

Tipografija u interaktivnoj računalnoj animaciji

Šubat, Tea

Master's thesis / Diplomski rad

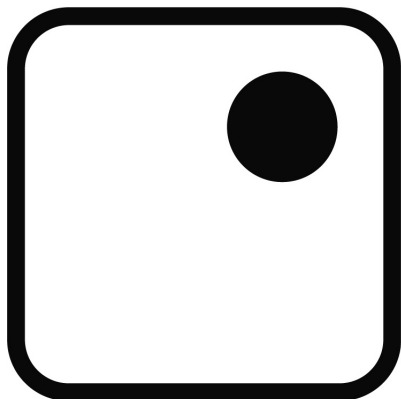
2015

Degree Grantor / Ustanova koja je dodijelila akademski / stručni stupanj: **University of Zagreb, Faculty of Graphic Arts / Sveučilište u Zagrebu, Grafički fakultet**

Permanent link / Trajna poveznica: <https://um.nsk.hr/um:nbn:hr:216:065137>

Rights / Prava: [In copyright](#)/[Zaštićeno autorskim pravom.](#)

Download date / Datum preuzimanja: **2024-09-18**



Repository / Repozitorij:

[Faculty of Graphic Arts Repository](#)



SVEUČILIŠTE U ZAGREBU
GRAFIČKI FAKULTET

TEA ŠUBAT

TIPOGRAFIJA U INTERAKTIVNOJ
RAČUNALNOJ ANIMACIJI

DIPLOMSKI RAD

Zagreb, 2015



Sveučilište u Zagrebu
Grafički fakultet

TEA ŠUBAT

TIPOGRAFIJA U INTERAKTIVNOJ RAČUNALNOJ ANIMACIJI

DIPLOMSKI RAD

Mentor:
Izv. prof. dr. sc. Ivana Žiljak Stanimirović

Student:
Tea Šubat

Zagreb, 2015

SAŽETAK

Kao glavna uloga tipografije u klasičnom smislu navodi se davanje vizualnog oblika sadržaju u funkciji prijenosa informacije. No, tipografija se primjenjuje na brojne druge načine, uključujući i njenu upotrebu kao samostalni element visoke dekorativnosti u vizualnim medijima. U tim slučajevima, slovni znakovi postaju izražajno sredstvo i nosioc informacija, neovisno o njihovoj čitljivosti koja je ponekad potpuno negirana, a naglasak se stavlja na vizualni dojam.

Dodatne mogućnosti izražavanja tipografijom uvelike su proširene njenim integriranjem u računalnu animaciju, gdje se umjesto statičnog, kratkotrajnog dojma izmjenama kroz vrijeme postiže mnogo dinamičniji efekt, a slovne je znakove moguće koristiti za stvaranje kompleksnih dinamičnih kompozicija. Potpuno novi pristup korištenju vizualnih elemenata u animaciji, pa tako i tipografije, dobiva se uvođenjem interakcije s gledateljem, koji tako iz pasivnog promatrača postaje aktivni korisnik aplikacije čiji krajnji izgled nije predefiniран, već proizlazi iz akcija svakog pojedinog korisnika.

U ovom radu sadržan je pregled nekih od postojećih primjera interaktivnih animacija koje kao svoje glavno izražajno sredstvo koriste tipografiju, te je prikazan proces izrade vlastitog primjera pomoću prethodno dizajniranog fonta temeljenog na fraktalima. Također se istražuju izražajne mogućnosti i prilagodljivost postojećih elemenata različitim transformacijama te način i mogućnosti uvođenja interakcije u kreiranu animaciju.

KLJUČNE RIJEČI: tipografija, računalna animacija, interaktivnost

ABSTRACT

The main role of typography in the classical sense is described as giving the written content visual form to convey information. However, typography is used in numerous other ways, including its use as a stand-alone, highly decorative element in the visual media. In such cases, the characters become both means of expression and carriers of information, regardless of their readability, which is sometimes completely negated, putting emphasis on their visual impression.

Further ways of expression using typography are made possible by its integration in computer animation. Instead of static, short term impression, changes over time let typography achieve a much more dynamic effect, and the characters can be used to create highly complex compositions in motion. An entirely new approach to using visual elements in animation, including typography, is obtained by introducing interaction with the viewer. The passive viewer then becomes an active user of an application whose final appearance is not predefined, but rather created by the actions of each individual user.

The theoretical part of this paper includes an overview of existing examples of interactive animation that use typography as their primary means of expression. The practical part describes the process of creating the author's own example, using a previously designed font based on the idea of fractals. It also explores the expressive potential and flexibility of the existing elements, as well as the methods and possibilities of introducing interactivity in the created animation.

KEYWORDS: typography, computer animation, interactivity

SADRŽAJ

1. UVOD.....	1
2. TEORIJSKI DIO	2
2.1. Tipografija u računalnoj animaciji.....	2
2.1.1. Izražajne mogućnosti i podjela animirane tipografije	3
2.1.2. Čitkost i čitljivost	6
2.1.3. Razvoj animirane tipografije kroz povijest	7
2.2. Vrste i način izrade računalne animacije.....	9
2.2.1. Kontrola pokreta	10
2.2.2. Definiranje ključnih okvira - <i>Keyframing</i>	11
2.2.3. Vrste animiranih sustava	12
2.2.4. Programski jezik Processing.....	14
2.3. Interaktivnost i korisnički generiran sadržaj.....	16
2.3.1. Interaktivna umjetnost	17
2.3.2. Interaktivna računalna animacija	18
2.3.3. Upotreba tipografije u interaktivnoj računalnoj animaciji.....	19
2.3.4. Korisnički generiran sadržaj.....	22
3. EKSPERIMENTALNI DIO	26
3.1. Metodologija i tijek rada	26
3.2. Prilagodba fonta.....	27
3.3. Izrada animacija.....	29
3.3.1. Objektno orijentirano programiranje	29
3.3.1. Animacija 1 - LetterVortex	30
3.3.1. Animacija 2 - WordVortex.....	36
3.4. Dodatne mogućnosti prilagodbe i primjena kreiranih animacija	41
4. ZAKLJUČAK	45
5. LITERATURA.....	46

1. UVOD

Osnovni zadatak tipografije u klasičnom smislu je davanje vizualnog oblika sadržaju. Ona tada ima funkciju apstraktnog nositelja informacija te podliježe određenim pravilima koja služe tome da se informacije prikazane putem teksta što brže i jasnije iščitaju, pritom težeći maksimalnoj mogućoj čitljivosti i čitkosti. Estetska funkcija tipografije tada pada u drugi plan.

Nasuprot tome, u području eksperimentalne tipografije njena se klasična uloga preispituje te nastaju djela koja poruke prenose prvenstveno vizualnim dojmom. I ovdje je riječ o informacijama prikazanim putem teksta, no njihova je čitkost i čitljivost svjesno narušavana kako bi se naglasak stavio na estetiku, a tipografija komunicirala s čitateljem na posve drugačiji način od klasičnog. Riječi i rečenice ne moraju nužno biti pročitane kako bi se postigao željeni dojam.

Koristeći statične medije, to je moguće postići na brojne načine, uključujući inovativne načine konstrukcije samih slovnih znakova, korištenje različitih materijala i metoda zapisa, ili neobične prijelome koji stranicu teksta pretvaraju u nečitljivo ali vizualno zanimljivo djelo.

Koristeći računalnu animaciju kao medij za stvaranje tipografskih kompozicija, autorima se otvaraju nebrojene, dotad nezamislive mogućnosti koje pruža dodatna vremenska komponenta, budući da se kompozicije mogu otkrivati i transformirati kroz vrijeme. Ipak, najveću pažnju promatrača privlači uvođenje interakcije, što omogućuje potpuno novi odnos s vizualnim elementima na ekranu, budući da iz pasivnog promatrača postaju aktivni korisnici aplikacije s kojom mogu komunicirati i mijenjati izgled kompozicije po svojoj želji.

U radu će biti prikazani neki do postojećih primjera interaktivnih animacija u kojima se tipografija koristi kao glavno izražajno sredstvo. Praktični dio rada koristit će autorske slovne znakove izrađene u sklopu završnog rada, uključujući njihovu prilagodbu za daljnju upotrebu, proces izrade interaktivne animacije pomoću programskog jezika Processing i mogućnosti njene primjene.

2. TEORIJSKI DIO

2.1. Tipografija u računalnoj animaciji

Animirana tipografija još se naziva i kinetičkom ili temporalnom tipografijom. Ona je po definiciji vještina animiranja teksta kako bi se povećao njegov utjecaj na čitatelja, odnosno gledatelja, pojačavajući emotivni i interaktivni doživljaj koji pobuđuje. Kombinacija pokreta i teksta, ponekad praćena zvukom, koristi se na određen način kako bi se publici prenijela željena ideja ili osjećaj. Tekst se na razne moguće načine predstavlja postupno kroz određeni period vremena, te postaje temporalan umjesto statičnog. Slično kako tradicionalna, statična tipografija, realizirana u medijima koji ne pružaju mogućnost promjene kroz vrijeme, proučava utjecaj koji kombinacije oblika slovnih znakova i negativnog prostora imaju na interpretaciju teksta, kinetička tipografija fokusira se na proučavanje utjecaja koji na isti doživljaj ima vrijeme.

Pokazalo se kako kinetička tipografija uvelike povećava mogućnost dodavanja snažnog emotivnog značaja i privlačnosti teksta, omogućujući mu da dobije neke od karakteristika koje se obično mogu naći u filmovima ili govoru. Njena je upotreba u medijima postala vrlo široka, uključujući televiziju, film te internetsko oglašavanje. Istraživanja u polju perceptualne psihologije koja se bave pažnjom, uspješnošću čitanja i razumijevanjem, pokazala su da se postepeno prikazivanje tekstualnog sadržaja kroz vrijeme može učinkovito koristiti kako bi se pridobila i dijelom izmanipulirala pažnja gledatelja, a u nekim slučajevima i poboljšala kvaliteta čitanja i razumijevanja sadržaja.

Osim za prenošenje tekstualnih informacija, kinetička se tipografija uspješno koristi i kao način umjetničkog izražavanja autora, gdje je čitljivost teksta negirana, a slovni se znakovi koriste poput bilo kojeg drugog likovnog elementa za izradu vizualno atraktivne animirane kompozicije. Koncepti, ideje i osjećaji mogu se jednako izražajno prenijeti vizualnim jezikom obogaćenim elementom promjene kroz vrijeme.

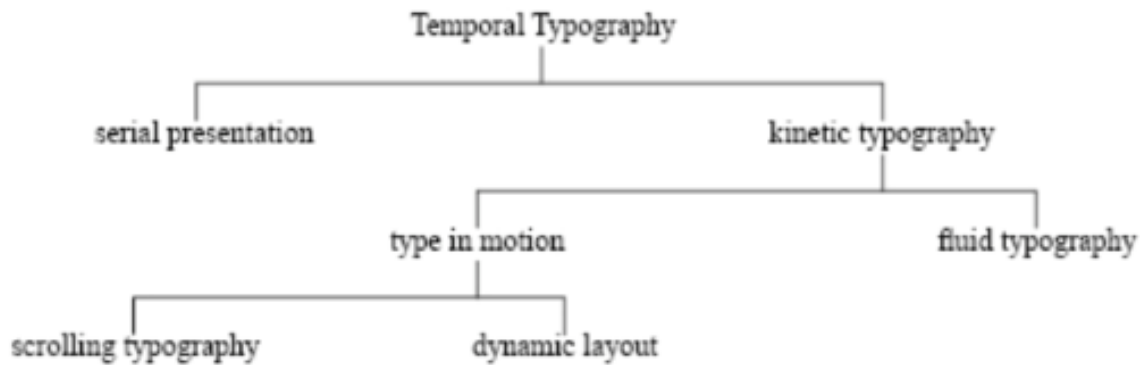
2.1.1. Izražajne mogućnosti i podjela animirane tipografije

Kao što je slučaj s bilo kojim drugim oblikom komunikacije, uspjeh u razumijevanju tekstualne poruke u sklopu animacije podrazumijeva njeno prenošenje pravovremeno, na razumljiv način, pogotovo u slučaju prenošenja određene emocije. Presentiranje teksta kroz vrijeme omogućava tekstu da dobije neke snažne izražajne mogućnosti obično rezervirane za film. Na primjer, dio teksta može biti povećan kako bi se naglasila njegova važnost, moguće mu je promijeniti boju kako bi se istakli željeni detalji, ili pozicionirati određeni element na mjesto koje promatraču jasno komunicira njegov značaj unutar hijerarhijske strukture informacija. Ovakvi načini isticanja dokazano imaju pozitivan utjecaj na brzinu i razumijevanje pročitanog sadržaja te zadržavanje pažnje. Jednako jasno mogu se prenijeti i emocije, pokretima kao što je pulsiranje dijela teksta koji je od iznimne važnosti.

U klasičnoj prirodi tipografije i vokabularu koji se koristi kako bi se opisali pojmovi uz nju vezani, pretpostavlja se da je medij u kojem je realizirana nepromijenjiv, budući da se primarno odnosi na pisanu, odnosno tiskanu riječ. Digitalni mediji su tipografiji, kao i brojnim drugim elementima, pružili pravu vremensku dimenziju u kojoj se slovni znakovi i njihove kombinacije mogu micati i transformirati.

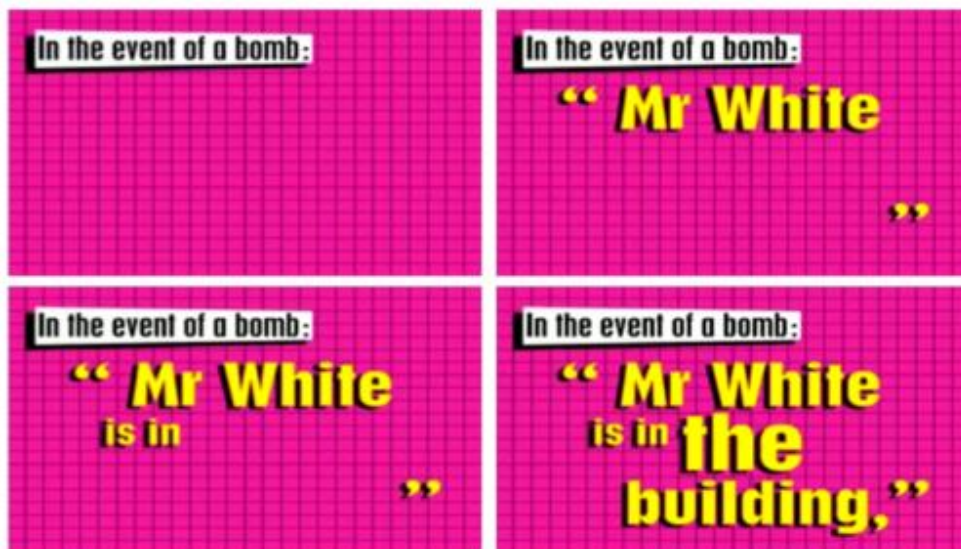
Dvije su glavne kategorije na koje se dijeli kinetička tipografija: pokretna (*motion*) i fluidna (*fluid*). Obje podrazumijevaju animaciju teksta, no razlika je bazirana na njegovom ponašanju. Kod pokretne kinetičke tipografije, elementi mijenjaju svoj položaj u odnosu na druge, dok se kod fluidne oblici slovni znakova transformiraju čak i kad tekst ne mijenja svoj položaj.

Prema dr. Barbari Brownie, koja je veliki dio svojih istraživanja provela upravo u ovom području te je autorica brojnih djela na temu tipografije, pogotovo eksperimentalne i animirane, podjela je nešto kompleksnija (slika 1.). Dr. Brownie kao najopćenitiji termin za bilo kakvu prezentaciju teksta na ekranu kroz vrijeme koristi naziv "temporalna tipografija" (*temporal typography*). Ona se dalje dijeli na serijsku prezentaciju (*serial presentation*) i kinetičku tipografiju.



Slika 1. Kategorizacija tipografije korištene u animaciji prema Barbari Brownie
<http://kinetictypography.dreshfield.com/tagged/kinetic-typography/chrono>

Serijska prezentacija odnosi se na animaciju koja koristi tipografiju, ali je ona unutar kadra posve statična. Riječi se ne transformiraju ni na koji način, a njihov položaj ostaje isti. Raspoređene su i prikazane upravo onako kako bi izgledale na otisnutoj stranici teksta, s jedinom razlikom što se ne pojavljuju sve odjednom već određenim redom, što sekvencu smješta u kategoriju animacije (slika 2.).

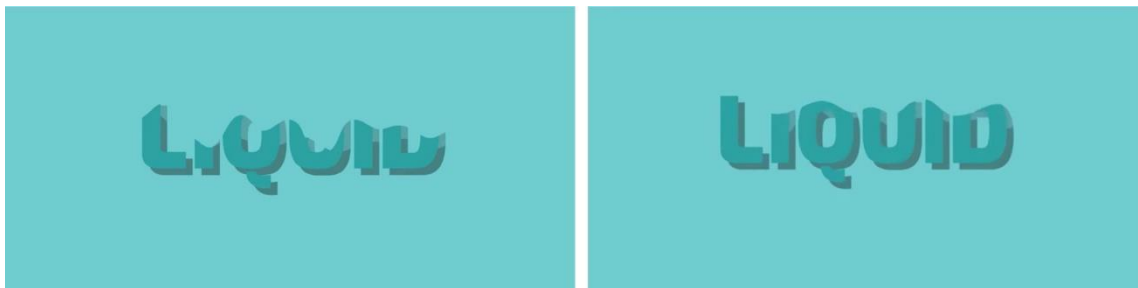


Slika 2. Serijska prezentacija
<http://kinetictypography.dreshfield.com/tagged/kinetic-typography/chrono>

Druga, najopsežnija kategorija temporalne kategorije je prema navedenoj teoriji kinetička tipografija. Ona uključuje sve animacije koje sadržavaju tekst koji nije statičan i koji se mijenja po bilo kojem parametru. Može se podijeliti prema razlici

u razini na kojoj se događa promjena u tekstu – je li ona globalna ili lokalna. Globalna promjena podrazumijeva promjenu u cjelokupnom rasporedu elemenata, odnosno slovnih znakova na ekranu, što kroz vrijeme rezultira promjenom čitave kompozicije, dok se lokalna promjena događa unutar individualnih slovnih znakova, rezultirajući transformacijom oblika. Većina kinetičke tipografije uključuje globalne promjene te time spada u kategoriju tipografije u pokretu (*type in motion*). Najjednostavnija vrsta ove promjene je kližući tekst (*scrolling type*), gdje se tekst jednostavno pomiče kao cjelina, najčešće s jedne strane ekrana na drugu, pri pokretu potpuno zadržavajući prvotni raspored svih elemenata. Kompleksnija promjena ovog tipa je dinamički raspored teksta (*dynamic layout*), u kojem se tekstualni elementi pomiču i u odnosu na ekran i jedan na drugog, mijenjajući početni raspored.

Lokalna promjena događa se u radovima koji spadaju u kategoriju fluidne tipografije. Slovni znakovi su nestalni i podložni promjenama svog oblika, potencijalno ali ne isključivo u kombinaciji s promjenom pozicije (slika 3.).



Slika 3. Fluidna tipografija

<http://www.naturebear.com/System/Landing-Pages/Animation>

Svaka od navedenih kategorija u današnjem je svijetu prisutna i u velikoj mjeri korištena, bilo da se radi o filmskim titlovima, sekvencama u televizijskim reklamama ili glazbenim spotovima. Izražajne mogućnosti tipografije u animaciji uvelike nadilaze samo prenošenje poruke čitljivim slovnim znakovima, već korištena u komercijalne, društveno korisne ili umjetničke svrhe ima mogućnost gledatelje motivirati da se angažiraju intelektualno i emocionalno.

2.1.2. Čitkost i čitljivost

Kao što je ranije spomenuto, u slučaju animacije teksta, često se pojavljuju situacije u kojima su čitkost i čitljivost drastično smanjene ili potpuno odsutne. Gotovo je uvijek to slučaj kod fluidne tipografije. Fluidnost se može opisati i kao izmjena faza čitkosti i nečitkosti, budući da slovni znakovi prilikom takvih transformacija nužno u nekim fazama preobrazbe postaju nečitki. Tijekom prijelaza iz jednog slovnog znaka u drugi, u više njih ili obrnuto, elementi moraju napustiti svoj prepoznatljivi oblik koji predstavlja određeno slovo.

Trajno je aktualno pitanje kako doživljavati tekst čiji je tipografski tretman toliko odmaknut od klasičnog da je tekst nemoguće pročitati, što se iz polja statične tipografije proteže i u animiranu. Slavni dizajner David Carson, koji je između ostalog poznat po svojim tipografskim eksperimentima, tvrdi da je čitkost samo jedna od mogućnosti kojima tipografija može komunicirati i preispituje njen značaj u odnosu na sve moguće načine izražavanja koje tipografija nudi. [1] S druge strane, brojni tekstovi koji tematiziraju kinetičku tipografiju stavljaju velik naglasak na teme koje su inače osnovna problematika klasične tipografije, pa tako grupa autora Engel, Ditterline i Yeung u svojim radovima proučavaju utjecaj koji animacija teksta ima na njegovo razumijevanje, a Heidi Specht se bazira na čitkosti, tvrdeći da su dinamični slovni znakovi koji podilaze promjenama kroz vrijeme potpuno beskorisni ukoliko se ne mogu pročitati. [2] Takve se tvrdnje kose s teorijama da je jedna od ključnih prednosti kinetičke tipografije upravo njena pojačana ekspresivnost, te da problemi kojima se tipografija uobičajeno bavi ne bi trebali biti glavna briga autora animacija.

Prilikom pokreta i transformacija, slovni znakovi prolaze kroz faze s raznim stupnjevima čitkosti, od potpuno jasnog obrisa slova do apstraktnog ili čak prepoznatljivog figurativnog oblika. Može se reći da je problematika čitkosti aktualna samo u fazama gdje se iz oblika još uvijek može do neke granice iščitati određeni slovni znak. Kasnije, kad se on transformira u nešto posve drugo i

postane slika, koncept čitkosti uopće ne može biti primijenjen te se postavlja pitanje mogu li se nastali oblici i dalje smatrati tekstem. [2]

Osim za fluidnu tipografiju, slično se može reći i za animacije koje spadaju u ostale kategorije. Svaka tipografska kompozicija može biti iščitana ili kao tekst, ili kao slika, a u slučaju da se njen izgled mijenja kroz vrijeme, i doživljaj promatrača se može mijenjati iz trenutka u trenutak.

2.1.3. Razvoj animirane tipografije kroz povijest

Osnovna namjena pisma, prijenos poruke, nije se mijenjala tisućama godina. Ipak, element koji neprestano evoluirao je medij kojim je pisana komunikacija prezentirana. Svaki od medija donio je nove mogućnosti izražavanja, a jedna od glavnih novosti koje je pružio razvoj računalne tehnologije je i mogućnost korištenja tipografije u animaciji.

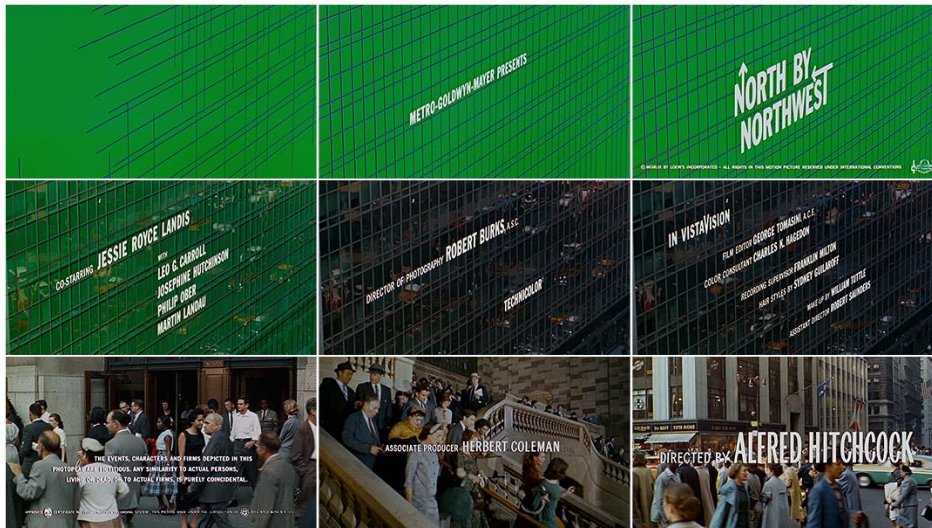
No, prve kombinacije teksta i pokreta nastale su i mnogo prije korištenja računala, zahvaljujući pojavi filma i grafičke animacije. Neki od prvih primjera animiranih slovni znakova pojavili su se još 1899. u oglasivačkim radovima Georgea Meliesa. Rani igrani filmovi sadržavali su kadrove s tekstem koji bi se mogli kategorizirati kao primjeri temporalne tipografije, točnije serijske prezentacije teksta. Tekst je bio prezentiran dio po dio i izmjenjivan korištenjem kinematičkih tranzicija (slika 4.).



Slika 4. Serijska prezentacija teksta, titlovi filma *The Maltese Falcon* iz 1941.

<http://www.artofthetitle.com/title/the-maltese-falcon/>

Tek u 60-im godinama 20. stoljeća počele su se izrađivati najavne špice filmova s tipografijom koja se može smatrati zaista kinetičkom. Prva dva filma čije su špice koristile nove tehnike i postavile buduće standarde su *North by Northwest* iz 1959. i *Psycho* iz 1960. redatelja Alfreda Hitchcocka. Autor obje sekvence je dizajner Saul Bass. Sadrže tekst koji dolazi izvan ekrana, mijenja poziciju i perspektivu u kojoj je prikazan i kasnije se stapa sa scenom filma (slika 5.).



Slika 5. Najavna špica filma *North by Northwest*

<http://www.artofthetitle.com/title/north-by-northwest/>

Tehnologija i razina na kojoj se animacija tipografije danas realizira uvelike je napredovala, no većina glavnih načina upotrebe ne razlikuje se previše od samih početaka, pa je tako danas ona neizostavan dio filmskih špica i reklamnih spotova. Sve je više područja u kojima kombinacija videa, teksta i zvuka može rezultirati izražajnom, pamtljivom porukom mnogo više nego upotrebom pojedinačnih elemenata. Istodobno, grafički dizajn evoluirao na način da se sve više fokusira na web i mobilne tehnologije koje omogućavaju interaktivnost, a upotreba kinetičke tipografije odličan je način da se pažnja promatrača zadrži dovoljno dugo da im se prenesu potrebne informacije.

Osim komercijalnih potreba, sve je veći broj samoiniciranih projekata i umjetničkih djela raznih autora koji su upravo u kombinaciji animacije i tipografije našli pravi način za izražavanje svojih ideja.

2.2. Vrste i način izrade računalne animacije

Tradicionalne, ručno rađene animacije nastaju na jedan od dva načina. Prvi je ručno crtanje sličica koje bi kasnije bile slagane u animaciju, dok druga koristi fizičke objekte, koji su fotografirani i nakon toga pomaknuti u sljedeći položaj, ponavljajući postupak u onoliko položaja koliko je potrebno za uvjerljivu animaciju. Takvu tehniku animacije danas nazivamo *stop motion*. U oba slučaja, zakvaljujući tromosti ljudskog oka, brza izmjena takvih sličica promatraču daje iluziju pokreta.

Kako bi se dobila iluzija neprekinutog pokreta, sličice se na ekranu moraju izmjenjivati brzinom od minimalno 12 sličica po sekundi (FPS – *frames per second*), dok je optimalno imati 24 ili 30. Ispod 12 FPS većina promatrača može zamijetiti određene trzaje prilikom izmjene sličica, što oku smeta kako bi nesmetano interpretiralo pokret. Tradicionalna ručno crtana animacija najčešće koristi brzinu od 15 FPS kako bi se smanjio broj potrebnih crteža, dok kod računalne postoji zahtjev za bržom izmjenom kako bi animirane scene što više dobile na realizmu.

Računalna animacija je velikim dijelom tehnološki napredniji sljedbenik istog postupka. Također se može služiti stvarnim trodimenzionalnim modelima ili dvodimenzionalnim ilustracijama, ovisno o vrsti animacije. Računalne animacije pružaju veću kontrolu nad događajima u sceni nego što je slučaj s klasičnom animacijom u nogo kraćem roku. Korištene npr. u filmovima daleko umanjuju cijenu snimanja određene scene, budući da je brojne elemente moguće naknadno generirati u računalnom programu umjesto angažiranja velikog broja statista ili dodatnog ulaganja u scenografiju.

Jedan od načina na koje je moguće proizvesti animaciju pomoću računala je moguće proizvesti i pomoću računala koristeći uređaje za iscrtavanje (*rendering machine*), koji sami iscrtavaju uzastopne sličice ukoliko se neki aspekt scene mijenja kroz vrijeme. To mogu biti jednostavni pokreti poput postpunog pomicanja kamere ili relativno gibanje objekata na sceni. Ovaj postupak ima određene sličnosti sa korištenjem fizičkih objekata u tradicionalnoj animaciji. Sofisticiranije

animacije i programi imaju mogućnost pomicanja kamere i objekata na kompleksnije načine koji uključuju zakrivljene staze u trodimenzionalnom prostoru, a koriste i zakone fizike kojima utvrđuju ponašanje elemenata. Često služe kao vizualizacije ovisnosti kompleksnih sustava o vremenu.

2.2.1. Kontrola pokreta

Značajan dio izrade animacije je kontrola pokreta. Sustavi korišteni na početku nisu bili dovoljno snažni kako bi mogli dozvoliti pregled animacije i interaktivne kontrole pokreta, a zahvaljujući neraširenosti tehnologije većina ljudi koja se na samom početku razvoja područja bavila računalnom animacijom nisu bili umjetnici ni dizajneri, već računalni znanstvenici. Stoga su se razvili sustavi za skriptiranje. Oni su korišteni kao računalni programski jezici visoke razine, gdje su autori animacije pisali skripte, odnosno programe, kojima bi kontrolirali tijek animacije. Takvi su programski jezici omogućavali definiciju kompleksnih tipova podataka, stvarajući tako objekte sa svojim vlastitim pravilima animacije.

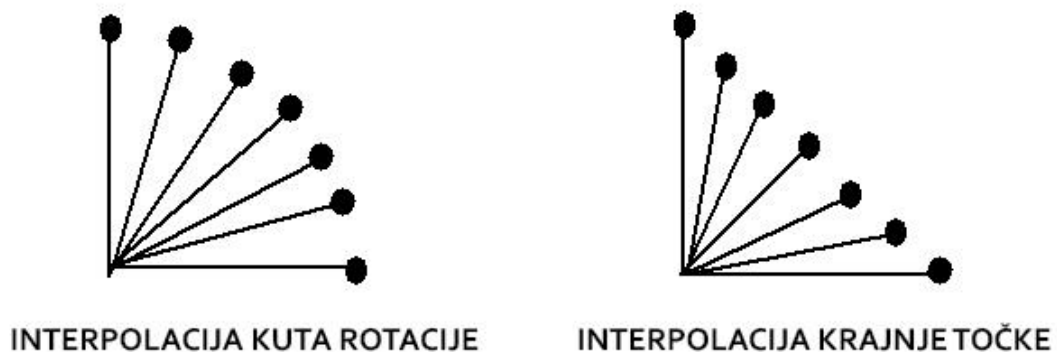
Kasniji, napredniji sustavi omogućili su različite vrste kontrole pokreta. Unatoč tome što ne postoji jednoznačna podjela vrsta računalne animacije, jedan od načina na koji se može klasificirati je prema razini apstrakcije u kontroli pokreta. Ovakva podjela stvara tri kategorije: sustave visoke, srednje i niske razine. Sustav niske razine zahtijeva od autora da precizno odredi svaki detalj pokreta, dok sustav visoke razine omogućava veću slobodu, odnosno korištenje apstraktnijih, općenitijih metoda. Što je objekt kompleksniji, to će imati više parametara pokreta koje bi autor trebao podesiti. Radi li se o grupi jednakih ili sličnih objekata, potrebno je izbjeći ponavljanje postupka za svaki pojedinačno, kako prilikom izrade animacije, tako i prilikom njenog izvođenja. Kako bi se to postiglo, potrebna je hijerarhijska kontrola, tako da su sustavi visoke razine kombinirani i preračunati u niže prilikom izvođenja animacije.

Osim korištenjem programskih jezika, ovaj je proces moguće izvesti i u računalnim programima za izradu animacija koji ne zahtijevaju pisanje koda, već se željeni efekti dobivaju upotrebom grafičkog sučelja.

2.2.2. Definiranje ključnih okvira - *Keyframing*

Uobičajena tehnika za izradu animacije s niskom razinom kontrole pokreta je *keyframing*, iscrtavanje ključnih okvira, odnosno sličica. Ova je tehnika razvijena za vrijeme klasične, ručne animacije i usavršena u studijima velikana kao što je Walt Disney.

Vodeći animatori nacrtali bi nekoliko ključnih sličica u kojima se definira izgled filma, a nakon toga bi pomoćni animatori iscrtavali sličice koje dolaze između njih i upotpunjuju sekvencu pokreta. Taj se proces naziva *inbetweening* ili *tweening*. Prilikom rada na računalu, autor također definira ključne sličice, a računalni program provodi *tweening*, odnosno popunjava međuprostor crtajući onoliko sličica koliko je potrebno da se dobije što prirodniji pokret, koristeći interpolaciju definiranih sličica. Kod interpolacije, moguće je modificirati brojne parametre, koje treba pažljivo prilagoditi ukoliko se teži realnom pokretu. Na primjer, prilikom rotacije linije, ne bi trebalo interpolirati poziciju njene završne točke unutar koordinatnog sustava, već kut koji zatvara u odnosu na ravninu početne točke, jer se u tom slučaju dobije pravo ponašanje linije (slika 6.).



Slika 6. Pravilna i nepravilna interpolacija položaja linije

Ovakva se vrsta animacije naziva animacijom uz pomoć računala (*computer-assisted animation*), budući da funkcionira na principu tradicionalne animacije, ponekad čak i koristeći digitalizirane rukom izrađene crteže, služeći se računalom kako bi se olakšao i ubrzao proces. Gotovo uvijek se koristi za dvodimenzionalnu animaciju, za razliku od računalno generirane animacije (*computer-generated*

animation). Objekti u ovom slučaju nemaju mogućnost interakcije niti im je moguće dodijeliti pravila ponašanja, već imaju eksplicitno definiran pokret. Neki od računalnih programa u kojima se ovakve animacije najčešće izrađuju su Adobe After Effects, Adobe Flash Professional, Apple Motion i FlipBook. Na tržištu postoje i brojni drugi, a zajedničko im je grafičko sučelje kojim se kontrolira ponašanje objekata na sceni, uključujući i vremensku crtu (*timeline*) na kojoj je moguće označiti ključne točke i pratiti tijek animacije (slika 7.). Većinom se radi o vrlo kompleksnim programima u kojima je izrada animacije samo jedna od mogućnosti, a animacije stvorene pomoću njih mogu biti vrhunske.



Slika 7. Sučelje programa Adobe After Effects

<https://jmalcolm2010.wordpress.com/kerry/motion-graphics/task-1-3/>

2.2.3. Vrste animiranih sustava

Prema sustavima kojima se stvaraju objekti i definira njihovo ponašanje, računalnu animaciju moguće je podijeliti na sljedeće vrste [3]: sustave za skriptiranje, proceduralnu, reprezentacijsku, stohastičku i bihevioralnu animaciju.

Sustavi za skriptiranje najstarija su vrsta sustava za kontrolu pokreta. Kako bi se njima mogli služiti, korisnici moraju naučiti programski jezik i njime kodirati

ponašanje objekata. Jedan takav sustav je ASAS (Actor Script Animation Language), koji predstavlja koncept glumca (*actor*), odnosno složenog objekta koji posjeduje svoja pravila za animiranje. Na primjer, u slučaju animacije automobila, kotači će se okretati prema definiranim pravilima unutar vlastitog koordinatnog sustava. Glumci imaju mogućnost komunicirati s ostalim glumcima slanjem poruka kako bi međusobno sinkronizirali pokrete. Slično ponašanje elemenata javlja se i u objektno orijentiranim programskim jezicima (OOP), kao što su C++, C# i Java.

Proceduralne animacije koriste procedure koje definiraju izvršenje određenog pokreta kroz vrijeme. Mogu se podvrgavati zakonima fizike ili metodama koje definira animator. Primjer ovakve animacije je pokret koji je posljedica nekog događaja, tzv. sekundarni pokret, kao što je bacanje lopte, koja udari u neki drugi objekt i uzrokuje njegov pomak.

Reprezentacijske animacije dozvoljavaju promjenu oblika objekta prilikom animiranja, a imaju tri potkategorije. Prva je animacija zglobnih elemenata, odnosno složenih objekata sastavljenih od povezanih krutih dijelova. Druga je animacija mekih objekata, primjerice kože napete na mišićima. Treća se naziva *morphing*, a odnosi se na preobrazbu jednog objekta u drugi, a može se dogoditi u dvodimenzionalnom ili trodimenzionalnom prostoru.

Stohastička animacija koristi stohastičke procese za kontrolu grupe objekata, kao što su čestični sustavi. Primjeri su vatra, vatromet ili slapovi.

Bihevioralna animacija objektima dodjeljuje pravila ponašanja kojima definira kako će reagirati na okolinu. Primjeri su jata ptica ili riba, gdje se svaka jedinka ponaša prema pravilima koje je definirao autor.

Ponašanje svakog računalno generiranog ili animiranog objekta određeni su programskim naredbama. Za svaki pokret potrebni su brojni proračuni i algoritmi, no nije u svakom slučaju nužno da autori animacija osobno pišu kodove, odnosno da uopće imaju znanje nekog programskog jezika, već se služe grafičkim sučeljem kako bi stvorili animaciju, koja se unutar programa interpretira u naredbe koje računalo razumije. Programiranje vlastite animacije, s druge strane, pruža autorima još potpuniju kontrolu nad izgledom i ponašanjem objekata. Izbor

programskih jezika vrlo je velik – to mogu biti neki od poznatih jezika široke namjene, kao što je Java, JavaScript, Python ili C odnosno njegovi sljedbenici C++ i C#, ili pak slabije raširenih jezika razvijenih za specifične namjene. Jedan od takvih jezika je i Processing, koji se pokazao kao pravi izbor za izradu ovog rada.

2.2.4. Programski jezik Processing

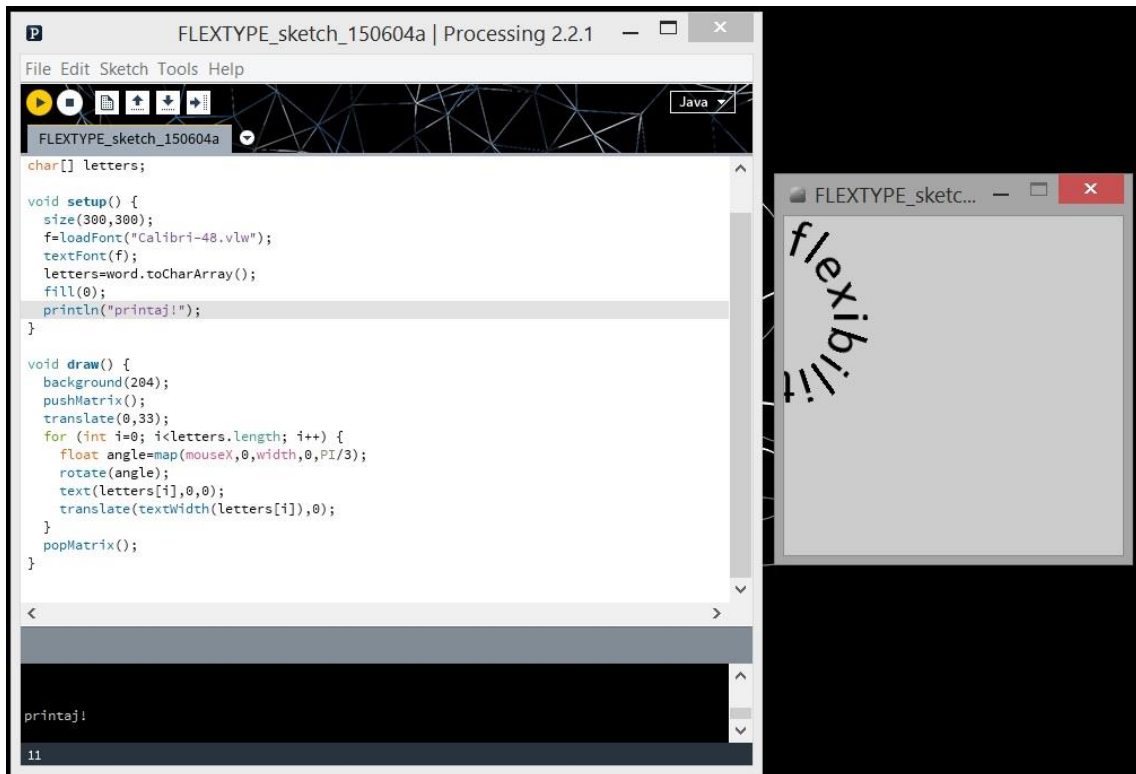
Processing je *open source* programski jezik i integrirano razvojno okruženje (IDE - *integrated development environment*) posebno osmišljen u svrhu realizacije radova u području računalne grafike i umjetnosti novih medija. Dodatna mu je svrha služiti zajednicama vizualnih umjetnika i dizajnera kao pomoć u podučavanju osnova računalnog programiranja u vizualnom kontekstu. Osmislili su ga Casey Reas i Benjamin Fry sa instituta MIT 2001. godine, od kada je konstantno nadograđivan i usavršavan. Temeljen je na programskom jeziku Java, no koristi jednostavniju sintaksu i grafički model programiranja.

Processing povezuje koncepte softvera s principima vizualnog oblikovanja, pokreta i interakcije. Integrira programski jezik, razvojno okruženje i metodologiju učenja u jedinstven sustav razumljiv početnicima. Služi i kao digitalna bilježnica za skice (*sketchbook*) i kao produkcijski alat za prototipe te konačne radove. Osmišljen je specifično kako bi se njime generirali i modificirali vizualni materijali, pazeći na ravnotežu jasnoće prezentacije i naprednih mogućnosti. Zahvaljujući tome, početnici mogu pisati kratke funkcionalne programe nakon samo nekoliko minuta uputa, dok iskusni korisnici imaju široke mogućnosti kreiranja vlastitih dodataka i nadograđivanja funkcija.

Omogućuje rad sa brojnim pojmovima i tehnikama računalne grafike i dizajna interakcija, uključujući vektorsku i rastersku grafiku, obradu slika, modele boja, reakciju na ulazne podatke iz miša i tipkovnice, mrežnu komunikaciju i objektno orijentirano programiranje. Proširivanje putem dodatnih biblioteka (*libraries*), kojih trenutno na Internetu postoji preko 100, nadograđuje Processing mogućnostima

kao što su generiranje zvuka, primanje i slanje podataka u raznim formatima, uvoz i izvoz raznih 2D i 3D datoteka.

Sučelje programa, odnosno njegovo razvojno okruženje (PDE – *Processing Development Environment*) jednostavno je i intuitivno za snalaženje (slika 8.). Na samom vrhu radnog prozora nalazi se izbornik i nekoliko kontrola za pokretanje, zaustavljanje, uvoz te izvoz programa, koji se u Processingu naziva skicom (*sketch*). Glavni dio prozora zauzima editor teksta, u kojem se upisuje kod. Donji dio prozora zauzima konzola, u kojoj se ispisuju poruke zadane naredbama, obično kontrolne, ili poruke o grešci u kodu. Drugi je prozor u kojem se izvršava program, dimenzija definiranih naredbom `size()`;, a otvara se pritiskom na Run.



Slika 8. Processing razvojno okruženje

Na prethodnoj slici moguće je uočiti da je sučelje postavljeno u mod Java. U padajućem je izborniku moguće odabrati neku drugu postavku, kao što je JavaScript ili Python, budući da je u programsko okruženje integrirano više programskih jezika. Za programiranje skica koje je potrebno postaviti na Internet, preporuča se koristiti JavaScript, uzevši u obzir da današnji internetski

preglednici sve manje podržavaju Javu. Za prevođenje postojećeg Processing Java koda u JavaScript postoji i tzv. prevoditelj Processing.js.

2.3. Interaktivnost i korisnički generiran sadržaj

Neovisno o tome radi li se o tipografiji ili drugim vizualnim elementima, animacije imaju sposobnost brže zadobiti te duže zadržati pažnju promatrača i obično ih zainteresirati na drugačiji način nego što to može statični medij. No, pogotovo u današnje vrijeme kad su ljudi okruženi i često zasićeni informacijama u svim mogućim oblicima, svaka se poruka bori za nekoliko sekundi pažnje koju često ni ne uspije dobiti. Ne gledajući komercijalne načine upotrebe, i u umjetnosti postoji potreba da djelo, umjesto pasivnog postojanja, komunicira s publikom i tako ostavi trajniji dojam na osobu koja je bila u kontaktu s njim. Tada se govori o interaktivnosti, koja je sve raširenija u brojnim područjima, od oglašavanja do umjetničkih instalacija.

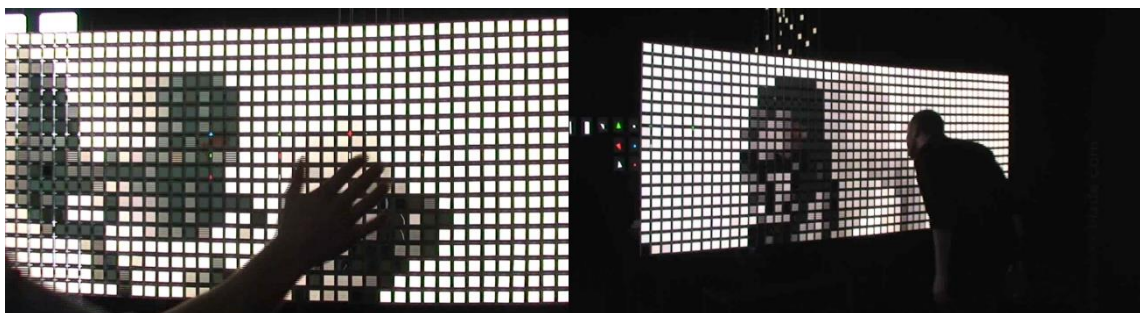
Interaktivnost je, u računalnoj terminologiji, komunikacija koja se ostvaruje između čovjeka i računalnog programa [4]. Programi koji se pokreću i izvršavaju bez direktne uključenosti korisnika nisu interaktivni, već se nazivaju pozadinskim programima. Računalne igrice, na primjer, moraju omogućiti visoku razinu interaktivnosti budući da se njihov tijek zasniva na ponašanju korisnika, no i osnovni programi s kojima se korisnici susreću svakodnevno vrlo su interaktivni – operativni sustav na računalu, aplikacije za pisanje teksta itd., samo što je njihovo ponašanje nešto ograničenije i određeno definiranim opcijama koje korisnici mogu izabrati. Na Internetu se tako simultano isprepliće nekoliko razina interaktivnosti, primjerice komunikacija s internetskim preglednikom i web stranicom. Najčešća vrsta interaktivnosti na web stranicama su hiperlinkovi, koji vode korisnike s jedne internetske lokacije na drugu. Interaktivnost se može promatrati kroz ulazne i izlazne oblike, pa su tako ulazni tipkanje raznih naredbi putem tipkovnice ili klikanje mišem, dok su izlazni prikaz utipkanog teksta ili odabranih slika na ekranu, isprintani dokumenti, video i brojni drugi.

2.3.1. Interaktivna umjetnost

Interaktivnost, dakle, ima značajnu ulogu u svakodnevnom životu svake osobe koja se služi računalom. Promatrajući segment zabave i umjetnosti, treba obratiti pažnju na umjetnost novih medija. Ona objedinjuje razne vrste projekata iz područja digitalne, računalne i virtualne umjetnosti, video igrice, računalnih animacija, 3D printanja i njima sličnih. Iako ovi pojmovi nisu strogo definirani i za mnoge je radove diskutabilno odrediti u koju kategoriju pripadaju, postoji i područje nazvano interaktivnom umjetnošću, što podrazumijeva svako umjetničko djelo koje uključuje gledatelje kako bi ostvarilo svoju svrhu. [5]

Za razliku od tradicionalnih umjetničkih oblika u kojima se jedina interakcija djela s publikom događa samo mentalno unutar svakog promatrača, u interaktivnim djelima publika sudjeluje tako da svojim ponašanjem direktno utječe na ishod te mijenja izgled djela. Načini na kojima svaka osoba može doprinijeti ograničeni su samo maštom autora i daleko nadilaze čisti psihološki doživljaj.

Veliki dio takvih djela služi se računalima i oslanja na senzore kako bi detektirali pokret, promjenu temperature, udaljenosti ili nekog drugog vanjskog faktora. Primjer je prikazan na slici 9., instalaciji *You Fade To Light* londonskog studija *rAndom International*, koja se sastoji od OLED zaslona i senzora za detekciju pokreta gledatelja, stvarajući na zaslonu zrcalnu sliku koja prati njegov položaj.



Slika 9. Interaktivna instalacija You Fade To Light

<https://vimeo.com/4282941>

Načini ulaza podataka mogu biti i jednostavniji, poput tipkovnice i miša. Na detektirane ulazne signale objekti reagiraju određenom promjenom za koju su

programirani. Može se reći da u interaktivnoj umjetnosti čovjek i stroj rade skupa kako bi svakog puta stvorili posve jedinstvenu sliku kroz doživljaj koji je poseban i drukčiji za svaku osobu. Umjetničko djelo u tom slučaju uzima u obzir publiku i preuzima ulogu responzivnog okoliša koji se prilagođava njenim željama.

Neki od alata kojima se autori služe za realizaciju ovakvih projekata su Arduino i I-CubeX kao računalno elektronički sklopovi i sučelja za izradu interaktivnih instalacija te Max/MSP, Processing i Pure Data kao programski jezici kojima se definira ponašanje objekata.

2.3.2. Interaktivna računalna animacija

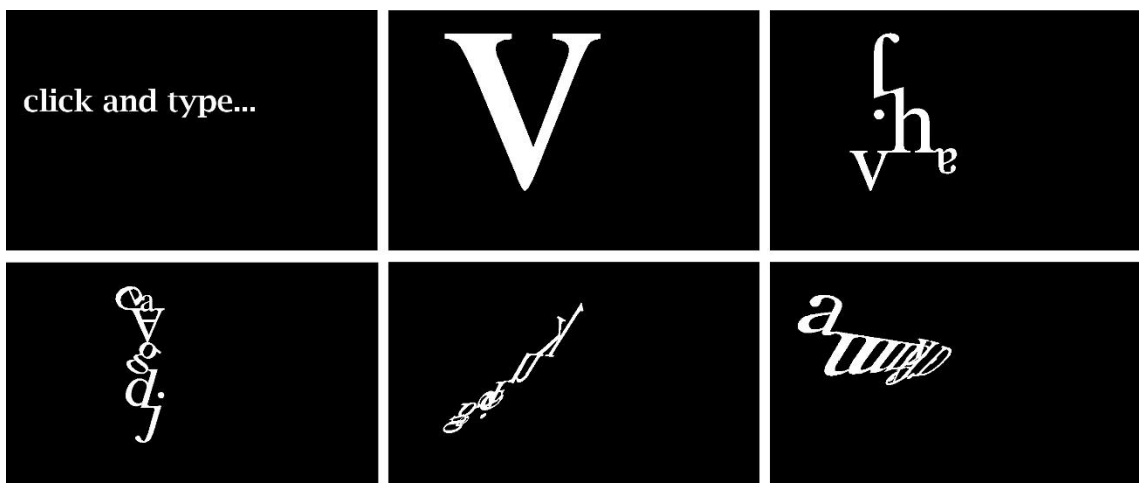
Od samog početka razvoja animacije pa sve do današnjih dana, njena generalna struktura može se definirati kao serija sličica koje su prvo izrađene, a onda korištene u svrhu stvaranja iluzije pokreta, umjesto da je pokret stvoren istovremeno sa sličicama i snimljen. Računalna je animacija omogućila nelinearnost, a autorima pružila dotad nezamislivu slobodu. Pojavio se pojam animacije u realnom vremenu (*realtime animation*), u kojem računalo izvršava programske naredbe i iscrtava animaciju jednakom brzinom kojom se ona prikazuje na ekranu. To znači da autor animacije može organizirati, programirati i unositi potrebne promjene u realnom vremenu i odmah vidjeti rezultate.

No, pojavila se i promjena u odnosu animacije i gledatelja. U tradicionalnom smislu, animacija je gledatelju predstavljena na isti način kao što se to događa sa slikom ili fotografijom, s razlikom u vremenskom trajanju, no on je i dalje pasivni promatrač koji pogledano može samo interpretirati. Iskustvo gledanja takve animacije je kontrolirano i u potpunosti predefiniрано onime što je autor želio reći. Često posjeduje narativ koji je predstavljen linearno i obuhvaća točno određeno vrijeme, završava uvijek na isti način, s jednom konačnom scenom nakon koje je prezentacija gledatelju završena. S druge strane, uvođenje interaktivnosti u animaciju pretvara ju u nelinearno djelo koje nije u potpunosti kontrolirano željama animatora, već gledatelja pretvara u koautora djela, budući da dolazi u poziciju kontrolirati određene parametre i time mijenjati izgled, odnosno ishod animacije. Može se reći da interakcija u animaciji omogućuje

dogadaje koji predstavljaju vezu između narativa i korisnika, koji je u mogućnosti prilagođavati narativ svojim željama i mijenjati ga svojim postupcima. Interaktivne animacije koriste se u brojne svrhe, od zabave i umjetnosti do edukacije. Klik mišem ili pritisak na tipku može biti od velikog značaja i potpuno promijenti korisničko iskustvo. Interaktivne animacije se često toliko odmiču od svoje početne definicije i postaju samostalne aplikacije, odnosno programi koji postoje s određenom svrhom.

2.3.3. Upotreba tipografije u interaktivnoj računalnoj animaciji

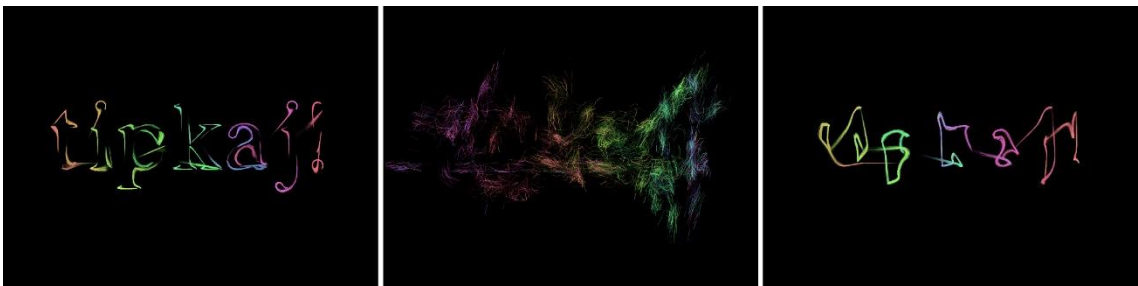
Tipografija se u interaktivnim animacijama može koristiti kao vrlo izražajan način za obogaćivanje korisničkog iskustva i pogodan materijal za eksperimente. Za razliku od rada u klasičnim programima poput MS Worda, unos teksta u tim slučajevima postaje zanimljivo iskustvo s neočekivanim rezultatima, gdje se njegova čitljivost često posve zanemaruje, a slovni znakovi prelaze u likovne elemente koji stvaraju neponovljive kompozicije svojstvene korisniku. Primjer takve aplikacije je Lettree (slika 10.). Korisnici imaju mogućnost unositi svoj tekst, čiji se slovni znakovi spajaju po određenim pravilima, nepredvidljivo i nasumično. Pritiskom na tipku Enter, kompozicija se počinje rotirati u trodimenzionalnom prostoru, pružajući nove, neobične poglede na upisani tekst.



Slika 10. Interaktivna tipografska animacija Lettree

<http://www.caligraft.com/exhibition/lettree>

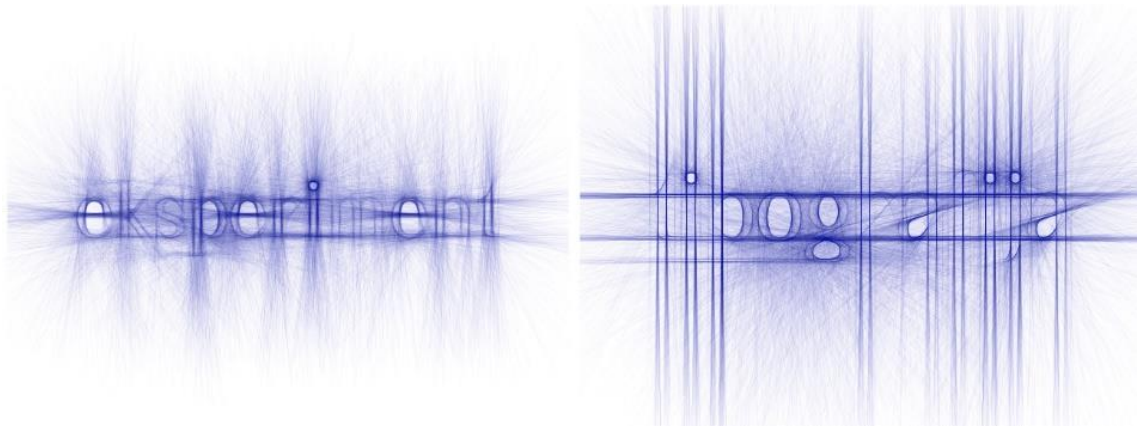
Ono što ovakve interaktivne aplikacije čini zanimljivima publici, čak i onom dijelu koji nije nužno zainteresiran za tipografiju, spoj je mogućnosti osobnog izražavanja i nasumičnosti programiranog dijela aplikacije. Čak i kad se više puta upiše ista riječ, konačan rezultat je različit, što publiku tjera na daljnje korištenje. U mnogo se slučajeva slovni znakovi koriste zbog svoje zanimljive forme, koja služi kao dobra podloga za razne eksperimente bazirane na geometriji. Dvije takve aplikacije su Hilos (slika 11.) i Sketching (slika 12.). Hilos omogućava korisniku da utipka tekst po želji, a nakon što stisne Enter, ispisuje ga na zaslonu u formi vijugavih obrisa. Boje linija nasumično su odabrane prilikom svakog pokretanja aplikacije, a njihov oblik korisnici kontroliraju mišem. Ovisno o položaju miša, linije su čvrsto zategnute oko obrisa slova ili lete po ekranu s velikom slobodom, postaju zakrivljene i fluidne ili se pak pretvaraju u kratke ravne crte.



Slika 11. Interaktivna tipografska animacija Hilos

<http://www.caligraft.com/exhibition/hilos>

Sketching također ispisuje riječ koju je korisnik utipkao tako da ju omeđuje ravnim tangentnim linijama koje konstruiraju obrise slovni znakova. Linije mijenjaju svoju duljinu i nagib ovisno o poziciji posljednjeg klika mišem.



Slika 12. Interaktivna tipografska animacija *Sketching*
<http://www.caligraft.com/exhibition/sketching>

U prethodnim primjerima korišteni su gotovi, standardni fontovi, a animacija se temeljila na nekonvencionalnom slaganju slovnih znakova ili drugačijem pristupu njihovim oblicima. Još jedan kreativan pristup interaktivnog animiranja tipografije podrazumijeva redefiniranje samog fonta, za koji je odličan primjer interaktivno, dinamično pismo Laika (slika 13.).



Slika 13. Dinamično pismo Laika i njegove varijacije
<http://www.laikafont.ch/>

Laika je projekt čiji su autori Michael Flückiger i Nicolas Kunz, a radi se o fontu kojeg mijenjaju i definiraju sami korisnici. Mijenjanjem parametara poput debljine linija, nagiba slova, duljine serifa i sličnih, iz jednog se predloška dobiva veliki broj različitih kombinacija. Pitanja koja postavljaju autori glase “*Zašto bi tipografija trebala biti rigidno definirana ako neće biti ispisana? U dinamičnom mediju, zašto i oblik i karakter pisma ne bi također bili shvaćeni dinamično? Zašto se i njegovi oblici ne bi mijenjali, transformirali i reagirali na okolnosti?*” [6]. Osim direktne transformacije od strane korisnika putem miša ili posebne konzole na izložbama,

potencijalna upotreba ovakvog pisma je u kombinaciji sa vizualizacijom podataka. Parametri Laike se tako mogu mijenjati zavisno o bilo kakvim ulaznim podacima, bili to meteorološki uvjeti, stanje na burzi, očitavanja s audiovizualnih senzora ili nešto posve drugo, stvarajući pismo koje u realnom vremenu reagira na svoju okolinu. (slika 14.)



Slika 14. Promjene karakteristika pisma ovisno o meteorološkim podacima iz različitih područja
<http://www.laikafont.ch/>

2.3.4. Korisnički generiran sadržaj

Interaktivne animacije objavljene na Internetu dostupne su širokoj publici te tako velik broj ljudi ima priliku isprobavati njihove mogućnosti. Budući da svaka osoba ima svoju viziju rezultata i tumačenje samog djela, koja se može uvelike razlikovati od ostalih, pogodne su za generiranje galerija korisnički generiranog sadržaja (UGC – *user generated content*).

Ovaj termin odnosi se na sav sadržaj čiji su autori korisnici određene aplikacije, servisa i sl., od komentara, objava na blogovima do medijskog sadržaja, a često se širi putem društvenih mreža. Korisnički generiran sadržaj može obuhvaćati gotovo cijelu web stranicu, kao što je slučaj kod Wikipedije, koju doslovno kreiraju korisnici i jedna je od najvećih postojećih baza ovakvo sadržaja na svijetu. U nekim slučajevima radi se o učinkovitom alatu za provođenje marketinških kampanja, gdje se za promociju proizvoda dijelom koristi sadržaj u čijem su nastajanju sudjelovali njihovi korisnici.

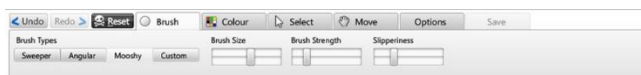
Brojne interaktivne animacije služe generiranju vizualno atraktivnog sadržaja pri čemu svaki korisnik ima priliku stvoriti nešto drugačije, pa je logično ponuditi opciju spremanja kreiranog sadržaja. Osim mogućnosti pohrane na korisnikovo

računalo, često uz aplikaciju postoji i galerija dosad spremljenih radova. Takav je pristup zanimljiv velikom dijelu korisnika jer imaju mogućnost pregledati bazu sadržaja koju su prije njih spremili ostali, još se više zainteresirati za aplikaciju vidjevši zanimljive primjere te na kraju i učiniti svoj rad dostupnim drugima.

Dodatak ovakvim galerijama može biti i omogućavanje komunikacije među korisnicima u vidu međusobnog komentiranja sadržaja, označavanja favorita i dijeljenja na društvenim mrežama.

Dva odlična primjera ove vrste su aplikacije za crtanje Viscosity i Myoats. Obje karakterizira neobičan pristup crtanju i brojne opcije za prilagodbu crtaćih alata i ostalih parametara, čime se omogućava stvaranje mnoštva različitih crteža koji su ipak prepoznatljivi za svaku od njih.

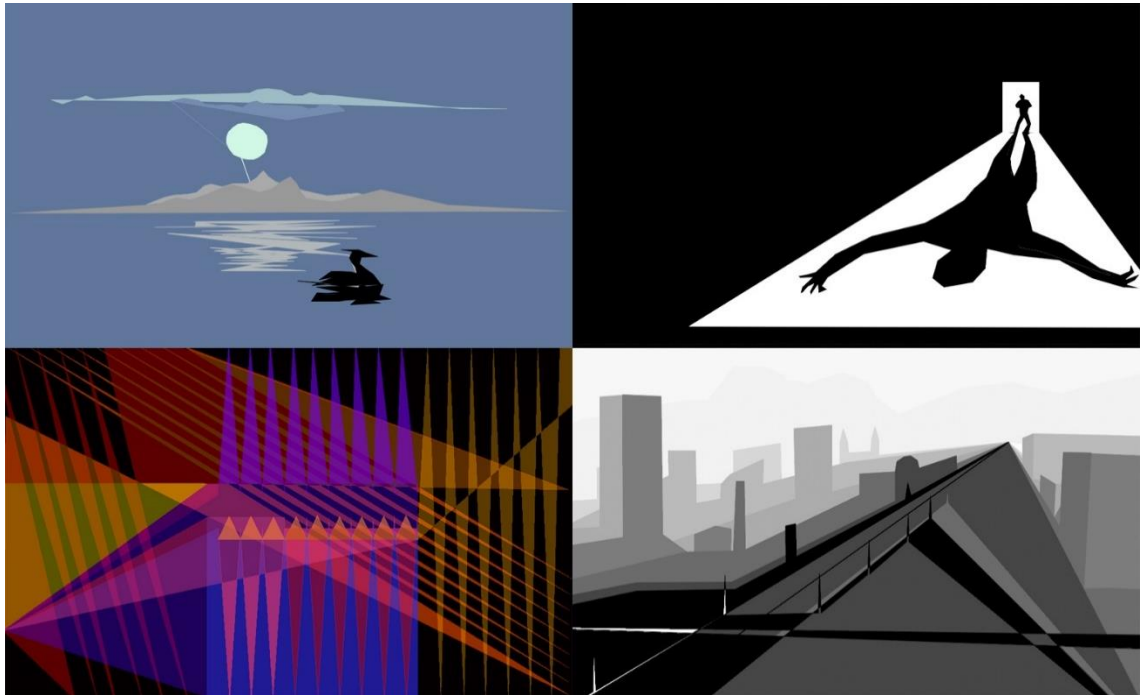
Viscosity na početnom ekranu prozora za crtanje postavlja jedan pravokutnik podijeljen na 9 horizontalnih traka u nijansama sive (slika 15.). Pomoću izbornika na vrhu, korisnici mogu mijenjati početne dimenzije, boje i poziciju pravokutnika, a nakon toga krenuti u njegovu deformaciju pomoću označavanja i micanja točaka ili kistovima koji izobličuju oblike na razne načine.



Slika 15. Viscosity: sučelje aplikacije i početno stanje

<http://windowseat.ca/viscosity/create.php>

Mogućnosti su neizmjerne i ovise samo o mašti korisnika, što je evidentno iz galerije koja trenutno broji oko 165 tisuća spremljenih radova koji variraju od potpuno apstraktnih do figurativnih, te od miimalističkih do izrazito kompleksnih (slika 16.).



Slika 16. Viscosity: četiri primjera korisničkih radova
<http://windowseat.ca/viscosity/browse.php>

Myoats je aplikacija čiji se crteži temelje na simetriji i često izgledaju poput bogato ukrašenih mandala. Korisnici mogu odabrati zrcaljenje preko jedne do šesnaest osi, koristiti desetak različitih alata za crtanje promjenjive debljine, transparentije i boje te utjecati direktno na ključne točke oblika. Radovi se mogu spremiti u galeriju koja trenutno broji više od 350 tisuća crteža, gdje postoji mogućnost komentiranja, dodavanja u favorite te dodjeljivanja bodova pomoću kojih se izdvajaju najbolji crteži unutar određenog perioda vremena (slika 17.).

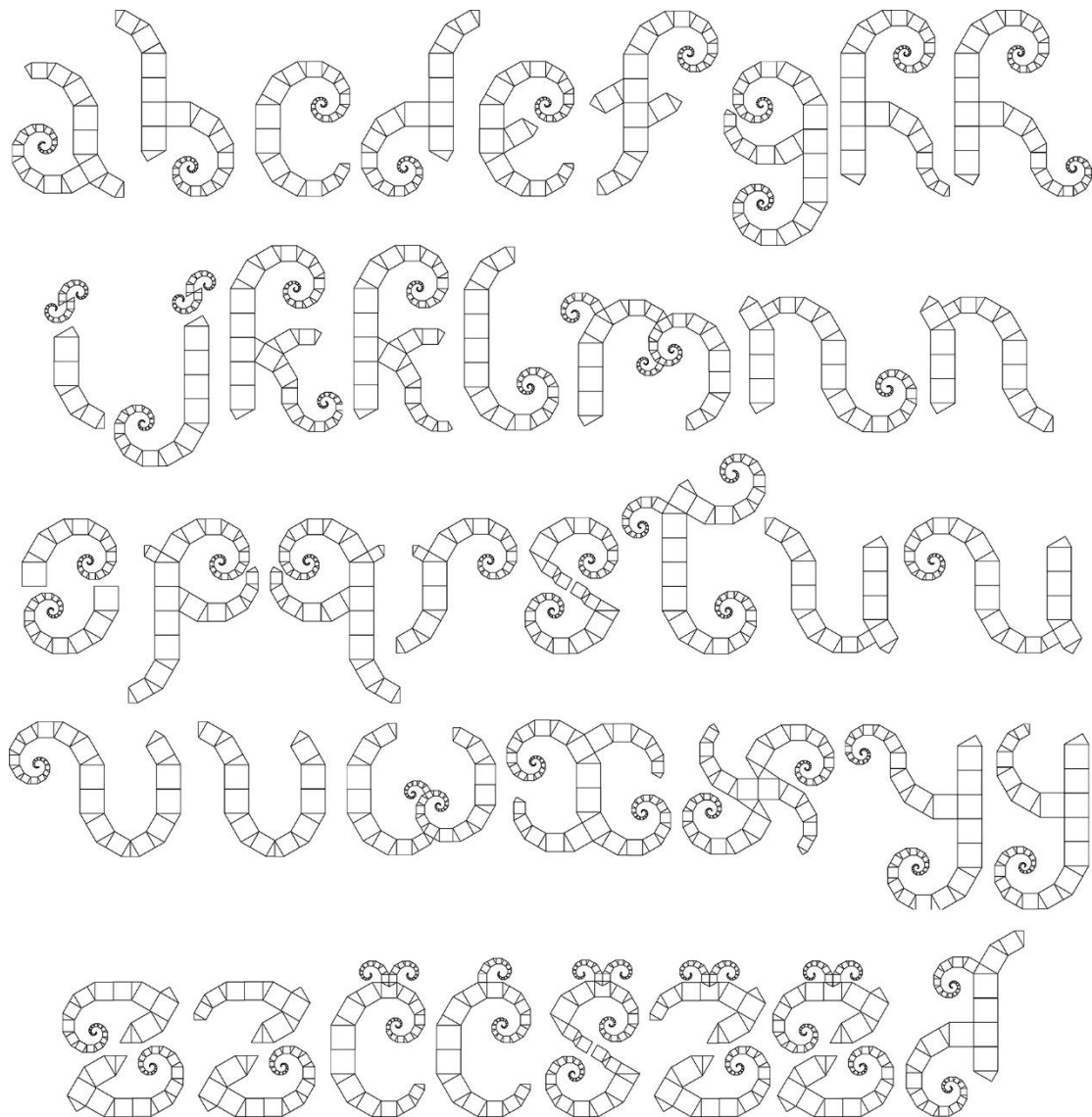


*Slika 17. Dva crteža nastala pomoću aplikacije Myoats
<http://www.myoats.com/>*

3. EKSPERIMENTALNI DIO

3.1. Metodologija i tijek rada

Za potrebu eksperimentalnog dijela rada izrađene su dvije interaktivne animacije u kojima se kompozicije slažu isključivo tipografijom. Korišteni font (slika 18.) je autorski dizajnirana tipografija koja je bila sastavni dio završnog rada [8], temeljen na ideji fraktala, a čine ga slovni znakovi sastavljeni od kvadrata i pravokutnih trokuta, prateći tako konstrukciju fraktala poznatog pod nazivom Pitagorino stablo, zahvaljujući čemu je i nazvan Typhagora.



Slika 18. Slovni znakovi fonta Typhagora korištenog za izradu animacija

Slovni znakovi fonta postojali su samo u vektorskom formatu, budući da su prethodno služili samo za ručno slaganje kompozicija. Stoga ih je trebalo malo prilagoditi i generirati funkcionalan font.

Pomoću programskog jezika Processing izrađene su dvije animacije koje korisnik može prilagođavati pomoću jednostavnog sučelja koristeći miš i tipkovnicu. Mogućnost unosa vlastitog teksta i kontrole nekoliko parametara stvara okruženje u kojem se generiraju zanimljive kompozicije. Bitno je napomenuti da način stvaranja kompozicija također prati karakter fonta, te tako zrcaljenjem, dupliciranjem i postepenom promjenom dimenzija naglašava fraktalnu prirodu konstrukcije. Uz samu izradu animacija, istražene su i mogućnosti njihove distribucije, prilagodbe i proširivanja te potencijalne upotrebe.

3.2. Prilagodba fonta

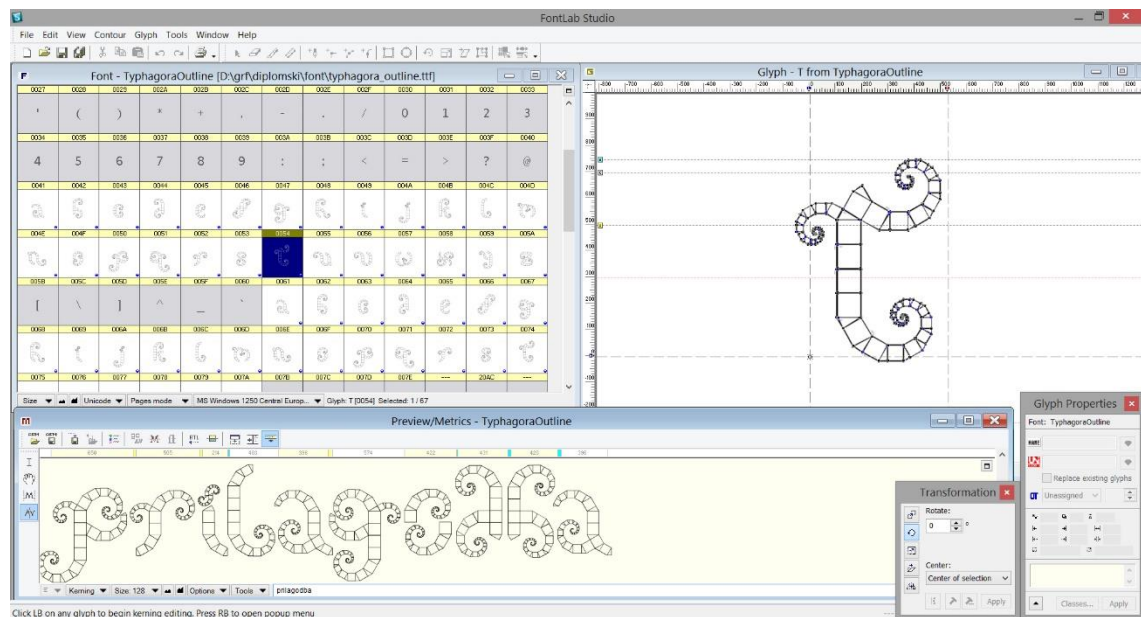
Kreirani slovni znakovi prethodno su služili samo za ručno slaganje statičnih kompozicija te nisu bili složeni u font datoteku koja se može instalirati na računalo i koristiti za unos teksta unutar bilo kojeg programa. Kako je unos teksta osnova zamišljenih animacija, pojavila se potreba za izradom klasičnog formata koji će reagirati na signale s tipkovnice.

Slovne je znakove trebalo prilagoditi na način da spojevi svih linija budu potpuno točni i osigurati da se linije i oblici ni na kojem mjestu ne preklapaju, budući da takve nesavršenosti stvaraju zbrku prilikom kopiranja u program za izradu fonta. Prilagodba slovnih znakova izvedena je u programu za vektorsku grafiku Adobe Illustrator CS6, dok je sam font kreiran u programu FontLab Studio 5.2.

Izrada fonta vrlo je zahtjevan i dugotrajan zadatak, no kako slovni znakovi nisu originalno dizajnirani u FontLab Studiu, ukratko će se navesti samo nekoliko osnovnih koraka koje je trebalo provesti kako bi se dobila potrebna datoteka spremna za daljnje korištenje.

Kao što je vidljivo na slici 19., sučelje dotičnog programa sastoji se od nekoliko ključnih prozora. U ovom slučaju, na lijevoj strani nalazi se tablica slovnih znakova (*Font window*). Klikom na neku poziciju u kodiranoj tablici, otvara se prozor u

kojem se dizajnira i modificira slovni znak koji će biti dodijeljen toj poziciji (*Glyph window*). Nakon što su svi željeni znakovi definirani, potrebno je urediti njihove međusobne prostorne odnose, odnosno dimenziju četverca za svaki pojedini znak i razmake između njih (*kerning*). U prikazanom primjeru, to se odvija u donjem dijelu sučelja programa (*Metrics window*).



Slika 19. Sučelje programa FontLab Studio

Ovako pripremljena tablica slovnih znakova dovoljno je uređena za planirani način upotrebe u animacijama te je nakon poduzetih koraka spremna za generiranje fonta u standardnom TrueType formatu.

3.3. Izrada animacija

Prvotni plan rada uključivao je izradu jedne animacije, no kasnije se istraživanjem i isprobavanjem brojnih mogućnosti došlo do više smjerova u kojima se ona može razvijati. Od njih su odabrana dva, pa su tako konačan rezultat dvije animacije koje imaju mnogo zajedničkih točaka, no po mnogočemu su i različite. Obje su napisane u programskom jeziku Processing, koriste isti font i kontroliraju se pomoću tipkovnice, kojom se unosi tekst, te miša, kojima se kontroliraju parametri i pomiče kompozicija. Detaljno će biti opisan postupak izrade prve animacije, dok će kod druge biti naglasak na razlikama.

Kako je Processing besplatan, *open-source* programski jezik kojem je jedan od ciljeva promicanje spajanja programiranja s vizualnim jezikom i kulturom, koristi ga velika zajednica ljudi koji međusobno surađuju, dijele i nadograđuju dijelove koda, pa je lako naći već gotove dijelove koda za neke osnovne funkcije, koji se prilikom pisanja novog programa mogu iskoristiti, i prilagoditi prema potrebama autora. U sklopu ovog rada korišten je samo jedan takav dio koda, a to je traka s klizačima za pomicanje (*scrollbar*), koja je prilagođena i integrirana u sučelje.

3.3.1. Objektno orijentirano programiranje

U svrhu boljeg razumijevanja procesa izrade animacija, treba spomenuti objektno orijentirano programiranje (*OOP – object oriented programming*) kao korišteni pristup prilikom definicije elemenata koji čine aplikaciju.

Ovakav pristup programiranju približava model programa načinu ljudskog razmišljanja. Umjesto klasičnog pisanja niza naredbi, temelji se na objektima, koji predstavljaju jedinice koje imaju svoje ponašanje, podatke i mogu međusobno djelovati. Objekt se može promatrati kao skup varijabli i potprograma za upravljanje tim varijablama [8].

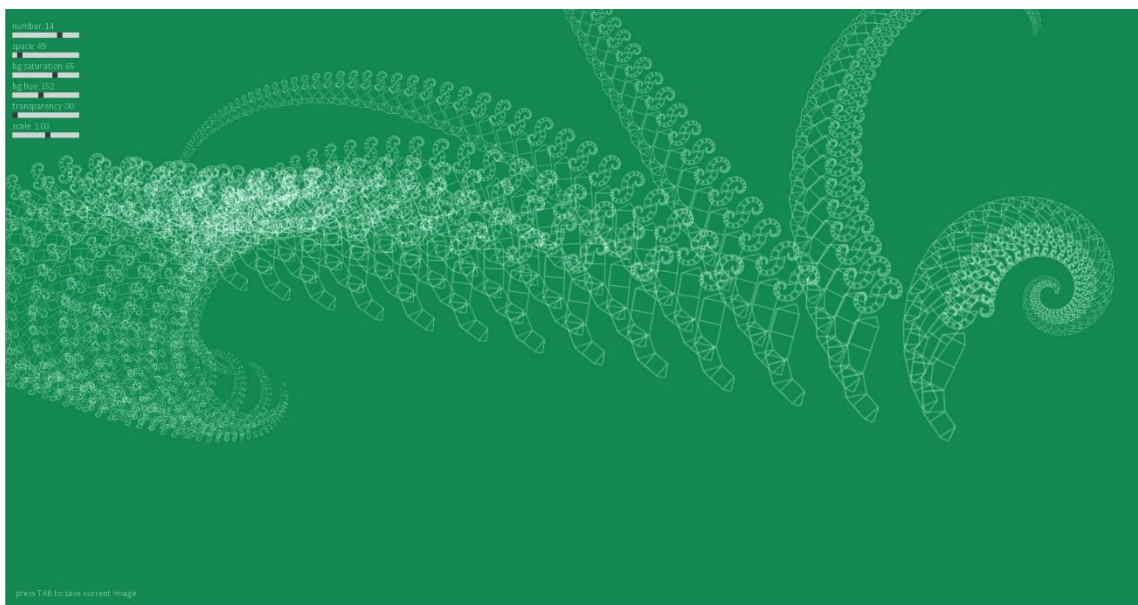
Dva su iznimno bitna pojma za razumijevanje OOP-a klasa i objekt. Klasa je način za opisivanje objekata. Nestatički, promijenjivi dijelovi klasa određuju koje će varijable i potprogrami, koji se nazivaju metodama, biti sadržani u objektima.

Prilikom izvođenja programa, objekti se stvaraju, izvršavaju svoje funkcije i uništavaju, a iz jedne klase moguće je stvoriti neograničen broj objekata. Objekt koji pripada klasi naziva se njenom instancom, a njegove varijable i metode varijablama i metodama instance. Stvarni podaci o svakom od objekata, odnosno instanci, poput dimenzija, boja ili položaja sadržani su u pojedinom objektu, ne u klasi. Prilikom pisanja programa, to omogućava brzo stvaranje novih objekata, a istodobno svaki objekt može biti po nečemu različit.

Budući da se Processing uvelike oslanja na Javu, koja je objektno orijentiran programski jezik, bilo je jednostavno i logično koristiti ovakav pristup.

3.3.2. Animacija 1 – LetterVortex

Prva se animacija (slika 20.) za izradu kompleksnih kompozicija u osnovi služi samo jednim slovnim znakom, odnosno onim koji je posljednji pritisnut na tipkovnici. Kontrolama u gornjem lijevom kutu sučelja moguće je kontrolirati broj ponavljanja osnovne kompozicije slova, razmak između njih, zasićenje i boju pozadine, transparentiju i relativnu veličinu elemenata. Kompozicija na ekranu mijenja se ovisno o poziciji miša, a slovo kojim se gradi moguće je mijenjati u bilo kojem trenutku te se animacija osvježava u realnom vremenu.



Slika 20. Izgled sučelja i funkcionalnost prve animacije

Ukoliko se ne radi o programiranju statične kompozicije, svaki program napisan u Processingu mora sadržavati dva glavna dijela, *setup()* i *draw()*.

Setup() sadrži one dijelove koda koji se ne mijenjaju, kao što je obično veličina prozora aplikacije ili model boja koji će biti aktivan. U njemu se definiraju i objekti te učitavaju fontovi koji će se koristiti. Ovaj se kod izvršava samo jednom, prilikom pokretanja programa. Zbog toga je ovaj dio dovoljan kad se radi o statičnoj slici. *Draw()* sadrži dijelove koda koji se izvršavaju prilikom svakog osvježavanja programa, što Processing predefinirano obavlja 60 puta u sekundi. To je ujedno i maksimalan broj osvježavanja koji je moguće postaviti, a može se smanjiti pozivom funkcije *frameRate()*, ovisno o tome koliko glatko animacija treba teći. Primjerice, animaciju koja bi ispisivala sat s točnošću do sekunde bilo bi dovoljno osvježavati jednom u sekundi.

Neki dijelovi koda pišu se izvan ove dvije kategorije. To se obično odnosi na one naredbe koje se izvršavaju kao posljedica nekog ulaznog događaja kao što je klik miša ili pritisak na određenu tipku na tipkovnici. Isto tako, izvan se definiraju i sve klase, objekti i varijable s kojima je potrebno raditi i u *setup()* i *draw()* dijelu programa, zbog toga što jedino na taj način postaju globalne i dostupne za poziv iz bilo kojeg dijela koda. Ovo se svojstvo naziva opsegom varijabli (*variable scope*). Ukoliko je varijabla definirana unutar nekog dijela koda, funkcionirat će samo unutar njega, dok je drugi dijelovi neće moći prepoznati.

Za izradu prve animacije, na početku je bilo potrebno definirati način izgradnje osnovne kompozicije slova, koja će se kasnije umnožavati i raspoređivati po ekranu. Prvi je uvjet osigurati da program registrira i prikazuje posljednje utipkano slovo, što omogućuju funkcije *key* i *text*. Nakon toga, treba osmisliti broj i način ponavljanja, što je izvedeno jednostavnom *for* petljom, u ovom slučaju s 50 ponavljanja. Svakim ponavljanjem slovo se pomiče, rotira i smanjuje, što stvara vrtložnu kompoziciju. Na kraju je trebalo uvesti mogućnost da korisnik kontrolira položaj elemenata putem miša. Koordinate miša prate se funkcijama *mouseX* i *mouseY*, no ne doslovno, već pomoću funkcije *atan2()*, koja izračunava kut u radijanima iz određene pozicije (u ovom slučaju položaja miša) do ishodišne točke koordinatnog sustava mjereno od pozitivne x-osi, a rezultat vraća kao

brojevenu vrijednost između π i $-\pi$. Definirana kompozicija kreirana je kao nova klasa pod nazivom LetterVortex i kasnije korištena za stvaranje objekata u animaciji.

Kako bi se kontrolirao broj, pozicija i ostali parametri objekata, bilo je potrebno izraditi jednostavno sučelje putem kojih ih korisnik može lako prilagoditi. To je riješeno pomoću traka s klizačima, prikazanim na slici 21.

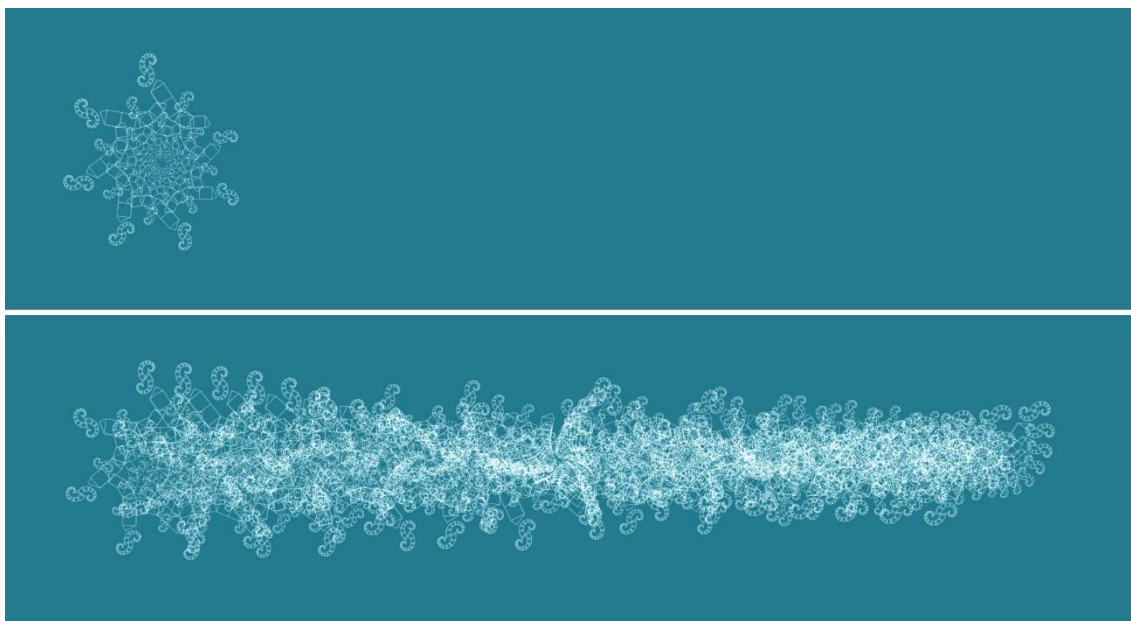


Slika 21. Klizači za kontrolu parametara animacije

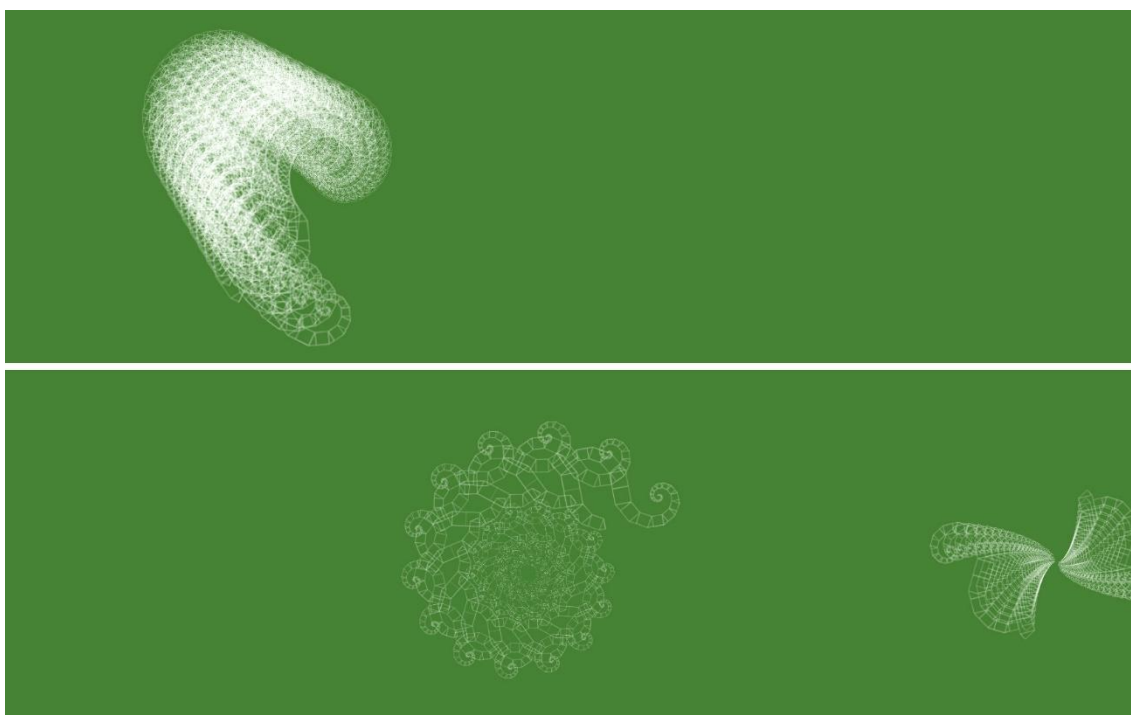
Prvim se klizačem određuje broj stvorenih objekata klase LetterVortex. Moguće ih je imati najmanje 1 i najviše 20, a stvaraju se izvršavanjem *for* petlje s onoliko ponavljanja koliko je određeno klizačem. Drugi određuje razmak između objekata, a u kombinaciji sa zadnjim klizačem, koji im određuje relativnu veličinu, mogu se rasporediti na razne načine. Treći i četvrti klizač mijenjaju zasićenje i ton boje pozadine, a peti transparentiju objekata.

Klizači su definirani klasom Scrollbar, predefiniranim elementom sučelja, koja je zadržala svoju osnovnu funkcionalnost, a izgledom je prilagođena okruženju.

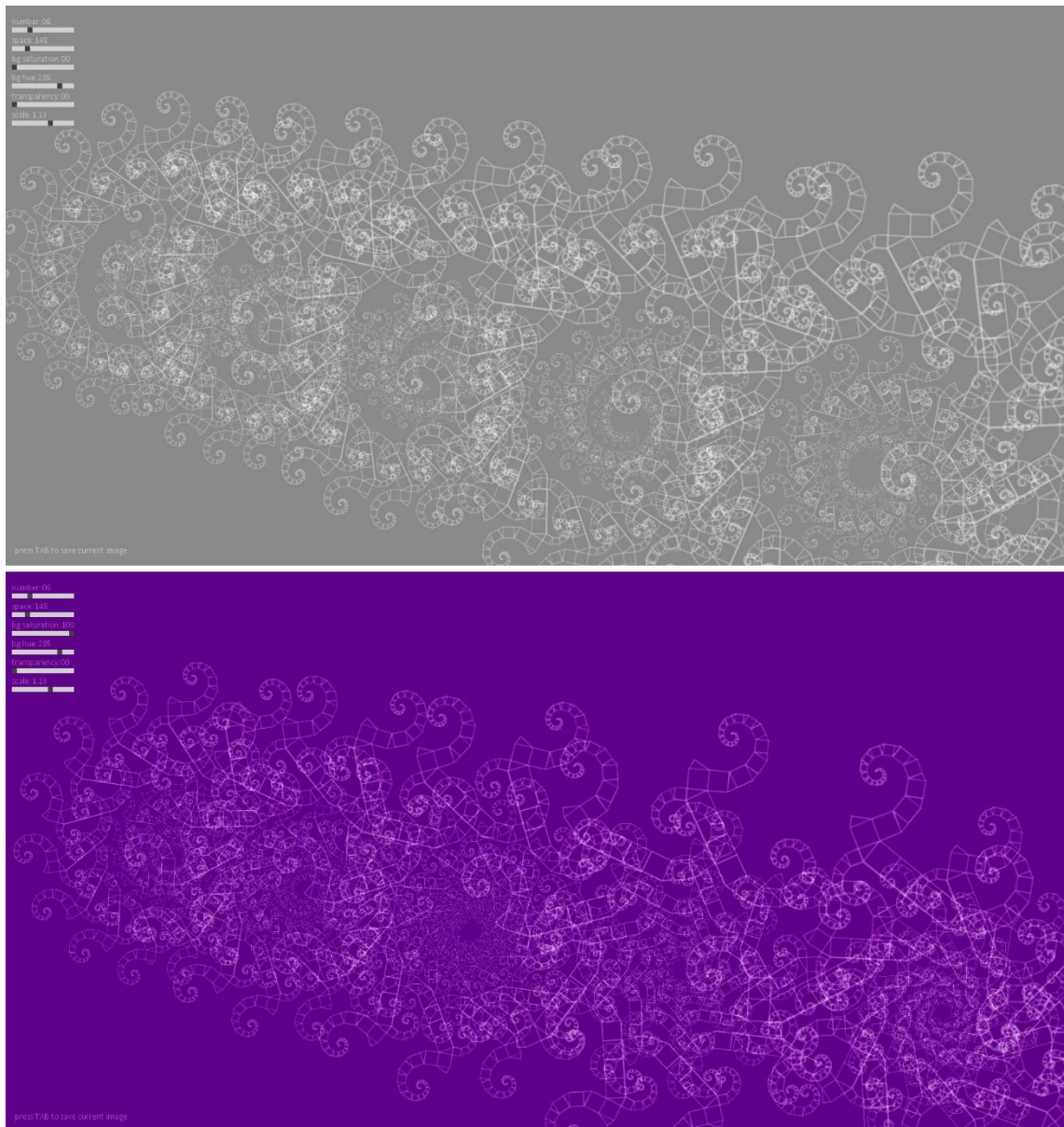
Integrirana je i mogućnost spremanja trenutne slike ekrana (*screenshot*), pa se pritiskom na tipku Tab slika sprema u lokalni folder aplikacije. Slike 22. do 26. prikazuju izgled animacije u krajnjim točkama pojedinog klizača.



*Slika 22. Slučaj sa 1 stvorenim objektom (gore) te sa maksimalnih 20 (dolje).
Relativna veličina je 1, a faktor udaljenosti 63.*

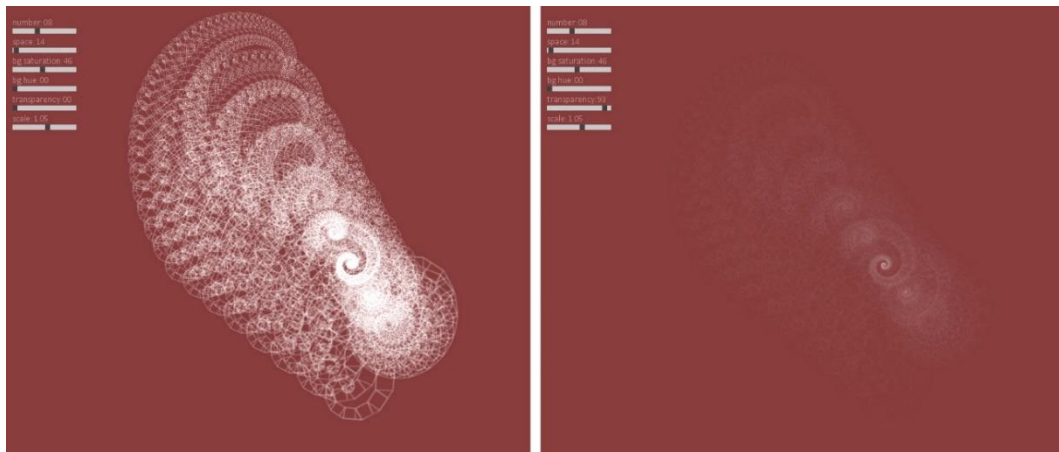


Slika 23. Razlika između minimalne (gore) i maksimalne udaljenosti objekata (dolje). Broj stvorenih objekata je 9, a dijagonalni pomak nastaje zbog blagog uvećanja – 1.04..



Slika 24. Razlika u zasićenju boje pozadine, od 0 (gore) koja rezultira monokromatskom slikom, do potpuno zasićene (dolje).

Promjenu u tonu boje pozadine nije potrebno slikovno prikazivati, budući da se ona po HSB (*Hue/Saturation/Brightness*) modelu kreće od 0 do 360 stupnjeva, gdje obje vrijednosti rezultiraju jednakim tonom crvene boje. Što se tiče veličine objekata, kao što je vidljivo na slici 26., nije ju bilo moguće definirati u apsolutnim jedinicama kao što su pikseli, već je bilo potrebno uvesti faktor povećanja, u ovom slučaju od 0.50 do 1.50. Razlog tome je način na koji Processing učitava i koristi fontove – pretvara svaki slovni znak u bitmapu točno određenih dimenzija, zbog čega pretjerano uvećavanje dovodi do pikselizacije i narušava izgled animacije.



Slika 25. Razlika u transparentiji objekata, od 0 (gore) do približno maksimalne vrijednosti, budući da bi maksimalna rezultirala potpuno nevidljivim objektima.



Slika 26. Razlika u relativnoj veličini objekata, od uvećanja 0.50 (gore) do 1.50 (dolje). Stvoreno je 20 objekata, a faktor udaljenosti je 42.

3.3.2. Animacija 2 – WordVortex

Druga je animacija nešto kompleksnija, počevši od toga da za stvaranje kompozicija ne koristi samo jedan slovni znak, već cijelu riječ ili rečenicu teoretski neograničene duljine. Zbog toga su u kodu prisutne neke bitne razlike, no s korisničke strane postoji mnogo sličnosti. Sučelje je slično kao u prethodnom primjeru (slika 27.). U gornjem lijevom kutu nalaze se klizači s kontrolama, s dodatkom trake u kojoj se prikazuje uneseni tekst na samom vrhu prozora. Parametri koje je moguće mijenjati klizačima u ovom su slučaju broj ponavljanja kompozicije, boja i transparentcija elemenata, svjetlina pozadine i stupanj pozadine, dok su tekstualni elementi ostajali bijeli, u ovoj je animaciji moguće detaljnije utjecati na boju elemenata, dok se pozadina mijenja samo po svjetlini, kako bi se osigurao željeni kontrast. U ovom se slučaju tekst upisuje jednom, a kompozicija se prikazuje tek pritiskom na Enter odnosno Return. Tipkom Backspace moguće je brisati tekst, a pritiskom na Tab sprema se slika na ekranu.

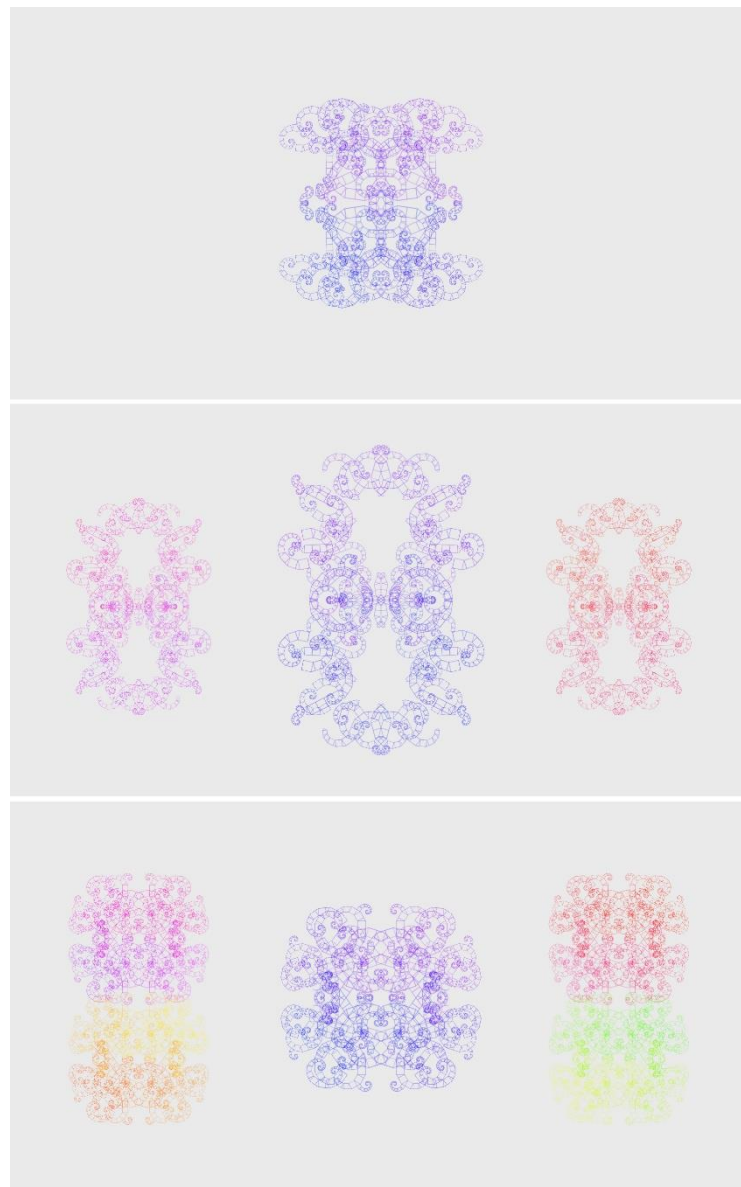


Slika 27. Elementi sučelja druge aplikacije

Za izradu ovog primjera također je stvorena nova klasa objekata, ovaj put nazvana WordVortex. Ne koristi funkciju *key* kako bi prikazala posljednje uneseno slovo, već preko *for* petlje dohvaća pojedina slova iz polja (*array*) koje aplikacija generira iz unesenog teksta koristeći funkciju *toCharArray()*. Svaki sljedeći slovni znak pomaknut je, zarotiran i za 0.05 manji u odnosu na prethodni. Dodatna se promjena događa u tonu boje. Ukoliko je na zadnjem klizaču, koji

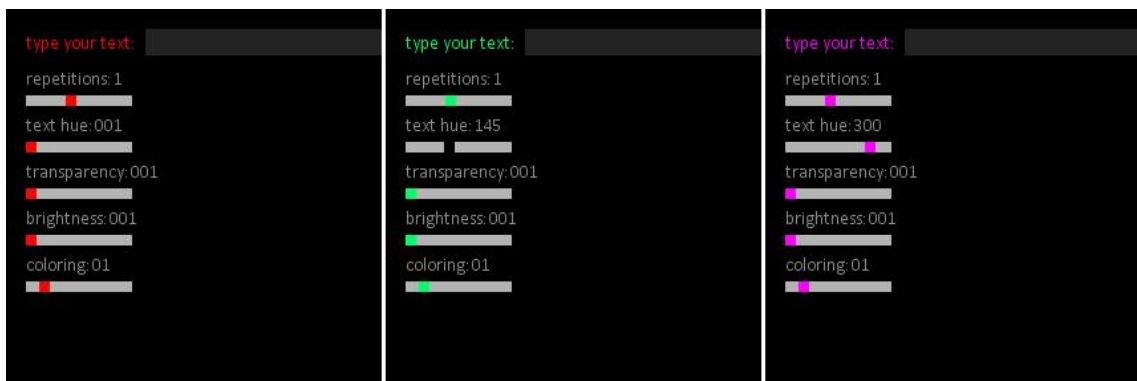
određuje stupanj variranja boje, označena vrijednost veća od 0, svako je slovo za toliko stupnjeva pomaknuto u krugu boja. Rotacija teksta opet je definirana pozicijom miša.

Ovako stvorena klasa integrirana je u glavni kod animacije, gdje se nalaze naredbe za njihovo zrcaljenje i dupliciranje. Logičkim uvjetima *if* definiran je tijekom izvršenja koda u ovisnosti o odabranom broju ponavljanja osnovne kompozicije (slika 28.) Promjenama parametara i unosom različitog teksta moguće je dobiti neponovljive, kompleksne kompozicije.

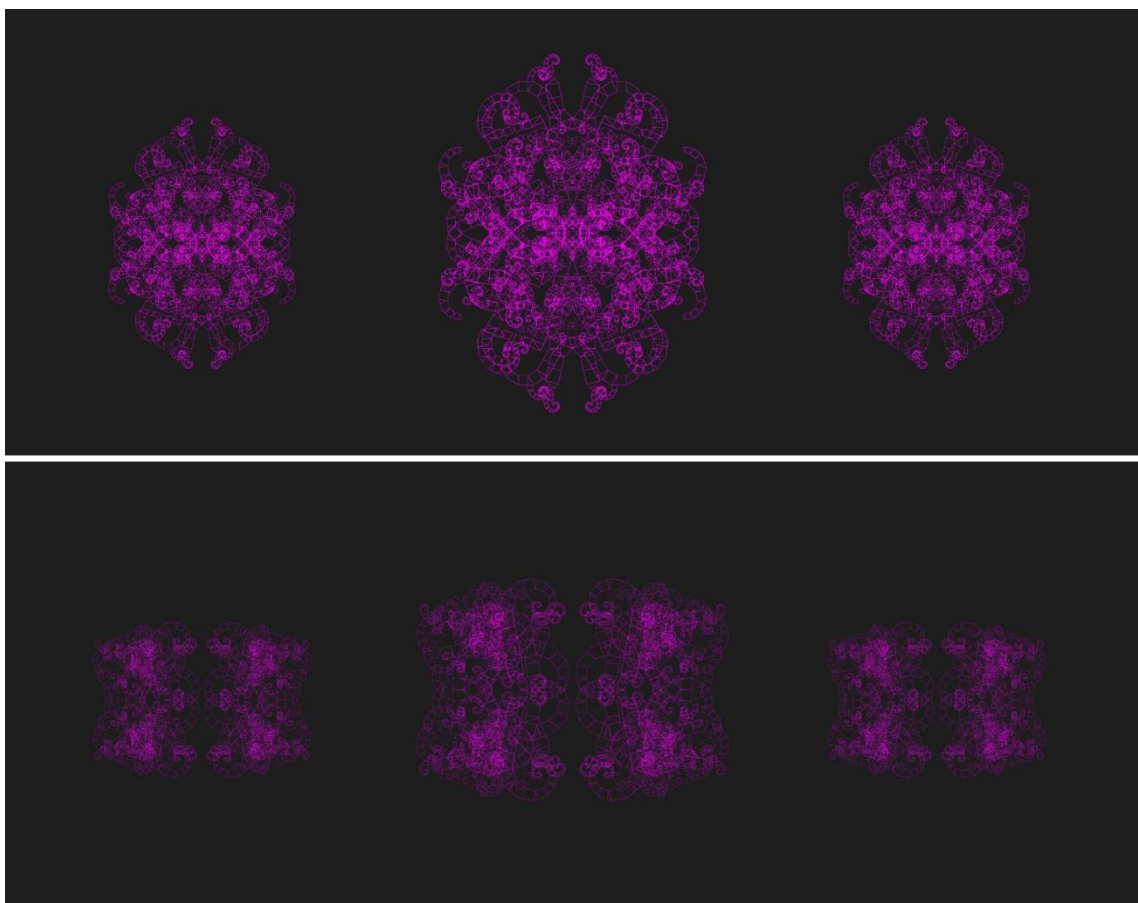


Slika 28. Utjecaj prvog klizača koji kontrolira broj zrcaljenja te stvara jedan, tri ili pet objekata

Kao što je bio slučaj s prvom animacijom, klizač koji kontrolira ton boje u oba ekstrema je jednak te ga nije potrebno prikazivati. Zanimljiva je funkcionalnost promjena boje elemenata sučelja, koja prati promjenu odabranog tona boje (slika 29.). Zahvaljujući tome, moguće je prije generiranja kompozicije vidjeti interakciju boje sa svjetlinom pozadine i po želji prilagoditi kontrast.



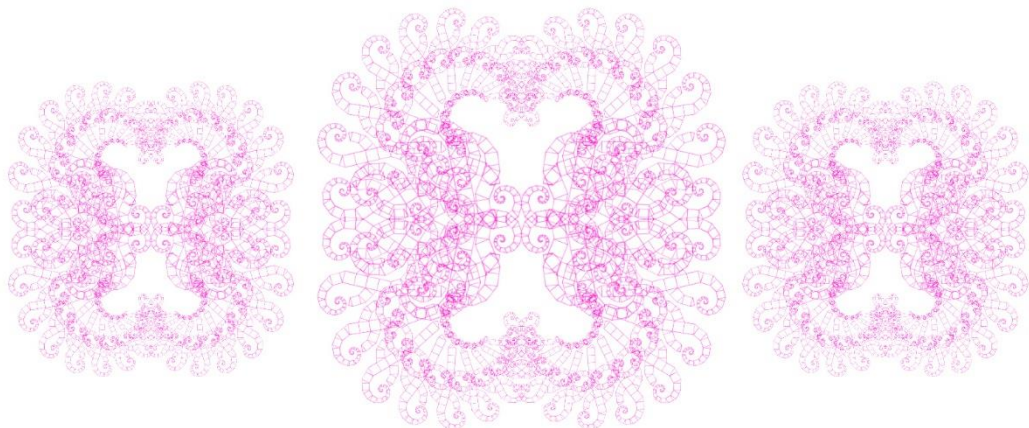
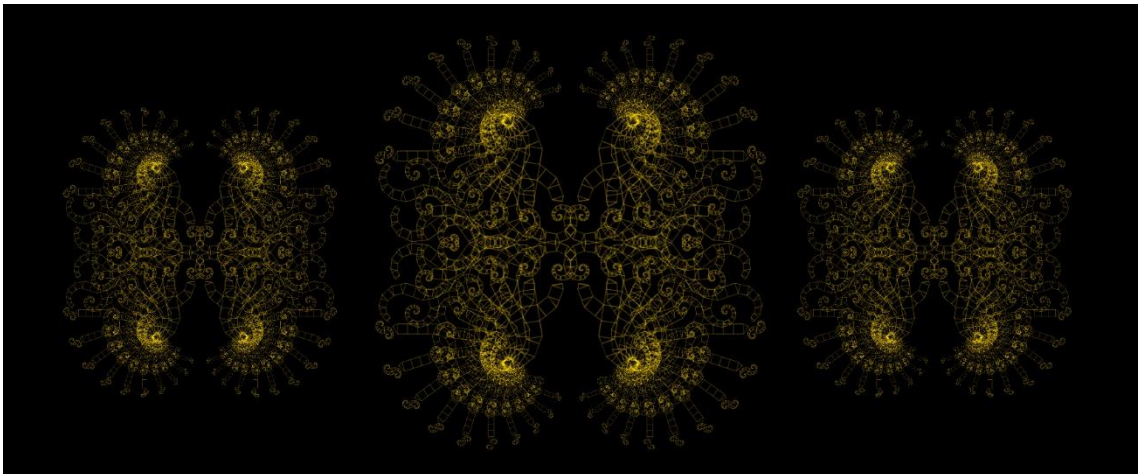
Slika 29. Promjena boje elemenata sučelja ovisno o odabranom tonu boje



Slika 30. Razlika u transparentiji objekata, od 0 (gore) do srednje vrijednosti 50 (dolje)

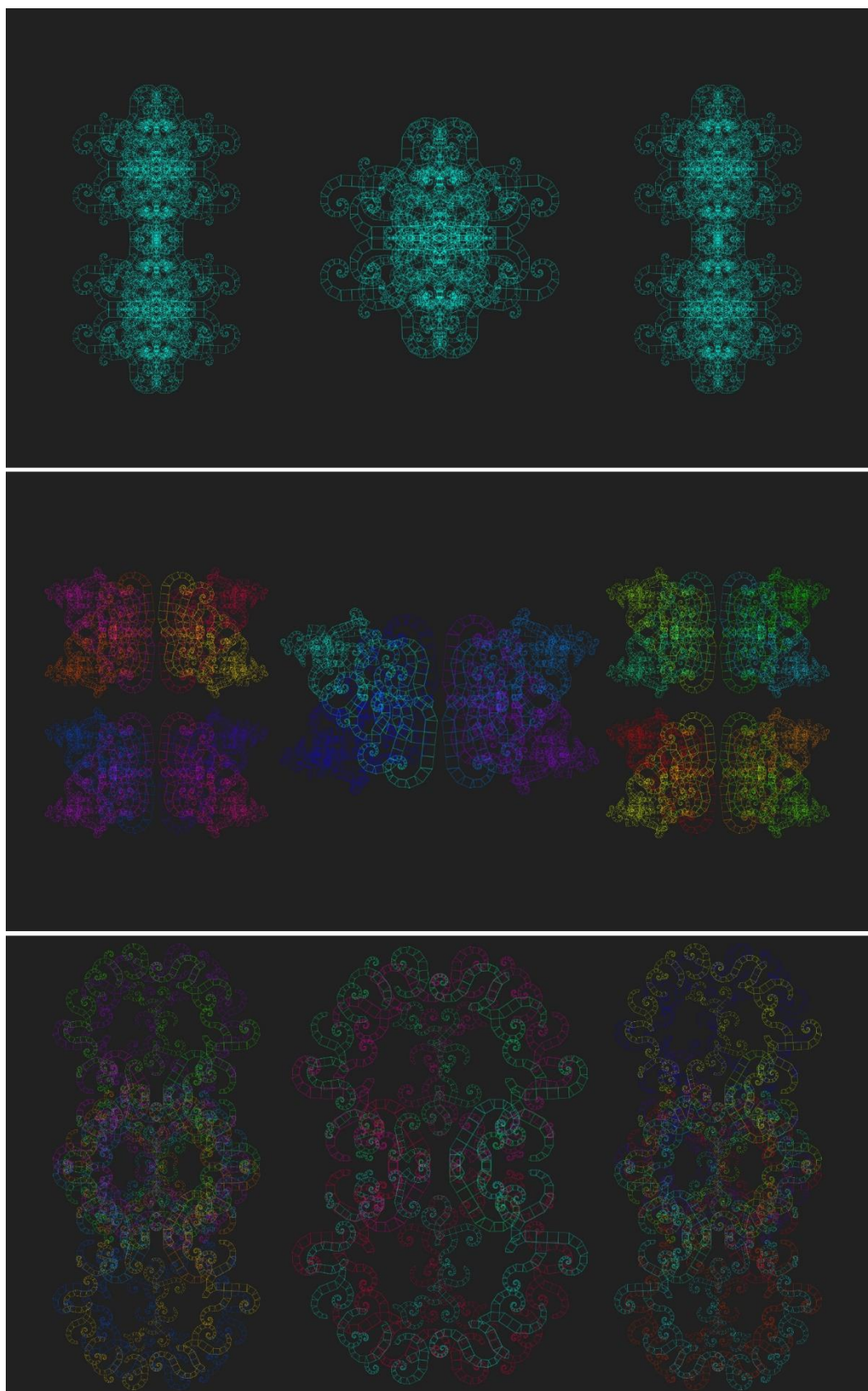
Trećim klizačem prilagođava se transparentcija teksta, čime se mogu dobiti zanimljivi rezultati, pogotovo kod dužih rečenica, gdje preklapanje slova stvara bogate psihodelične uzorke (slika 30.).

Četvrti klizač utječe na svjetlinu pozadine, čijom se promjenom dobiva velika razlika i stvara dojam pozitiva i negativa (slika 31.).



Slika 31. Kompozicija s najmanjom (gore) i najvećom vrijednošću svjetline (dolje)

Posljednji klizač u nizu utječe na karakteristiku koja uvelike pridonosi atraktivnosti i jedinstvenosti generiranih slika. Radi se o postupnoj promjeni boje, gdje je u jednom nizu svako slovo za odabrani broj stupnjeva pomaknuto u krugu boja. Kad izračun prođe 360 stupnjeva, naredba kreće ispočetka. Utjecaji ovog klizača prikazani su na slici 32.

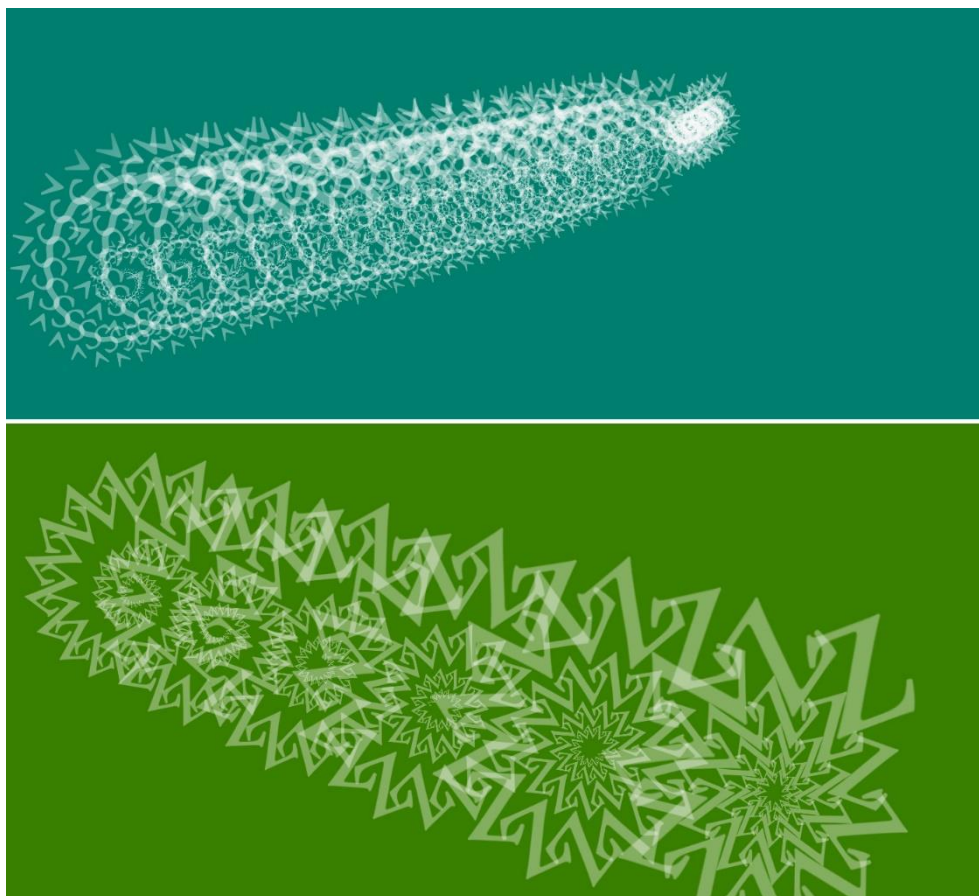


Slika 32. Utjecaj posljednjeg klizača – kompozicija bez varijacija u tonu boje (gore), s malim stupnjem pomaka 3 (sredina) te s maksimalnom vrijednošću 10 (dolje)

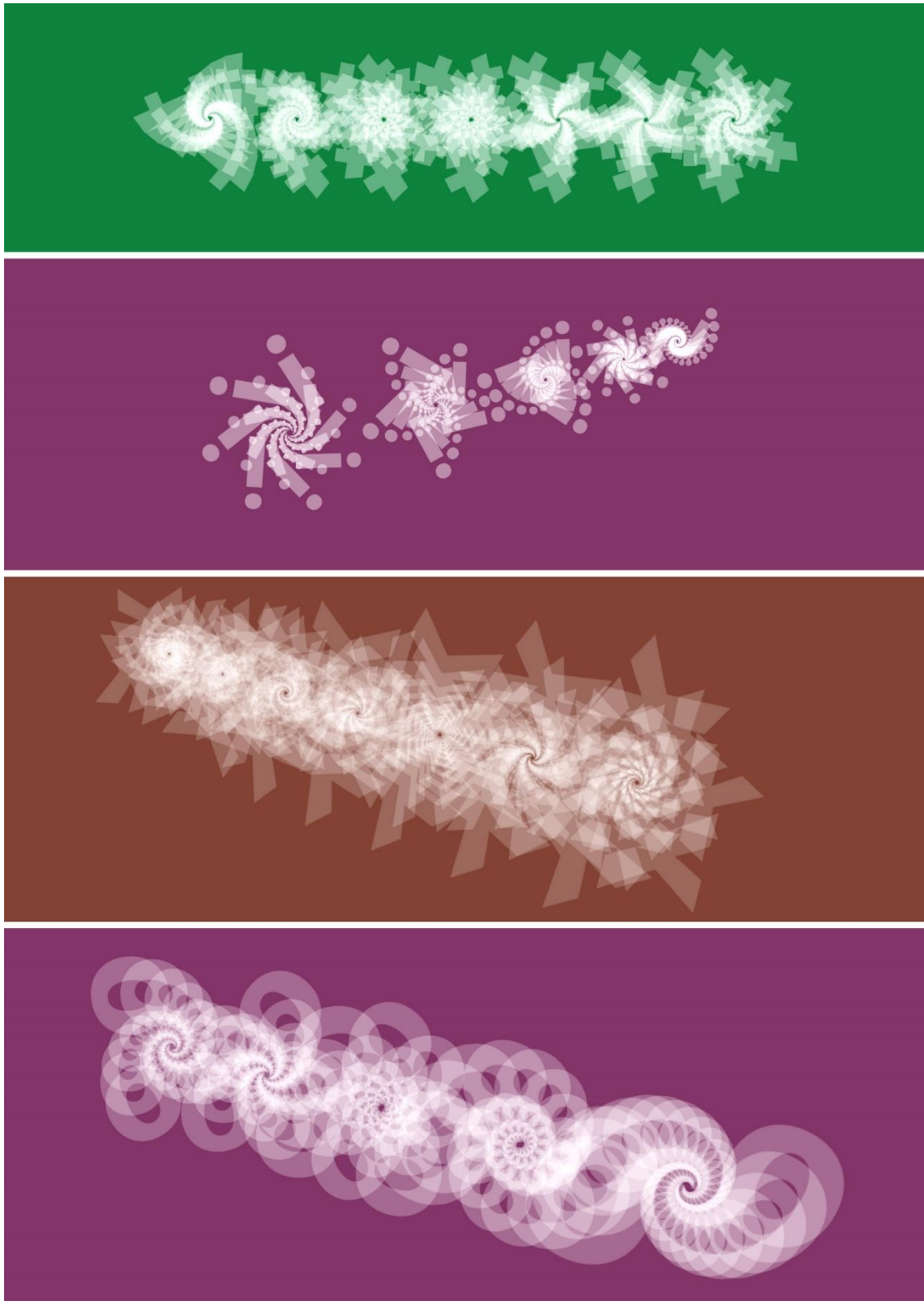
3.4. Dodatne mogućnosti prilagodbe i primjena kreiranih animacija

Jedan od ciljeva izrade animacija bio je stavljanje prethodno izrađenog fonta u upotrebu. Stoga su one i realizirane imajući na umu upravo njegove karakteristike i izgled. Karakter pokreta, slaganja i konačnog izgleda kompozicija upotpunjuje koncept samog fonta. Ipak, to ne znači da neki drugi, klasičniji fontovi ne mogu jednako dobro funkcionirati i uklopiti se u izgled animacija.

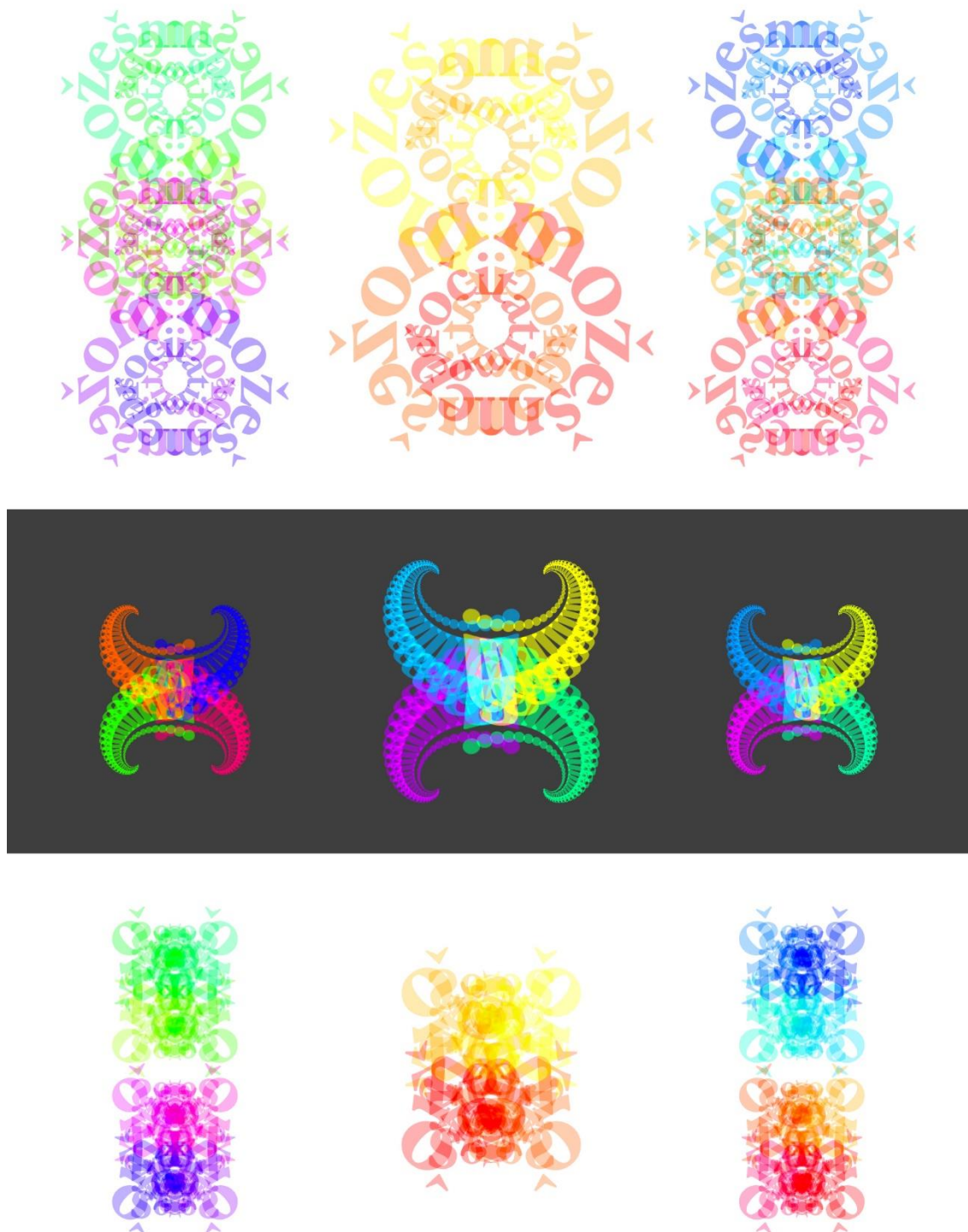
Upotrebom fontova sa većim stupnjem čitkosti i čitljivosti smanjuje se razina apstrakcije dobivenih kompozicija, pa je tako u velikom broju slučajeva moguće razaznati slovne znakove od kojih se sastoje. Naravno, namjena animacija ni tada nije ispisivanje potpuno čitljivog teksta, već se stvara zanimljiva ravnoteža između jasnoće slovnog znaka i njegovog pretvaranja u isključivo likovni element, a estetika oblika je naglašena. Primjeri su prikazani na slikama 33. do 35.



Slika 33. Kompozicije nastale upotrebom fonta Adobe Garamond Pro Regular



Slika 34. Kompozicije nastale upotrebom fonta Myriad Pro Bold



Slika 35. Kompozicije nastale upotrebom fonta Adobe Caslon Pro Bold

Vidljivo je da animacije imaju iznimno visok stupanj prilagodljivosti te da pružaju mogućnost realizacije međusobno vrlo različitih rezultata, pritom zadržavajući konzistentan karakter. Zahvaljujući tome, njihova potencijalna primjena nadilazi samo izradu estetski atraktivnih kompozicija, bilo statičnih ili dinamičnih, koje nisu stavljene u funkciju.

Jedan od načina upotrebe u kojima bi ostvarile svoj potencijal je u kombinaciji s glazbom, u vidu video spota ili projekcija na raznim događanjima. U tom slučaju, kod bi bilo potrebno prilagoditi tako da reagira na određenu komponentu zvuka, poput frekvencije, ritma ili glasnoće, te se ovisno o njima transformira. Postojeći način funkcioniranja dobra je polazišna točka za promišljanje o naprednijim načinima interakcije.

Potpuno drugačiji način primjene, za koji su generirane kompozicije izrazito pogodne, je u području zaštitnog tiska. Veliki broj zrcaljenja i ponavljanja, prilagođavanje putem personaliziranog teksta i generiranje putem koda rezultiraju iznimno kompleksnim grafičkim elementima koje bi bilo iznimno teško falsificirati, čak i putem korištene aplikacije, budući da je podešavanjem parametara moguće dobiti nekoliko stotina milijuna različitih kombinacija.

Animacije su objavljene na Internetu, na adresi <http://typhagora.wordpress.com>, gdje ih je moguće preuzeti na svoje računalo i preko obrasca na stranici poslati vlastite uratke. Stranica sadrži galeriju generiranih slika koja će se kroz vrijeme nadopunjavati, a jedna od bitnih stavki galerije je i skupina animiranih GIF-ova, koji su naknadno izrađeni iz slika kreiranih putem aplikacija, što čini dodatnu mogućnost njihove upotrebe.

4. ZAKLJUČAK

Upotreba tipografije u računalnoj animaciji izvrstan je primjer situacije u kojoj tipografija komunicira isključivo svojom formom u multimedijском okruženju. Za razliku od klasičnog načina primjene, gdje su prioriteti čitkost i čitljivost, estetika i vizualni dojam slovnih znakova stavljeni su u prvi plan. Upotrebom tipografije kao glavnog izražajnog sredstva animacije, otvaraju se brojne mogućnosti za stvaranje dinamičnih kompozicija koje nedvojbeno privlače publiku i zadržavaju njihovu pažnju više od statičnih, što se dodatno proširuje uvođenjem interakcije s korisnikom.

Izrađene animacije stavile su naglasak na fleksibilnost i interaktivnost konačnog rezultata, budući da se kombinacijom relativno malog broja parametara dolazi do raznovrsnih kompozicija koje je nemoguće ponoviti. Istražene su brojne razine kompleksnosti kompozicija te pretvaranja teksta u potpunu apstrakciju. Prethodno dizajnirani autorski font uklopio se u osnovni koncept i estetiku animacija, time zadržavajući konzistenciju ideje. Osim toga, istražena je i mogućnost korištenja ostalih, standardnih fontova (Garamond, Myriad, Caslon) u istu svrhu, što je rezultiralo interesantnim pretvaranjem uobičajenih, dobro poznatih slova u apstraktne likovne elemente.

Primjena interaktivnih animacija poput ovih sve je šira razvojem novih tehnologija i medija, te obuhvaća raznolika područja, od umjetnosti do onih nešto funkcionalnijih, kao što su oglašavanje ili edukacija. Njihovi eventualni produkti, kao što su u ovom slučaju generirane slike, također nadilaze okvire umjetničkog izražavanja mogu naći novu praktičnu svrhu u područjima poput zaštitnog tiska.

5. LITERATURA

1. Pipes A. (2005). *Production for Graphic Designers*, Laurence King Publishing, London
2. *** <http://www.fusion-journal.com/issue/001-fusion/fluid-characters-in-temporal-typography/> - *Fluid Characters in Temporal Typography*, 8. kolovoza 2015.
3. *** <https://www.siggraph.org/education/materials/HyperGraph/animation/anim0.htm> – *Computer Animation*, 8. kolovoza 2015.
4. Dinç N. (2006). *Traditional and Interactive Animation: From Perspective of Storytelling and Production Process*, dostupno na: http://newmedia.yeditepe.edu.tr/pdfs/isimd_06/06.pdf, 9. kolovoza 2015.
5. *** https://en.wikipedia.org/wiki/Interactive_art - Wikipedia, Interactive art, 8. kolovoza 2015.
6. *** <http://www.laikafont.ch> – Laika, 8. kolovoza 2015.
7. Šubat T. (2013). *Eksperimentalna tipografija*, završni rad, Grafički fakultet Sveučilišta u Zagrebu
8. Čić M. (2003). *Uvod u programiranje*, dostupno na http://laris.fesb.hr/java/blokovi_instance.htm, 10. kolovoza 2015.