

Uvod

Još od najranijih vremena postojanja ljudske civilizacije, čovek je nastojao da pojave oko sebe naslika i na taj način ovekoveči. Sa pojavom prvih slika i crteža, kod čoveka se javila potreba da, pored statičnih, nacrtanih predstava, dočara i magiju pokreta. Upravo takav proces, kojim se stvara iluzija pokreta, naziva se **animacija**. U kursu pred vama biće ilustrovane najrelevantnije tehnike za postizanje animacije na webu, počevši od onih najjednostavnijih. Tako će na početku biti prikazane tehnike koje omogućavaju unapređivanje korisničkog okruženja, suptilnom animacijom UI elemenata. Kako kurs bude odmicao, biće razmatrane i složenije tehnike, koje će nam omogućiti da animiramo grafiku koju samostalno budemo crtali. Na kraju, kao najsloženiji i najefektniji vid animacije, kurs će predstaviti i kreiranje jedne igre koja će se izvršavati unutar web pregledača.

Animacija kroz vreme

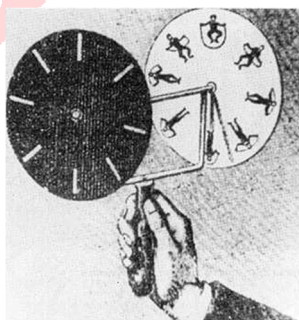
Tokom vremena, ljudi su koristili različite tehnike za postizanje animacije. Najstariji primer animacije jeste serija uzastopno povezanih crteža, starih preko 5.200 godina, koji su pronađeni u gradu *Shahr-e Sūkhté*, na jugoistoku Irana. Pronađeno delo sačinjeno je od 5 povezanih crteža koji ilustruju persijsku pustinjsku kozu koja skače kako bi dohvatila list sa grane (slika 1.1).



Slika 1.1. Jedan od najstarijih primera animacije (izvor https://en.wikipedia.org/wiki/History_of_animation)

Svaki od crteža ilustruje jedan trenutak u vremenu, a svi crteži zajedno prikazuju hronološki proces koji simulira dešavanje, odnosno kretanje.

Najraniji primeri animacije, kakav je i ovaj ilustrovan slikom 1.1, oslanjali su se na percepciju gledaoca u stvaranju iluzije prikaza pokreta. Kako je čovečanstvo napredovalo, javila se ideja o kreiranju uređaja (mašine) koja bi obavljala naizmeničnu promenu slika, čime bi se dobila iluzija kretanja. Prvi uređaj takve vrste bio je **fenakistiskop** (slika 1.2).



Slika 1.2. Fenakistiskop (izvor <https://sh.wikipedia.org/wiki/Fenakistoskop>)

Fenakistiskop je prvi, široko rasprostranjeni uređaj za stvaranje animacije. Reč je o vrlo jednostavnoj mehaničkoj napravi, koja se sastojala iz dva okrugla kartona, spojena osovnom. Osovina je omogućavala da se jedan od kartona neometano okreće oko svog centra. Na ivici takvog kartona nalazili su se crteži koji su ilustrovali kadrove animacije. Drugi karton posedovao je vertikalni prorez koji je korišćen za posmatranje animacije. Sve što je bilo potrebno uraditi jeste zavrteti karton sa slikama i kroz otvor na drugom kartonu posmatrati slike, koje međusobnim smenjivanjem stvaraju iluziju kretanja (animacija 1.1).

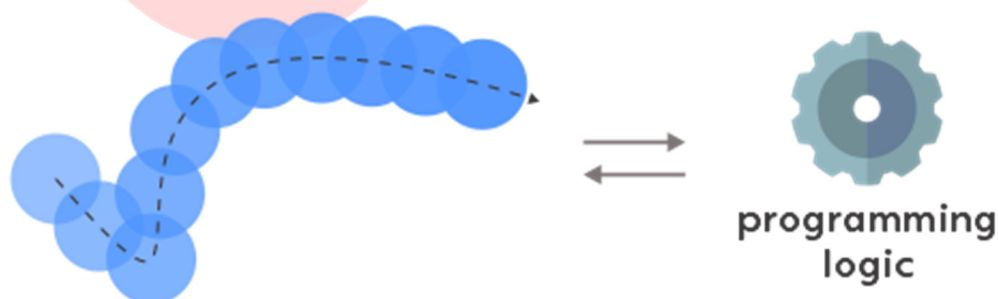


Animacija 1.1. Simulacija efekta koji je proizvodio fenakistiskop (izvor <https://en.wikipedia.org/wiki/Phenakistiscope>)

Kompjuterska animacija

Kratak uvod u istoriju animacije na početku ove lekcije nije naveden bez razloga. Osnovni princip na kom se zasniva rad fenakistiskopa uopšte se ne razlikuje od principa koji se koriste za postizanje kompjuterske animacije. Jedina razlika se ogleda u oruđu koje se koristi za postizanje animacije.

Za ono za šta su čoveku bili potrebni dani i meseci danas se korišćenjem kompjutera realizuje za nekoliko minuta. Kompjuterska animacija se reprodukuje na monitorima, odnosno displejima, a izvodi je hardver računara koji kontroliše softverska logika. Tako je fenakistiskop i njemu slične naprave u digitalnom svetu zamenjen programskom logikom, napisanom nekim od programskih jezika. Crteži tradicionalne animacije na kartonu ili na nekom drugom materijalu zamenjeni su elektronskim prikazima sačinjenim iz mreže piksela.



Slika 1.3. Princip postizanja kompjuterske animacije

Slika 1.3. ilustruje osnovni princip funkcionisanja kompjuterske animacije, gde programska logika igra presudnu ulogu u stvaranju iluzije kretanja. Slikom je ilustrovana animacija kojom se dočarava kretanje jednog kruga. Programska logika, u veoma kratkim vremenskim intervalima, generiše prikaz scene koja se prikazuje na monitoru. Sukcesivne scene se međusobno veoma malo razlikuju tako što se pozicija kruga za nekoliko piksela pomera duž definisane trajektorije. Zbog brzog smenjivanja susednih scena gledalac stiče utisak kretanja.

Statička i dinamička animacija

Animacija predstavlja veoma širok pojam. Postepena promena boje pozadine nekog HTML elementa unutar web pregledača svakako je jedan od primera animacije. Kretanje Flappy Bird karaktera u istoimenoj popularnoj igri takođe je jedan od primera animacije. Ipak, između dva navedena primera animacije, postoje značajne razlike. Prvi navedeni scenario najčešće predstavlja statički tip animacije. To znači da je animacija definisana mnogo pre nego što je reprodukovana, čak i pre samog pokretanja aplikacije. Njene osobine su tačno utvrđene od strane programera, te je stoga ona svaki put identična. Razlikuje se samo trenutak njenog pokretanja.

Sa druge strane, kretanje Flappy Bird karaktera predstavlja primer dinamičke animacije. To je animacija čije osobine nisu unapred poznate, već se određuju tokom izvršavanja aplikacije (engl. *runtime*), a sve u zavisnosti od reakcija igrača.

Pitanje

Animacija unutar igara predstavlja vrstu:

- a) **dinamičke animacije**
- b) statičke animacije
- c) fokusirane animacije
- d) dirigovane animacije

Objašnjenje

Animacija unutar igara predstavlja primer dinamičke animacije. To je animacija čije osobine nisu unapred poznate, već se generišu tokom izvršavanja aplikacije (engl. runtime).

Animacija na webu

U svetu modernog web razvoja, animacija zauzima veoma važno mesto. Mi ćemo se u nastavku ovoga kursa baviti kreiranjem animacije sa dva osnovna mesta primene:

- **Animacija korisničkog okruženja** (engl. *UI Animation*) – Prilikom razvoja korisničkog okruženja web aplikacija, animacija se može koristiti da korisniku, na suptilan način ukaže na ono što se događa prilikom izvršavanja aplikacije, te da na taj način podigne opšti korisnički ugođaj. Animacija je posebno korisna kada je potrebno dočarati promenu stanja korisničkog okruženja aplikacije. Takve situacije se mogu stvoriti prilikom učitavanja ili unosa podataka, prelaska sa jednog na neki drugi prikaz, ali i u situacijama u kojima korisniku postaje dostupna neka nova akcija. Animacija se na kraju može koristiti i za postizanje sofisticiranijeg izgleda korisničkog okruženja, što kod korisnika rezultuje osećanjem kvalitetnije aplikacije. Jedan deo ovoga kursa biće posvećen različitim pristupima za kreiranje animacije korisničkog okruženja web aplikacija.

- **Animacija unutar igara** (engl. *Game Animation*) – Osnova gotovo svake igre jeste animacija. Kažemo, gotovo svake igre, zato što postoje i igre koje se mogu u potpunost realizovati bez upotrebe animacije (na primer, igre koje podrazumevaju upotrebu table kao što su šah ili iks-oks). Ipak, mnogo je veći broj onih igara koje u većoj ili manjoj meri podrazumevaju upotrebu animacije. Pritom je animacija unutar igara klasičan primer dinamičkog tipa animacije, zato što korisnik, odnosno igrač ima mogućnost da tokom izvršavanja igre utiče na osobine animacije. Drugi deo ovog kursa biće posvećen kreiranju web igara korišćenjem različitih pristupa.

Tehnike za postizanje animacije na webu

Animaciju je na webu moguće postići na različite načine, a sve u zavisnosti od efekta koji je potrebno postići. Ipak, svi tipovi animacije podrazumevaju postojanje grafičkog elementa koji se animira. To može biti neki ugrađeni HTML element, slika, vektorska grafika, ali i grafika koja je nacrtana potpuno programabilno, korišćenjem JavaScript koda i specijalnog canvas HTML elementa koji omogućava crtanje.

Animacija se unutar web aplikacija može kreirati korišćenjem dva jezika, pa se tako može reći da postoji:

- **CSS animacija,**
- **JavaScript animacija.**

CSS animacija

CSS jezik poseduje dva sistema koja omogućavaju kreiranje animacije:

- CSS tranzicije (engl. *CSS transitions*),
- CSS animacije (engl. *CSS animations*).

Ova dva sistema CSS animacije primarno su namenjeni za animiranje UI elemenata, pa samim tim za unapređivanje opšteg korisničkog doživljaja prilikom korišćenja web aplikacija i sajtova. CSS animacije su vrlo jednostavne za kreiranje i potrebno ih je koristiti uvek kada se željena animacija može kreirati korišćenjem funkcionalnosti CSS jezika. Neki od klasičnih scenarija za primenu CSS animacije mogu biti:

- animacija prilikom prelaska miša preko nekog HTML elementa (engl. *hover*),
- animacija koja se koristi da simbolizuje učitavanje stranice ili neku drugu vremenski zahtevnu operaciju (engl. *loader animation*),
- animacija galerije ili portfolija,
- animacija zaglavlja, navigacije, modalnih prozora...

Oba spomenuta CSS sistema za postizanje animacije biće predmet jedne od narednih lekcija.

JavaScript animacija

Iz prethodnih redova, potpuno je jasno da su CSS sistemi za animiranje primarno namenjeni postizanju UI animacije. Dinamičnost je ograničena na mogućnost kontrolisanja početka i završetka animacije. To praktično znači da su osobine animacije unapred utvrđene i da je jedina nepoznanica kada će animacija započeti, što zavisi od korisnika web sajta, koji će takvu animaciju nekim od svojih postupaka da pokrene. Ipak, ovaj kurs će se, pored statičke animacije, baviti i različitim pristupima za postizanje dinamičke animacije, odnosno animacije čije se osobine mogu kontrolisati tokom njenog izvršavanja. Za obavljanje takvog posla neophodno je korišćenje JavaScript jezika.

Kurs pred vama biće primarno posvećen kreiranju animacije upotrebom JavaScript jezika. Takav posao, pored JavaScript jezika, podrazumeva korišćenje još nekih pojmova, koji u simbiozi omogućavaju da se kreira dinamička animacija unutar web pregledača:

Tajming funkcije

Svaka animacija zasniva se na kontinuiranoj, postepenoj promeni vizuelnih osobina elemenata koji se animiraju. Osobine elemenata koji se animiraju, ažuriraju se i po više desetina puta u jednoj sekundi, čime se dobija privid kretanja. Stoga su osnova svake JavaScript animacije, tajming funkcije koje omogućavaju da se određeni blok koda ciklično ponavlja. Web pregledači nama na korišćenje izlažu nekoliko takvih funkcija:

- `setTimeout()`,
- `setInterval()`,
- `requestAnimationFrame()`.

Navedene tajming funkcije biće jedna od najznačajnijih tema lekcija koje slede, s obzirom na to da one predstavljaju osnovni motor animacije.

- **Canvas element i Canvas API** – JavaScript se retko koristi za animiranje HTML elemenata. Iako je tako nešto moguće uraditi, savetuje se korišćenje CSS animacija ili tranzicija ukoliko je potrebno postići neku animaciju unutar korisničkog okruženja aplikacija. Ipak, za naprednije animacije i za kreiranje igara pribegava se korišćenju JavaScript jezika za animiranje grafike koja se crta unutar jednog posebnog HTML elementa. Reč je o elementu `canvas`. `canvas` koji u bukvalnom prevodu znači platno na kome je moguće crtati proizvoljnu grafiku. Crtanje grafike obavlja se korišćenjem jednog posebnog aplikativnog programskog interfejsa web pregledača koji se zove Canvas API. Grafika nacrtana unutar `canvas` elementa može se animirati korišćenjem JavaScript jezika. Nakon upoznavanja CSS animacije, u ovom kursu će posebna pažnja biti posvećena crtanju grafike unutar `canvas` elementa i njenom animiranju korišćenjem JavaScript jezika.
- **Canvas element i WebGL** – Pored skupa funkcionalnosti unutar Canvas API-ja, crtanje grafike unutar `canvas` elementa može se obaviti korišćenjem još jednog aplikativnog programskog interfejsa web pregledača – WebGL. WebGL (Web Graphics Library) je JavaScript biblioteka koja omogućava da se unutar `canvas` HTML elementa obavi crtanje i animiranje 2D i 3D grafike i da se tom prilikom iskoriste sve pogodnosti grafičkog hardvera koji postoji na korisničkom uređaju. WebGL tako nešto omogućava oslanjajući se u pozadini na OpenGL ES biblioteku koja je posebno koncipirana za kreiranje hardverski pogonjene, interaktivne grafike visokih performansi.

Rezime

- Proces kojim se stvara iluzija pokreta naziva se animacija.
- Još od najstarijih vremena, ljudska vrsta pokušava da dočara kretanje.
- Fenakistiskop je jedan od najstarijih uređaja koji su se koristili za naizmeničnu promenu slika, te dobijanje iluzije kretanja.
- Moderna, digitalna animacija oslanja se na hardver kompjuterskih uređaja, a kontrolisana je od strane programske logike.
- Animaciju na webu je moguće kreirati korišćenjem dva jezika, CSS-a i JavaScripta, i različitih pripadajućih skupova funkcionalnosti.

- CSS animacija podrazumeva korišćenje CSS jezika za kreiranje animacije; CSS jezik poseduje dva sistema za animiranje: CSS tranzicije i CSS animacije.
- JavaScript animacija podrazumeva kontinuirano, programabilno ažuriranje osobina vizuelnih elemenata koji se animiraju.
- Osnova svake JavaScript animacije jeste neka od tajming funkcija: `setTimeout()`, `setInterval()` ili `requestAnimationFrame()`, koje omogućavaju da se logika animacija ciklično ponavlja.
- HTML element `canvas` omogućava crtanje proizvoljne grafike.
- Unutar `canvas` HTML elementa grafiku je moguće crtati korišćenjem JavaScript jezika.
- Canvas API obezbeđuje skup funkcionalnosti koje omogućavaju crtanje dvodimenzionalne grafike unutar HTML elementa `canvas`.
- WebGL API omogućava da se unutar `canvas` HTML elementa obavi crtanje i animiranje 2D i 3D grafike i da se tom prilikom iskoriste sve pogodnosti grafičkog hardvera.

