

Komunikacija bez prepreka – izrada inkluzivne mobilne aplikacije u MIT App Inventoru

Scenarij poučavanja Komunikacija bez prepreka namijenjen je nastavnicima informatike i osmišljen je za provedbu kroz više nastavnih susreta (2-3 blok sata). Temelji se na projektnoj nastavi u kojoj učenici razvijaju inkluzivnu mobilnu aplikaciju od osnovne funkcionalnosti do naprednijih rješenja koristeći MIT App Inventor. Cilj scenarija je upoznati učenike s osnovama razvoja mobilnih aplikacija, primjenom umjetne inteligencije u obliku prepoznavanja govora i pretvorbe teksta u govor te ih potaknuti na razmišljanje o ulozi tehnologije u inkluziji i društvenoj odgovornosti. Aktivnost se temelji na upoznavanju i razumijevanju komunikacijskih prepreka s kojima se susreću osobe s oštećenjem sluha i vida te na razvoju rješenja koje kombinira vizualne i auditivne elemente kojima tehnologija postaje inkluzivno sredstvo komunikacije.

Očekivani ishodi:

Učenici će moći:

- objasniti kako funkcioniraju komponente za prepoznavanje govora i pretvorbu teksta u govor te njihovu primjenu u mobilnim aplikacijama
- razlikovati Designer i Blocks dio u MIT App Inventoru te objasniti njihovu ulogu u razvoju aplikacije
- izraditi funkcionalnu mobilnu aplikaciju koja integrira vizualne i auditivne elemente
- prepoznati i opisati komunikacijske poteškoće s kojima se susreću osobe s oštećenjem sluha i vida
- povezati tehničko rješenje s inkluzivnim scenarijem i objasniti kako aplikacija doprinosi pristupačnosti i smanjenju komunikacijskih prepreka
- analizirati ulogu digitalnih tehnologija u unapređenju kvalitete života i ravnopravnosti korisnika

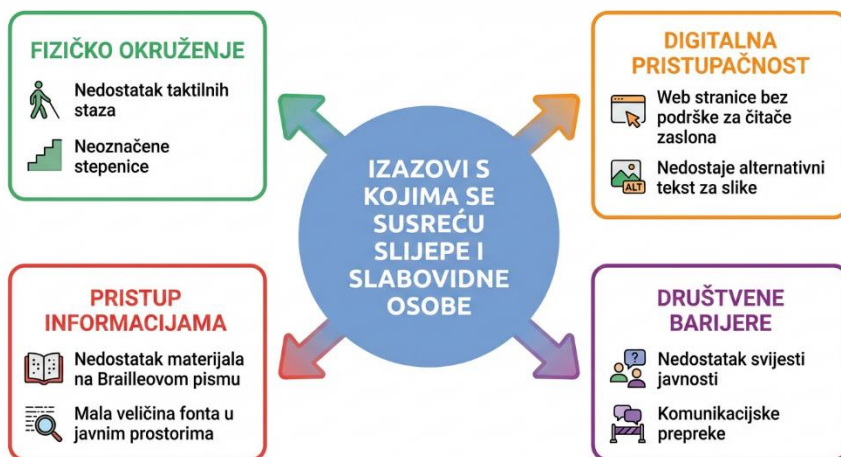
Scenarij obuhvaća sedam dijelova: uvod i analiza problema komunikacije, upoznavanje MIT App Inventora, planiranje aplikacije, izradu aplikacije, testiranje i doradu te završnu refleksiju.

1. UVOD – PROBLEM KOMUNIKACIJE

Motivacija: nastavnik postavlja pitanja:

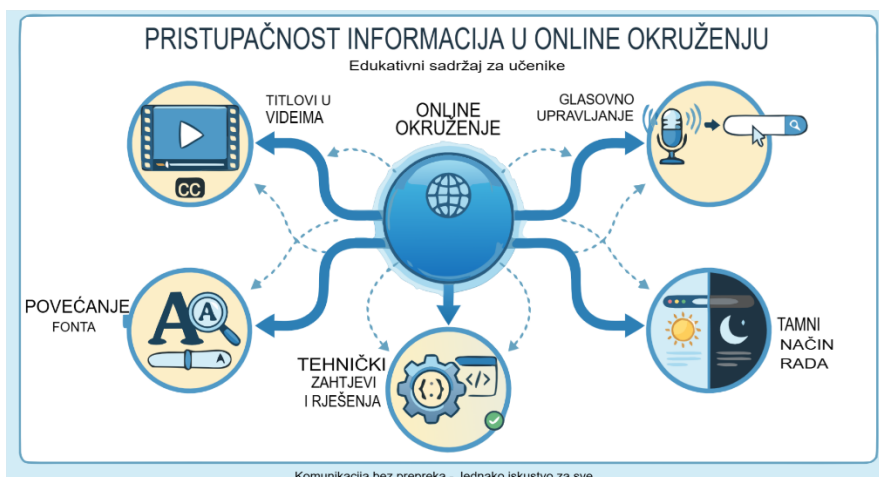
- „Zamislite da ne čujete. Kako biste komunicirali u trgovini?“
- „Zamislite da ne vidite. Kako biste pročitali poruku na mobitelu?“
- „Može li mobitel postati alat koji povezuje ove dvije situacije?“

Učenici iznose primjere iz svakodnevnog života.



Nastavnik uvodi pojam inkluzivne tehnologije i objašnjava da se radi o digitalnim rješenjima koja su osmišljena tako da budu pristupačna i korisna osobama s različitim sposobnostima i potrebama. Ukazuje da inkluzivna tehnologija ne podrazumijeva samo prilagodbu osobama s invaliditetom, već dizajn koji unaprijed uzima u obzir raznolikost korisnika poput različite dobi, jezične barijere, privremene poteškoće ili situacijske prepreke.

Učenici raspravljaju o primjerima iz svakodnevnog života (titlovi u videima, glasovno upravljanje, povećanje fonta, tamni način rada) te uočavaju kako su takva rješenja već prisutna u tehnologiji koju svakodnevno koriste.



Predstavlja se projektni zadatak:

Izraditi mobilnu aplikaciju koja:

- prepoznaje govor
- prikazuje tekst
- izgovara tekst naglas
- daje vibracijsku ili zvučnu potvrdu

Naglašava se da aplikacija treba služiti osobama koje:

- ne čuju (čitaju tekst)
- ne vide (slušaju tekst i osjećaju vibraciju)

Učenici postupno uviđaju da tehnologija nije samo sredstvo zabave ili komunikacije, već alat koji može smanjiti svakodnevne prepreke i povećati samostalnost korisnika. Kroz razgovor zaključuju da pravilno osmišljena aplikacija može omogućiti razumijevanje ondje gdje postoji komunikacijska barijera; pretvoriti govor u tekst, tekst u govor ili pružiti dodatnu osjetilnu potvrdu pri čemu kvaliteta digitalnog rješenja ne ovisi isključivo o tehničkoj izvedbi, već i o tome koliko odgovara stvarnim potrebama ljudi.

2. UPOZNAVANJE MIT APP INVENTORA

Prijava i spajanje mobilnog uređaja

Početni koraci:

- App Inventoru pristupamo na adresi: <https://ai2.appinventor.mit.edu/>
- Kod prve prijave možemo odabrati s kojim korisničkim računom želimo pristupiti (možemo odabrati neki koji često koristimo npr. google korisnički račun).
- Nakon prijave pojavljuje se početni prozor iz sustava pomoći u kojem možemo odabrati želimo li informacije o spajanju mobitela s računalom ili korištenja tzv. emulatora. Možemo odabrati i da se prozor više ne pojavljuje. Emulator je program koji "glumi" mobitel na našem računalu.

Učenici odlaze na <https://ai2.appinventor.mit.edu/>

Skidanje i instaliranje programa MIT AI2 Companion App na mobitel.

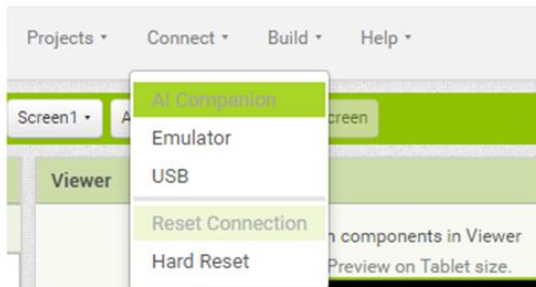
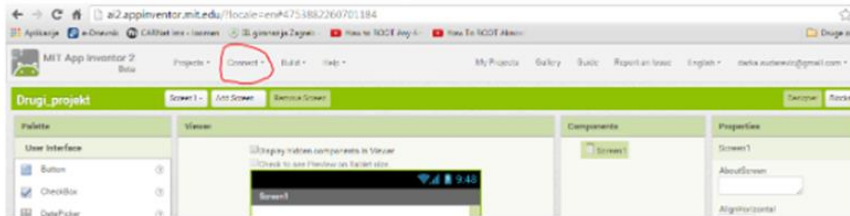
Skinuti ga s Play Stora koristeći QR kod



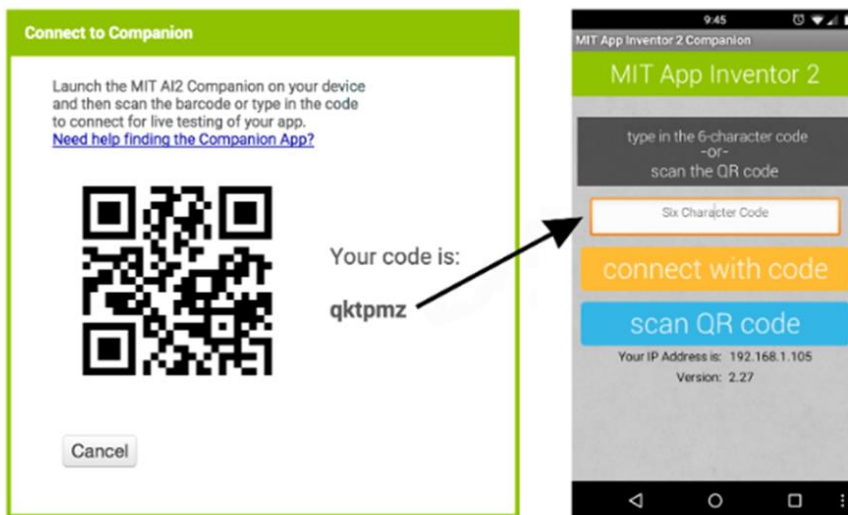
ili izravnim upisivanjem adrese

<https://play.google.com/store/apps/details?id=edu.mit.appinventor.aicompanion3>.

Otvoriti App Inventor i u izborniku Connect odabrati AI Companion.



Pojavit će se QR kod pa možemo nakon pokretanja AI Companiona na mobilnu odabrati spajanje skeniranjem koda ili upisivanjem koda.

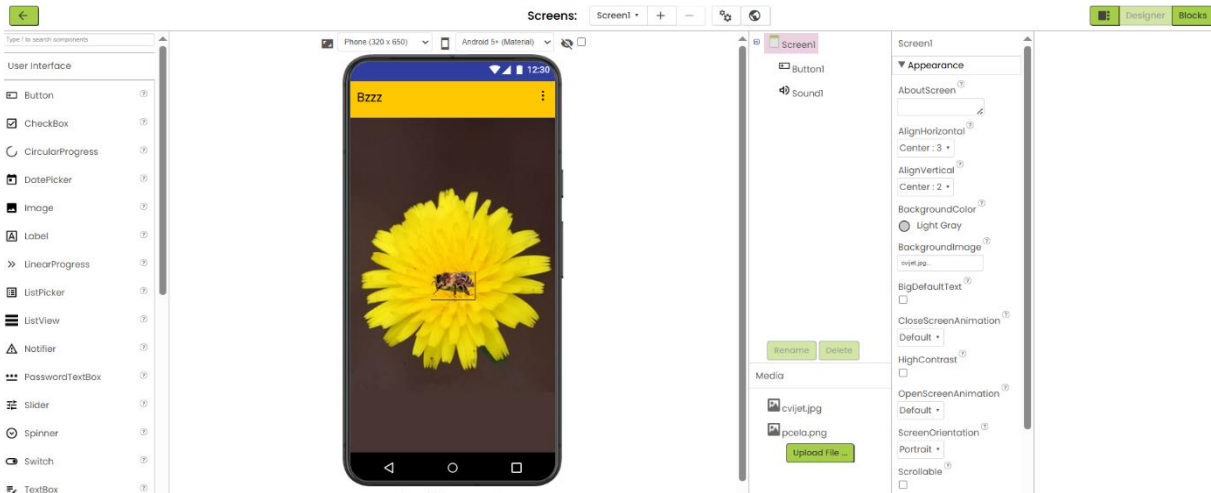


Na taj smo način spojili naše računalo sa mobilnim uređajem. Nastavnik objašnjava da se sve promjene u aplikaciji odmah prikazuju na mobitelu.

Upoznavanje sučelja

Nastavnik objašnjava dijelove sučelja:

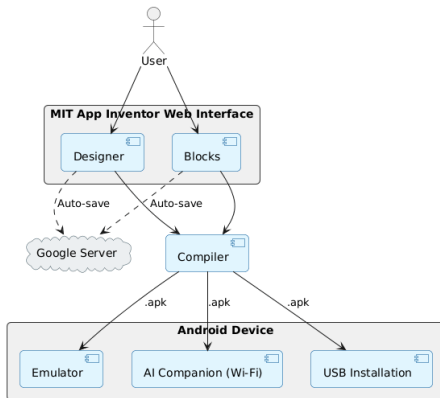
- Designer – izrada sučelja
- Palette – odabir komponenata
- Viewer – prikaz aplikacije
- Components – popis elemenata
- Properties – postavke
- Blocks – programiranje logike



Kada želimo dodati ponašanje određenih elemenata (objekata) naše buduće aplikacije odabiremo dio Blocks.

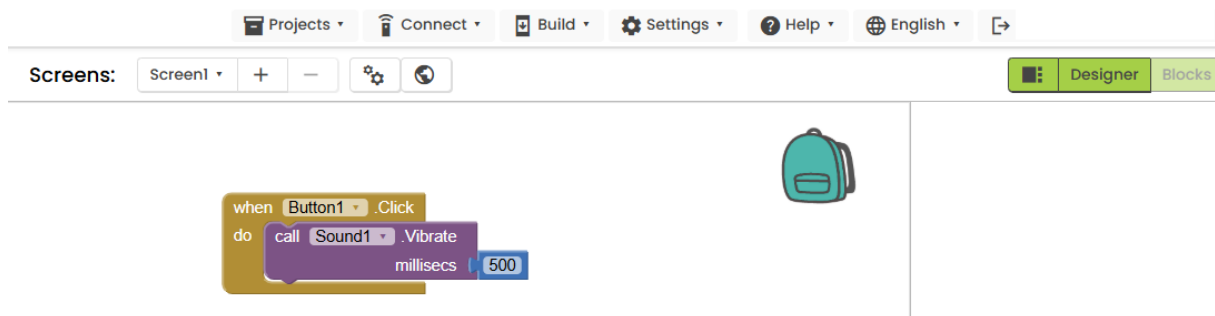
Prelazimo u "programerski dio" koji se temelji na programiranju povlačenjem pojedinih instrukcija.

Svi događaji koji su mogući na pojedinom elementu aplikacije (objektu) nalaze se u popisu u dijelu Blocks.



Kroz kratki primjer učenici:

- dodaju Button
- dodaju Sound
- programiraju vibraciju

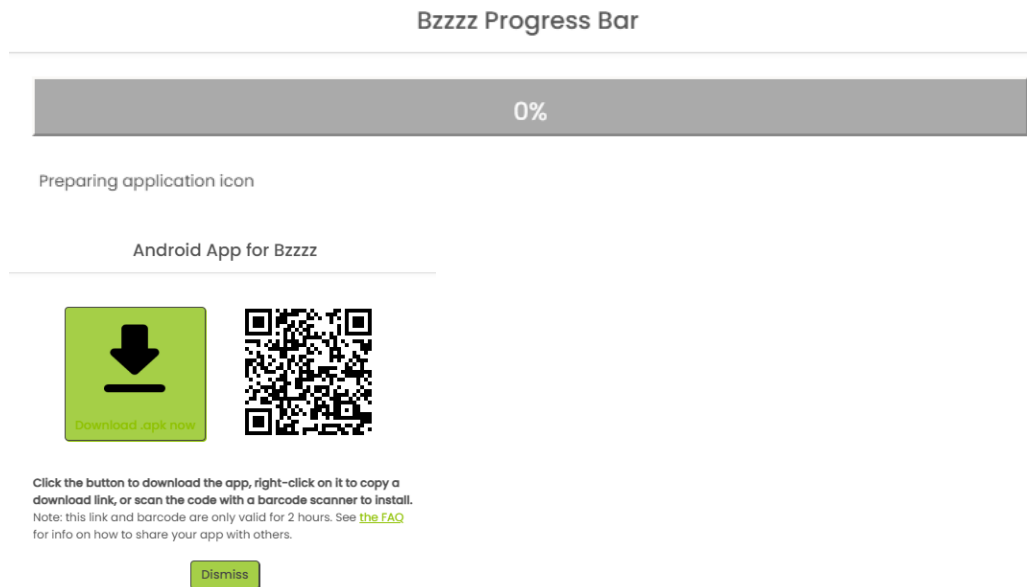


Cilj je razumjeti logiku:

when Button.Click → izvrši radnju

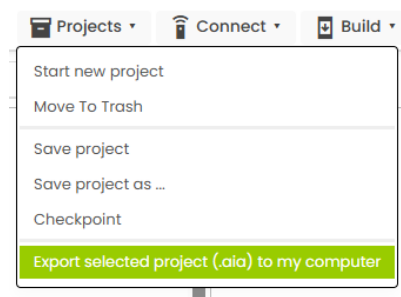
Distribucija programa drugim korisnicima

Kad u App Inventoru napravimo svoju mobilnu aplikaciju vjerojatno ju želimo podijeliti s drugim korisnicima. Za početak treba napraviti verziju programa koja se može instalirati na drugom mobitelu. Za to ćemo odabrati naredbu App (save .apk to my computer) iz izbornika Build.



Nakon nekog vremena datoteka s nazivom projekta i ekstenzijom .apk preuzet će se na naše računalo. Sada datoteku možemo određenoj osobi poslati npr. mailom ili objaviti na Play Store. Kod instaliranja aplikacije na mobitel možda će trebati dati neka dopuštenja za instalaciju (možda će trebati pogledati postavke mobitela).

U MIT App Inventoru koriste se dva formata datoteka: .aia i .apk, koji imaju različitu namjenu. Datoteka .aia predstavlja projekt (izvornu) datoteku aplikacije i sadrži sve elemente projekta – komponente iz Designer dijela, blokove programskog koda iz Blocks dijela te dodane medijske sadržaje. Ona služi za daljnje uređivanje, doradu i nadogradnju aplikacije te se može ponovno učitati u MIT App Inventor.



S druge strane, .apk je instalacijska datoteka namijenjena Android uređajima i predstavlja gotovu, izvršnu verziju aplikacije spremnu za instalaciju na mobilni uređaj.

Za razliku od .aia datoteke, .apk se ne može uređivati, već služi isključivo za korištenje aplikacije.

3. PLANIRANJE APLIKACIJE

Nastavnik objašnjava zadatak i dijeli upute.

Opis zadatka: Izradite aplikaciju koja omogućuje osnovnu komunikaciju između osoba s različitim potrebama.

Korisnik izgovara riječ ili rečenicu → aplikacija prepoznaje govor i:

- Prikazuje tekst na ekranu.
- Izgovara tekst naglas (TextToSpeech).
- Daje osjetilnu povratnu informaciju (vibracija/zvuk) kao potvrdu.

Scenarij inkluzije:

- Osobe koje ne čuju mogu vidjeti tekst na ekranu.
- Osobe koje ne vide mogu čuti izgovor i osjetiti vibraciju kao potvrdu unosa.
- Time aplikacija postaje most komunikacije i primjer inkluzivne tehnologije.

Napomene za pomoć:

U Designer dijelu dodajte:

- Button za pokretanje snimanja govora.
- Label za prikaz prepoznatog teksta.
- Komponente SpeechRecognizer, TextToSpeech i Sound (ili Vibrate).

U Blocks dijelu povežite:

- kada korisnik klikne gumb → pokreni SpeechRecognizer
- nakon što se govor prepozna → prikaži rezultat u Label, izgovori ga pomoću TextToSpeech i pokreni vibraciju/zvuk.
- Možete dodati ListPicker s izborom jezika za TextToSpeech (npr. engleski, njemački, hrvatski)
- Aktivirajte Caption Mode za automatski veliki tekst (npr. LabelResult.FontSize = 32)
- Koristite “tamnu temu” radi boljeg kontrasta i pristupačnosti
- Povratna funkcionalnost: dugo držanje gumba može pokrenuti “ponovi izgovor”

Rad predajete u formatu .aia i .apk



Učenici u parovima definiraju:

- Naziv aplikacije
- Ciljne korisnike
- Minimalne funkcionalnosti
- Dodatne funkcionalnosti

Raspravljaju:

- Kako osigurati čitljivost?
- Koja veličina fonta je prikladna?
- Treba li tamna tema?
- Treba li izbor jezika?

Nastavnik usmjerava učenike da razmišljaju o pristupačnosti, a ne samo o funkcionalnosti.

4. IZRADA APLIKACIJE – DESIGNER DIO

Učenici započinju izradu korisničkog sučelja aplikacije u Designer dijelu MIT App Inventora. Cilj je oblikovati jednostavno, pregledno i funkcionalno sučelje koje će biti prilagođeno korisnicima s različitim komunikacijskim potrebama.

Učenici dodaju:

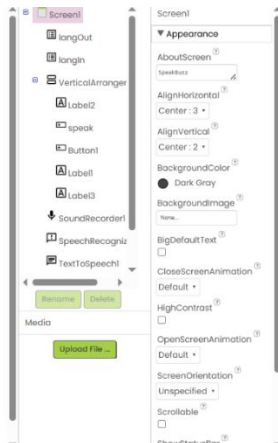
- Button (Start Listening)
- Label (prikaz rezultata)
- SpeechRecognizer
- TextToSpeech
- Sound ili Vibrate

Opcionalno:

- ListPicker (odabir jezika)
- Caption Mode (povećanje fonta)

Učenici najprije dodaju Button koji služi za pokretanje prepoznavanja govora (npr. naziv „Start Listening“). Nastavnik ih potiče na razmišljanje o veličini gumba, kontrastu boja i njegovom položaju na zaslonu kako bi bio lako uočljiv i dostupan svim korisnicima. Zatim dodaju Label komponentu koja će služiti za prikaz prepoznatog teksta. Pažnju posvećuju veličini fonta, poravnanju teksta i čitljivosti, budući je ova komponenta važna za korisnike s oštećenjem sluha. Potom iz kategorije Media dodaju komponente SpeechRecognizer i TextToSpeech, koje omogućuju primjenu umjetne inteligencije u aplikaciji, te komponentu Sound ili koriste funkciju vibracije radi dodatne osjetilne povratne informacije. Opcionalno mogu dodati i ListPicker za odabir jezika te implementirati Caption Mode, kojim se povećava veličina fonta radi bolje čitljivosti. Ove opcije predstavljaju dodatnu razinu pristupačnosti i prilagodbe korisniku.

Tijekom rada nastavnik obilazi učenike, pomaže im u organizaciji elemenata na zaslonu, upozorava na važnost jednostavnog dizajna te potiče razmišljanje o tome kako će aplikacija izgledati iz perspektive osobe s poteškoćama.



4.1 IZRADA APLIKACIJE – BLOCKS DIO

Nakon oblikovanja sučelja učenici prelaze u Blocks dio, gdje definiraju ponašanje aplikacije. Ovdje povezuju korisničke radnje s odgovarajućim funkcijama aplikacije. Osnovna logika aplikacije temelji se na dva ključna događaja. Kada korisnik klikne na gumb (when Button.Click), aplikacija pokreće komponentu SpeechRecognizer.Start, čime započinje proces prepoznavanja govora. Istovremeno se može aktivirati vibracija ili zvučni signal kako bi korisnik dobio potvrdu da je snimanje započelo.

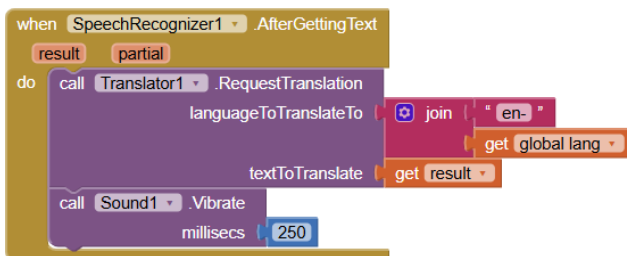
when Button.Click

- SpeechRecognizer.Start
- Vibrate

Nakon što sustav prepozna govor, aktivira se događaj when SpeechRecognizer.AfterGettingText. U tom trenutku prepoznati tekst (result) postavlja se u Label komponentu, čime postaje vidljiv na zaslonu. Zatim se isti tekst proslijeđuje komponenti TextToSpeech.Speak, koja ga izgovara naglas, a aplikacija ponovno daje vibracijsku potvrdu uspješno izvršene radnje.

when SpeechRecognizer.AfterGettingText

- set Label.Text to result
- TextToSpeech.Speak result
- Vibrate



Nastavnik objašnjava da komponenta SpeechRecognizer koristi umjetnu inteligenciju za pretvaranje govora u tekst, dok komponenta TextToSpeech koristi algoritme za generiranje govora iz teksta. Time učenici razumiju da njihova aplikacija ne sadrži samo klasične programske naredbe, već integrira AI funkcionalnosti koje omogućuju naprednu obradu podataka.

U ovoj fazi učenici uočavaju povezanost između događaja (klik gumba), obrade podataka (prepoznavanje i generiranje govora) i povratne informacije korisniku (tekst, zvuk, vibracija), čime se ostvaruje cjeloviti komunikacijski ciklus aplikacije.

5. PRIMJER GOTOVE APLIKACIJE

Nastavnik prikazuje nekoliko dovršenih rješenja..

Primjer_1

Ovaj primjer prikazuje osnovnu funkcionalnu verziju aplikacije za inkluzivnu komunikaciju. Logika rada temelji se na tri događaja. Kada korisnik klikne gumb (when Button1.Click), pokreće se komponenta SpeechRecognizer1.GetText, čime aplikacija započinje prepoznavanje govora. Nakon što sustav prepozna izgovoreni tekst, aktivira se događaj SpeechRecognizer1.AfterGettingText. U tom trenutku prepoznati tekst (Result) prikazuje se u oznaci

(Label1.Text), zatim se isti tekst izgovara naglas pomoću komponente TextToSpeech1.Speak, a aplikacija daje dodatnu osjetilnu povratnu informaciju putem vibracije u trajanju od 500 milisekundi. Dodatna funkcionalnost implementirana je kroz događaj when Button1.LongClick. Dugim pritiskom na gumb korisnik može ponovno čuti zadnji prepoznati tekst bez ponovnog snimanja govora. I u tom slučaju aktivira se TextToSpeech te vibracija kao potvrda radnje. Ovaj primjer predstavlja minimalnu, ali potpuno funkcionalnu verziju aplikacije koja ostvaruje inkluzivni scenarij: osoba koja ne čuje može pročitati tekst na ekranu, osoba koja ne vidi može čuti izgovoreni sadržaj, vibracija služi kao potvrda uspješno izvršene radnje.

Primjer je didaktički vrlo pogodan jer jasno prikazuje osnovni princip rada: pokretanje događaja → obrada rezultata → višestruka povratna informacija (vizualna i auditivna).

Primjer_2

Ovaj primjer predstavlja nadograđenu verziju osnovne aplikacije za inkluzivnu komunikaciju, u kojoj su uz osnovne funkcionalnosti dodani elementi vizualne prilagodbe sučelja. Kada korisnik klikne na Button1, pokreće se komponenta SpeechRecognizer1.GetText, čime započinje proces prepoznavanja govora. Nakon što se govor obradi, aktivira se događaj SpeechRecognizer1.AfterGettingText. U tom trenutku prepoznati tekst prikazuje se u Label1, zatim se izgovara pomoću komponente TextToSpeech1, a aplikacija daje vibracijsku povratnu informaciju u trajanju od 400 milisekundi. Uz to, dolazi do vizualne promjene – mijenja se boja pozadine zaslona (Screen1.BackgroundColor) i povećava veličina fonta oznake (Label1.FontSize), čime se dodatno naglašava da je radnja uspješno izvršena. Drugi gumb (Button2) služi kao dodatna funkcionalnost. Njegovim pritiskom ponovno se aktivira TextToSpeech te vibracija, a istovremeno se prilagođava vizualni prikaz – mijenja se boja pozadine zaslona i povećava veličina fonta u tekstualnom okviru (TextBox1.FontSize = 32). Time se omogućuje bolja čitljivost sadržaja, što je posebno važno za korisnike s poteškoćama vida. Ovaj primjer pokazuje višu razinu prilagodbe jer kombinira: prepoznavanje govora, pretvorbu teksta u govor, vibracijsku potvrdu, dinamičku promjenu vizualnih elemenata (boja i veličina fonta). U odnosu na osnovnu verziju, ovdje se jasnije vidi kako aplikacija može reagirati na korisničku radnju ne samo funkcionalno, već i vizualno, čime se povećava razina inkluzivnosti.

Primjer_3

Rješenje uključuje:

- izbor jezika
- prijevod teksta
- promjenu veličine fonta (Caption Mode)
- vibraciju kao potvrdu
- dugotrajni pritisak za dodatne opcije

```
when Translator1 .GotTranslation
  responseCode  translation
do
  set TextToSpeech1 .Language to get global lang
  call TextToSpeech1 .Speak
    message get translation
  call Sound1 .Vibrate
    millisecs 250
  set Label1 .Text to get translation
```

```

when speak .Click
do
  call SoundRecorder1 .Start
  call SpeechRecognizer1 .GetText
  call Sound1 .Vibrate
  milliseconds 250

when speak .LongClick
do
  call langOut .Open

initialize global lang to "en"

when langOut .AfterPicking
do
  set global lang to langOut .Selection
  set Label2 .Text to join get global lang

when Button1 .Click
do
  if get global but = 1
  then
    set Label1 .FontSize to 30
    set Label1 .BackgroundColor to black
  else
    set Label1 .FontSize to 15
    set Label1 .BackgroundColor to make color make a list 0 0 0 0
  set global but to get global but × -1

```

Primjer_3 predstavlja naprednu i funkcionalno složeniju verziju inkluzivne aplikacije za komunikaciju, koja uz osnovno prepoznavanje i izgovor teksta uključuje i mogućnost prijevoda, odabir jezika te prilagodbu prikaza sadržaja. Na početku se inicijalizira globalna varijabla jezika (npr. „en“), a korisnik ga može promijeniti putem ListPicker komponente, čime se nova vrijednost sprema i koristi u daljnjoj obradi teksta. Klikom na glavni gumb pokreće se proces prepoznavanja govora (SpeechRecognizer), uz zvučnu i vibracijsku potvrdu početka rada, dok dugim pritiskom korisnik može otvoriti izbornik za odabir jezika, čime se ostvaruje dodatna razina funkcionalnosti kroz različite tipove korisničke interakcije. Nakon prepoznavanja govora, dobiveni tekst šalje se komponenti Translator, gdje se prevodi na odabrani jezik, a po primitku rezultata postavlja se odgovarajući jezik u komponenti TextToSpeech te se prevedeni tekst izgovara naglas i prikazuje na zaslonu. Time aplikacija ostvaruje cjelovit komunikacijski tijek: govor → prijevod → prikaz → izgovor. Dodatno je implementirana mogućnost prilagodbe veličine fonta i izgleda sučelja (tzv. Caption Mode) pomoću globalne varijable i uvjetne logike, čime se omogućuje lakša čitljivost za osobe s

oštećenjem vida te jasna vizualna potvrda promjene načina rada. Ovaj model objedinjuje rad s globalnim varijablama, više događaja, integraciju više AI komponenti te prilagodbu korisničkog sučelja, čime predstavlja najvišu razinu složenosti rješenja i primjer napredne, inkluzivne mobilne aplikacije.

Učenici analiziraju:

Koje su dodatne funkcionalnosti?

Kako je aplikacija unaprijeđena?

Što bi oni dodali?

Naglašava se da su ovo samo neki od mogućih primjera, a ne jedina rješenja.

6. TESTIRANJE I DORADA

U ovoj fazi učenici sustavno testiraju izrađene aplikacije u različitim scenarijima uporabe. Testiranje se provodi radi provjere tehničke ispravnosti, kao i radi procjene funkcionalnosti, jasnoće sučelja i razine pristupačnosti. Učenici ispituju kako aplikacija reagira kada korisnik govori tiho, brzo ili nerazgovijetno te analiziraju koliko je prepoznavanje govora točno i pouzdano. Provjeravaju reagira li aplikacija jednako uspješno pri promjeni jezika, uključivanju ili isključivanju caption mode-a te pri korištenju dugog i kratkog pritiska gumba. Provjeravaju čitljivost teksta (veličina fonta, kontrast, boja pozadine) i jasnoću povratnih informacija (zvuk, vibracija).

Učenici testiraju aplikaciju u različitim scenarijima:

- govor tiho
- govor brzo
- promjena jezika
- uključivanje Caption Mode-a

Tijekom testiranja bilježe uočene nedostatke, poput pogrešnog prepoznavanja govora, kašnjenja u obradi teksta, neadekvatne veličine fonta ili nedovoljno jasnih vizualnih promjena. Nakon toga pristupaju doradi aplikacije te prilagođavaju veličinu fonta, trajanje vibracije, raspored elemenata ili pojednostavljaju sučelje kako bi poboljšali korisničko iskustvo.

Bilježe uočene probleme/nedostatke:

- pogrešno prepoznavanje
- kašnjenje
- neprikladna veličina fonta

Ova faza omogućuje učenicima razumijevanje da razvoj aplikacije nije jednokratan proces, već ciklus planiranja, izrade, testiranja i unaprjeđenja. Učenici uočavaju da kvaliteta digitalnog rješenja ovisi o kontinuiranom poboljšavanju i prilagodbi stvarnim potrebama korisnika, osobito kada je riječ o inkluzivnim tehnologijama.

7. ZAVRŠNA REFLEKSIJA

U završnom dijelu projekta učenici se osvrću na izrađene aplikacije, proces rada i naučene sadržaje. Refleksija se provodi kroz vođeni razgovor, individualno promišljanje i kratku raspravu u razredu. Nastavnik potiče učenike da razmišljaju o tehničkoj izvedbi kao i o svrsi onoga što su izrađivali.

Pitanja za učenike:

Kako ova aplikacija može pomoći osobama s poteškoćama?

Učenici prepoznaju da aplikacija može poslužiti kao most komunikacije između osoba koje ne čuju i osoba koje ne vide. Osobe s oštećenjem sluha mogu pročitati izgovoreni tekst na zaslonu, dok osobe s oštećenjem vida mogu čuti sadržaj putem pretvorbe teksta u govor i dobiti vibracijsku potvrdu. Učenici uočavaju da kombinacija vizualnih i auditivnih elemenata povećava pristupačnost i omogućuje veću samostalnost korisnika.

Je li teško razvijati inkluzivna rješenja?

Učenici zaključuju da tehnička izvedba nije jedini izazov. Potrebno je razmišljati o korisnicima, njihovim potrebama, čitljivosti, kontrastu, jednostavnosti sučelja i jasnoći funkcionalnosti. Shvaćaju da inkluzivni dizajn zahtijeva empatiju, planiranje i testiranje, a ne samo programiranje blokova.

Što biste poboljšali?

Neka od mogućnih poboljšanja/nadogradnje...

- dodavanja više jezika
- spremanja povijesti razgovora
- poboljšanja točnosti prepoznavanja govora
- dodatnih opcija prilagodbe (veličina fonta, tamna tema)
- jednostavnijeg sučelja za starije korisnike

Kroz projekt učenici su naučili:

- kako funkcioniraju komponente za prepoznavanje govora i pretvorbu teksta u govor
- razliku između Designer i Blocks dijela
- kako povezati događaje i logiku rada aplikacije
- kako tehničko rješenje prilagoditi stvarnim potrebama korisnika

Vrednovanje aplikacije temelji se na:

- funkcionalnosti aplikacije (radi li prepoznavanje i izgovor)
- jasnoći i pristupačnosti sučelja
- primjeni inkluzivnog scenarija
- logičkoj povezanosti blokova
- urednosti i dokumentaciji (.aia i .apk)

Učenici sudjeluju i u samovrednovanju kroz završnu refleksiju.

Projektna aktivnost dokazuje da tehnologija sama po sebi nije ni dobra ni loša te da njezin utjecaj ovisi o načinu na koji je koristimo i svrsi kojoj služi. Jednako tako, programiranje nije samo tehničko znanje, već i društvena odgovornost. Razvijanje aplikacije za inkluzivnu komunikaciju primjer je koji pokazuje kako digitalna rješenja mogu doprinijeti jednakim mogućnostima i smanjenju komunikacijskih prepreka u društvu.