice nude linearni raspon u dvije dimenzije, pa su zato pogodne za kretanju. Kontroleri također mogu biti ujedno i izlazni uređaji [19], prenošenjem vibracija igraču.
- Trkaći volan i periferija: specificirani skup ulaznih uređaja kojem je cilj emulirati kontrole u stvarnim vozilima poput automobila, kamiona, traktora i slično. Također, mogu biti i izlazni uređaji te prenositi vibracije korisniku [20].
- Ekрани na dodir: mobilne videoigre su jako popularne, pa tako jako puno igrača igra s ekranima na dodir. Oko 77% svih prihoda u industriji igara dolazi od korisnika koji igraju ovom metodom [21]. Problem kod ove metode unosa jest taj što prsti ujedno prekrivaju ekran pa je bitno tokom razvoja dobro se pobrinuti o rasporedu tipki na ekranu.

- VR Oprema: oprema koja služi za igranje virtualne stvarnosti. Uglavnom je kombinacija ulaznih i izlaznih uređaja.
- Palice za zrakoplove: oprema koja se koristi za igranje simulatora letenja
- Pokretni kontroleri: oprema koja služi za detekciju pokreta. Dolazi u više formi sa različitim načinima rada, primjeri *Wii Remote* se drži u ruci i pomiče, dok je *Xbox Kinect* skup senzora koji stoji u mjestu i promatra igrača. Pokretni kontroleri mogu biti samostalni ili u sklopu opreme za virtualnu stvarnost
- Kontroleri namijenjeni osobama s invalidnostima: služe kako bi nudili igračima sa fizičkim invalidnostima mogućnosti igranja. Mogu sadržavati i ulazne i izlazne uređaje.

Nije rijetko da igre nude više metoda unosa igraču na izbor. Primjer igre koja nudi više izbora unosa bi bio *Fortnite* (Epic Games). Pošto je *Fortnite* dostupan na PC-u mora podržavati igranje mišem i tipkovnicom, pošto je dostupan na konzolama mora podržavati unos kontrolerom, a pošto je dostupan na Androidu [22], mora podržavati unos ekranom na dodir [22]. Međutim nisu sve metode unosa jednako prikladne zadatku. Zbog toga puno pucačina implementira pomoć pri ciljanju (engl. *aim assist*) za igrače koji ne igraju s tipkovnicom [23]. Tome je cilj da igrači koji igraju s tipkovnicom i ostali igrači se mogu boriti jednako pravedno.

2.6. Postojeća rješenja

Na tržištu već postoje, ili su postojali, uređaji koji nude alternativne metode unosa. Razlikuju se u pristupima, a tako i u ciljanim skupinama.

2.6.1. Sony EyeToy

Sony EyeToy je USB kamera koju je razvio Sony Computer Entertainment. EyeToy je bio projekt Richarda Marksa, i postao je dostupan javnosti 1999 [24][25]. Playtoy se prodao u preko 10 milijuna kopija [24], i bio je dostupan za konzolu Playstation 2. Kamera je imala rezoluciju od 320 x 240 piksela i svrha kamere je da detektira pokrete, odnosno da igraču nudi nove načine igranja, bez da on mora koristiti kontroler. Izgled EyeToy-a vidljiv je na slici 2.3.



Slika 2.3. Sony EyeToy kamera [26]

Za EyeToy posebno su razvijene igre. Neke od njih su:

- *EyeToy Play*: igra koja se sastoji od 12 mini igara. Igre se igraju tako da igrač miče svoje tijelo, a neke od njih su:
 - “*Disco star*”: igrač mora kopirati plesača na ekranu (detekcija cijelog tijela)
 - “*Wishi Washi*”: igrač mora čistiti prozore pomicanjem ruku (detekcija ruku/tijela)
 - “*Soccer craze*”: igrač mora održavati loptu u zraku sa svojom glavom (detekcija glave)
 - “*Slap Stream*”: igrač mora udarati miševе, ali ne smije zečevе (detekcija ruku)
- *EyeToy Groove*: igra u kojoj je cilj plesati. Igrač mora u ritmu pozadinske glazbe dirati tipke koje se nalaze oko ekrana (detekcija ruku)
- *EyeToy Kinetic*: igra kojoj je cilj pomoć pri tjelovježbi. Igrači mogu odabrati 12-tjedni trening program i igra analizira njihovu uspješnost u treningu. (detekcija tijela)

2.6.2. Xbox Kinect

Kinect je digitalni uređaj koji je napravio Microsoft za *Xbox 360* i *Xbox One* konzole. Prvi puta dostupan javnosti 2010 [27][28]. Postojale su dvije generacije, *V1* i *V2* gdje je prva imala rezoluciju kamere od 640x480 piksela, a druga 1920x1080 [29]. Sam uređaj je zapravo skup senzora i kamera s kojim se ostvaruje detekcija osobe i pokreta. Kinect sustav prikazan je na slici 2.4.



(a) (b)
Slika 2.4. (a) – Xbox Kinect V1 [28], (b) Xbox Kinect V2 [29]

Microsoft je tokom razvoja *Xbox One* konzole shvatio da će se za Kinect razviti manje igara ako ne bude svaki igrač imao Kinect. Također, igrači nisu voljni kupiti Kinect ako za njega nema igara. Zbog toga Microsoft je odlučio da svaka *Xbox One* konzola dolazi sa Kinectom. [28]

Neki od primjera igara koje su razvijene za Kinect bi bile:

- *Fruit Ninja Kinect* – na ekranu igraču lete voćke, a on ih mora prerezati. Rezanje radi detekcijom pokreta [30].
- *Kinectimals* – igrač u virtualnom svijetu može interagirati sa životinjama. Jedna od interakcija je maženje koja se odvija detekcijom pokreta [30].
- *Dragon Ball Z for Kinect* – igrač se može boriti u igri. Igra se odvija u prvom licu i koristi prepoznavanje pokreta za upravljanje udarcima [30].
- *Just Dance* – igrač pleše dok svira glazba. Cilj mu je oponašati virtualnog lika kojeg vidi na ekranu [30].

2.6.3. Wii Remote

Wii Remote je uređaj koji je napravio Nintendo za igraču konzolu Wii. Uređaj oblika daljinskog upravljača u sebi ima senzore koji služe za detekciju pokreta. Prednost ovog pristupa je što uređaj oblika daljinskog upravljača može emulirati ostale naprave iz stvarnog života [32], poput reketa za tenis. Wii Remote je napravljen za široku uporabu u igrama, a nudi mogućnosti ekstenzija pomoću dodataka (npr. dodatak koji pretvara Wii Remote u volan za igranje trkaćih igara). Wii Remote prikazan je na slici 2.5.

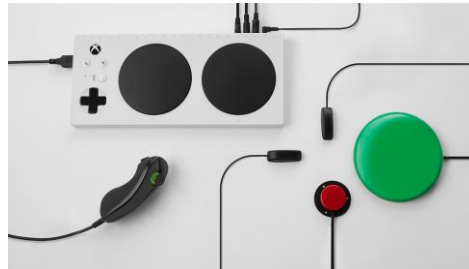


Slika 2.5. Wii Remote (lijevo) [32]

Najpopularnija igra koja koristi Wii Remote je *Wii Sports*, koja dolazi zajedno sa konzolom. Unutar igre postoji 5 igara: tenis, bejzbol, golf, boks i kuglanje. U svih 5 igara Wii Remote se koristi za detekciju pokreta potrebnu za igranje. Može se primijetiti kako su igre dizajnirane oko toga da Wii Remote pokušava dočarati osjećaj pravog objekta (reket, golf palica).

2.6.4. Xbox Adaptivni Kontroler

Xbox adaptivni kontroler (engl. *Xbox Adaptive Controller*) je uređaj napravljen u svrhu nuđenja alternativnih metoda unosa korisnicima sa fizičkim invalidnostima [33]. Najveća prednost Xbox adaptivnog kontrolera jest ta što je jeftiniji od alternativa (košta otprilike 100 američkih dolara), te što zbog svoje arhitekture može se lagano nadograditi sa uređajima za proširivanje. Xbox Adaptivni kontroler prikazan je na slici 2.6.



Slika 2.6. Xbox adaptivni kontroler i dodatni uređaji za proširivanje [33]

Xbox adaptivni kontroler nije napravljen kako bi nudio nove vrste igara, već mu je svrha da se postojeće igre mogu postati pristupačnije osobama s fizičkim invalidnostima.

3. DIZAJN I IZRADA VIDEOIGRE ZA USPOREDBU METODA UNOSA

Interakcija čovjeka i računala je složena, a jedan od dijelova interakcije je ulaz. Za ulaz moguće je koristiti više metoda unosa, a njihovo dodavanje prednost je videoigri jer potencijalno alternativne metode unosa mogu igraču nuditi nova iskustva i izazove. Također, alternativne metode unosa nude igračima koji su spriječeni u igranju videoigara zbog fizičkog invaliditeta, mogućnost igranja. Zbog toga, potrebno je dizajnirati videoigru koja nudi više alternativnih metoda unosa.

3.1. Odabir žanra i osnovne funkcionalnosti

Prilikom dizajniranja videoigre bitno je razumjeti kome je igra namijenjena te na koji način ona pokušava zabaviti igrača. Cilj igre je da bude relativno jednostavna za razumjeti, kako bi ljudi koji će ju testirati mogli lagano svladati i dati povratnu informaciju o metodama kontrole, odnosno interakcije sa videoigrom. Zbog toga su za žanr videoigre odabrana dva žanra; *infinite runner* i *side scroller*. Kombinacijom ta dva žanra dobiva se igra gdje igrač stalno trči u stranu, a odabrano je da igrač trči prema desno. Igra je 2D (što nalaže žanr *side scroller*), zbog lakše izrade i zbog lakoće igranja. Također, prednost ovog žanra i izvedbe je ta što je videoigra prikladna većem broju uzrasta. Pošto je cilj rada istražiti interakciju čovjeka i računala u pogledu videoigara, cilj je napraviti zabavnu igru, ne nužno grafički kompleksnu.

Igrač mora izbjegavati prepreke te mu je cilj postići što bolji rezultat. Na svom putu može pokupiti bonuse (engl. *Power Up*) kako bi si, ili olakšao igranje, ili povećao rezultat. Prilikom neuspjeha, odnosno kraja igre, igrač može odmah igrati ispočetka. U igri igrač trči sve brže što mu je rezultat viši, odnosno igra se s vremenom ubrzava.

Potrebno je odrediti koje alternativne metode unosa ponuditi igraču. Za alternativne metode moguće je koristiti više uređaja, a za ulazni uređaj odabrana je kamera, zbog svojstva da se preko nje može implementirati više alternativnih metoda unosa. Na slici koju daje kamera moguće je detektirati lice (kao i druge dijelove tijela) i boje, a obje su odabrane kao metoda unosa u videoigri. Obje metode unosa su fleksibilne, odnosno nude više načina na koje ih je moguće izvesti te više načina na koje igrač može igrati.

3.2. Korišteni alati i tehnologije

Za izradu videoigre potrebno je koristiti određene alate kako bi se olakšala sama izrada. Potrebno je izabrati alate koji nude funkcionalnosti potrebne za izradu videoigre.

3.2.1. Unity

Za izradu ovog projekta korišten je Unity. Unity je pokretač igre (engl. *game engine*), koji nudi vizualno slaganje scena te programiranje funkcionalnosti u C# [34]. Videoigra koristi scene za glavni izbornik kao i za glavni nivo videoigre, a sve funkcionalnosti napisane su u C#. Najveća prednost Unity-a je što omogućuje da prilikom izrade videoigre, nije potrebno se zamarati implementacijom računalne grafike, 3D prostora i sličnog, već je moguće potpuno se fokusirati na implementiranje funkcionalnosti koje su bitne za konkretnu videoigru, odnosno u slučaju ovog rada, moguće se potpuno fokusirati na implementiranje alternativnih metoda unosa i značajki konkretnih za videoigru.

3.2.2. OpenCV plus Unity

OpenCV plus Unity je resurs koji omogućuje korištenje *OpenCV* biblioteke unutar Unity-a. Korištenjem ove biblioteke omogućuje se pristup kameri, kao i pokretanje modela zasnovanih na računalnom vidu. U ovom radu to je korisno zbog korištenja tri pristupa interakcije koja zahtijevaju pristup kameri, dok dva od njih zahtijeva pokretanje modela detekcije lica.

3.3. Prikaz osnovnih dijelova videoigre

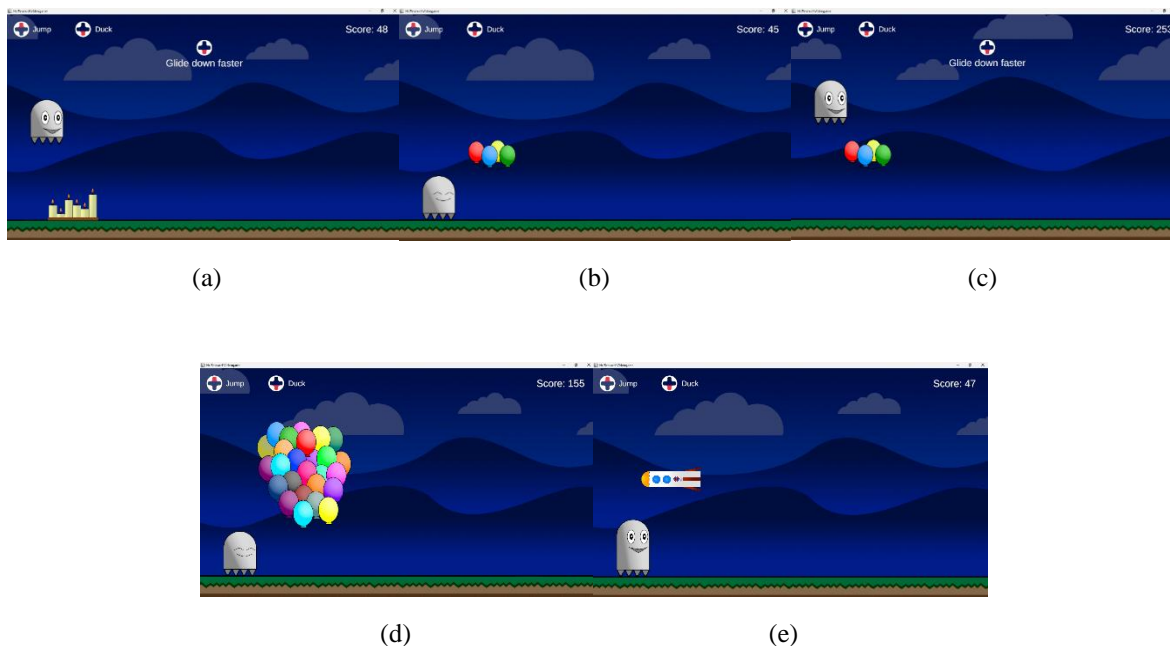
Osnovni dijelovi igre su prepreke, bonusi i rezultat. Igraču je cilj skupiti što više bonusa, izbjegavati prepreke te postići najbolji rezultat.

3.3.1. Prepreke i uvjeti neuspjeha

U igri se nalaze 4 prepreke i svaka od njih ima cilj omesti igrača na drugačiji način. Ovisno o prepreci ona se može izbjeći tako da ju igrač preskoči ili da se sagne ispod nje. Prepreke su podijeljene na sljedeće:

- Svijeće – prepreka koja se može izbjeći samo preskakanjem.
- Mali baloni – prepreka koja se može izbjeći i preskakanjem i saginjanjem.
- Veliki baloni – prepreka koja se može izbjeći samo saginjanjem.
- Raketa – prepreka koja će se izbjeći uvijek osim ako igrač ne skoči namjerno na nju. Njen cilj je zbuniti igrača u napetim trenucima.

Prikaz prepreki dostupan je na slici 3.1.



Slika 3.1. Prepreke u igri, (a) – svijeće, (b) – preskakanje balona, (c) – saginjanje ispod balona, (d) – saginjanje ispod velikih balona, (e) – prolaz ispod rakete

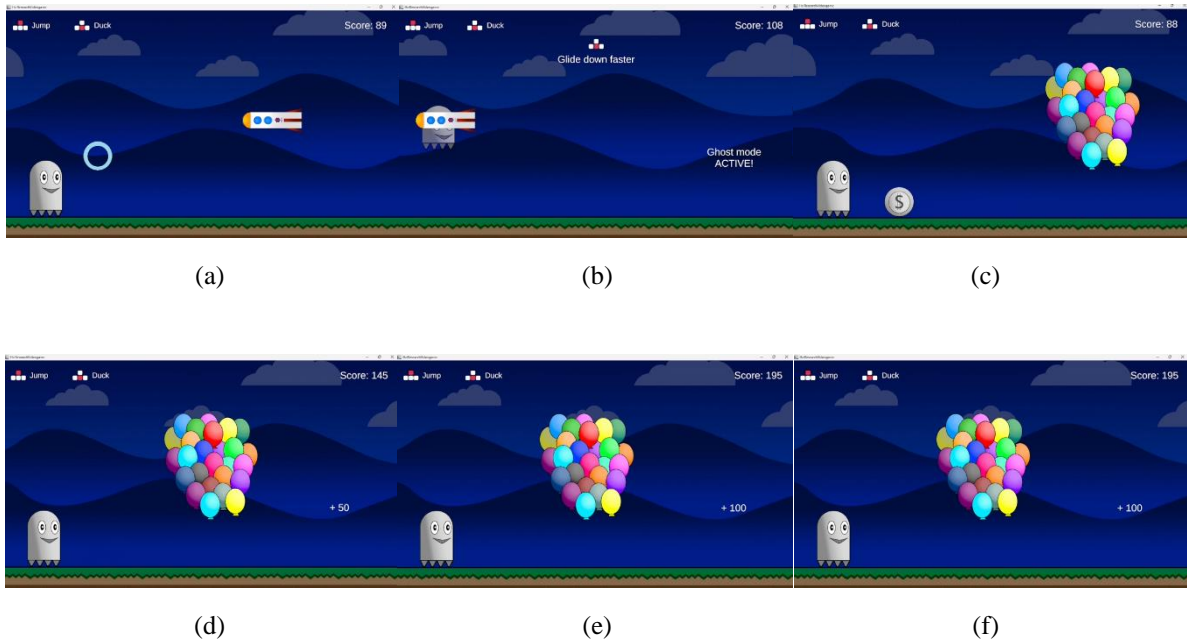
Igrač ne smije s nijednim dijelom svog lika doći u kontakt s preprekom, jer se u tom slučaju igra zaustavlja i u sučelju se ispisuje rezultat.

3.3.2. Bonusi i rezultat igrača

U igri se nalaze 3 bonusa. Igraču je u interesu da pokupi bonuse jer oni mu mogu ili olakšati igru ili poboljšati rezultat. Bonusi su podijeljeni na sljedeće:

- Plavi prsten – bonus koji igraču daje privremenu otpornost na prepreke. Za vrijeme trajanja bonusa igrač se smije sudarati u prepreke i njegova igra neće biti prekinuta. Efekt nije trajan.
- Srebrni novčić – bonus koji igraču nadodaje manju vrijednost na rezultat
- Zlatni novčić – bonus koji igraču nadodaje veću vrijednost na rezultat

Prikaz bonusa nalazi se na slici 3.2.



Slika 3.2. (a) – prije kupljenja plavog prstena, (b) – nakon kupljenja plavog prstena, (c) – prije kupljenja srebrnog novčića, (d) – nakon kupljenja srebrnog novčića, (e) – prije kupljenja zlatnog novčića, (f) – nakon kupljenja zlatnog novčića

Kako bi igrač pokupio bonus on se mora bilo kojim dijelom svog lika sudariti s njim. Nakon toga će se na sučelju ispisati informacije o pokupljenom bonusu.

Rezultat igre (engl. *score*) je brojčana vrijednost koju igrač nastoji poboljšati, odnosno postići najveću vrijednost. Ona se računa na sljedeći način:

Neka je n broj ciklusa (engl. *tick*) koji su prošli od početka igre, a s_n vrijednost prirasta rezultata/ciklusa, tada se rezultat s računa prema formuli 3-1

$$s = s_n * n \quad (3-1)$$

Vrijednost s_n je proizvoljna i ovisi o programeru. Za male vrijednosti s_n postojati će male razlike u rezultatu za različite pokušaje, dok za velike vrijednosti će se također pojaviti problem gdje je igračima teško pamtit i njihov najbolji rezultat jer broj sadrži puno znamenki.

Neka je v brzina igre u koordinate/ažuriranju (u konkretnom slučaju igre, ažuriranje se obavlja fiksno svakih 1/60 sekunde), b osnovna brzina (konstanta), m skala po kojoj se skalira brzina (konstanta), a s rezultat igre, tada je brzina v izračunata prema formuli 3-2.

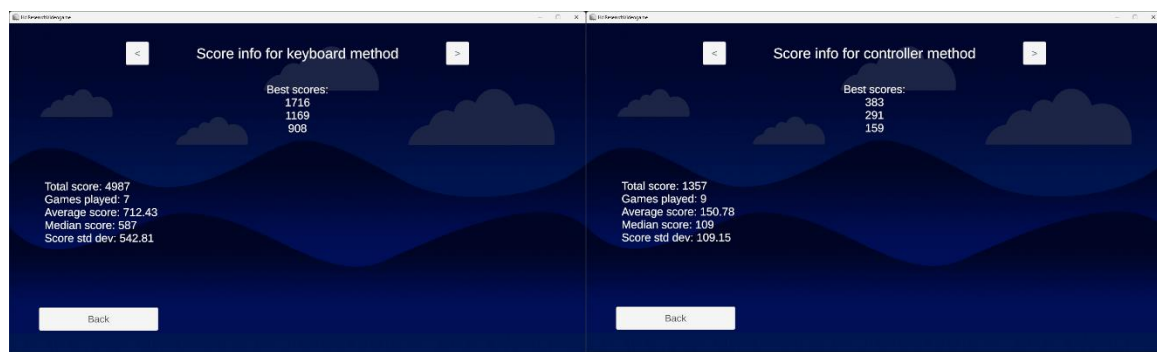
$$v = \frac{b + s}{m} \quad (3-2)$$

Formula 3-2 je proizvoljno odabrana zbog želje da se usko poveže brzina igre sa rezultatom. Moguće je dobiti sličan efekt tako da se brzina v računa preko vremena koje je prošlo od početka

igre, međutim tada kupljenje bonusa ne utječe na brzinu igre. Ovim pristupom je osigurano to da je igrač prisiljen napraviti odluku: pokupiti bonus i poboljšati rezultat ili ne pokupiti i imati sporiju igru.

Na slici 3.3. prikazano je kako rezultate igre igrač može pratiti u izborniku (engl. *menu*) “High Scores & Data”. Izbornik nudi igraču da izabere način igranja, te mu ispiše sljedeće informacije o njegovim rezultatima (ovisno o metodi unosa):

- Najbolja 3 rezultata (engl. *Best scores*)
- Ukupni rezultat (engl. *Total score*)
- Broj igri koje je igrao (engl. *Games played*)
- Prosječni rezultat (engl. *Average score*)
- Medijan rezultata (engl. *Median score*)
- Standardna devijacija rezultata



(a)

(b)

Slika 3.3. prikaz rezultata, (a) – igranje tipkovnicom, (b) – igranje kontrolerom

Kako bi se rezultati igre spremili između različitih paljenja igre, svi rezultati pohranjeni su u datoteku `highScores.json`. Ta datoteka nalazi se u glavnom direktoriju igre, pa igrač može odlučiti učitati datoteku u neki drugi program te dodatno obraditi podatke. Prilikom završetka igre poziva se metoda koja ažurira listu rezultata te sinkronizira stanje objekta `HighScores` sa `highScores.json` datotekom.

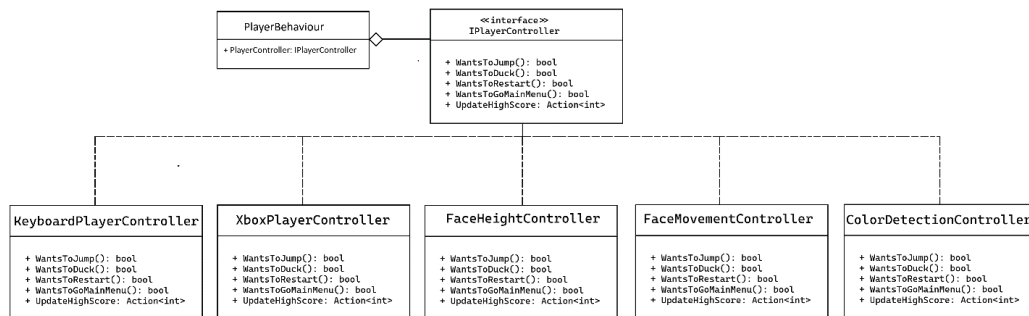
3.4. Sustav interakcije

U ovom poglavlju će biti opisan dizajn sustava interakcije, kao i njihov rad, te mogućnosti i mane.

3.4.1. Dizajn sustava za odabir interakcije

Kako bi igrač mogao birati kojom metodom interakcije želi raditi, dizajniran je sustav koji omogućuje biranje metode interakcije, a da u isto vrijeme ne dovodi do problema dupliciranja koda

te čvrste spregnutosti. Korišten je oblikovni obrazac “*Strategija*” [31] (engl. *Strategy*), a na slici 3.4. prikazan je UML dijagram. Strategija omogućuje ubrizgavanje ovisnosti [31] (engl. *dependency injection*) u objektno orijentiranom okruženju.



Slika 3.4. UML prikaz oblikovnog obrasca strategije – konkretni primjer

Klasa *PlayerBehaviour* koristi sučelje *IPlayerController* kako bi moglo raditi s bilo kojom od 5 klasa: *KeyboardPlayerController*, *XboxPlayerController*, *FaceHeightController*, *ColorDetectionController*, *FaceMovementController*.

Zadace/odgovornosti tih klasa su sljedeće:

- *KeyboardPlayerController* – proučiti stanje tipkovnice i sukladno odlučiti koju akciju želi napraviti igrač
- *XboxPlayerController* – proučiti stanje Xbox kontrolera i sukladno odlučiti koju akciju želi napraviti igrač
- *FaceHeightController* – proučiti stanje *FaceDetector* klase, te sukladno o informacijama o detektiranom licu odlučiti koju akciju želi napraviti igrač
- *ColorDetectionController* – proučiti stanje *ColorDetector* klase, te sukladno o informacijama o detekciji željene boje odlučiti koju akciju želi napraviti igrač
- *FaceMovementController* – proučiti stanje *FaceDetector* klase, te sukladno o informacijama o detektiranom licu odlučiti koju akciju želi napraviti igrač

Svaka od prethodnih klasa izvedena je također preko oblikovnog obrasca “*Singleton*” [31]. Singleton obrazac u objektno orijentiranom okruženju jamči da će postojati samo jedna instanca klase [31]. Pošto ove klase nemaju potrebe instancirati više objekata (jer samo jedan uređaj upravlja likom u svakom trenutku) poželjno je koristiti *singleton*.

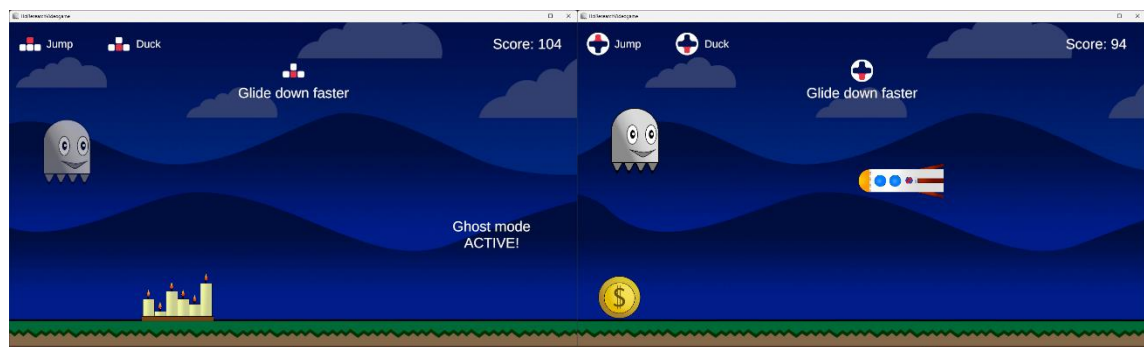
Sučelje *IPlayerController* propisuje sljedeće:

- *WantsToJump* – metoda koja vraća *bool* vrijednost koja označava namjeru da igrač skoči

- *WantsToDuck* – metoda koja vraća *bool* vrijednost koja označava namjeru da se igrač sagne
- *WantToRestart* – metoda koja vraća *bool* vrijednost koja označava namjeru igrača da ponovno krene igrati nakon što je igra završila.
- *WantsToGoMainMenu* – metoda koja vraća *bool* vrijednost koja označava namjeru igrača da ode na glavni izbornik igre nakon što je igra završila.
- *UpdateHighScore* delegat tipa *Action<int>* na koji će se postaviti referenca na metodu za ažuriranje rezultata

Ovisno o metodi igranja, prikaz na zaslonu (engl. *HUD – Heads Up Display*) se mora mijenjati. Primjerice, kada igrač igra tipkovnicom treba mu biti prikazano koja tipka se koristi za skakanje, a koja za saginjanje, koja za odlazak na glavni izbornik... Međutim, kada igra kontrolerom trebaju biti prikazane slike na kontroleru.

Na slici 3.5. vidljiv je prikaz na zaslonu koji se koristi kako bi se komuniciralo sljedeće:



(a)

(b)

Slika 3.5. prikaz na zaslonu, (a) – igranje tipkovnicom, (b) igranje kontrolerom

Prema slici 3.5. (a):

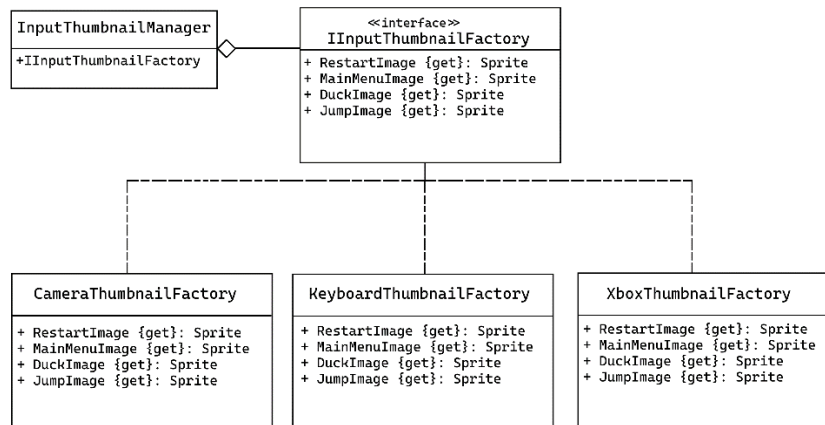
- Igrač može skočiti sa tipkom *strelica gore*
- Igrač se može sagnuti sa tipkom *strelica dolje*
- Igrač može brže pasti nakon skoka ako koristi *strelicu dolje*

Prema slici 3.5. (b):

- Igrač može skočiti s tipkom *d-pad gore*
- Igrač se može sagnuti sa tipkom *d-pad dolje*
- Igrač može brže past nakon skoka ako koristi *d-pad dolje*

Također, postoji kolekcija ikona i za metode gdje se koristi kamera (a to su metode igranja visinom lica, pokretima lica te detekcijom boje), kao i ikone za povratak na glavni izbornik i ponovno pokretanje igre.

Sustav koji osigurava pravilno korištenje ikona također koristi oblikovni obrazac *Strategija* [31], a UML diagram je prikazan na slici 3.6.



Slika 3.6. UML prikaz oblikovnog obrasca strategije – konkretni primjer

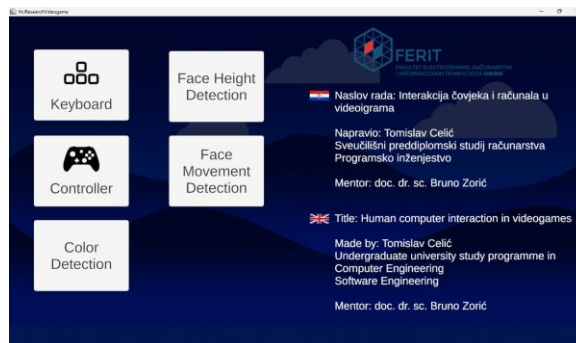
Klasa *InputThumbnailManager* koristi objekt koji implementira sučelje *IinputThumbnailFacotry* kako bi odredila koje slike se prikazuju na korisničkom sučelju:

- Slika *RestartImage* igraču govori kako da ponovno pokrene igru
- Slika *MainMenuImage* igraču govori kako da ode na glavni izbornik igre
- Slika *DuckImage* igraču govori kako da se sagne
- Slika *JumpImage* igraču govori kako da skoči

Klase koje implementiraju ovo sučelje su:

- *CameraThumbnailFactory* – prikazuje igraču bitne informacije dok on igra detekcijom visine lica, detekcijom pokreta lica ili detekcijom boja
- *KeyboardThumbnailFactory* – prikazuje igraču bitne informacije dok on igra tipkovnicom
- *XboxThumbnailFactory* – prikazuje igraču bitne informacije dok on igra kontrolerom

U grafičkom sučelju, nakon pritiska na play, igrač ima opciju izbora metode unosa, a to je prikazano na slici 3.7.



Slika 3.7. Prikaz grafičkog sučelja za odabir metode unosa

3.4.2. Interakcija tipkovnicom/Xbox kontrolerom

Interakcija tipkovnicom i Xbox kontrolerom su dvije standardizirane metode unosa. Slične su jer obje zahtijevaju isčitavanje stanja tipki. Na tipkovnici igrač pritisne strelicu gore te time izražava svoju namjeru da skoči, dok pritiskom strelice prema dolje izražava želju da se sagne. Igrač koji igru igra Xbox kontrolerom, mora pritisnuti D-pad gore kako bi skočio, odnosno D-pad dolje kako bi se sagnuo. Razlika u ove dvije metode unosa (engl. *input method*) jest ta što igrač koji igra na tipkovnici može pritisnuti obje tipke u isto vrijeme, dok igrač na Xbox kontroleru to ne može jer D-pad fizički to ne može napraviti, jer iako se D-pad na prvu čini kao 4 odvojene tipke, to je zapravo jedan tipka koja se miče u dvije osi. U slučaju da igrač pritisne obje tipke na tipkovnici doći će do nedefiniranog ponašanja, odnosno akcija koja se izvrši ovisiti će o tome kako Unity procesuirala metode unosa.

Velika prednost kod ove dvije metode unosa jest ta što igrač vrlo vjerojatno ima prethodnih iskustva s njima, te to što može prenijeti vještine sa, ili na, druge igre. Također, broj potencijalnih igrača koji već posjeduju kontroler ili tipkovnicu je veći. Mana ovih metoda unosa potiče od istih razloga, odnosno ove metode unosa mogu biti manje interesantne igraču jer je on već igrao s njima, te mu ne predstavljaju nikakva nova iskustva.

3.4.3. Detekcija lica

Igranje detekcijom lica može se ostvariti u više načina. Dva načina koja su implementirana u videoigri su: detekcija pozicije lica te detekcija pokreta lica. Igranje detekcijom lica nudi igraču mogućnosti da igra igru bez korištenja ruku. Igrač može odabrati igrati igru tako da pomiče svoju glavu, ili da pomiče cijelo svoje tijelo (npr. skakanje ili čučanj). Iako postoji jako puno potencijalnih metoda unosa, igranje detekcijom lica se dodatno razlikuje od drugih rješenja, jer ista nude da igrač igra bez tipkovnice/kontrolera, ali i dalje zahtijevaju korištenje ruku. Igranje bez

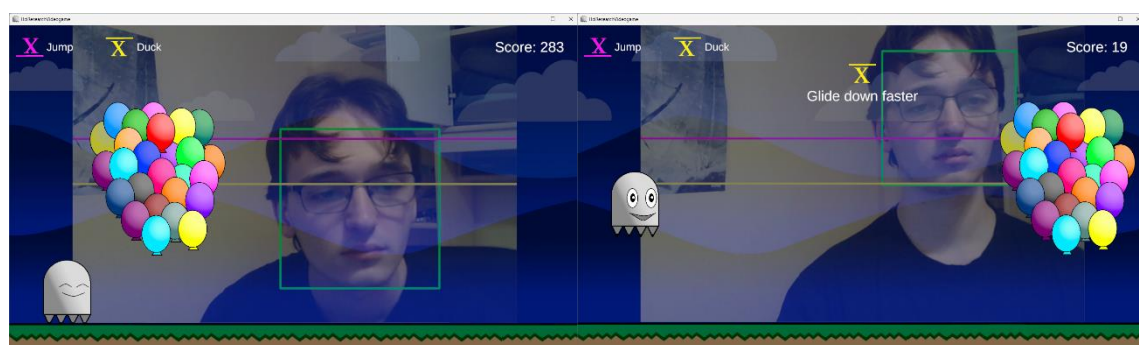
ruku može nadodati na efekt novog iskustva, te može nuditi igračima sa tjelesnim invalidnostima da igraju igru koju bi inače bili spriječeni igrati.

Velika mana koja vrijedi za sve metode igranja gdje je kamera ulazni uređaj, je ta što se u interakciju između čovjeka i računala uvodi nepredvidljivi faktor okruženja u kojem se on nalazi. Naime, igračevo okruženje može sadržavati objekte koji će biti krivo detektirani. Sama kvaliteta kamere, kao i osvjetljenje u okruženju utječe na njega. Kada igrač igra tipkovnicom, osvjetljenje u njegovoj prostoriji nije važno. Nije važno koliko slika ljudi ima na zidovima. Također, kod detekcije lica, problem se dešava kada druga osoba slučajno uđe u kadar kamere, te možda ista bude detektirana umjesto igrača.

Također, kod detekcija lica, stvara se problem povratne informacije. Prilikom pritiska tipke, igrač ima taktilan osjećaj, dok toga nema pri detekciji lica. Taj problem riješen je prikazom slike kamere, te iscrtavanjem zelenog kvadrata oko mjesta detekcije. To nije idealno, svugdje primjenjivo rješenje, jer potencijalno može utjecati na igračev doživljaj, odnosno može ometati umjetnički dizajn igre.

3.4.4. Interakcija detekcijom pozicije lica

Interakcija detekcijom pozicije lica radi na osnovi prepoznavanja lica, te ovisno o visini lica na slici odabire se akcija koju će lik napraviti. Igraču je prikazana ružičasta i žuta linija. Ukoliko igrač stavi centar svog lica iznad ružičaste linije, lik na ekranu će skočiti, a ukoliko stavi centar svog lica ispod žute linije lik će se sagnuti. Slika 3.8. prikazuje igranje pozicijom lica i sučelje koje prikazuje pragove/poziciju detekcije.



(a)

(b)

Slika 3.8. Igranje detekcijom pozicije lica: (a) sagnjanje, (b) skakanje

Prednost ovog pristupa je što je jednostavan i za implementirati i za komunicirati igraču što mora raditi. Također, nudi opciju da igrač sam sebi namjesti vrijednosti pragova, što je iznimno važno za uspješnost implementacije ove metode unosa, jer prostorno okruženje, kao i fizičke sposobnosti

svakog igrača nisu jednake. Igrači koji su niži ili su fizički udaljeniji od kamere će zahtijevati da pragovi budu niži, jer ne mogu visinom svog lica dosegnuti gornje dijelove slike. Problem je što ukoliko više igrača igra na jednom računalu naizmjenice, ili ukoliko isti igrač želi promijeniti svoju fizičku poziciju, mora ponovno namještat granice.

3.4.5. Interakcija detekcijom pokreta lica

Interakcija detekcijom pokreta lica radi na osnovu prepoznavanja lica, te na osnovu trenutne pozicije lica i prijašnjih pozicija lica. Razlika u pogledu na interakciju detekcijom pozicije lica je ta što igrač nema apsolutne pragove koje mora zadovoljiti, već relativne. Relativni pragovi znače da igrač ne mora nužno biti na istoj fizičkoj lokaciji tokom igranja, odnosno može se slobodno micati u okruženju te promijeniti pristup tokom igranja. Mana ovog pristupa je to što igraču iz prve ne mora nužno biti jasno koliko brzo se mora kretati licem, niti koliko mora pomaknuti lice da bi se detektirao pokret, već on to mora sam shvatiti. Pošto se koriste prethodne pozicije lica, te relativna razlika, može se dogoditi da kriva detekcija lica uzrokuje „trzaje“ lika u igri. Za uklanjanje istih potrebno je koristiti algoritam za uklanjanje pogrešno detektiranih vrijednosti. U videoigri korišten je sljedeći algoritam:

Za rad potrebni su sljedeći parametri:

n_m = broj uzoraka visine lica

t_m = odstupanje koje je potrebno da se registrira pokret

t_e = granica greške – ukoliko dva susjedna uzorka imaju razliku veću od ove vrijednosti smatra se da je došlo do greške

n_e = broj uzoraka koji se koristi za detekciju pogreške

Q_h = stog u koji pohranjujemo vrijednosti visine lica.

Matematički prikaz algoritma za uklanjanje krivo detektiranih vrijednosti, koristeći ranije navede oznake je prikazan u postupcima u formulama 3-3, 3-4, 3-5, 3-6 te krajnje na 3-7.

$$\forall i \in [1, \text{len}(Q_h)], n_e < i < n_m - n_e \quad (3-3)$$

Suma a_b uzoraka prije trenutnog uzorka je prema formuli 3-4.

$$a_b = \frac{\sum_{j=i-n_e}^{i-1} q_h[j]}{n_e} \quad (3-4)$$

Suma a_a uzoraka poslje trenutnog uzorka je prema formuli 3-5.

$$a_a = \frac{\sum_{j=i+1}^{i+q_n} q_h[j]}{n_e} \quad (3-5)$$

Prosječna vrijednost a je prema formuli 3-6.

$$a = \frac{a_b + a_a}{2} \quad (3-6)$$

Konačno, vrijednost $q_h[i]$ zamjenjuje se prema formuli 3-7.

$$q_h[i] = \begin{cases} a, & |q_h[i] - a| > t_e \wedge |a_a - a_n| < t_m, \\ q_h[i], & \text{inače} \end{cases}, \quad (3-7)$$

gdje je \wedge logički operator i.

Uklanjanjem loše detektiranih vrijednosti osigurano je da lik ne radi nagle, kratke radnje. Ovo je ključno jer može rezultirati da igrač „izgubi“ igru zbog greške koja nije njegova, što uzrokuje frustraciju. Korištenje algoritma za uklanjanje loše detektiranih vrijednosti nije idealno rješenje, jer može produljiti vrijeme od igračevog pokreta lica do radnje u igri. Kašnjenje unosa (engl. *input lag*) nije poželjna stvar jer može igraču narušiti dojam igranja igre. Također, pošto je izrađena igra natjecateljskog duha, odnosno potiče igrača da postigne što bolji rezultat, te od njega traži brze reakcije u napetim situacijama, moguće je isfrustrirati igrača jer ne može postići jednako dobar rezultat kakav bi postigao da je kašnjenje unosa manje.

3.4.6. Interakcija detekcijom boje

Interakcija detekcijom boje radi na osnovi prepoznavanja odabrane boje. Igrač navigira u postavke gdje odabire boju tako da stavi objekt čija se boja prepoznaje u centar prikaza kamere. Za objekt igrač može koristiti neke od sljedećih opcija:

- Boju odjeće – ovu opciju igrač može odabrati ukoliko želi kontrolirati lika na ekranu sukladno sa svojim pokretima (kada skoči uživo, skoči i lik na ekranu...). Kako bi ovaj pristup radio potrebno je da igrač nosi odjeću unikatne boje te da stoji dovoljno daleko od kamere da se ne bi njegova odjeća raspinjala cijelim vidikom kamere. Prednost ovog pristupa je što igrač ne mora koristiti ruke, ali mana je zahtijevanje većeg fizičkog prostora, odnosno udaljenost između kamere i igrača mora biti relativno velika (u usporedbi s drugim metodama unosa). Također, veća udaljenost može rezultirati time da igrač slabije vidi što se događa na ekranu.
- Boju objekta u ruci – ovu opciju igrač može odabrati ukoliko želi kontrolirati lika na ekranu sukladno sa pokretima ruke (kada podigne ruku, skoči lik na ekranu). Za ovakav objekt igrač može odabrati bilo koji objekt s bojom koja je unikatna njegovom okruženju, a tokom

razvoja su se koristili objekti poput bilježnica, flomastera, bojica i sličnih. Prednost ovog pristupa je što igrač ima sposobnost bržeg reagiranja, odnosno lakše mu je brzo pomaknuti ruku s objektom nego cijelo tijelo. Mana pristupa je što igrač i dalje mora koristiti ruke, tako da ovaj pristup ne olakšava igranje igračima sa invalidnostima koje ih sprječavaju u istom.

- Boja objekta na tijelu – ova opcija slična je opciji sa bojom odjeće, a izvodi se tako da se zalijepi mali objekt (najčešće papirić) unikatne boje na tijelo. Iako se na prvu ova opcija ne čini znatno drugačija od prve, rješava problem gdje se odjeća prostire cijelim vidikom kamere, što forsira igrača da se potencijalno udalji toliko daleko da teže vidi što se događa na ekranu. Tada igrač može igrati igru pomicanjem svog tijela (kada skoči uživo, skoči i lik na ekranu), međutim kod ovog pristupa bitno je paziti da objekt na tijelu ne otpadne, što nije problem kod prijašnja dva pristupa.

Prednost koja vrijedi za sve potencijalne metode igranje metodom detekcije boja jest ta što su pogrešne detekcije jako rijetke. Za implementaciju logike odlučivanja radnje moguće je koristiti detekciju pozicije boje, ili detekciju pokreta obojanog objekta. U videoigri je implementirana samo jedna metoda, a to je pozicijom. Razlog tomu je taj što za razliku od svog lica, igrač ima sposobnost promijeniti gdje se objekt nalazi u relativnom odnosu na sebe, odnosno, igrač može nalijepiti objekt više ili niže na svom tijelu, držati ga u ruci, koristiti boju koja se nalazi na određenom dijelu njegove odjeće i slično. Međutim ne može promijeniti gdje je njegovo lice u usporedbi s njegovim ramenima. Zbog toga, problemi koji postoje prilikom detekcije pozicije lica (igrač će možda promijeniti fizičku lokaciju, promijeniti će se igrač i drugi) nisu toliko izraženi prilikom igranja metodom detekcije bojom.

Algoritam pronalaska boje radi na principu da prolazi kroz sliku, odnosno matricu boja, na sljedeći način:

Neka je matrica M takva prema formuli 3-8

$$\begin{bmatrix} C_{00} & \cdots & C_{0m} \\ \vdots & \ddots & \vdots \\ C_{n0} & \cdots & C_{nm} \end{bmatrix}, \quad (3-8)$$

gdje je:

h = visina slike

w = širina slike

$$n = w - 1$$

$$m = h - 1$$

C = struktura $\{r, g, b, a\}$ koja predstavlja $RGB\alpha$ model boje, gdje je svaki kanal realni broj na intervalu $[0,1]$

Prolazom kroz matricu traži se pojavljivanje strukture C_{ij} koje se poklapa sa traženom strukturom. Za potrebe usporedbe moguće je ignorirati α kanal. Sam algoritam otkriva jednu manu igranja detekcijom boja. Algoritam analizira cijelu sliku, a kreće od ruba slike. Međutim igrač prilikom odabira boja možda ne primijeti određene objekte koji se nalaze u samim rubovima slike, pa tako zbog fokusiranja na centar ekrana odabere krivu boju. Također, treba imati na umu da se nijansa boje mijenja u ovisnosti o svjetlu, a određeni objekti kompletno mogu promijeniti boju. Ovo se događa prirodno, jer kako igrač pomiče objekt, izvor svjetla će imati drugačiji kut sa objektom. Još jedna mana je ta što prethodno navedeni efekt nije jednak na svim objektima. Objekti obojani metalik bojama mogu na kameri izgledati potpuno bijelo pod određenim kutom svjetla, a ovo ne mora biti očigledno igraču prilikom odabira boje, odnosno igrač može izabere „lošiji“ objekt te bude frustriran ukoliko se isto ponovi više puta. Samo detektiranje boje mora se obaviti s tolerancijom.

Neka je C_t boja tolerancije, C_u boja koja se trenutno uspoređuje, a C_p boja koju se pokušava naći. Tada se usporedba s tolerancijom vrši na prema formuli 3-9.

$$b = (|r_u - r_p| < r_t) \wedge (|g_u - g_p| < g_t) \wedge (|b_u - b_p| < b_t) \quad (3-9)$$

gdje je:

\wedge aritmetičko-logički operator “i”

b rezultat usporedbe, vrijednosti *bool* {istina (engl. *true*), laž (engl. *false*)}

Tolerancija boje je nešto što igrač sam odabire prilikom odabira boje. Iako je prednost što igrač ima opciju da si poboljša detekciju, postoji šansa da zbog prevelike tolerancije detekcija boje bude neuspješna, što je mana. U usporedbi s klasičnim metodama unosa, igranje detekcijom boja znači da će igrač više vremena potrošiti u postavkama, kako bi si namjestio odabranu boju, što nije cilj igraču, pa se isto smatra manom.

Iako ova metoda igranja potencijalno može nuditi alternativni način igranja osobama s invaliditetom, mana je što za razliku od klasičnih metoda unosa ne može se igrati, ili je otežano igranje ukoliko osoba pati od daltonizma. Također, postoji problem povratne informacije. U videoigri taj problem je riješen prikazom slike kamere te iscrtavanjem malog zelenog kvadratića

na mjestu detekcije. Mana ovog pristupa je što u igri se tada nalazi indikator koji može brzo „skakutati“ okolo po slici, te ometati igrača.

3.5. Vrednovanje igre i metode unosa

U svrhu vrednovanja igre i alternativnih metoda unosa, napravljen je upitnik koristeći *Google Forms*. 32 ispitanika sudjelovalo u anketi, a isti su dobili igru nakon što je utvrđeno da posjeduju kameru za računalo, kontroler i tipkovnicu. Drugim riječima, ispitanici su prihvaćeni u istraživanje nakon što je potvrđeno da mogu koristiti sve metode unosa. Nakon što je potvrđeno da su odigrali sve metode unosa, dobili su poveznicu na anketu te istu ispunjavaju anonimno.

3.5.1. Pitanja

U upitniku postavljeno je 13 pitanja podijeljenih u 4 cjeline. Pitanja su prikazana u tablici 3.1.

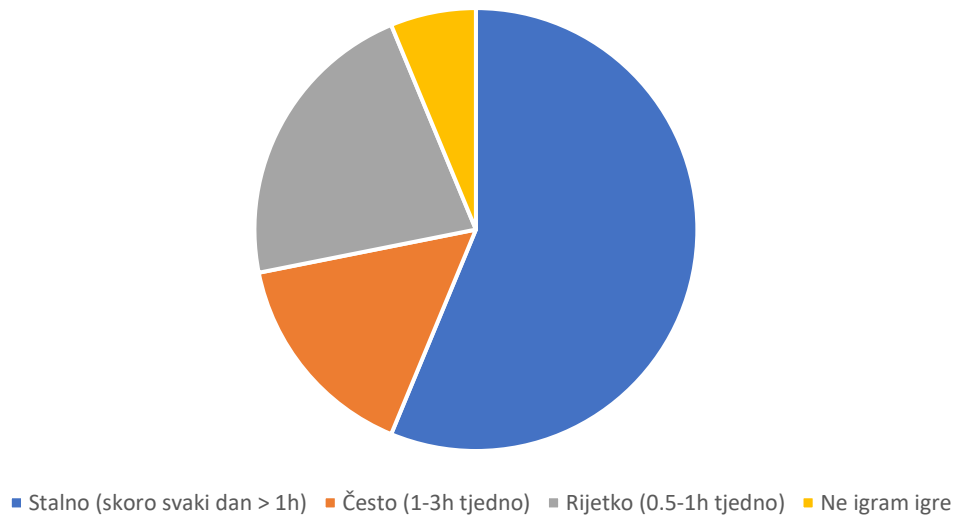
Tablica 3.1. Pitanja postavljena sudionicima

Broj	Pitanje
Osnovno o igraču	
1.	Koliko često igrate videoigre?
2.	Koju metodu igranja/unosa najviše preferirate?
Pravednost metode igranja	
1.	Od 1 do 5, ocijenite metodu igranja tipkovnicom
2.	Od 1 do 5, ocijenite metodu igranja kontrolerom
3.	Od 1 do 5, ocijenite metodu igranja pozicijom lica
4.	Od 1 do 5, ocijenite metodu igranja bojom
5.	Od 1 do 5, ocijenite metodu igranja pokretima lica
Označavanje istinitih tvrdnji	
1.	Označite tvrdnje koje vrijede za metodu igranja sa tipkovnicom
2.	Označite tvrdnje koje vrijede za metodu igranja sa kontrolerom
3.	Označite tvrdnje koje vrijede za metodu igranja pozicijom lica
4.	Označite tvrdnje koje vrijede za metodu igranja bojom
5.	Označite tvrdnje koje vrijede za metodu igranja pokretom lica
Dojmovi	
1.	Vaša povratna informacija i dojmovi

3.5.2. Rezultati upitnika i interpretacija

Na upitnik odgovorila su 32 igrača. Na slici 3.9. prikazana je podjela po učestalosti igranja videoigara grafom.

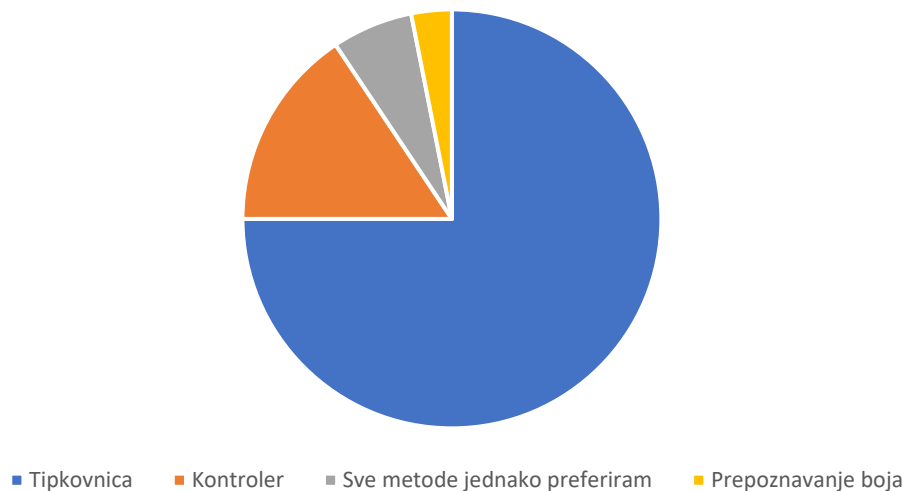
Koliko često igrate videoigre?



Slika 3.9. Graf podjele učestalosti igranja

Iz podjele jasno je da će zaključci o igri biti najtočniji za populaciju koja igra igre stalno, dok će biti teško pouzdano zaključiti što misle ljudi koji ne igraju igre. Na slici 3.10. vidljiva je podjela preferencija prema metodi unosa.

Koju metodu igranja/unosa najviše preferirate?



Slika 3.10. Preference metode unosa

Jasno je kako igrači preferiraju stare metode unosa s kojima su već upoznati, od kojih tipkovnica ima značajnu prednost nad kontrolerom. Jedina nova metoda unosa koja je uopće bila preferirana jest metoda igranja detekcijom boje. Za bolji uvid u zašto igrači preferiraju klasične metode, potrebno je pogledati za koje metode unosa ispitanici misle da su pravedne (pravednost je u ovom

kontekstu sposobnost da igrač postigne jednaki rezultat svim metodama), jer igrači iako im je nešto zabavno, možda neće nužno htjeti igrati ako se osjećaju da neće moći postići željene rezultate. Prikaz distribucije pravednosti nalazi se na Tablici 3.2.

Tablica 3.2. Distribucija pravednosti (sposobnosti da se ostvari jednaki rezultat)

Metoda igranja/ocjena	1	2	3	4	5	Prosjek
Tipkovnica	0	0	0	15.63%	84.38%	4,84
Kontroler	0	3.13%	6.25%	21.88%	68.75%	4,56
Visina lica	0	12.5%	59.38%	25%	3.13%	3,19
Boja	0	25%	25%	28.13%	21.88%	3,47
Pokreti lica	3.13%	12.5%	43.75%	34-38%	6.25%	3,28

Jasno je da da igrači misle da je metoda igranja tipkovnicom najpravednija, dok je druga po redu metoda igranja s kontrolerom. Alternativne metode unosa koje su rađene u ovom radu su uglavnom imale osrednje rezultate. Metoda igranja detekcijom boje je jedina dobila rezultate koji ne prate normalnu distribuciju, a to može biti jer metoda jako ovisi od igrača do igrača. Izbor objekta koji se koristi za boju, kao i očekivanje igrača moglo je grupirati ispitanike u različite skupine. I dalje, metoda igranja detekcijom boje dobila je bolje rezultate od dvije metode igranja licem, iako prosječni rezultat nije bio puno bolji. Dublji uvid može se dobiti preko treće skupine pitanja, a to su tvrdnje s kojima se slažu ispitanici, koje su prikazane u tablici 3.3.

Tablica 3.3. Tvrdnje s kojima se slažu ispitanici

Tvrdnja	Tipkovnica	Kontroler	Visina lica	Boje	Pokreti lica
Nepouzdana	0	3.13%	25%	12.5%	34.38%
Puno namještanja	0	6.25%	68.75%	50%	53.13%
Intuitivna	81.25%	71.88%	25%	31.25%	28.13%
Potrebno puno prostora	3.13%	6.25%	46.88%	31.25%	37.5%
Zabavna	87.5%	75%	46.88%	53.13%	46.88%

Na tablici 3.3. vidljivo je koliko se ispitanika slagalo s određenom tvrdnjom u odnosu na metodu unosa. Tvrdnje za koje se može reći da smetaju igraču pri interakciji sa igrom (zahtijevanje puno prostora, nepouzdanost, potreba za puno namještanja) češće su kod metoda unosa koje su uvedene kod ovog rada, dok se iste pojavljuju rijetko u tradicionalnim metodama unosa (tipkovnica, kontroler). Pozitivne tvrdnje (metoda je zabavna za igranje, metoda je intuitivna) češće su kod tradicionalnih metoda unosa. Dio toga može biti zbog prikaza korisničkog sučelja, no ne treba zanemariti kako su ispitanici skoro sigurno imali prethodnih iskustva s tradicionalnim metodama unosa, dok isto nije slučaj s alternativnim metodama. Od alternativnih metoda, metoda igranja detekcijom boje najbolje je prošla u ovom krugu pitanja, međutim razlike nisu velike.

U konačnici potrebno je proučiti dojmove koje su ostavili ispitanici. 21 od 32 ispitanika (65,63%) ispunilo je ovo pitanje, no nisu svi odgovori pogodni za analizu. Komentari poput “dosta zanimljiva igra” i “Zanimljiva igra.” nisu pogodni za analizu. Također, postoji i šansa da su ga ispitanici ostavili jer su mislili da je pitanje obvezno, a ne zato što su se osjećali kao da trebaju reći još nešto. Prikaz odgovora nalazi se na tablici 3.4.

Tablica 3.4. Dojmovi

<i>Broj</i>	<i>Odgovor</i>
1	Malo lošiji komentari za metode igranja sa visinom lica, odabirom boje i pokretima lica dolaze iz činjenice da nisam dovoljno upoznat s takvim vrstama kontrole, kao u primjeru gdje se pita za pravednost različitih metoda igranja. Smatram da su ovakve vrste kontroli zastupljenije u videoigrama da bi odgovori bili potpuno drugačiji, tj. njihova pravednost je usko vezana sa korisnikovim mogućnostima i koliko se osjećaju upoznato sa istima.
2	Zabava za cijelu obitelj. Trebalo mi je par pokušaja da shvatim da detektiranje visine lica radi najbolje kad se malo udaljim. Detektiranje pomicanja lica je bilo u redu, ali nije uvijek pohvatalo moje pokrete. Imao sam malo više problema sa detektiranjem boje, možda nisam dobro namjestio parametre u postavkama. "Color Detection Pixels To Skip" slider bi mogao imati malo veći kontrast. I znam da malo cjepidlačim sada, ali bila bi dobre tipke za izlaz i za puni zaslon. Videoigra je opuštajuća, ali i dovoljno izazovna da bude zabavan, a zaokret korištenja kamere i micanje glave donosi sa sobom dodatan izazov i puno zabave. Glavni lik je sladak, i volio bih da dobije neko ime (ako ga već nema). Za kraj želim naglasiti kako je ovo prva igrica ikad koju sam igrao sa uključenom kamerom, i da sam ugodno iznenađen sa svojim iskustvom.
3	Bez obzira na lošu kvalitetu moje kamere, igranje koristeći metoda visine lica i pokreta lica je bilo vrlo zabavno. Iako je bilo teže i ponekad lik nije radio ono što ja želim, nikad prije nisam igrao igru na ovaj način. Da je igra bila samo sa tipkovnicom ne bih ju ni probao.
4	Videoigra je zanimljiva. Podsjeća na neke popularne igre koje postoje, ali s novim zaokretom - mogućnost igranja na različite načine. Svaki način je intuitivan i dobro prikazan kako se igra. Videoigra dobro prikazuje kako je zahtjevnije igrati na druge načine koji nisu tipkovnica ili kontroler, no kako na taj način se igrač mora i više potruditi što može učiniti igru zanimljivijom.
5	Metoda tipkovnicom i kontrolerom se čini sasvim uredu, no upravljanje licem i bojama je pomalo nezgrapno
6	Igranje sa tipkovnicom ili kontrolerom se mogu postići najbolji rezultati, metoda sa bojama je zanimljiv koncept, ali smatram da je puno zabavnije igrati sa nekom od metoda koja koristi lice.

Ispitanici su u komentarima istaknuli kako su manje upoznati s alternativnim metodama unosa, što je očekivano, jer ipak to su „alternativne“ metode unosa. Najveći problemi kod alternativnih metoda unosa su problemi s detekcijom (kašnjenje, kriva detekcija i sl.). Iako postoji prostor za unapređenje detekcije lica (naprijedniji modeli, bolje podešavanje modela u početku), ovo je značajan problem jer će se praktički uvijek pojavljivati. Također, činjenica je da igrači smatraju da ne mogu postići isti rezultat alternativnim metodama unosa kao sa klasičnim metodama unosa. Ovaj problem nije nužno moguće riješiti, zbog činjenice da pokreti tijela (lice, ruka koja drži objekt sa bojom, skakanje i sl.) su sporiji od pokreta prstiju na tipkovnici ili kontroleru. Još jedan uočeni problem je to što igrači iz prve ne znaju kako pristupiti alternativnim metodama unosa (npr. osoba koja je ostavila komentar 2 nije znala koliko se udaljiti od kamere). Ovo je greška u dizajnu

grafičkog sučelja, a nije jedina te vrste. Greške grafičkog sučelja moguće je ispraviti, no zbog prethodno istaknutih mana nije nužno da će se postići bolji rezultati. Bitna prednost je to što igrači smatraju da su alternativne metode unosa zabavnije, međutim ukoliko usporedimo komentare u tablici 3.4. sa slikom 3.10. gdje su prikazane preference prema metodama unosa, dolazi do kontradikcije. Kontradikciju moguće je objasniti teorijom da igrači kojima se svidjela ideja videoigre i alternativnih metoda unosa su osjećali veću potrebu da ostave komentar, dok igrači kojima se nije svidio koncept ili izvedba videoigre/alternativnih metoda unosa su htjeli odraditi anketu što prije.

Za kraj, može se interpretirati kako alternativne metode unosa poput metoda igranja sa licem i metode igranja detekcijom boje nisu intuitivne i zabavne kao klasične metode unosa. Također, ove metode donose nove mane, poput potrebe za dodatnim namještanjem i zahtjevom za više prostora, zbog čega je još teže opravdati dodavanje ovih metoda u videoigre. Iako su u teoriji alternativne metode unosa mogu nuditi veću pristupačnost igraču, zbog dodatnih mana to nije nužno slučaj. Potrebno je naglasiti kako testiranje videoigre je, zbog okolnosti, najviše provedeno nad populacijom koja već igra videoigre, dok su osobe koje ne igraju videoigre znatno manje zastupljene.

4. ZAKLJUČAK

Interakcija čovjeka i računala je važna znanstvena disciplina zbog toga što danas ljudi sve više i više imaju interakcije s računalima u svakodnevnom životu. Normanovi ciklusi kao i okvir za interakciju opisuju načine interakcije čovjeka sa računalom. Osim za produktivne aktivnosti, računala se koriste i za zabavu, pri čemu se mogu istaknuti videoigre. HCI je važna disciplina za izradu videoigara te nudi načine za alternativne metode unosa, a neke od alternativnih metoda unosa su dostupne na tržištu.

U radu je izrađena videoigra čija je svrha vrednovanje novih alternativnih metode unosa u usporedbi s klasičnim metodama unosa. Zbog toga je dizajnirana da bude jednostavna i prepoznatljiva igraču, a da u isto vrijeme nudi alternativne metode unosa koje će igrači moći isprobati bez potrebe da se istovremeno privikavaju na novu igru. Izrađena videoigra ima 5 različitih metoda unosa, od koje su 3 alternativne. Testiranjem videoigre na 32 igrača dobiven je jasan zaključak da igrači preferiraju klasične metode unosa zbog lakoće igranja te zabave. Mogućnost postizanja manjeg rezultata kao i teškoće pri samom igranju su glavni uzroci prethodnih problema.

Trenutni pristup ima mane, poput činjenice da se za sve alternativne metode unosa koristi kamera kao ulazni uređaj. Kamera ima nekoliko mana, poput činjenice da igrač ne može igrati u mraku ili sa više igrača oko sebe, koje su jednake za sve alternativne metode unosa u izrađenoj videoigri. Dodavanjem alternativnih metoda unosa koje se ne oslanjaju na kameru potencijalno je moguće dobiti drugačije rezultate. Također, mana trenutnog pristupa je to što igrači imaju mogućnost isprobati alternativne metode unosa samo na jednoj videoigri, dok je poželjno izraditi ih više. Tim pristupom moguće je bolje vrednovati metode unosa, jer ih igrači ne koriste samo u kontekstu jedne videoigre. Dodatno, potrebno je igru puno bolje testirati među populacijom koja ne igra videoigre često. Iako alternativne metode mogu nuditi dodatni stupanj pristupačnosti, alternativne metode unosa napravljene u ovom radu su pokazale da imaju dodatne prepreke koje sprječavaju igrača u igranju.

LITERATURA

- [1] Association of Human-Computer Interaction, What is human-computer interaction? [online], Association of Human-Computer Interaction, Ujedinjeno Kraljevstvo, dostupno na: <https://www.hci.org.uk/human-computer-interaction/> [22.08.2024.]
- [2] J. M. Carroll, Human Computer Interaction - brief intro [online], Interaction Design Foundation – IxDF, Aarhus, dostupno na: <https://www.interaction-design.org/literature/book/the-encyclopedia-of-human-computer-interaction-2nd-ed/human-computer-interaction-brief-intro> [22.08.2024.]
- [3] PYMNTS, Smartphones, Smart TVs Drive Surge in Connected Device Ownership in 2023 [online], PYMNTS, Boston, 2023., dostupno na: <https://www.pymnts.com/connectedeconomy/2023/smartphones-smart-tvs-drive-surge-in-connected-device-ownership-in-2023/> [22.08.2024.]
- [4] B. Blumenschied, 10 Real Life Examples of Embedded Systems [online], DIGI, 2021, Minneapolis, Minnesota, dostupno na: <https://www.digi.com/blog/post/examples-of-embedded-systems> [22.08.2024.]
- [5] A. Dix, J. E. Finlay. G. D. Abowd, Human-Computer Interaction – Third edition, Pearson, London, 2006.
- [6] D. A. Norman, The Design of Everyday Things: Revised and Expanded Edition, Basic Books, New York, 2013.
- [7] G. D. Abowd, R. Beale, Users, systems and interfaces: a unifying framework for interaction, People and Computers Sv. VI, str. (73-87), Cambridge University Press, Cambridge, 1991.
- [8] Accesibility [online], Interaction Design Foundation – IxDF, Aarhus, 2016., dostupno na: <https://www.interaction-design.org/literature/topics/accessibility> [07.09.2024.]
- [9] R. Pagulayan, K. Keeker, T. Fuller, D. Wixon, R. Romero, D. Gunn, “User-Centered Design in Games,” Human–Computer Interaction Handbook, pp. 795–822, Svibanj 2012.
- [10] Game Developer, The Psychological Perspective on Game Design [online], Game Developer, 2019, London, dostupno na: <https://www.gamedeveloper.com/design/the-psychological-perspective-on-game-design#close-modal> [21.08.2024.]

- [11] A. Brazie, Video Game Balance: A Definitive Guide [online], Game Design Skills, dostupno na: <https://gamedesignskills.com/game-design/game-balance/> [21.08.2024.]
- [12] M. Figgett, What do “nerf” and “buff” mean in gaming? [online], The Big Tech Question, London, 2022., dostupno na: <https://bigtechquestion.com/2022/01/31/gaming/nerf-buff-mean/> [23.08.2024.]
- [13] E. Makuch, Why Rainbow Six Siege Doesn't Let You Respawn [online], Gamespot, San Francisco, 2014., dostupno na: <https://www.gamespot.com/articles/why-rainbow-six-siege-doesnt-let-you-respawn/1100-6422962/> [23.08.2024.]
- [14] A. Brough, Adder [online], gtacars.net, Ujedinjeno Kraljevstvo, 2022, dostupno na: <https://gtacars.net/gta5/adder> [23.08.2024.]
- [15] V. Horde, J. Petty, J. Ryan, How to Make Money in GTA 5 [online], IGN, San Francisco, Kalifornija, 2023., dostupno na: <https://www.ign.com/wikis/gta-5/How-to-Make-Money-in-GTA-5> [23.08.2024.]
- [16] Rocky Mountain College of Art + Design, The Psychology of Game Art: How Colors and Design Affect Player Behavior [online], Rocky Mountain College of Art + Design, Denver, 2024., dostupno na: <https://www.rmcad.edu/blog/the-psychology-of-game-art-how-colors-and-design-affect-player-behavior/> [23.08.2024.]
- [17] O. Toms, Fortnite Storm guide (V8.00) - Fortnite Storm Eye, Storm stats, Storm tips and tricks [online], Rock Paper Shotgun, Halstead, 2019., dostupno na: <https://www.rockpapershotgun.com/fortnite-storm-guide-v7-30-fortnite-storm-eye-storm-stats-storm-tips-and-tricks> [23.08.2024.]
- [18] K. Franek, Microsoft Revenue Breakdown by Product, Segment and Country [online], Kamil Franek, dostupno na: <https://www.kamilfrank.com/microsoft-revenue-breakdown/> [23.08.2024.]
- [19] S. Butler, Controller vs. Mouse and Keyboard: Which Is Better? [online], HowToGeek, Sterling, 2023., dostupno na: <https://www.howtogeek.com/873011/controller-vs-mouse-and-keyboard/> [23.08.2024.]
- [20] OC Racing, What is Force Feedback?, OC Sim Racing, Rockville, [online], dostupno na: <https://www.ocsimracing.com/reviews/what-is-force-feedback> [23.08.2024.]

- [21] J. Flynn, 20+ Mobile Game Statistics [2023]: Revenue, Monetization, Downloads, Users [online], Zippia, San Francisco, 2023., dostupno na: <https://www.zippia.com/advice/mobile-game-statistics/> [23.08.2024.]
- [22] J. Hollander, Is Fortnite Cross-Platform? An Essential Fortnite Platforms Guide [online], setup.gg, 2024., dostupno na: <https://www.setup.gg/game/fortnite/fortnite-cross-platform/> [23.08.2024.]
- [23] J. Uke, What Is Aim Assist In Gaming? [online], Gametree, Los Angeles, dostupno na: <https://gametree.me/gaming-terms/aim-assist/> [24.08.2024.]
- [24] T. Kim, In-Depth: Eye To Eye - The History Of EyeToy [online], Gamasutra, San Francisco, dostupno na: https://web.archive.org/web/20170823204426/http://www.gamasutra.com/php-bin/news_index.php?story=20975 [24.08.2024.]
- [25] R. Marks, EyeToy, Innovation and Beyond [online], Sony, San Francisco, 2010., dostupno na: <https://blog.playstation.com/2010/11/03/eyetoy-innovation-and-beyond/> [24.08.2024.]
- [26] Caméra EyeToy - PlayStation 2 [online], Reference-Gaming , Oraison, dostupno na: <https://www.reference-gaming.com/fiche-produit/camera-eyetoy-playstation-2> [25.08.2024.]
- [27] E. Chan, Xbox Kinect History [online], Gadgetmates, Las Vegas, 2021., dostupno na: <https://gadgetmates.com/gaming-history-a-look-back-at-the-xbox-kinect> [25.08.2024.]
- [28] J. Corden, Farewell, dear sweet Kinect [online], Windows Central, Bath, 2018., dostupno na: <https://www.windowscentral.com/ode-kinect> [25.08.2024.]
- [29] M. Yam, Xbox One Review: Unifying Your Living Room Experience [online], Tom's Hardware, New York, 2013., dostupno na: <https://www.tomshardware.com/reviews/microsoft-xbox-one-console-review,3681-6.html> [25.08.2024.]
- [30] A. Wallace, 15 Best Microsoft Kinect Games Of All Time [online], Retro Dodo, New York, dostupno na: <https://retrododo.com/best-microsoft-kinect-games/> [25.08.2024.]
- [31] E. Gamma, R. Helm, R. Johnson. J. Vlissides, Design Patterns: Elements of Reusable Object-Oriented Software, Prentice Hallm, Upper Saddle River, 1995.

- [32] A. Vandervell, Nintendo Wii Review [online], Trusted Reviews, London, 2006., dostupno na: <https://www.trustedreviews.com/reviews/nintendo-wii> [10.09.2024]
- [33] Xbox, Xbox Adaptive Controller [online], Xbox, Redmond, dostupno na: <https://www.xbox.com/en-US/accessories/controllers/xbox-adaptive-controller?msocid=1c278cdcee656ed237c198beefd46faa> [11.09.2024]
- [34] J. Brodtkin, How Unity3D Became a Game-Development Beast [online], Dice, New York, 2013., dostupno na: [How Unity3D Became a Game-Development Beast | Dice.com Career Advice](#) [20.08.2024.]

SAŽETAK

U teorijskom dijelu rada napravljen je pregled znanstvene discipline Interakcije čovjeka i računala. Interakcija čovjeka i računala je znanstvena disciplina koja proučava kako čovjek koristi računala te kako dizajn istih utječe na lakoću korištenja. Rad daje pregled modela i okvira za opisivanje HCI-a, te govori o sučelju (spoju izlaza i ulaza) i njegovom utjecaju na pristupačnost. Također, su prikazane razlike u videoigrama koje utječu na interakciju čovjeka i računala. Napravljen je pregled tehnologija koje pokušavaju nuditi alternativan načina unosa u videoigrama. U praktičnom dijelu izrađena je videoigra čija je svrha vrednovanje 3 alternativne metoda unosa u usporedbi sa 2 klasične metode unosa. Videoigra je dizajnirana s ciljem da bude poznata širem broju igrača, kako bi se lakše vrednovale alternativne metode unosa. Napravljeno je istraživanje na 32 osobe te su interpretirani rezultati.

Ključne riječi: interakcija čovjeka i računala, kontrole, prepoznavanje aktivnosti, videoigre

ABSTRACT

Human computer interaction in videogames

In the theoretical part of the thesis, a review of the scientific discipline of Human-Computer Interaction is presented. Human-Computer Interaction is a scientific discipline that studies how humans use computers and how computer design affect ease of use. The thesis provides an overview of a model and a framework for describing human-computer interaction and discusses the interface (input and output) and its impact on accessibility. Differences in video games that affect human-computer interaction are also presented. A review of technologies that offer alternative input methods in video games is provided. In the practical part, a video game was developed with the aim of evaluating three alternative input methods in comparison with two "classic" input methods. The video game was designed to be familiar to a wider audience of players, in order to more accurately evaluate alternative input methods. Research was conducted on 32 participants, and the results were interpreted.

Keywords: human-computer interaction, controls, activity recognition, video games

PRILOZI

1. Interakcija čovjeka i računala u videoigrama u pdf formatu
2. Izvorni kod programskog rješenja dostupan na: <https://github.com/celkiboi/HCIResearch>