

Inovativna uporaba GUI-tutorja pri projektih iz robotike in elektronike v pedagoškem kontekstu

David Rihtaršič

Univerza v Ljubljani

david.rihtarsic@pef.uni-lj.si

Rok Gabrovšek

Univerza v Ljubljani

Rok.Gabrovsek@pef.uni-lj.si

Prispevek obravnava inovativno uporabo generativne umetne inteligence (GUI) pri predmetih Robotika in Elektronika na Pedagoški fakulteti Univerze v Ljubljani. V okviru obeh predmetov, ki ju obiskujejo študenti dvopredmetnega študija s smerjo vezave s tehniko, smo v zimskem semestru 2024/2025 uvedli nov pristop k samostojnemu delu študentov pri izdelavi končnih projektov. Uporabljen je bil prilagojen GUI-agent – tutor, nadgrajen z didaktičnimi priporočili (aktivno učenje, Kolbov učni cikel) in opremljen z literaturo, ki je bila obravnavana pri predavanjih. Za razliko od splošnih generativnih modelov je agent upošteval razpoložljivo opremo študentov ter tehnične specifike projektov. Tako je zagotavljal specifične in relevantne odgovore, prilagojene konkretni situaciji. Namen sistema je bil usmeriti študente skozi individualizirano učno pot, pri čemer je agent s svojimi odgovori nadomeščal učiteljevo prisotnost, kadar je bil ta zaseden. Zbrana vprašanja in odgovori, ki so nastali med interakcijami GUI-agenta – tutorja s študenti, bodo služili za nadaljnji razvoj agenta in izboljšanje njegove pedagoške učinkovitosti.

Ključne besede: generativna umetna inteligenca, GUI-tutor, personalizirano učenje, projekti iz elektronike in robotike, Kolbov učni cikel

 © 2025 David Rihtaršič in Rok Gabrovšek

<https://doi.org/10.26493/978-961-293-431-6.13>

Uvod

Umetna inteligenca (UI) ima vse večjo vlogo v izobraževanju, saj omogoča personalizirano podporo študentom pri učenju, kar je še posebej pomembno pri tehničnih predmetih, kjer je treba združiti teoretične osnove in praktične veščine. Tehnologija GPT (angl. *Generative Pre-trained Transformer*) uporablja metode UI za učinkovito tvorjenje besedila. Različni modeli te tehnologije (npr. 01, GPT-4o, GPT-3.5 ...) ponujajo priložnost za zagotavljanje interaktivne podpore študentom, vendar njihova splošna zasnova pogosto ne ustreza specifičnim potrebam tehničnih projektov.

Pregled dostopnih podatkov na področju raziskav, ki povezujejo izobraževanje in modele generativne umetne inteligence (GUI), nam daje nekaj zani-

mivih iztočnic. Dosedanje raziskave so zaznale predvsem pozitivno mnenje o orodjih GUI, zlasti v visokošolskem izobraževanju in disciplinah STEM (iz angl. *Science Technology Engineering and Math*). Ti področji sta v vseh raziskavah največkrat omenjeni. Navdušenje nad aplikacijami GUI, kot so ChatGPT, Google Bard, Bing Chat in Claude, kaže na to, da je skepticizem, ki se je pojavljal, ko so se generativni modeli pojavili prvič, premagan in da je izobraževalna skupnost pripravljena sprejeti tehnologije, ki obetajo preobrazbo izobraževanja. Večina raziskav je uporabljala kvalitativne metode za globlji vpogled v integracijo GUI v izobraževanje (Jahic idr., 2024). Med pregledom opazimo pomanjkanje podatkov za nekatera področja. Tako sklepamo, da je implementacija GUI v osnovnih šolah bistveno manjša v primerjavi z višjimi stopnjami izobraževanja (Fatimah idr., 2024), kar odpira številne možnosti za nadaljnje aktivnosti na tem področju. Primanjkuje tudi kvantitativnih raziskav, še posebej takih, ki bi podale odgovor na to, kako uporaba GUI vpliva na učne dosežke učencev, ki uporabljajo GUI (Jahic idr., 2024).

Rezultati trenutnih raziskav o tem, kako uporaba GUI vpliva na učne dosežke učencev, so zelo deljeni. Za primer si pogledjmo nekaj kontradiktornih eksperimentov. Qureshi (2023) opisuje eksperiment, ki je vključeval dve skupini študentov računalništva, ki sta morali rešiti naloge iz programiranja. Kontrolna skupina je lahko za pomoč pri nalogah uporabljala le učbenike in zapiske, eksperimentalna skupina pa tudi ChatGPT. Naloge so bile sestavljene tako, da so ocenjevale znanje uporabe algoritmov in podatkovnih struktur. Skupine so tekmoval v reševanju nalog z različnimi zahtevnostmi, merili pa so število uspešno opravljenih nalog in čas reševanja. Eksperimentalna skupina, ki je uporabljala ChatGPT, je dosegla boljše povprečne rezultate pri vseh nalogah v primerjavi s kontrolno skupino, največje razlike so bile opazne pri lažjih nalogah. Pri zahtevnejših nalogah se je prednost zmanjšala, saj so člani eksperimentalne skupine potrebovali več časa za odpravljanje napak pri kodi, ki jo je ustvaril ChatGPT. Izkazalo se je, da sama uporaba ChatGPT ni zagotovila trajnega prenosa znanja, ki bi ga razvili s tradicionalnimi metodami in samostojnim učenjem (Qureshi, 2023).

V nekem drugem eksperimentu so Weeks idr. (2024) analizirali vpliv uporabe modelov GUI na akademsko uspešnost študentov ekonomije in podjetništva v uvodnem predmetu računovodstva. Študente so razdelili v dve skupini, ena je uporabljala orodja GUI, druga pa ne. Uporabnike so identificirali s pomočjo orodja ZeroGPT za zaznavanje GUI v oddanem delu. Rezultati so pokazali, da so uporabniki GUI na izpitih v povprečju dosegali za 6,71 točke (vseh možnih točk je bilo 96,67) nižje rezultate kot neuporabniki, zaradi česar menijo, da se je pokazal zaviralni učinek uporabe GUI pri globljem učenju.

Hamsa Bastani idr. (2024) v eksperimentu, izvedenem med srednješolci, analizira vpliv GUI, specifično GPT-4, na učenje matematike v srednji šoli. Učenci so bili razdeljeni v tri skupine: kontrolno skupino brez dostopa do GUI, skupino z dostopom do GPT-4 v obliki običajnega pogovornega modela (splošno dostopnega) in skupino z dostopom do prilagojenega modela GPT Tutor, ki je učence do rešitve popeljal po korakih s sprotnimi namigi. Med vajami sta skupini z dostopom do GUI dosegli boljše rezultate, učenci z dostopom do GPT Tutorja so celo za 127 % bolje reševali vaje od kontrolne skupine. Na končnem izpitu nobena izmed skupin ni imela dostopa do GUI. Skupina, ki je uporabljala splošen model, je dosegla za 17 % slabše rezultate od ostalih dveh. Rezultati kažejo na tveganje, ki ga prinaša prekomerno zanašanje na GUI.

Sklepamo lahko, da GUI izboljša znanje in učne dosežke učencev, vendar pod določenimi pogoji. Pogoji za njen pozitiven vpliv vključujejo ustrezno pedagoško podporo, ki GUI usmerja kot orodje za ustvarjanje znanja namesto zgolj porabe informacij. Izobraževalni sistem mora ustvariti okolje, ki spodbuja etično in premišljeno uporabo GUI ter študente usposobi za samostojno učenje in kritično ocenjevanje virov, kar pomaga preprečevati zlorabe in povečati avtonomijo pri učenju (Saude idr., 2024). Čeprav se raziskovalci še ne strinjajo povsem o učinkih GUI na pridobljeno znanje, pa je več strinjanja o tem, da ima GUI pozitivne vplive na drugih področjih izobraževanja. Visokošolski učitelji, ki so se ukvarjali s spletnimi tečaji, so v večini potrdili pozitivne vplive GUI tako z vidika učinkovitosti učiteljev kot dela študentov (Kotsiovos idr., 2024). Ena glavnih prednosti GUI je prilagoditev povratnih informacij za vsakega posameznika, zaradi česar ima široko uporabnost v izobraževanju in omogoča podporo tako učencem kot učiteljem. Kot učni pomočnik običajno deluje v obliki pogovornih agentov oz. »chatbotov«, ki učence podpirajo pri učenju in opravljanju domačih nalog. Orodja GUI so uporabna pri branju, računanju in pisanju, predlagajo izboljšave za slovnico in slog, povzamejo zahtevnejša besedila ter spodbujajo kritično razmišljanje. Poleg tega GUI omogoča ustvarjanje prilagojenih učnih poti, kar zagotavlja unikatno učno izkušnjo, prilagojeno sposobnostim in tempu učenca. Kot inteligentni tutor prepozna vrzeli v znanju učencev in ponuja prilagojene razlage ter dodatne naloge, s čimer spodbuja globlje razumevanje snovi in podpira individualne učne potrebe. Pomembna prednost je tudi, da je vedno dostopen, kar pomeni, da učenci lahko dobijo odgovor takoj, ko se jim pojavi vprašanje. Vedno nepogrešljivejši je tudi za učitelje. Omogoča jim, da lažje pripravijo učne materiale, kot so vaje, predstavitve ali kvizi, ki so prilagojeni različnim stopnjam in potrebam študentov. Učitelji lahko GUI uporabijo za generiranje več verzij učnih načrtov, ustvarjanje razprav o določeni temi ali generiranje dodatnih

vaj. Na ta način omogoča boljšo pripravo na pouk in zmanjšuje čas, ki ga učitelji namenijo administrativnim nalogam, pregledom nalog, ocenjevanju in podajanju povratnih informacij (Rashid idr., 2024).

GUI torej omogoča bolj personalizirano, poglobljeno in dostopnejše učenje za vse učence, hkrati pa zmanjšuje obremenitve učiteljev in prinaša spremembe v učenje in poučevanje na vseh ravneh izobraževanja. Pri predmetih, kot sta robotika in elektronika, se študentje pogosto srečujejo z ovirami, ki jih sami težko premagajo, čeprav so te včasih banalne narave. Npr., pri pisanju programske kode v jeziku C++ lahko že pozabljeni znak »;« za študenta povzroči nepremostljivo težavo. Taka napaka za izkušenega učitelja predstavlja dokaj enostaven poseg, ki pa študentu ni na voljo v vsakem trenutku. Takšne ovire ne le upočasnijo delo, temveč tudi zmanjšujejo motivacijo študentov, saj brez sprotne podpore težko napredujejo. Z vidika izobraževanja s področij STEM omenimo še, da uporaba GUI učence uvaja v najsodobnejše tehnologije in jih opremlja s ključnimi veščinami za reševanje kompleksnih realnih izzivov, s čimer krepí njihovo inovativnost in zmožnost za reševanje problemov (Rashid idr., 2024). Napisano se dobro povezuje z ugotovitvijo, da predmeti STEM učencev ne pripravljajo zgolj na specifične poklice, temveč naj bi jih opremljali z veščinami, znanjem in miselnostjo, ki so ključni za osebni uspeh ter napredek družbe. Področja STEM igrajo eno osrednjih vlog pri oblikovanju prihodnosti, saj se soočajo z globalnimi izzivi in ustvarjajo inovacije, ki prinašajo rešitve za boljši svet (Marzuki idr., 2024).

Raziskava na področju robotike je pokazala, da je uporaba UI pri učenju robotike izboljšala učne rezultate za 20 %. Poleg tega so se pri učencih izboljšale tudi sposobnosti reševanja kompleksnih problemov ter razumevanje inženirskih in programskih konceptov, povezanih z robotiko (Chipantiza idr., 2024).

Informacije o popularnosti GUI v STEM-izobraževanju in pri dosežkih v poučevanju robotike kažejo, da bi GUI lahko prinašala korist na tem področju. Predvsem so se za zanimive izkazali pogovorni agenti, ki bi učence vodili pri projektnem delu, kar pri robotiki predstavlja velik del učnega procesa. Agenti za pogovor so aplikacije, tisti virtualni roboti, ki se danes pojavljajo na številnih spletnih straneh, da nam pomagajo najti, kar potrebujemo. Agent je v osnovi aplikacija, ki komunicira z ljudmi z uporabo njihovega naravnega jezika skladno z informacijami, ki jih vsebuje baza podatkov, in modulom UI, ki interpretira vprašanja in išče odgovore. Na podlagi tega osnovnega shematskega modela lahko gradimo tako zapletenejše sisteme kakor tudi preprostejše. Zapletenejši sistemi se učijo iz interakcij. Uporaba agentov v izobraževanju ni nekaj novega, nekateri so jih uporabljali že na začetku tega stoletja

(Paredes Rizo, 2021). Zagotovo pa so v zadnjem obdobju dobili nov zagon z razvojem GUI.

Povezava med UI in robotiko ne ponuja zgolj priložnosti za pridobivanje naprednih tehnoloških spretnosti, temveč spodbuja tudi razvoj kognitivnih, tehničnih, socialnih in čustvenih kompetenc pri učencih. Raziskovalci iz Ekvadorja so pregledali različne raziskave in predstavili načine za integracijo UI v poučevanje robotike. Poudarjajo, da vključitev agentov v izobraževalno robotiko odpira nove priložnosti za izboljšanje dostopnosti, učinkovitosti in učne izkušnje, saj učencem omogoča interaktivno in praktično raziskovanje ter razumevanje zapletenih konceptov. V tem kontekstu agent ne deluje le kot virtualni pomočnik, temveč tudi kot personalizirani učni partner in tutor, ki pomembno prispeva k usposabljanju nove generacije navdušencev in strokovnjakov za robotiko. Uporaba ChatGPT, ene izmed najbolj razširjenih platform GUI, poenostavi iskanje informacij, odpravi dvome ter učence usmerja pri pridobivanju znanja iz robotike. To orodje učencem omogoča jasne in jedrnat razlage temeljnih konceptov, reševanje zapletenih problemov in raziskovanje novih idej na področju robotike. Pri praktičnih projektih ChatGPT deluje kot podporno orodje, ki ponuja namige, nasvete in ustvarjalne rešitve. S sposobnostjo razumevanja učenčevih sporočil se prilagodi potrebam vsakega posameznika ter s tem zagotavlja prilagojeno in učinkovito učenje (Chipantiza idr., 2024).

Nekateri raziskovalci, med drugim Wojciechowski idr. (2023) ter Tlili idr. (2023), so izrazili skrb glede zanesljivosti odgovorov generativnih modelov UI, saj ti pogosto podajajo zavajajoče ali celo lažne informacije. Zavedati se moramo, da informacije, ki jih poda GUI, niso vedno pravilne, kar lahko privede do napačnega razumevanja ali celo napačnih zaključkov. Ta tveganja še posebej izstopajo pri uporabi GUI kot tutorjev, kjer je natančnost ključnega pomena za uspeh učnega procesa.

Pri projektnem delu lahko preverjanje delovanja (ali nedelovanja) posameznega sklopa projekta v fizični obliki služi kot način za validacijo pravilnosti namigov GUI-tutorja. To učencem omogoča, da aktivno preverjajo pridobljene informacije in razvijajo razumevanje skozi praktično delo. Vendar pa GUI prinaša tudi tveganje, da pri učencih pride do pretiranega zanašanja na tehnologijo, kar zmanjšuje njihovo sposobnost samostojnega reševanja težav. Namesto da bi se osredotočili na analizo problema in iskanje lastnih rešitev, se lahko preveč zanašajo na hitro dostopne odgovore. Takšno vedenje lahko negativno vpliva ne le na pridobljeno znanje, temveč tudi na razvoj ustvarjalnosti in kritičnega mišljenja, ki sta ključni veščini pri reševanju kompleksnih problemov. Da bi bil GUI čim učinkovitejši, je nujno, da učenci

in učitelji razvijejo kompetence, kot sta digitalna pismenost ter sposobnost preverjanja informacij iz različnih virov (Kharrufa in Johnson, 2024).

Uporaba GUI mora biti premišljena in uravnotežena, saj lahko le tako prispeva k izboljšanju izobraževanja. Hkrati je treba paziti, da se izogne negativnim vplivom, kot sta zmanjšanje ustvarjalnosti in oblikovanje napačnih prepričanj (Tlili idr., 2023).

Glede na trenutni teoretični pregled lahko ugotovimo, da osnovnemu orodju ChatGPT primanjkuje didaktičnih in specifičnih navodil za izvedbo različnih projektov. Zato je bil cilj te raziskave razviti in oceniti prilagojenega GUI-tutorja, ki bi nudil hitro in specifično pomoč študentom pri projektih iz robotike. Ključno raziskovalno vprašanje, ki ga naslavljamo, je: Kako lahko izboljšamo izkušnjo samostojnega učenja s prilagojenimi GUI-tutorji?

Metodologija

Raziskava je bila osredotočena na uporabo GUI-tutorja pri projektih iz robotike. Njen namen je bil preučiti, kako lahko prilagojeni GUI-tutor podpira študente pri samostojnem reševanju tehničnih izzivov in prispeva k njihovemu učnemu procesu. Raziskava je vključevala uvedbo GUI-tutorja v praktični del predmeta Robotika, ki je bil zasnovan kot projektno usmerjeno učenje.

Cilji raziskave so bili:

- preveriti uporabnost in učinkovitost GUI-tutorja pri podpori študentom pri tehničnih projektih;
- identificirati morebitne pomanjkljivosti GUI-tutorja in jih sproti nasloviti z ustreznimi prilagoditvami;
- zbrati povratne informacije od študentov o njihovem dožemanju GUI-tutorja, njegovi uporabnosti in vplivu na njihov učni proces.

Projekti iz robotike

Raziskava je potekala v okviru rednih laboratorijskih srečanj pri predmetu Robotika in je bila namenjena testiranju osnovnih funkcionalnosti GUI-tutorja ter ocenjevanju njegovega vpliva na samostojno delo študentov. Študenti so pred začetkom projektnega dela deset tednov obravnavali ključne vsebine preko izkustvenega učenja, kjer so spoznavali osnove programskega jezika C++, uporabo krmilnika Arduino UNO z modulom RobDuino, senzorje in aktuatorje ter osnovna načela delovanja mobilnih robotov.

GUI-tutorja smo vključili v zadnjih štirih tednih izvajanja predmeta, ko so študentje začeli z izdelavo svojih projektov. V tej fazi je bil tutor zasnovan tako, da je zagotavljal takojšnjo podporo pri reševanju tehničnih težav in raz-

jasnjevanju konceptov. Uporabljeni GUI-tutor je bil sproti prilagajan glede na povratne informacije, zbrane med interakcijami s študenti. S prilagoditvami smo poskušali izboljšati njegovo uporabnost, natančnost odgovorov in sposobnost vodenja študentov skozi učni proces.

Prilagoditev GUI-tutorja

GUI tutor (v nadaljevanju tudi tutor), uporabljen pri projektih iz robotike, je bil naš prvi namenski tutor, posebej zasnovan za podporo študentom pri tehničnih projektih. Zaradi inovativnosti pristopa in omejenega časa za razvoj je bil tutor zasnovan kot prilagodljivo orodje, ki smo ga sproti izboljševali na podlagi povratnih informacij in opažanj iz interakcij s študenti. Ključne prilagoditve so vključevale vsebinsko in didaktično obogatitev tutorja, da bi čim bolje ustrezal specifičnim potrebam predmeta Robotika. Izpostavimo lahko naslednje ključne prilagoditve s primeri navodil¹ v umaknjenih odstavkih:

Vsebinski dodatki: Tutor je bil opremljen z literaturo in gradivom, ki so ga študentje že obravnavali v uvodnem delu predmeta. To je zagotovilo, da so bili odgovori tutorja v skladu z obstoječim znanjem študentov in s tehničnimi zahtevami projekta. Takšen pristop je zmanjšal tveganje za zmedo zaradi neskladij med obravnavano vsebino in vsebino, ki jo najdemo drugod po spletu.

[V]sebino črpa iz knjige: *Experiential Learning of Robotics* in iz spletne strani: https://davidrihtarsic.github.io/Experiential_Learning_of_Robotics/. Pri podajanju primerov, razlag ali kodnih izsekov naj uporabi le komponente in materiale, ki so del učnega načrta [...].

Didaktična nadgradnja: Upoštevali smo pomanjkljivosti splošnega orodja ChatGPT, ki mu primanjkuje specifičnih didaktičnih navodil. Tutor je bil obogaten s teorijo izkustvenega učenja in z integracijo Kolbovega učnega cikla (KUC). Cilj je bil podpirati študente v štirih ključnih fazah učenja: pri konkretni izkušnji, refleksiji, konceptualizaciji in eksperimentiranju.

[P]omaga študentu pri izvedbi projekta iz robotike, s poudarkom na sledenju Kolbovem učnemu ciklu. Pomočnik vodi študenta skozi štiri ključne faze [...]. Vse faze naj bodo eksplicitno izpostavljene in očitne.

Struktura navodil: Navodila tutorja so bila zasnovana tako, da so študenta vodila korak za korakom pri reševanju problema. Najprej ga je tutor seznanil

¹ Primeri so del računalniškega programa AI tutor, ki je poleg ostale kode vseboval tudi vsebinsko-didaktična navodila, s katerimi smo ga prilagodili našim potrebam poučevanja. Ta navodila praktično pokažejo, kako lahko isto orodje prilagodimo svojim potrebam. Program je dostopen na: https://github.com/davidrihtarsic/AI_tutor

z enostavnim primerom, ki je služil kot izhodišče razumevanja bistvenih vsebin. Študent je bil nato spodbujen k eksperimentiranju (po KUC) na podlagi predlaganega primera, kar je omogočilo postopno usvajanje znanja in razvijanje veščin skozi praktično delo.

Odgovori naj bodo podani kot »minimal working example« zato, da jih študentje lažje razumejo in ne prihaja do »kognitivne preobremenitve«. Četudi uporabnik postavi kompleksnejše vprašanje, naj ga pomočnik razčleni na manjše podprobleme in vodi študenta pri reševanju vsakega posebej.

Preverjanje znanja: Na koncu vsakega sklopa je tutor študentu zastavil vprašanja, ki so bila zasnovana tako, da preverjajo razumevanje in sposobnost aplikacije naučenega znanja. Namen teh vprašanj je bil spodbuditi kritično razmišljanje in utrditi usvojeno snov.

[P]ostavi kompleksnejše vprašanje za preverjanje razumevanja, kot je to predvideno po Bloomovi taksonomiji. Vprašanje naj bo napisano v krepkem slogu, odgovori izbirnega tipa A, B, C in D pa naj bodo razvrščeni v seznam spodaj. Asistent naj počaka na odgovor študenta in naj njegov odgovor ovrednoti, po potrebi poda tudi povratno informacijo.

Postopek izvedbe

Tutorja smo preizkusili pri samostojnih projektih iz robotike, ki so bili zasnovani predvsem za gostujoče Erasmusove študente. Ti študentje predznanja s področja robotike niso imeli, kar je raziskavi dodalo poseben izziv: zagotoviti podporo, ki bi omogočila učinkovito delo tudi začetnikom. Po obravnavi osnovnih vsebin so študentje prejeli projektne naloge, zasnovane tako, da so vključevale ključne teme predmeta Robotika: načrtovanje in izdelavo preprostih senzorjev, pisanje programske kode za krmilnik, konstrukcijo mobilnega robota z modularno zbirko Fischertechnik ter povezovanje in uporabo ustreznih aktuatorjev.

Projektne naloge so obsegale enake osnovne vsebine, vendar so bile aplikacije nalog različne, da bi spodbudile kreativnost in prilagoditev specifičnim zahtevam:

- *Sumo robot:* Naloga je vključevala zasnovo in izdelavo robota, ki je moral prepoznati najbližji predmet v tekmovalnem ringu in ga izriniti. Projekt je zahteval uporabo senzorjev za zaznavanje razdalje ter algoritmov za usmerjanje gibanja.
- *Primitivni organizem:* Robot je moral najti najsvetlejšo točko v prostoru in se premakniti do nje na določeno razdaljo. Ko je to svetilo po nekaj

časa ugasnilo, je moral robot najti novo najsvetlejšo točko in ponoviti algoritme.

- *Robotski sesalnik*: Robot je moral naključno raziskovati prostor, prepoznati temnejša področja podlage in se tam zadržati dlje časa, hkrati pa obiti morebitne ovire.
- *Cestni čistilec*: Robot je sledil črti in s poti odstranjeval ovire, nato pa nadaljeval s sledenjem črti.
- *Reševalec labirinta*: Robot je navigiral po labirintu do označenega cilja s kombinacijo algoritmov za sledenje liniji in zaznavanje sten. Ko je našel označeno mesto v labirintu, pa se je lahko ustavil in prekinil z delovanjem.

Za izdelavo svojih projektov so imeli študentje na voljo štiri tedne po štiri šolske ure. Med celotnim obdobjem so lahko uporabljali ali pomoč učitelja, če jim je bil ta na voljo, ali aplikacijo za pogovor s tutorjem. Vsak teden smo pregledali in analizirali pogovore med študenti in tutorjem. Ta analiza je omogočila identifikacijo težav in potreb po prilagoditvah slednjega. Ključne izboljšave, izvedene med procesom, so vključevale:

- razčlenjevanje nalog: naloge so bile razdeljene na manjše, obvladljivejše sklope, kar je študentom omogočilo postopno reševanje problemov;
- vprašanja za samoevalvacijo: tutor je po zaključku vsakega sklopa študentom postavil vprašanja za preverjanje razumevanja, kar je spodbudilo kritično razmišljanje;
- usmerjanje v literaturo: dodana so bila usmerjena navodila, ki so študente opozorila na specifične vsebine v gradivu, obravnavanem med predavanji;
- prilagoditev uporabniškega vmesnika: tutorjeva komunikacija se je v začetnih tednih odvijala preko terminalnega okna, kar se je izkazalo za manj prijazno do uporabnikov; sredi projekta smo prešli na spletno aplikacijo, ki je omogočala preglednejšo interakcijo z uporabo označb »markdown« za bolj strukturirane odgovore; poleg tega je bila dodana funkcionalnost za učitelje, ki jim je omogočala sprotno spremljanje pogovorov med študenti in tutorjem – ta funkcija je omogočila hitrejše prepoznavanje morebitnih težav in usmerjanje študentov, kadar so se soočili z napačnimi razlagami ali koncepti.

Proces smo zaključili z vprašalnikom, kjer so študentje podali kvalitativno mnenje o uporabi tutorja GUI. Vprašalnik je zajemal oceno uporabnosti tutor-

ja, njegovih sposobnosti reševanja tehničnih težav in podpore pri učenju. Povratne informacije so služile kot osnova za nadaljnje izboljšave sistema in oceno njegovega vpliva na učni proces.

Zbiranje podatkov

Glavni vir podatkov so bili zapisi pogovorov med študenti in tutorjem. Na začetku raziskave, ko je tutor komuniciral preko terminalnega okna, so se pogovori shranjevali lokalno na računalnikih, ki so jih uporabljali posamezni študentje. Takšen pristop je bil sicer funkcionalen, vendar je oteževal centralizirano zbiranje podatkov in njihovo analizo.

Z uvedbo spletne aplikacije in prehodom na strežniško arhitekturo so se vsi pogovori začeli samodejno shranjevati na strežnik. Ta prilagoditev je učitelju omogočila sproten dostop do zapisov pogovorov v realnem času, kar je bistveno olajšalo spremljanje interakcij in omogočilo hitre prilagoditve tutorjevih odgovorov ter izboljšanje učnega procesa.

Poleg zapisov pogovorov smo ob zaključku učnega procesa študentom razdelili kvalitativni vprašalnik, ki je zajemal njihovo mnenje o uporabnosti tutorja, njegovih funkcionalnostih in sposobnosti podpore pri reševanju tehničnih izzivov. Podatki iz vprašalnika so bili zbrani in obdelani v Googlovih preglednicah, kar je omogočilo nadaljnjo analizo povratnih informacij in sintezo ključnih ugotovitev o učinkovitosti sistema.

Analiza podatkov

V analizi podatkov smo pregledali vse zbrane pogovore med tutorjem in študenti ter podatke iz končnega kvalitativnega vprašalnika. Zaradi majhnega števila študentov (pet študentov: dva iz Finske, eden iz Slovaške, eden iz Turčije in eden iz Srbije) smo lahko podrobno analizirali vse interakcije za vsak teden posebej in za vsakega študenta. Takšen pristop je omogočil identifikacijo vzorcev in ključnih izzivov, ki so se pojavljali med uporabo tutorja.

Na podlagi teh ugotovitev smo uvedli več ključnih sprememb, vključno s krajšanjem odgovorov tutorja, z razčlenjevanjem nalog na manjše enote in izboljšanjem uporabniškega vmesnika. Te prilagoditve so bile izvedene sproti med potekom projekta, kar je omogočilo takojšen odziv na opažene težave.

Analiza zbranih podatkov je omogočila sistematično identifikacijo izzivov, s katerimi so se študentje soočali, in prilagoditev tutorja za izboljšanje njegove uporabnosti. Kombinacija pregledov pogovorov in povratnih informacij iz vprašalnika je zagotovila celovit vpogled v učni proces in interakcijo z GUI-tutorjem, kar bo služilo kot osnova za nadaljnji razvoj sistema.

Rezultati

Analiza podatkov, zbranih v raziskavi, razkriva vzorce in ključne vidike uporabe tutorja pri projektih iz robotike. Rezultati temeljijo na pregledih pogovorov med študenti in tutorjem ter povratnih informacijah iz kvalitativnega vprašalnika. Ti podatki osvetljujejo, kako so študentje uporabljali tutorja, s kakšnimi izzivi so se soočali ter kako so prilagoditve tutorja vplivale na njihov učni proces in uspešnost pri projektih. V nadaljevanju so predstavljene ugotovitve glede interakcije med študenti in tutorjem ter drugi ključni rezultati raziskave.

Interakcije med študenti in tutorjem

Na podlagi analize pogovorov med študenti in tutorjem so bili opaženi naslednji vzorci, ki prikazujejo tako učinke kot izzive pri uporabi tutorja v projektu robotike:

- *Ambiciozen začetek vprašanj*: Študentje so pogosto začeli z vprašanji, ki so od tutorja pričakovali celovite rešitve že v prvem odgovoru, npr. »What codes do I need to make a robot vacuum?« To nakazuje začetno težnjo študentov, da iščejo neposredne rešitve brez lastne analize problema.
- *Slaba definicija vprašanj*: Pogosto zastavljena vprašanja, kot npr. »How do I turn on the 2 resistors?«, so bila premalo specifična ali celo napačna, nelogična. To kaže na nezadostno predznanje študentov, ki niso znali jasno opredeliti, kaj točno želijo od tutorja. Tako lahko celo ločimo študente po znanju. Npr., pri enem študentu, ki si s tutorjem niti ni pomagal do konca projekta, lahko že v prvih vprašanjih najdemo zelo ciljno usmerjena vprašanja oz. trditve: »I am using RobDuino modul and I wired the motors to D7, D6 and D5, D4.«
- *Izogibanje vprašanjem za samoevalvacijo*: Študentje so pogosto ignorirali tutorjeva vprašanja za preverjanje njihovega znanja; če jim je tutor npr. zastavil vprašanje »What do you think will happen if you set the PWM value to 255?«, nanj niso odgovorili, pa čeprav bi jim tutor še enkrat razložil vsebino. To nakazuje nizko motivacijo za analizo odgovorov in pridobivanje novega znanja ter preferiranje hitrega napredovanja k rešitvam. Nekateri so celo tutorja opremili z dodatnimi navodili, naj teh vprašanj niti ne postavlja: »Can you skip the questions at the end?«
- *Miselna preobremenitev zaradi kompleksnosti odgovorov*: Daljši odgovori, ki so vključevali več novih konceptov ali kompleksne programske

rešitve, so študente pogosto preobremenili in tega so se tudi zavedali: »This example is too complicated.« Študentje so zato kopirali le bistvene dele kode in jih neposredno vključili v projekt.

- *Pristop »copy-paste« k reševanju težav:* Študentje so pogosto izvajali tehniko »copy-paste« brez predhodnega analiziranja kode. Tako so naleteli na pogoste napake podvajanja imena funkcij, niso razumeli, na katero mesto naj novo kodo vgradijo ...
- *Personalizacija interakcije:* Erasmusovi študentje so izpostavili prednost uporabe tutorja v svojem maternem jeziku (npr. »Can you say that in Slovak?«), kar je izboljšalo njihovo razumevanje in omogočilo bolj personaliziran učni proces. Prav tako so si upali zastaviti vprašanja, ki bi jih pri profesorju sicer morda preskočili, ker so se zavedali, da gre za osnovna vprašanja, ki so jih na predavanjih obravnavali večkrat: »How can I measure the value from the IR sensor?«

Interakcija med študenti in tutorjem je bila večinoma takšna, kot bi jo pričakovali v komunikaciji med študenti in učiteljem. Študentje so pogosto iskali konkretne rešitve za svoje težave, postavljali osnovna vprašanja in se opirali na tutorjeve odgovore pri napredovanju v projektu. Ključna razlika pa je bila v zmanjšanju neposrednega bremena za učitelja, ki je v preteklih letih težko zagotavljal sprotno pomoč vsem študentom. V tem eksperimentu je tutor prevzel velik del podpornih nalog, kar je učitelju omogočilo, da je delo študentov spremljal v bolj opazovalni vlogi. Namesto da bi neprestano reševal težave in odgovarjal na vprašanja, je lahko učitelj prepoznal širše vzorce in se osredotočil na strateške prilagoditve učnega procesa. Ta sprememba je prispevala k učinkovitejšemu in manj stresnemu okolju tako za študente kot za učitelja.

Povratne informacije študentov

Študentje so tutorja v povprečju ocenili s 3,8 na lestvici od 1 do 6, pri čemer so se ocene gibale od najnižje, 1, do najvišje, 6, kar kaže na raznolikost izkušenj.

Ena izmed prednosti uporabe tutorja je bila možnost komunikacije v različnih jezikih, kar je bilo še posebej pomembno za skupino študentov iz različnih držav. Ta funkcionalnost jim je omogočila lažje razumevanje strokovnih izrazov v maternem jeziku in s tem boljše razumevanje nalog. Poleg tega so pohvalili takojšnjo odzivnost tutorja, kar je bilo še posebej koristno pri postavljanju preprostih vprašanj ali v situacijah, ko učitelj ni bil na voljo. Tutor jih je spodbujal k premisleku o reševanju težav in ponujal smernice za premaganje konkretnih izzivov. Študentje so bili prav tako zadovoljni z razlagami

ključnih konceptov, ki jih je podajal tutor in so pomembni za delovanje robota. Ti vključujejo načine povezovanja elementov in razumevanje programske kode. Podane razlage so jim pomagale bolje razumeti zahtevne tehnične vidike in jih uspešno uporabiti pri svojih projektih. Primeri pozitivnih izkušenj vključujejo študenta, ki je navedel, da mu je tutor pomagal napisati kodo za izogibanje oviram. Po prejetju osnovne kode jo je prilagodil in vključil svoje ideje, kar mu je omogočilo boljše razumevanje programske logike. Drug študent je omenil, da mu je tutor predstavil in razložil delovanje prej nepoznatih programskih funkcij, ki jih je uporabil za upočasnitev robota. Študentje so izpostavili še en ključen vidik, in sicer podajanje referenc iz študijskega gradiva in spletnih virov, kar jim je omogočilo boljše razumevanje kompleksnih tematik. Ta funkcionalnost je bila ocenjena kot ena najuporabnejših za uspešno samostojno delo.

Nekateri študentje so imeli težave z obsežnostjo odgovorov, saj so bili ti pogosto predolgi in nepregledni, kar je oteževalo iskanje bistva. Poleg tega so nekateri opozorili na nezadostno specifičnost odgovorov, kar je povzročilo frustracije pri reševanju nalog. Eden od študentov je izpostavil, da je tutor pogosto ponujal preveč informacij, ki niso bile neposredno uporabne za njegovo specifično vprašanje. Študent z negativno izkušnjo je tutorja ocenil z 1. Poudaril je, da je bila uporaba prezapletena in da ni imel časa za razumevanje dolgotrajnih odgovorov.

Kljub nekaterim kritikam je večina študentov tutorja ocenila kot koristno orodje, ki je izboljšalo njihovo samostojnost in razumevanje tehničnih konceptov. Večina bi ga še naprej uporabljala za namene učenja robotike ter ga priporočila drugim študentom. Zaključimo lahko, da je tutor dobro dopolnil tradicionalno podporo učiteljev in prispeval k njihovemu napredku.

Glavne ugotovitve

Rezultati raziskave kažejo, da so študentje s pomočjo tutorja uspešno reševali težave, s katerimi so se soočali pri projektih iz robotike. Kljub pogostemu pristopu »copy-paste« je narava nekonsistentnih rezultatov, ki jih tak način dela pogosto prinese, študente prisilila v podrobnejšo analizo programske kode, kar je pripomoglo k njihovemu razumevanju in razvoju tehničnih veščin. Tutor se je izkazal kot ključna podpora, ki je študentom omogočila samostojnejše delo in hitreje napredovanje.

Največja pridobitev pa je bila na strani učitelja, ki se je iz vloge gasilca težav premaknil v opazovalca in analitika učnega procesa. Sprostitev časa in energije učitelja je omogočila celostnejši pregled nad projektnim delom, s čimer je lahko bolje prepoznal širše vzorce in strateško prilagodil učni proces. To

predstavlja pomemben premik v dinamiki projektnega dela, ki prispeva k bolj strukturiranemu in učinkovitejšemu učenju.

Tudi študentje so tutorja prepoznali kot koristno orodje, ki je izboljšalo njihovo samostojnost in razumevanje tehničnih konceptov. Posebej so pohvalili možnost komunikacije v maternem jeziku, takojšnjo odzivnost in podajanje referenc na študijsko gradivo. Glavni pomanjkljivosti sta bili predolgi odgovori in pomanjkanje specifičnosti pri nekaterih nalogah. Kljub tem izzivom je tutor pomembno dopolnil podporo učiteljev in prispeval k napredku študentov.

Razprava

Iris Ma idr. (2024) so ugotovili pozitiven vpliv uporabe tutorjev na osnovi tehnologije GUI na področju poučevanja programiranja. Podobno potrjuje tudi naša analiza pogovorov med študenti in tutorjem, in sicer da je tutor učinkovito reševal nekatere banalne tehnične težave, kot so manjkajoči znaki »;« v kodi, dvojno definirane funkcije in podobni programerski izzivi. Ti rezultati so bili skladni s pričakovanji, saj je tutor, opremljen z ustreznimi navodili, lahko hitro identificiral in predlagal rešitve za takšne napake. Presenetljivo pa je bilo, da se je tutor dobro odrezal tudi pri kompleksnejših izzivih, povezanih z ožičenjem in diagnostiko odpovedi pogonskega motorja. Kljub omejitvi na tekstovno komunikacijo je tutor z natančnimi in strukturiranimi vprašanji študente vodil skozi postopek odpravljanja težav, kar kaže na njegov potencial za podporo tudi pri praktičnejših vidikih projektnega dela.

Dodatne možnosti analize pogovora

Raziskava je pokazala, da bi lahko vprašanja, ki jih študentje zastavljajo tutorju, služila kot osnova za ocenjevanje njihovega znanja. Analiza vprašanj bi omogočila vrednotenje po dveh ključnih merilih: (a) pravilnost: ali je vprašanje smiselno in natančno; (b) relevantnost: ali vprašanje odraža razumevanje študenta o problemu.

Vrednotenje vprašanj bi lahko služilo kot orodje za spremljanje napredka študentov in izboljšanje učnega procesa. Vendar je pomembno poudariti, da bi lahko prekomerno osredotočanje na ocenjevanje vprašanj povzročilo zadržanost študentov pri spraševanju. Yu Chen idr. (2024) v raziskavi razložijo, da nekateri učenci lažje postavijo vprašanje GUI-tutorju kot profesorju, saj s tem ne razkrijejo svojega znanja (ali pomanjkanja le-tega). Ena ključnih prednosti tutorja je prav v tem, da študentje nimajo zadržkov zastavljati vprašanj, tudi če so ta banalna ali nepovezana z že obravnavano snovjo. Tutor deluje kot orodje brez čustev, ki ne presoja in ne zameri, kar omogoča odprto in sproščeno komunikacijo. S tako komunikacijo so bili študentje motivirani za

reševanje problemov. Vsako zastavljeno vprašanje odraža trenutni fokus in potrebo študenta, kar prispeva k njihovi angažiranosti. Ta vidik interakcije je ključen, saj motivacija igra pomembno vlogo pri usvajanju znanja. Tutorjeva sposobnost, da zagotavlja podporo brez zadržkov, povečuje njegov potencial za spodbujanje aktivnega učenja. V prihodnje bi bilo smiselno raziskati, kako lahko tutorji hkrati spodbujajo kakovost zastavljenih vprašanj in ohranjajo sproščenost komunikacije, kar bi še dodatno povečalo njihovo dodano vrednost v učnem procesu.

Tutor je zapolnil vrzel v podpori študentom pri tehničnih projektih, vendar so rezultati pokazali potrebo po izboljšavi metod za preverjanje študentskega znanja. Prehod z neposrednega preverjanja znanja na prilagodljivo ocenjevanje, ki temelji na vprašanju študenta, bi lahko povečal učinkovitost tutorja.

Uporabniški vmesnik:

Lim idr. (2011) so menja, da je uporabniško prijazen vmesnik ključnega pomena za uspešno interakcijo med uporabniki in digitalnimi učnimi orodji. Migracija iz terminalnega okna v komunikacijo preko spletnega brskalnika je bila bistvena za izboljšanje uporabniške izkušnje študentov. Terminalske načine komunikacije je za večino študentov predstavljal neznano in neprijazno okolje, kar je oteževalo interakcijo s tutorjem. Prehod na spletno aplikacijo pa je študente prestavil v znano okolje, kjer so se počutili samozavestnejši in bolj sproščeni. To je bistveno vplivalo na kakovost in pogostost njihovega sodelovanja s tutorjem. Velik prispevek novega vmesnika je bila tudi možnost uporabe priljubljene tehnike »copy-paste«. Študentje so lahko enostavno kopirali besedilo napak, ki jih je izpisal prevajalnik programske kode, in jih posredovali tutorju, ki jim je podal ciljno usmerjene odgovore. Ta funkcionalnost je izboljšala učinkovitost komunikacije in omogočila hitrejšo reševanje tehničnih težav.

Še večji učinek pa je prinesel prehod na strežniško obliko, ki je učitelju omogočila spremljanje pogovorov študentov s tutorjem v realnem času. Čeprav ta funkcionalnost v začetnem načrtu raziskave ni bila predvidena, se je razvila kot zelo uporabna prilagoditev tekom raziskave. Ta sprememba je prinesla več ključnih prednosti:

- *Pregled nad učnim procesom:* Učitelj je imel boljši vpogled v potek projektnega dela in je lahko prepoznal, kdaj in kje se pojavljajo težave.
- *Možnost posredovanja:* Ob opažanju napačnih interpretacij ali težav je učitelj lahko pravočasno posredoval in zagotovil usmeritev študentom, da so se izognili napačnim učnim tokovom.

- *Ocena študentovega znanja*: Analiza pogovorov je omogočila vpogled v raven razumevanja študentov na podlagi njihovih vprašanj in odzivov.
- *Zaznavanje napačnih predstav* (angl. *misconceptions*): Strežniški vmesnik je olajšal identifikacijo napačnih razumevanj, ki so ovirala nadaljnjo gradnjo znanja.
- *Možnosti analize pouka*: Zbrani podatki omogočajo poglobljeno analizo učnega procesa in načrtovanje izboljšav za naslednje izvedbe predmeta.

Poleg sprotne podpore je strežniški vmesnik omogočil vzpostavitev robustnega sistema, ki ne le podpira učni proces, temveč zagotavlja stalno izboljševanje na podlagi zbranih podatkov. Ta pristop krepi tako kakovost izobraževanja kot tudi zmožnost prilagajanja učnega procesa specifičnim potrebam študentov.

Omejitve raziskave

Raziskava ima več omejitev, ki jih je treba upoštevati pri interpretaciji rezultatov. Ena ključnih omejitev je majhno število študentov, ki so sodelovali v raziskavi (skupaj pet). Takšen omejen vzorec lahko vodi do neobičajnih ugotovitev, ki niso nujno reprezentativne za širšo populacijo. Zaradi tega je treba rezultate obravnavati previdno in jih upoštevati predvsem kot izhodišče za nadaljnje raziskave na večjih vzorcih.

Druga pomembna omejitev je odsotnost kontrolne skupine. Ker raziskava ni vključevala študentov, ki bi izvajali projekt brez uporabe tutorja, je primerjava učinkov tutorja mogoča le z izkušnjami učitelja iz preteklih let. Takšen pristop, čeprav uporaben, ne predstavlja znanstvene metode in ne omogoča neposrednega merjenja vpliva tutorja v primerjavi z alternativnimi oblikami podpore.

Te omejitve kažejo na potrebo po nadaljnjih raziskavah, ki bi vključevale večje vzorce študentov, uporabile kontrolne skupine in omogočile bolj strukturirano oceno vpliva tutorja na učni proces. Kljub temu trenutni rezultati ponujajo dragocene vpoglede in osvetljujejo potencial GUI-tutorjev v projektno usmerjenem učenju.

Zaključek

Raziskava o uporabi tutorja pri projektih iz robotike je pokazala, da so študentje orodje uporabljali na način, ki je v veliki meri odražal običajne vzorce uporabe takšnih podpornih tehnologij. Študentje so prednostno iskali neposredne rešitve z najmanjšim možnim vložkom, namesto da bi se osredotočali na iskanje in utrjevanje znanja. Kljub našim naporom, da tutorja opremimo z

vprašanji za samoevalvacijo, se študentje pogosto niso odzvali na ta vprašanja, kar kaže na pomanjkanje motivacije za refleksijo in poglobljeno učenje. Ta izziv odpira vprašanje, kako dodatno prilagoditi tutorja, da bi spodbujal analitičnejši in bolj angažiran pristop k učenju.

Ena ključnih ugotovitev raziskave je bila, da je prehod na strežniško obliko, čeprav prvotno ni bil načrtovan, izjemno povečal vrednost orodja. Strežniška arhitektura je učiteljem omogočila poglobljenejši vpogled v učni proces, kar je še posebej pomembno pri projektnem delu, kjer je proces močno individualiziran. Ta oblika je učiteljem omogočila spremljanje napredka v realnem času, prepoznavanje težav in napačnih predstav ter prilagajanje pouka na podlagi zbranih podatkov. Strežniška oblika je tudi ponudila priložnost za sistematično analizo pogostih vprašanj, kar odpira možnosti za izboljšanje učnih načrtov in metodologij v prihodnjih izvedbah predmeta.

Na podlagi ugotovitev raziskave bi bilo smiselno prilagoditi projektne naloge tako, da bi bile razčlenjene na več manjših, obvladljivih delov. Napredovanje v naslednjo fazo projekta bi bilo pogojeno s pravilno rešenim vprašalnikom ali nalogo, kar bi študente spodbudilo k razumevanju in refleksiji. Ta pristop bi lahko zmanjšal uporabo pristopa »copy-paste« in prispeval k poglobljenejšemu usvajanju znanja.

Raziskava ponuja dragocene vpoglede v uporabo tutorja, vendar tudi osvetljuje izzive, ki jih je treba nasloviti v prihodnjih raziskavah in implementacijah. Kljub omejitvam raziskave trenutni rezultati potrjujejo velik potencial GUI-tutorjev za podporo projektno usmerjenemu učenju, zlasti v tehničnih disciplinah. V prihodnje bo treba nadalje raziskovati, kako optimizirati funkcionalnosti tutorja in izboljšati učne izkušnje študentov.

Literatura

- Bastani, H., Bastani, O., Sungu, A., Ge, H., Kabakçı, O., in Mariman, R. (2024). *Generative AI can harm learning*. Social Science Research Network. https://papers.ssrn.com/sol3/papers.cfm?abstract_id=4895486
- Chen, Y., Jensen, S., Albert, L. J., Gupta, S. in Lee, T. (2023) Artificial intelligence (AI) student assistants in the classroom: Designing chatbots to support student success. *Information Systems Frontiers*, 25, 161–182.
- Chipantiza, N., Palma, H., in Morocho, L. (2024). Systematic analysis of artificial intelligence integration in robotics learning in secondary education. *Universidad Ciencia y Tecnología*, 28(123), 111–121.
- Fatimah, S., Sarwi, S., in Haryani, S. (2024). Artificial intelligence in STEM education: A bibliometric analysis. *International Journal of Multicultural and Multireligious Understanding*, 11(1), 186–200.

- Jahic, I., Ebner, M., Schön, S., in Edelsbrunner, S. (2024). Exploring the use of generative AI in education: Broadening the scope. *Lecture Notes in Computer Science*, 14724, 283–304.
- Kharrufa, A., in Johnson, I. (2024). The potential and implications of generative AI on HCI education. V *Proceedings of the 6th annual symposium on hci education (EduCHI '24)* (str. 1–8). Association for Computing Machinery.
- Kotsiovos, J., Chicone, R., in Doyle, S. (2024). Optimizing student success: The impact of generative AI in teaching and learning. *TCC 2024 Conference Papers*.
- Lim, C., Song, H. D., in Lee, Y. (2012). Improving the usability of the user interface for a digital textbook platform for elementary-school students. *Educational Technology Research and Development*, 60(1), 159–173
- Ma, I., Krone-Martins, A., in Videira Lopes, C. (2024). Integrating ai tutors in a programming course. In *Proceedings of the 2024 on ACM virtual global computing education conference* (str. 130–136). Association for Computing Machinery.
- Marzuki, O. F., Lih, E., Abdullah, W., Khairuddin, N., Inai, N., MD Saad, J., in Aziz, M. (2024). Innovating education: A comprehensive review of STEM education approaches. *International Journal of Academic Research in Progressive Education and Development*, 13, 614–631.
- Paredes Rizo, C. (2021). *Chatbots en educación secundaria: retos y propuestas para su aplicación en el aula*. Universidad de Valladolid.
- Qureshi, B. (2023). ChatGPT in computer science curriculum assessment: An analysis of its successes and shortcomings. V *2023 9th International conference on e-society, e-learning and e-technologies* (str. 7–13). Association for Computing Machinery.
- Rashid, S. F., Duong-Trung, N., in Pinkwart, N. (2024). Generative AI in education: Technical foundations, applications, and challenges. V S. Kadry (ur.), *Artificial intelligence and education*. IntechOpen. <https://doi.org/10.5772/intechopen.1005402>
- Saude, S., Barros, J. P., in Almeida, I. (2024). Impacts of generative artificial intelligence in higher education: Research trends and students' perceptions. *Social Sciences*, 13(8), 410.
- Tlili, A., Shehata, B., Adarkwah, M. A., Bozkurt, A., Hickey, D. T., Huang, R., in Agyemang, B. (2023). What if the devil is my guardian angel: ChatGPT as a case study of using chatbots in education. *Smart Learning Environments*, 10(1), 15.
- Wecks, J. O., Voshaar, J., Plate, B., in Zimmermann, J. (2024). *Generative AI usage and exam performance*. Social Science Research Network. <https://doi.org/10.2139/ssrn.4812513>
- Wojciechowski, A. in Korjonen-Kuusipuro, K. (2023). How artificial intelligence affects education? *Human Technology*, 19(3), 302–306.

Innovative Application of a GEN-AI Tutor in Robotics and Electronics Projects in a Pedagogical Context

This paper explores an innovative use of generative artificial intelligence (GEN-AI) in the courses “Robotics” and “Electronics” at the Faculty of Education, University of Ljubljana. In these courses, attended by dual-major students with a focus on technology, a new approach to independent student work on final projects was introduced in the winter semester of the academic year 2024/2025. A tailored GEN-AI agent-tutor was employed, enhanced with didactic principles (active learning, Kolb’s learning cycle) and equipped with literature covered in lectures. Unlike general generative models, the agent accounted for the students’ available equipment and the technical specifics of the projects. This allowed it to provide specific and relevant responses tailored to the given situation. The system aimed to guide students through a personalised learning pathway, with the agent’s responses substituting for the teacher’s presence when the teacher was unavailable. Questions and responses generated during interactions between the Gen-UI agent-tutor and students will serve as a basis for further development and improvement of the agent’s pedagogical effectiveness.

Keywords: generative artificial intelligence, Gen-AI tutor, personalised learning, electronics and robotics projects, Kolb’s learning cycle