



59. državno srečanje mladih raziskovalcev Slovenije

VKLJUČEVANJE 3D SVETOV V IZOBRAŽEVANJE NA PRIMERU VIRTUALNEGA HODNIKA MODROSTI IN SOBE POBEGA

Aplikativni inovacijski predlogi in projekti

Sekundarno področje: računalništvo

Avtorica: Anja Podbrežnik

Mentorica: Mojca Borin

Somentorica: Nina Murks

Šola: Srednja ekonomska šola in gimnazija Maribor

Maribor 2025

KAZALO VSEBINE

| | | |
|--------|---|----|
| 1 | UVOD | 6 |
| 2 | METODOLOGIJA DELA..... | 7 |
| 2.1 | Izbira orodja | 7 |
| 2.2 | Delo v orodju..... | 9 |
| 2.3 | Načrt, drevesna rešitev iz sobe pobega | 9 |
| 2.4 | Izdelava programa | 11 |
| 2.4.1 | Postavitev delovnega okolja | 11 |
| 2.4.2 | Izdelava zgradbe hodnika in sobe pobega | 12 |
| 2.4.3 | Osvetlitev hodnika in sobe pobega..... | 12 |
| 2.4.4 | Dodajanje tekstur..... | 12 |
| 2.4.5 | Dodajanje prvoosebnega krmilnika | 13 |
| 2.4.6 | Namestitev slik na stene hodnika | 13 |
| 2.4.7 | Dodajanje vrat..... | 14 |
| 2.4.8 | Odpravljanje napak prehajanja prvoosebnega krmilnika skozi vrata | 14 |
| 2.4.9 | Opremljanje sobe pobega | 15 |
| 2.4.10 | Ustvarjanje gradiva s pomočjo orodja Canva..... | 15 |
| 2.4.11 | Dodajanje vnosa prvoosebnega krmilnika..... | 16 |
| 2.4.12 | Dodajanje pozdravnega okna..... | 19 |
| 2.4.13 | Dodajanje gumbov na predmete v sobi | 20 |
| 2.4.14 | Dodajanje vrat v sobo pobega | 23 |
| 2.4.15 | Dodajanje gumba za izhod | 25 |
| 2.4.16 | Nadgraditev spreminjanja besedila v programu tudi po izvozu s pomočjo Google preglednic | 27 |
| 2.4.17 | Odpravljanje napak..... | 32 |
| 2.4.18 | Izvoz igre | 32 |
| 2.5 | OPOMBE | 33 |
| 2.6 | POVZETEK METODOLOGIJE DELA | 33 |
| 3 | REZULTATI IN INTERPRETACIJA REZULTATOV | 35 |

| | | |
|---|--|----|
| 4 | DRUŽBENA ODGOVORNOST IN TRAJNOST | 36 |
| 5 | PRISPEVEK K NAPREDKU..... | 37 |
| 6 | ZAKLJUČEK / SKLEPI | 38 |
| 7 | LITERATURA | 39 |

KAZALO TABEL

| | |
|---|---|
| Tabela 1: Primerjava programskega okolja Unity z Unreal Engine (Johns, 2025)..... | 8 |
|---|---|

KAZALO SLIK

| | |
|---|----|
| Slika 1: Skica hodnika in sobe pobega (lasten vir)..... | 9 |
| Slika 2: Drevesni načrt pravilnega zaporedja vprašanj na posameznih predmetih v sobi pobega (lasten vir)..... | 11 |
| Slika 3: Prikaz izdelka po dodajanju snovi in osvetlitve (lasten vir) | 13 |
| Slika 4: Prikaz prilagajanja velikosti luknje v zidu, glede na velikost vrat (lasten vir) | 14 |
| Slika 5: Prikaz vstavljenega kode pod objektom [Dialog UI] (1) in pravilno povezanega platna in gumba za zapiranje (lasten vir) | 20 |
| Slika 6: Prikaz pravih povezav med objekti (glejte obkroženo vsebino na sliki) (lasten vir)..... | 21 |
| Slika 7: Prikaz pravilne povezave »NaKlik« funkcije (lasten vir)..... | 21 |
| Slika 8: Izgled panela za vnos gesla (lasten vir) | 23 |
| Slika 9: Tabela v Google preglednicah za črpanje podatkov v Unity (lasten vir) | 27 |
| Slika 10: Slika scene s prikazanimi podatki iz tabele in preglednik z ustreznimi povezavami in naloženimi podatki iz Google tabele (lasten vir) | 31 |
| Slika 11: Na sliki so označeni popravki, ki smo jih naredili (lasten vir) | 32 |
| Slika 12: Kako izvoziti igro (lasten vir)..... | 33 |

KAZALO IZVLEČKOV KODE

| | |
|--|----|
| Izvleček kode 1: Programska koda za odpiranje spletnih povezav do gradiv (Clip Collection Vault, 2022) | 15 |
| Izvleček kode 2: Programska koda s skripto Interaktivno (Chris Nolet, 2022)..... | 17 |
| Izvleček kode 3: Programska koda za interakcijo igralca (Chris Nolet, 2022) | 18 |
| Izvleček kode 4: Programska koda za zaprtje pozdravnega okna (Herbou, 2021)..... | 19 |
| Izvleček kode 5: Programska koda za odprtje plošče ob kliku na gumb (Jayanam, 2019) | 22 |
| Izvleček kode 6: Programska koda za zaprtje panela ob ponovnem kliku na gumb (Jayanam, 2019) 22 | |

| | |
|---|----|
| Izveček kode 7: Programska koda za izpis uporabniškega vnosa v polje za vnos gesla. (Gemini, 2025) | 24 |
| Izveček kode 8: Posodobljena programska koda »Sistem za geslo« z dodanim preverjanjem števila črk in gesla. (ThirteenNov, Unity "Enter Correct Passcode" Tutorial - Part 3 : The wrong and correct passcode, 2024) (ThirteenNov, Unity "Enter Correct Passcode" Tutorial - Part 2 : Displaying each number entered, 2024) | 25 |
| Izveček kode 9: Programska koda za izhod iz igre (ChatGPT, 2025) (Gemini, 2025) (Kap Koder, 2021) | 25 |
| Izveček kode 10: Posodobljena programska koda »Sistem za geslo« z dodanim gumbom za izhod ob vnosu pravilnega gesla (ChatGPT, 2025) (Gemini, 2025) (Kap Koder, 2021) | 26 |
| Izveček kode 11: Koda za shranjevanje podatkov (ChatGPT, 2025) | 28 |
| Izveček kode 12: Koda za nalaganje in prikaz vprašanj (ChatGPT, 2025) | 30 |
| Izveček kode 13: Koda za prikaz in razvrščanje podatkov po indeksu (ChatGPT, 2025) | 31 |

POVZETEK

Predstavljamo vam inovacijski predlog razvoja interaktivne učne 3D igre v razvijalskem orodju Unity. Cilj projekta je bil ustvariti 3D sobo pobega in hodnik modrosti. Hkrati pa smo omogočili spreminjanje vsebine in prilagajanje le te v že izvoženi igri. S tem smo omogočili spreminjanje učne snovi in besedil ter povečali uporabnost igre z možnostjo njene posodobitve izven razvijalskega orodja Unity. Igra deluje tako, da v hodniku modrosti učenci s pomočjo gumbov dostopajo do učnih gradiv izdelanih v grafičnem orodju Canva, nato pa svoje novo pridobljeno znanje uporabijo za pobeg iz sobe pobega. V postopku izvedbe inovacijskega predloga smo se ukvarjali s primerjalno analizo razvijalskih orodij Unity in Unreal Engine, ter za naš projekt izbrali Unity. Prav tako smo izdelali načrte razporeditve posameznih učnih gradiv in vprašanj po celotnem 3D objektu. Pri razvoju projekta v Unity pa smo uporabili orodji ProBuilder za grajenje 3D objektov in TextMesh Pro za oblikovanje besedila, ter implementirali številne programske kode, pri katerih smo preizkušali tudi zmožnosti pisanja programskih kod ChatGPT – ja in Geminija. Na koncu smo ugotovili, da so 3D igre inovativnejši način učenja in povečajo motivacijo učencev.

Ključne besede: 3D učna igra hodnika modrosti in sobe pobega, razvijalsko orodje Unity, programski jezik C#, ChatGPT, Gemini

SUMMARY

We present to you an innovative proposal for the development of an interactive educational 3D game in the Unity development tool. The goal of the project was to create a 3D escape room and a corridor of wisdom. At the same time, we allowed for content modification and customization in the already exported game. This enabled us to modify the learning material and texts and increase the usability of the game with the possibility of updating it outside the Unity development tool. The game works in such a way that in the corridor of wisdom, students use buttons to access learning materials created in the Canva graphics tool, and then use their newly acquired knowledge to escape from the escape room. In the process of implementing the innovative proposal, we made a comparative analysis of the Unity and Unreal Engine development tools, and chose Unity for our project. We also created plans for the distribution of individual learning materials and questions throughout the entire 3D object. When developing the project in Unity, we used ProBuilder for building 3D objects and TextMesh Pro for text formatting, and implemented numerous code snippets, testing the coding capabilities of ChatGPT and Gemini. In the end, we found out that 3D games make a more innovative way of learning and increase student motivation.

Keywords: 3D educational game the corridor of wisdom and escape room, Unity development platform, C# programming language, ChatGPT, Gemini

1 UVOD

V današnjem času imajo mladi precej negativen odnos do šolanja. Šola jim predstavlja nujno zlo, najraje so na mobilnih omrežjih in računalniških igrah. Vsi se zavedamo, kako pomembno je izobraženo prebivalstvo za državo, saj ji omogoča večjo gospodarsko rast z inovativnimi podjetji in visoko specializiranimi delavci. V ta namen sem se odločila izvesti ta inovacijski predlog, s katerim želim navdušiti mlade generacije za učenje in dvigniti njihovo motiviranost.

Najlažje se je mladim približati s tehnološkimi novostmi, še lažje z računalniškimi igrami. Zakaj potem ne bi učne snovi v šoli predelovali skozi računalniške igre? Učenje je lahko tudi zabavno! A vendar katera igra bo bolj pritegnila učence, 2D ali 3D? Odgovor sem izvedela v raziskavi, opisani v članku Struktura igralcev ploščadnih iger (Clack, 2023), v kateri so med drugim preverjali priljubljenost 2D in 3D iger. Rezultati so pokazali večjo priljubljenost 3D iger v primerjavi z 2D igrami. To je precej razumljivo, saj nam 3D igre omogočajo poglobljene in bolj realistične izkušnje. (Egor, 2025) Glede na navedeno sem se odločila, da bo moja igra 3D, saj si želim doseči čim večjo motiviranost učencev za učenje in povečanje njihovega znanja.

3D okolja se le redko uporabljajo v namen izobraževanja, kar pa bi pripomoglo tudi k večji računalniški pismenosti učencev. S tem inovacijskim predlogom, pa ne bom oblikovala le inovativnega učnega okolja, ki se bo uporabljalo v namen izobraževanja na naši šoli, ampak bom pripomogla tudi k večji lastni računalniški pismenosti, saj se bom prvič seznanila z izdelavo projekta v Unity-u in programskim jezikom C#. Do zdaj sta moje edino znanje programiranja poletni šoli Python programerke (Fakulteta za elektrotehniko, računalništvo in informatiko, 2024) in Programiranje C++ za dekleta (Fakulteta za elektrotehniko računalništvo in informatiko, 2024), ki ju je organizirala Fakulteta za elektrotehniko, računalništvo in informatiko Maribor. Moj cilj v projektu je tudi preizkusiti umetni inteligenci ChatGPT in Gemini za pomoč pri pisanju kod. Moj največji osebni cilj pa je, preizkusiti sebe ali zmorem narediti celotno igro povsem sama. In ali je na spletu dovolj informacij, ki zadostujejo začetniku za izdelavo celotne 3D igre?

Razmišljala sem, kako bi lahko v 3D okolju pripravila igro z učnim gradivom pri predmetu informatike na naši šoli. Učno gradivo se nanaša na zgradbo računalnika in zgodovino računalništva. Po razmisleku sem se odločila, da bi lahko naredila sobo pobega. Na spletu je že nekaj virtualnih sob pobega v namen učenja kot so sobe pobega s spletne strani »Madison Country Public Library« ali predloge virtualnih sob pobega na »Genially«, pa vendar so vse te sobe 2D, kar nam ne da poglobljene in vživete izkušnje. Morda se na spletu res najde kakšna virtualna 3D soba pobega, pa vendar ta ni namenjena učenju, bolj zabavi. Svojo sobo pobega sem si resnično želela imeti povsem inovativno, zato sem ji dodala še hodnik modrosti. Našla še nisem nobene sobe pobega, ki bi vsebovala tudi učno snov, ki se jo učenci naučijo pred vstopom v sobo pobega. Večinoma so le sobe pobega, ki služijo preverjanju že pridobljenega znanja. Kako se lahko potem iz sobe pobega rešiš, če nimaš pridobljenega znanja? Dokončna odločitev je bila, da bom imela najprej hodnik modrosti, ki bo imel na stenah različne slike povezane z učno snovjo. Poleg vsake slike bo gumb, ob kliku na katerega se ti bo odprlo učno gradivo narejeno v Canvi. Po tem takem je cilj naloge tudi spoznati, ali je možno posodabljati gradivo in besedilo v hodniku modrosti in sobi pobega, tudi po izvozu igre? Čisto na koncu pa bodo po sprehodu skozi hodnik učenci vstopili v sobo pobega in iz nje poizkusili pobegniti s pridobljenim znanjem.

Prav tako je moj cilj ustvariti igro, ki jo bo lahko katerikoli učitelj/ica tudi brez računalniškega znanja iz programiranja prilagajal/a svoji učni snovi in tako motiviral/a učence, da bi se skozi igro naučili novo učno snov.

2 METODOLOGIJA DELA

2.1 Izbira orodja

Ko smo vedeli, kaj želimo narediti, smo morali izbrati ustrezno orodje za izdelavo projekta. Na spletu smo poiskali katera orodja so najprimernejša za ustvarjanje 3D iger. Našli smo Unity in Unreal Engine. Naredili smo primerjavo med obema orodjema, ki je prikazana v tabeli. (Tabela 1) (Johns, 2025)

Tabela 1: Primerjava programskega okolja Unity z Unreal Engine (Johns, 2025)

| KRITERIJ | PROGRAMSKO OKOLJE UNITY | PROGRAMSKO OKOLJE UNREAL ENGINE |
|--|---|---|
| UPORABNOST ZA ZAČETNIKE | Enostavnejši za začetnike, široka dokumentacija in skupnost. | Manj prijazen začetnikom, vendar ima obsežno dokumentacijo. |
| PROGRAMSKI JEZIK | C# (enostaven za učenje, široka podpora). | C++ (večja zahtevnost) ali Blueprint vizualno programiranje. |
| PODPORA ZA 3D IN 2D IGRE | Močna podpora za 3D in 2D igre in simulacije. | Močna podpora za 3D igre in simulacije, bolj osredotočen na 3D. |
| VIZUALNA KAKOVOST | Dobra vizualna podpora, vendar nekoliko slabša v primerjavi z Unreal Engine. | Izjemna vizualna kakovost, primerna za visokokakovostne igre. |
| ENOSTAVNOST ZA IMPLEMENTACIJO INTERAKTIVNOSTI | Enostavna implementacija, veliko virov in orodij za interaktivnost. | Prav tako dobra podpora, vendar pogosto zahteva več napora za začetnike. |
| ZMOGLJIVOST | Zelo dobro optimizirana za mobilne naprave in manjše igre. | Izjemna zmogljivost, ki je primerna za kompleksne igre, a nekoliko težja za optimizacijo. |
| PODPORA ZA PLATFORME | Izvoz za skoraj vse platforme (Windows, macOS, iOS, Android itd.). | Izvoz za podobne platforme, vendar Unity nudi več fleksibilnosti pri manjših napravah. |
| CENA | Brezplačna različica, vendar z možnostjo nakupa naprednih funkcij. | Brezplačna različica, vendar s 5% provizijo na prihodke po določeni meji. |
| SKUPNOST IN VIRI | Ogromna skupnost, veliko brezplačnih virov in vsebin. | Velika skupnost, vendar manjša kot pri Unity; več primerov za visoko kakovostne igre. |
| ORODJA IN DODATKI | Širok spekter orodij, vključno s številnimi vtičniki in trgovino s sredstvi (ang. Asset store). | Širok spekter orodij, predvsem za visokokakovostne igre. |
| PODPORA ZA UMETNO INTELIGENCO | Enostavna integracija umetne inteligence in razširitev z zunanjimi orodji. | Ponuja možnosti napredne umetne inteligence, vendar lahko zahteva več napora pri izvedbi. |
| UČENJE IN RAZVOJ | Hiter začetek, številni viri za začetnike, veliko spletnih tečajev. | Potreben čas za učenje, še posebej C++ je lahko zahtevnejši. |

Za svoj projekt smo izbrali Unity, saj ponuja lahko razumljiv uporabniški vmesnik in obsežno podporo skupnosti. Omogoča tudi enostavno dodajanje interaktivnosti s pomočjo programiranja v programskem jeziku C#, ter prilagoditev fizikalnih lastnosti in osvetlitve, kar nam je omogočilo večjo ustvarjalno svobodo. Pomemben dejavnik pri izbiri je bila tudi možnost izvoza projekta na različne platforme.

Unreal Engine se nam je zdel manj primeren, saj je bolj osredotočen na razvoj grafično zahtevnih 3D iger in ima kompleksnejši delovni proces, kar bi podaljšalo čas učenja in izdelave našega projekta. (Johns, 2025)

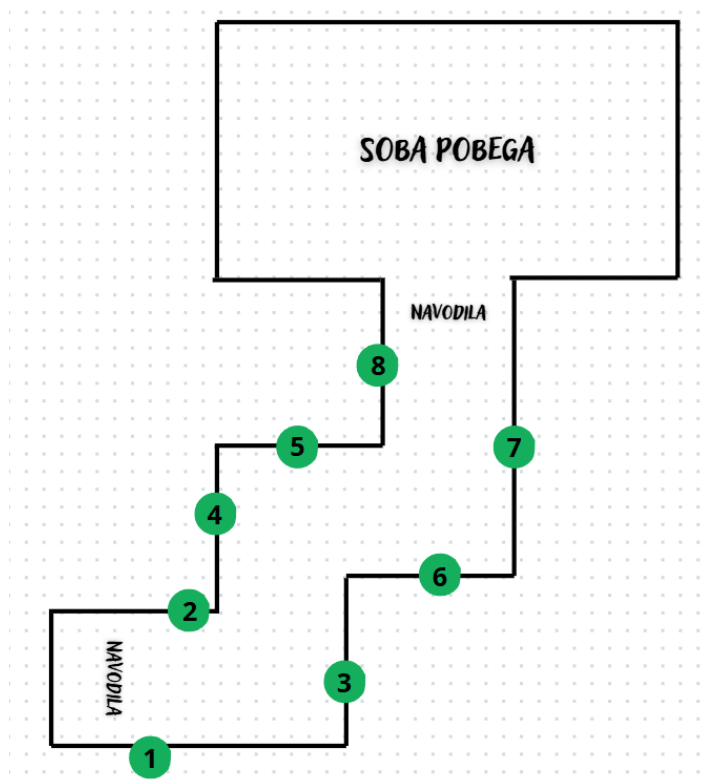
2.2 Delo v orodju

Za začetek razvoja projekta smo najprej prenesli in namestili programsko orodje Unity z uradne spletne strani. Na voljo je bila brezplačna različica, Unity 6, ki je povsem zadostovala našim potrebam.

Najprej smo se prijavili v platformo Unity Hub in nato ustvarili nov projekt. Izbrali smo predlogo "Univerzalno 3D jedro", določili ime projekta in ga ustvarili. Ko je bil postopek zaključen, se je odprlo glavno delovno okolje, kjer smo lahko začeli z izdelavo našega projekta. (Unity, 2024)

2.3 Načrt, drevesna rešitev iz sobe pobega

Pripravili smo si načrt končnih izdelkov. Najprej smo si razdelili gradivo, ki se ga bodo dijaki skozi igro naučili, in ga razporedili po hodniku modrosti, pri čemer smo določili tudi prikaz gradiva v oklepajih. Tematsko smo se odločili za učno snov, ki obravnava zgradbo računalnika in zgodovino računalništva. Skica hodnika in sobe pobega je prikazana spodaj v prikazu Slika 1.



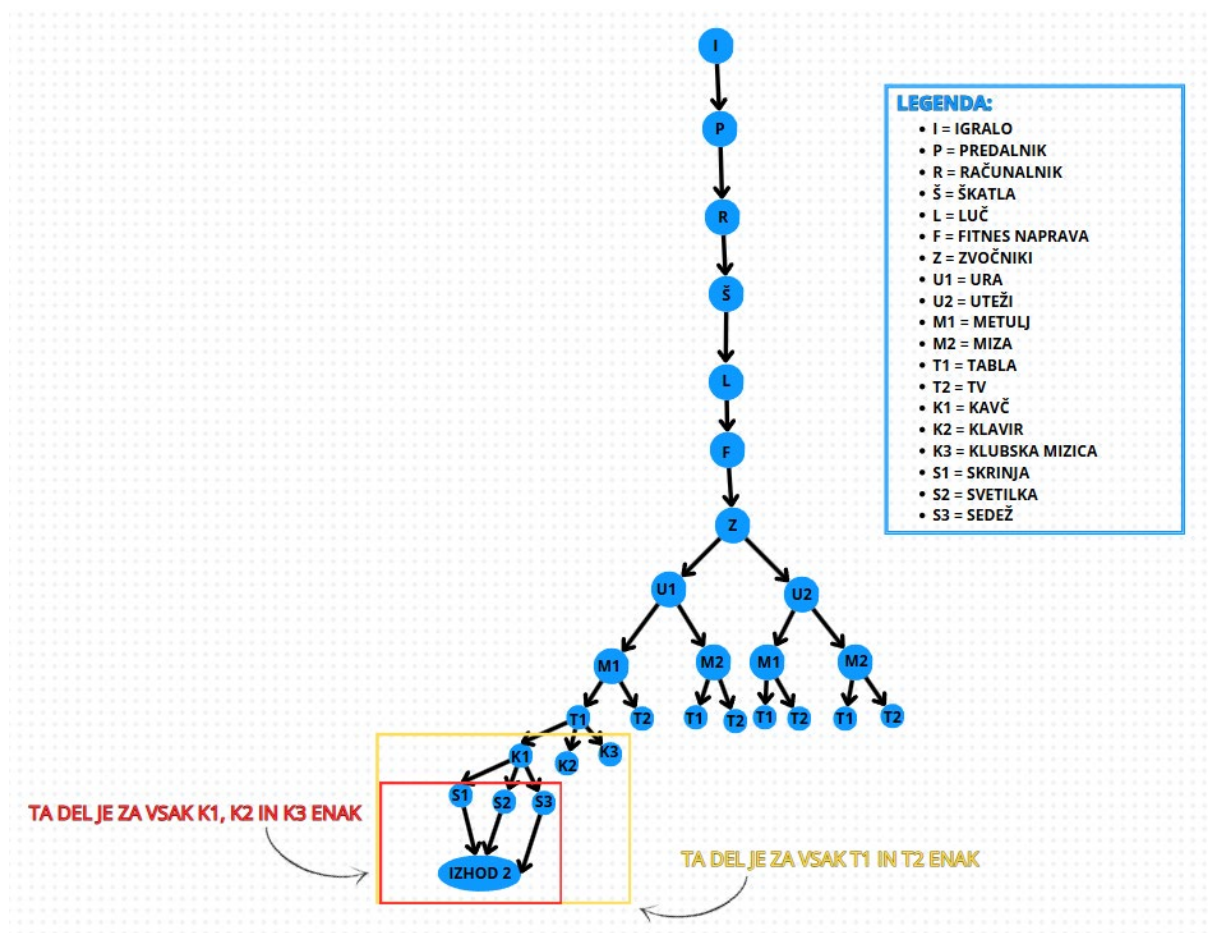
Slika 1: Skica hodnika in sobe pobega (lasten vir)

Legenda številik na skici (Slika 1).

- 1 - Časovnica razvoja računalništva (časovni trak).
- 2 - Delitev zgradbe računalnika na strojno in programsko opremo (miselni vzorec).
- 3 - Matična plošča (video).
- 4 - Procesor (predstavitev).
- 5 - Pomnilnik (predstavitev).
- 6 - Vhodne / Izhodne enote (slikovni prikaz).
- 7 - Von Neumannov model računalnika (samo slikovni prikaz v hodniku).
- 8 - Osnovne funkcionalnosti računalnika (miselni vzorec).

Vsaka številka predstavlja sliko na steni in gumb, ki vodi do gradiva ustvarjenega v Canvi. (Canva, 2025)

Naredili smo še načrt za sobo pobega (Slika 2). Soba pobega bo vsebovala veliko pohištva in drugih predmetov. Nekateri predmeti bodo imeli na sebi majhen gumb z vprašanjem, ob kliku na katerega se bo odprlo okno z imenom predmeta, vprašanjem in tremi možnimi odgovori. Le en odgovor bo pravilen, črka pred pravilnim odgovorom pa bo tvorila prvo črko naslednjega predmeta, pred katerim bo naslednje vprašanje. Le pravilno zaporedje črk pravih odgovorov bo predstavljalo geslo, ki ga bo moral uporabnik vnesti v prava vrata za izhod iz igre.



Slika 2: Drevesni načrt pravilnega zaporedja vprašanj na posameznih predmetih v sobi pobega (lasten vir)

Prikazan je drevesni prikaz pravilnega zaporedja vprašanj pri predmetih, ki tvori geslo: IPRŠLFZUMTKS, le tega morajo igralci vpisati v Izhod 2.

2.4 Izdelava programa

2.4.1 Postavitev delovnega okolja

Na računalnikih smo že odprli delovno okolje, v katerem smo delali. Za začetek smo v naš projekt prenesli paketa "Naknadna obdelava" in "ProBuilder" ter sredstvo "Mini prvoosebni krmilnik". Paketa smo prenesli tako, da smo odprli zavihek "Okno", poiskali "Upravljalnik paketov", v iskalno polje vpisali ime vsakega paketa posebej in ju prenesli. Sredstvo pa smo prenesli tako, da smo obiskali trgovino sredstev Unity, se prijavi s svojim računom, ki smo ga uporabljali v Unity Hub, poiskali sredstvo in ga dodali v svojo knjižnico. Nato smo v našem projektu v Unity-ju kliknili na "Trgovina

sredstev", izbrali "Moja sredstva", poiskali dodano sredstvo in ga prenesli. (Simon Serge Pasi, 2021)

2.4.2 Izdelava zgradbe hodnika in sobe pobega

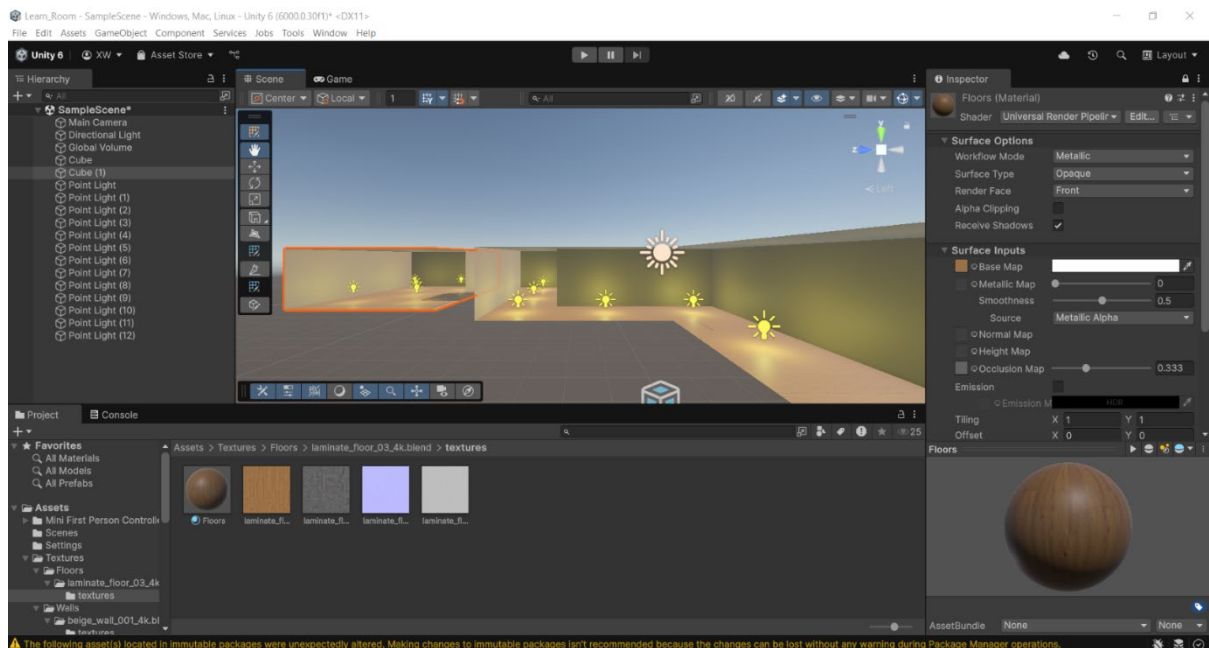
Odprli smo zavihek "Orodja" > "ProBuilder" > "Urejevalniki" > "Ustvari obliko" > "Kocka" in vstavili zeleno obliko, pri čemer smo se zavedali, da ena kocka meri 1 meter. To kocko smo s pomočjo puščic raztegnili v kvader. Ker smo želeli, da bi bil naš hodnik razvejan in ne zgolj raven, smo obliko razvejali tako, da smo kliknili na rob, kjer smo jo želeli podaljšati, ter nato izbrali "Orodja" > "ProBuilder" > "Geometrija" > "Vstavi robno zanko". Nato smo z orodjem "Izbira površin" in uporabo puščic ter tipke Shift podaljšali hodnik. Ko smo oblikovali zeleno razvejano strukturo hodnika, smo zraven dodali še en večji kvader, ki je predstavljal sobo pobega. Na koncu smo odprli "Orodja" > "ProBuilder" > "Objekt" > "Obrni normale objekta", s čimer smo dosegli, da so naši kvadri postali votli.

2.4.3 Osvetlitev hodnika in sobe pobega

Hodnik in sobo smo osvetlili tako, da smo v drevesni strukturi datotek z desnim klikom izbrali "Svetilo" > "Točkovna osvetlitev". S tem smo ustvarili novo luč, ki smo ji nastavili pravilno lego, barvo, jakost in območje osvetlitve. Način osvetlitve je bil nastavljen na "realno-časovni" način. Nato smo luč večkrat podvojili in uredili osvetlitev celotnega prostora.

2.4.4 Dodajanje tekstur

Odprli smo spletno mesto Poly Haven (Poly Haven, 2025) in prenesli dve teksturi – eno za tla (Poly Haven, 2023) in eno za stene (Poly Haven, 2022). Preneseni teksturi smo povlekli v svoj projekt med sredstva. Nato smo poleg teh tekstur ustvarili dve novi snovi (desni klik > "Ustvari" > "Snov"). V eno snov smo povlekli eno teksturo, v drugo snov pa drugo teksturo. Nato smo vsako snov povlekli na ustrezne površine našega kvadra, s čimer smo pridobili končen videz prostora, kot prikazuje slika. (Slika 3)



Slika 3: Prikaz izdelka po dodajanju snovi in osvetlitve (lasten vir)

2.4.5 Dodajanje prvoosebnega krmilnika

Prvoosebni krmilnik je sistem, ki omogoča premikanje in nadzor kamere iz perspektive prve osebe v 3D okolju. V našem projektu se uporablja za realistično navigacijo, interaktivnost in poglobljeno uporabniško izkušnjo. Je ključen element našega projekta, saj želimo uporabniku omogočiti občutek prisotnosti in neposredno raziskovanje prostora.

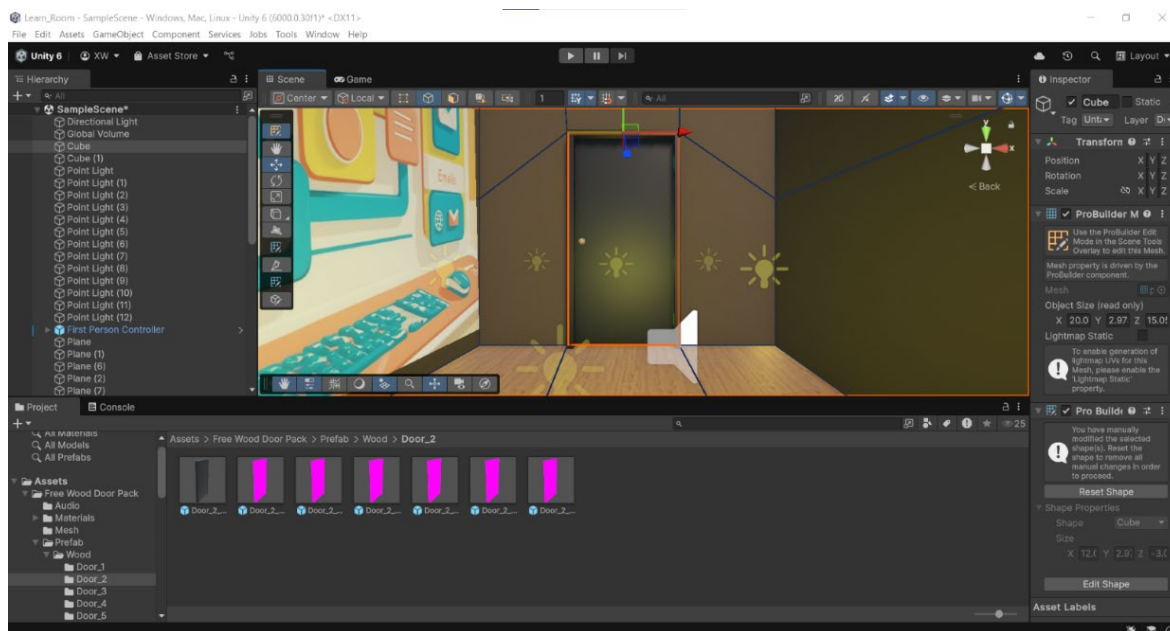
V drevesni strukturi datotek smo dodali »Mini prvoosebni krmilnik« in izbrisali glavno kamero iz seznama objektov. Tako smo krmilnik lahko videli v sceni. Nastavili smo mu ustrezno pozicijo (na začetku hodnika) in zeleno rotacijo.

2.4.6 Namestitev slik na stene hodnika

Na stene hodnika smo namestili slike. To smo storili tako, da smo najprej zeleno sliko povlekli med sredstva, nato smo ustvarili novo snov. Sliko smo povlekli na snov, kot smo to storili za stene in tla, ter na koncu ustvarili še nov 3D objekt, tokrat ploskev. Na ta objekt smo povlekli snov, objektu smo spremenili zeleno velikost in ga postavili tik ob steno. To smo ponovili še za vse preostale slike. (The Online Learning Space, 2021)

2.4.7 Dodajanje vrat

Dodali smo vrata, tako da smo prenesli brezplačen paket lesenih vrat iz trgovine sredstev Unity. Ko smo ga uspešno dodali v naš projekt (kot smo že pred tem dodali »Mini prvoosebni krmilnik«), smo morali v zidu med sobo pobega in hodnikom narediti luknjo (Slika 4). Naredili smo jo tako, kot smo razvejali hodnik, le da tokrat nismo držali tipke Shift in vlekli puščic, ampak smo samo izbrisali del ploskve. Ko smo v projekt vstavili vrata, smo morali velikost luknje prilagoditi velikosti vrat. To smo prilagodili s pomočjo puščic. Vrata so bila odprta, zato smo obkljukali gumbek, da so vrata ostala odprta. (Biostart, 2024)



Slika 4: Prikaz prilagajanja velikosti luknje v zidu, glede na velikost vrat (lasten vir)

2.4.8 Odpravljanje napak prehajanja prvoosebnega krmilnika skozi vrata

Kljub odprtim vratom naš prvoosebni krmilnik (ang. FPS) ni mogel skozenj. Zato smo povečali "Trkalnik" vrat, ki omogoča prehod prvoosebnemu krmilniku, ter obkljukali "Sprožilec", saj to omogoča prehod prvoosebnemu krmilniku skozenj. Ker to še vedno ni pomagalo, smo spremenili še velikost prvoosebnemu krmilniku. Tedaj se je prvoosebni krmilnik lahko premaknil skozenj, vendar je lahko pod določenim kotom videl skozi steno v druge prostore. To smo odpravili tako, da smo pri kameri našega

prvoosebnega krmilnika nastavili "Obrezovanje ravnin v bližini" na 0,1. (ChatGPT, 2025) (Gemini, 2025)

2.4.9 Opremljanje sobe pobega

Potem smo opremili sobo pobega, tako da smo prenesli paket »Brezplačna oprema – Nizkopoligonski paketi«. Po prenosu, ko smo uspešno uvozili predmete, smo enega dodali na prizor, pri čemer je bil cel vijoličen. Napako smo odpravili tako, da smo pri materialu spremenili "Senčilo" v "ProBuilder6/Standard Vertex Color", na isti način smo dodali še preostale predmete na želeno mesto. (ithappy, 2025)

2.4.10 Ustvarjanje gradiva s pomočjo orodja Canva

Ustvarili smo gradivo s pomočjo orodja Canva. Najprej smo se prijavili, izbrali ustrezno predlogo ali naredili svojo in nato urejali pisave, dodajali grafike, slike, zvoke itd. Na koncu smo projekt izvozili ali samo ustvarili povezavo za dostop. V našem primeru smo ustvarjali povezave za dostop, preko katerih bo igra s programsko kodo dostopala do našega oblaka v Canvi, ki nam bo omogočal tudi posodobitev gradiva. Zgolj pri gradivu o Von Neumannovi arhitekturi smo izvozili gradivo kot sliko, ki smo jo nato dodali na steno.

Napisali smo kodo, ki je ob kliku na gumb odprla želeno povezavo. Kliknili smo v Sredstva, desni klik > "Ustvari" > "Skripto za programsko kodo" > "Prazna skripto za programsko kodo v programskem jeziku C#". Poimenovali smo jo »OpenUrl«. Da bi lahko pisali kodo, smo morali prenesti Visual Studio Code z njihove uradne spletne strani. Po prenosu in namestitvi smo odprli skripto, ki smo jo ustvarili, in vanjo napisali kodo, ki je prikazana v spodnjem izvlečku (Izvleček kode 1).

```
using UnityEngine;

public class OpenUrl : MonoBehaviour
{
    public void Openurl()
    {
        Application.OpenURL("https://www.google.com/");
    }
}
```

Izvleček kode 1: Programska koda za odpiranje spletnih povezav do gradiv (Clip Collection Vault, 2022)

Nato smo shranili kodo in se v seznamu objektov postavili na prvo ploskev, ki smo jo že ustvarili za slike. Kliknili smo desni klik > "Uporabniški vmesnik" > "Gumb – TextMesh Pro". Ker »TextMesh Pro« še nismo uporabljali, smo ga morali najprej uvoziti z oknom, ki se nam je prikazalo. Nato se nam je prikazal gumb, ki je bil zelo velik, zato smo spremenili njegovo velikost in velikost njegovega platna. Prav tako smo izboljšali ločljivost gumba tako, da smo povečali število pikslov na enoto (ang. pixel per Unit Multiplier). Označili smo, da je gumb interaktiven, in pri metodi, ki se je sprožila na klik smo pritisnili plus. Zgornjo ikono smo pustili, v spodnjo ikono pa smo povlekli iz seznama objektov objekt »Platno«, v katerega smo prej dodali skripto »OdpriPovezavo«. Ko smo povlekli Platno na želeno mesto, smo lahko v skrajno desnem okencu izbrali funkcijo »OpenUrl« iz »Platna«. (Clip Collection Vault, 2022)

2.4.11 Dodajanje vnosa prvoosebnega krmilnika

Gumbi so bili funkcionalni, vendar smo morali omogočiti, da bo prvoosebni krmilnik lahko kliknil nanje. Dodali smo novo komponento »Vnos igralca« na naš prvoosebni krmilnik. Ob dvojnem kliku na okence "Akcije" se je prikazal seznam dejanj, kjer smo dodali novo dejanje za levi klik miške (interakcija s predmeti). Pri dejanjih smo kliknili plus in ga poimenovali »Levi gumb miške«. Nato smo ob kliku na puščico poleg »Levi gumb miške« vstavili »Miška« > »Levi gumb«.

Nato smo ustvarili Interakcije, kjer smo izbrali »Le pritisnjeno«, in omogočili samodejno shranjevanje v zgornjem desnem kotu.

Po teh nastavitvah smo ustvarili »Upravljalnik igre« in »Upravljalnik igralca« kot prazna igralna predmeta.

Prenesli smo sredstvo »Quick Outline« in ustvarili skripto »Interaktivno«, ki smo jo povezali z gumbi. Skripta »Interaktivno« je prikazana spodaj v izvlečku (Izvleček kode 2).

```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;
using UnityEngine.Events;

public class Interactable : MonoBehaviour
{
```

```

Outline outline;
public string message;

public UnityEvent onInteraction;

void Start()
{
    outline = GetComponent<Outline>();
    DisableOutLine();
}

public void Interact()
{
    onInteraction.Invoke();
}

public void DisableOutLine()
{
    outline.enabled = false;
}

public void EnableOutLine()
{
    outline.enabled = true;
}
}

```

Izveček kode 2: Programska koda s skripto Interaktivno (Chris Nolet, 2022)

Dodali smo novo značko »Interaktivno« in to značko nastavili za gumb v pregledniku. Nato smo napisali novo skripto »Interakcija igralca«, ki smo jo povlekli na prvoosebni krmilnik. V skripto smo zapisali programsko kodo (Izveček kode 3).

```

using System.Collections;
using System.Collections.Generic;
using UnityEngine;

public class PlayerInteraction : MonoBehaviour
{
    public float playerReach = 3f;
    Interactable currentInteractable;

    void Update()
    {
        CheckInteraction();
        if (Input.GetKeyDown(KeyCode.F) && currentInteractable != null)
        {
            currentInteractable.Interact();
        }
    }

    void CheckInteraction()
    {
        RaycastHit hit;
        Ray ray = new Ray(Camera.main.transform.position, Camera.main.transform.forward);
    }
}

```

```

if (Physics.Raycast(ray, out hit, playerReach))
{
    if (hit.collider.tag == "Interactable")
    {
        Interactable newInteractable = hit.collider.GetComponent<Interactable>();

        if (currentInteractable && newInteractable != currentInteractable)
        {
            currentInteractable.DisableOutLine();
        }

        if (newInteractable.enabled)
        {
            SetNewCurrentInteractable(newInteractable);
        }
        else
        {
            DisableCurrentInteractable();
        }
    }
    else
    {
        DisableCurrentInteractable();
    }
}
else
{
    DisableCurrentInteractable();
}
}

void SetNewCurrentInteractable(Interactable newInteractable)
{
    currentInteractable = newInteractable;
    currentInteractable.EnableOutLine();
}

void DisableCurrentInteractable()
{
    if (currentInteractable)
    {
        currentInteractable.DisableOutLine();
        currentInteractable = null;
    }
}
}

```

Izvleček kode 3: Programska koda za interakcijo igralca (Chris Nolet, 2022)

Ko smo skripto shranili, so gumbi delovali, vendar smo jih morali še ustrezno poimenovati, spremeniti barvo in nastaviti njihove povezave. Vsak gumb smo povezali z ustrezno spletno povezavo v skripti, ki je vodila do vsebin, povezanih z gumbom. Spletne povezave, ki so prej vodile do Googla, smo zamenjali z novimi spletnimi

naslovi. Tako smo ustvarili za vsak gumb svojo kodo, pri tem pa smo spreminjali samo povezavo in ime kode. (Chris Nolet, 2022)

2.4.12 Dodajanje pozdravnega okna

Na začetku igre smo dodali pozdravno okno. To okno smo podvojili in ga postavili tudi pred vhod v sobo pobega, nato pa smo napisali skripto za zaprtje tega okna. Programska koda je zapisana v izvlečku (Izvleček kode 4).

```
using UnityEngine;
using UnityEngine.UI;
using UnityEngine.Events;

namespace EasyUI.Dialogs{

    public class Dialog{}

    public class DialogUI : MonoBehaviour {
        [SerializeField] GameObject canvas;
        [SerializeField] Button closeButton;
        Dialog dialog = new Dialog ( );

        public static DialogUI Instance;

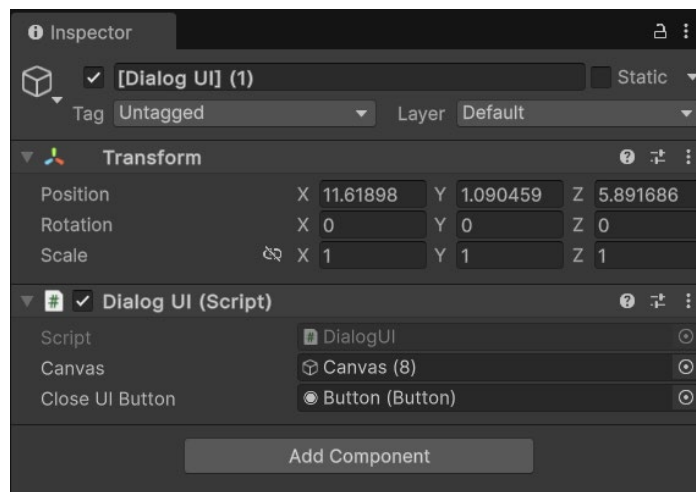
        void Awake ( ) {
            Instance = this;
            closeButton.onClick.RemoveAllListeners ( );
            closeButton.onClick.AddListener ( Hide );
        }

        public void Show ( ) {
            canvas.SetActive (true);
        }

        public void Hide ( ) {
            canvas.SetActive (false);
            dialog = new Dialog ( );
        }
    }
}
```

Izvleček kode 4: Programska koda za zaprtje pozdravnega okna (Herbou, 2021)

V »Dialogu uporabniškega vmesnika« smo ustvarili novo komponento in povezali gumb z oknom (Slika 5). (Herbou, 2021)

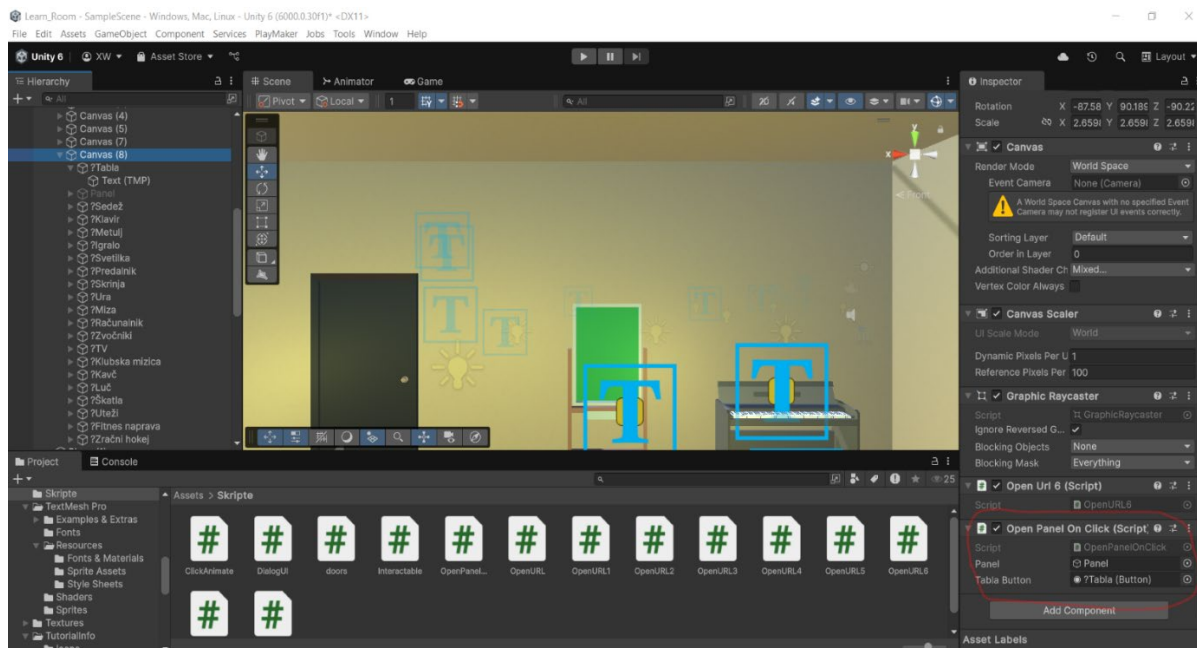


Slika 5: Prikaz vstavljenega kode pod objektom [Dialog UI] (1) in pravilno povezanega platna in gumba za zapiranje (lasten vir)

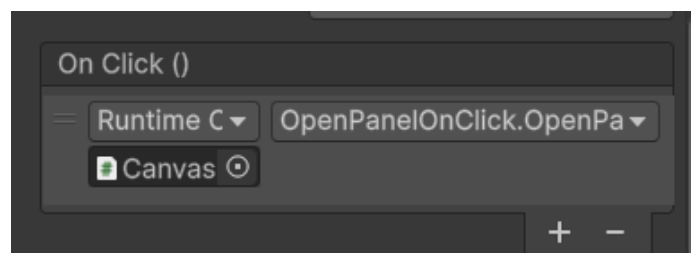
2.4.13 Dodajanje gumbov na predmete v sobi

Kopirali smo obstoječ gumb, ga pomanjšali, spremenili besedilo v "?", prilagodili pisavo in ga nato večkrat podvojili, da smo dobili skupno 20 gumbov. Ob kliku na gumb se je odprlo polje z besedilom. V drevesni strukturi datotek pod platnom smo ustvarili ploščo, ji nastavili črno ozadje in dodali besedilo. Besedilo je vsebovalo vprašanje s tremi možnimi odgovori, pri čemer je prva črka pravilnega odgovora vodila do naslednjega predmeta z vprašanjem. Plošča je bila ob začetku skrita in se ni prikazala takoj. Prikaz nastavitve povezav med objekti je viden na spodnji sliki (Slika 6).

Nato smo napisali programsko kodo, ki bo ob kliku na gumb odprla ploščo oz. naše besedilo. Ustvarili smo novo skripto »MonoBehaviour« in jo poimenovali »OdpriPanelNaKlik«. Preizkusili smo ali ChatGPT zmore napisati preprosto kodo, ki bi odprla ploščo ob kliku na gumb. ChatGPT nam je napisal kodo, ki smo jo nato dodali na platno in platno povlekli v funkcijo »NaKlik«. (Slika 7) (ChatGPT, 2025)



Slika 6: Prikaz pravih povezav med objekti (glejte obkroženo vsebino na sliki) (lasten vir)



Slika 7: Prikaz pravilne povezave »NaKlik« funkcije (lasten vir)

Pa vendar, ko smo zagnali program, se ob kliku na gumb ni zgodilo nič. Programsko kodo smo že velikokrat poskusili napisati s ChatGPT-jem, vendar nam je vsakič napisal napačno programsko kodo. Včasih nam je konzola javljala napake, včasih pa ne, vendar se po zagonu nič ni zgodilo, kot bi se moralo. Iz izkušenj smo vedeli, da je najbolje uporabljati programsko kodo, ki so jo napisali ljudje. Zato smo izbrisali ChatGPT-jevo kodo in jo nadomestili z novo, prepisano iz video vodiča. Oseba iz video vodiča je za to napisala preprosto programsko kodo, ki je prikazana v spodnjem izvlečku kode (Izvleček kode 5).

```
using UnityEngine;
```

```
public class OpenPanelOnClick : MonoBehaviour
{
    public GameObject Panel;
```

```

public void OpenPanel()
{
    if(Panel != null)
    {
        Panel.SetActive(true);
    }
}

```

Izvoleček kode 5: Programska koda za odprtje plošče ob kliku na gumb (Jayanam, 2019)

Nato smo na mesto vstavljenega koda umetne inteligence vstavili to novo kodo. Ko smo zagnali program, je vse skupaj delovalo! Vendar, kaj če bi dodali, da se ob ponovnem kliku na gumb panel zapre, tako bi ohranili večjo preglednost nad prostorom. Dodali smo skripto, ki je zapisana v izvlečku (Izvoleček kode 6).

```

using System.Collections;
using System.Collections.Generic;
using UnityEngine;

public class OpenPanelOnClick : MonoBehaviour
{
    public GameObject Panel;

    public void OpenPanel()
    {
        if(Panel != null)
        {
            bool isActive = Panel.activeSelf;

            Panel.SetActive(!isActive);
        }
    }
}

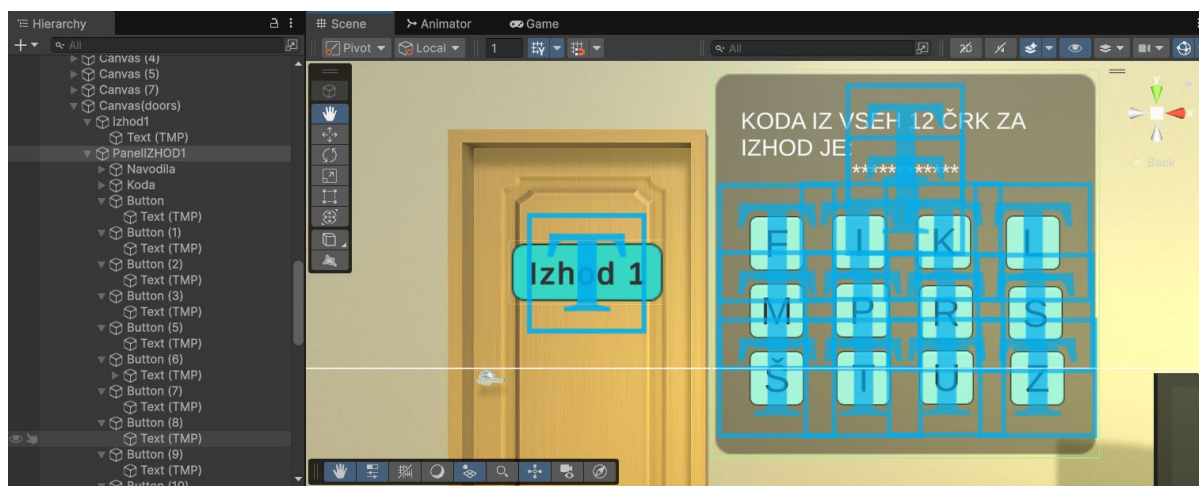
```

Izvoleček kode 6: Programska koda za zaprtje panela ob ponovnem kliku na gumb (Jayanam, 2019)

Ploščo smo podvojili za vseh dvajset gumbov in prilagodili besedilo vsakega panela z unikatnim vprašanjem ter odgovori. Igralci bodo s pravilnimi odgovori sestavljali 12-črkovno geslo, pri čemer vsaka pravilna izbira razkrije novo črko, ki jih vodi do predmeta, katerega ime se začne na to črko. Ker se zavedamo, da lahko predmete poimenujemo s sopomenkami, ima vsak predmet ob odprtju panela zapisano tudi ime predmeta. Na koncu bodo morali na ploščo pri pravilnih izhodnih vratih vnesti celotno geslo, da pobegnejo iz sobe pobega. (Jayanam, 2019)

2.4.14 Dodajanje vrat v sobo pobega

Na tri stene sobe pobega smo dodali vrata, vsaka s svojim gumbom, ki prikazuje številko izhoda. Že ustvarjen gumb (gumb pri predmetu) smo kopirali, ga preimenovali, prilagodili obliko in besedilo ter dodali nov panel z navodili in poljem za vnos kode, prikazanim kot zvezdice. Katera vrata so pravilna izvemo pri zadnjem vprašanju, pri katerem so namesto črk pred odgovori navedeni: »Izhod 1«, »Izhod 2« in »Izhod 3«. Pravilen odgovor stoji za »Izhod 2«. V kolikor bodo igralci ugotovili pravilen odgovor bodo lahko svoj odgovor vnesli v pravilna vrata (Izhod 2), ki jim bodo v primeru vnosa pravilnega gesla ponudila gumb za izhod. Nato smo dodali 12 gumbov za črke, vse oblikovali in premaknili na pravo mesto. Izgled panela je prikazan na spodnji sliki (Slika 8). Celotno platno smo dvakrat skopirali za preostala vrata ter vse panele skrili, kar pomeni, da se ob zagonu igre ne pokažejo.



Slika 8: Izgled panela za vnos gesla (lasten vir)

Ustvarili smo skripto »Sistem za geslo« in jo dodali v vsako platno posebej. Nato smo v pregledniku pri vsakem črkovnem gumbu nastavljal vrednost, ki jo predstavlja. Ko smo povezali polje za prikaz kode, smo ugotovili, da skripta uporablja starejše komponente, zato smo jo prilagodili za »TextMeshPro« gumbe in besedilo. Posodobljena skripta je prikazana spodaj (Izvleček kode 7).

```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;
using UnityEngine.UI;
using TMPro;
```

```

public class PasscodeSystem : MonoBehaviour
{
    public TextMeshProUGUI textDisplay;
    string userText = "";

    void Start()
    {

    }
    void Update()
    {

    }

    public void buttonPress(string txt){
        userText += txt;
        textDisplay.text = userText;
    }
}

```

Izveček kode 7: Programska koda za izpis uporabniškega vnosa v polje za vnos gesla. (Gemini, 2025)

Nato smo omejili vnos na 12 črk in dodali preverjanje pravilnosti kode po vnosu. Posodobljena koda je prikazana v spodnjem izvlečku (Izveček kode 8).

```

using System.Collections;
using System.Collections.Generic;
using UnityEngine;
using UnityEngine.UI;
using TMPro;

public class PasscodeSystem : MonoBehaviour
{
    public string correctPasscode = "IPRŠLFZUMTKS";
    public TextMeshProUGUI textDisplay;
    string userText = "";

    void Start()
    {

    }

    void Update()
    {

    }

    public void buttonPress(string txt){

        userText += txt;
        textDisplay.text = userText;

        if(userText.Length == 12){
            if(userText == correctPasscode){

```

```

        textDisplay.text = "ČESTITAMO!";

    }else{
        textDisplay.text = "NAPAČNO!";
    }
}else if(userText.Length > 12) {
    userText = "";
    textDisplay.text = userText;
}
}
}
}

```

Izvleček kode 8: Posodobljena programska koda »Sistem za geslo« z dodanim preverjanjem števila črk in gesla. (Thirteenov, Unity "Enter Correct Passcode" Tutorial - Part 3 : The wrong and correct passcode, 2024) (Thirteenov, Unity "Enter Correct Passcode" Tutorial - Part 2 : Displaying each number entered, 2024)

2.4.15 Dodajanje gumba za izhod

Dodali smo gumb za izhod, ki se pojavi ob napisu »ČESTITAMO!«. V Unity smo ustvarili nov gumb (lahko smo kopirali obstoječega), nato smo napisali kodo za izhod in jo povezali v platno, ki je bilo »starš« gumba (hierarhično gledano). Koda za izhod je zapisana v spodnjem izvlečku (Izvleček kode 9).

```

using System.Collections;
using System.Collections.Generic;
using UnityEngine;

public class Quit : MonoBehaviour
{
    public void QuitGame()
    {
        Debug.Log("Quit!");
        Application.Quit();
    }
}

```

Izvleček kode 9: Programska koda za izhod iz igre (ChatGPT, 2025) (Gemini, 2025) (Kap Koder, 2021)

Ker funkcija »Quit()« ni delovala v urejevalniku Unity, vendar je kasneje delovala pri izvoženi verziji programa, smo delovanje metode »Quit()« testirali z izpisovanjem sporočil v konzoli. Slednje smo storili s pomočjo funkcije »Debug.Log«.

Funkcijo za gumb smo dodali v našo kodo »Sistem za geslo« (Izvleček kode 10).

```

using System.Collections;
using System.Collections.Generic;
using UnityEngine;
using UnityEngine.UI;
using TMPro;

```

```

public class PasscodeSystem : MonoBehaviour
{
    public string correctPasscode = "IPRŠLFZUMTKS";
    public TextMeshProUGUI textDisplay;
    public Button successButton;
    string userText = "";

    //Start is called before the first frame update
    void Start()
    {
        successButton.gameObject.SetActive(false);
    }

    //Update is called once per frame
    void Update()
    {

    }

    public void buttonPress(string txt){

        userText += txt;
        textDisplay.text = userText;

        if(userText.Length == 12){

            if(userText == correctPasscode){
                textDisplay.text = "ČESTITAMO!";
                successButton.gameObject.SetActive(true);
            }
            else{
                textDisplay.text = "NAPAČNO!";
            }
        }
        else if(userText.Length > 12) {
            userText = "";
            textDisplay.text = userText;
        }
    }
}

```

Izvleček kode 10: Posodobljena programska koda »Sistem za geslo« z dodanim gumbom za izhod ob vnosu pravilnega gesla (ChatGPT, 2025) (Gemini, 2025) (Kap Koder, 2021)

Pri tem smo samo malo spremenili kodo tako, da smo na začetek dodali gumb. Na začetku smo nastavili, da je gumb skrit in na koncu pri pogojnem stavku pod "ČESTITAMO!" dodali, da se pojavi še ta gumb. Nato smo šli v drevesno strukturo datotek, kjer smo poiskali naše platno, ki že ima to skripto, in v mesto za gumb povlekli naš gumb za izhod.

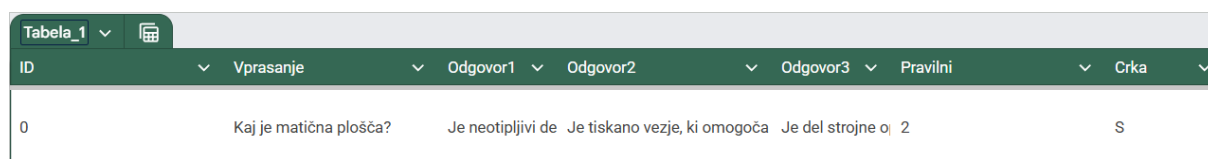
Ker se nam je prekrivalo "ČESTITAMO!" in gumb za izhod, smo na panelu z zvezdicami nastavili, da bo pomaknjen bolj na levo stran. V sistem za geslo smo dodali,

da se gumb prikaže ob pravilnem geslu. V platno smo povlekli gumb in prilagodili postavitev, da se ni prekrival z besedilom. Ker se je "ČESTITAMO!" prikazalo tudi pri napačnih vratih, smo podvojili skripto in jo prilagodili, da tam izpiše "NAPAČNA VRATA!". Nato smo v obeh platnih zamenjali staro skripto, ponovno povezali geslo in besedilo ter nastavili funkcije »OnClick« za črkovne gumbe.

Na začetni pogled kamere smo dodali besedilo "Premakni miško in poišči navodila.", da smo igralca usmerili. (ChatGPT, 2025) (Gemini, 2025) (Kap Koder, 2021)

2.4.16 Nadgraditev spreminjanja besedila v programu tudi po izvozu s pomočjo Google preglednic

V hodniku modrosti smo že omogočili spreminjanje učnega gradiva tudi po izvozu programa, saj nas ob kliku na gumb preusmeri v oblak v Canvi, kjer se gradivo vsakič ponovno posodobi. Razmišljali smo, da bi želeli spreminjati tudi vsa besedila po izvozu programa. V ta namen smo se odločili ustvariti direktno povezavo na Google preglednice, ki nam bo omogočila spreminjanje besedila po izvozu igre iz Unity - ja. To smo naredili tako, da smo ustvarili v Google preglednicah novo tabelo. V njo smo vpisali naslove in pod njih vpisali ustrezno vprašanje, indeks vprašanja (ID) in možne odgovore, na koncu pa še številko pravilnega odgovora in črko le tega. Postopek vam bomo predstavili na primeru besedila panela, ki se odpre ob kliku na gumb pri tabli v sobi pobega. Preglednico si lahko ogledate na sliki. (Slika 9)



| ID | Vprašanje | Odgovor1 | Odgovor2 | Odgovor3 | Pravilni | Crka |
|----|------------------------|-------------------|------------------------------|------------------|----------|------|
| 0 | Kaj je matična plošča? | Je neotipljivi de | Je tiskano vezje, ki omogoča | Je del strojne o | 2 | S |

Slika 9: Tabela v Google preglednicah za črpanje podatkov v Unity (lasten vir)

Sledilo je pisanje programskih kod, s katerimi smo povezali Unity s CSV podatki iz tabele. Za to smo naredili tri kode. Prva koda je za shranjevanje podatkov in smo jo poimenovali »QuestionData«. (Izvleček kode 11) (ChatGPT, 2025)

```
using System;  
using System.Collections.Generic;  
  
[Serializable]  
public class QuestionData
```

```

{
    public int ID;
    public string Vprasanje;
    public List<string> Odgovori;
    public int Pravilni;
    public string Crka;
}

```

Izvleček kode 11: Koda za shranjevanje podatkov (ChatGPT, 2025)

Nato smo napisali programsko kodo za nalaganje in prikaz vprašanj, ki smo jo poimenovali »QuestionManager«. (Izvleček kode 12) (ChatGPT, 2025)

```

using System.Collections;
using System.Collections.Generic;
using UnityEngine;
using UnityEngine.Networking;
using TMPro;
using System.Text.RegularExpressions;

public class QuestionManager : MonoBehaviour
{
    public string sheetUrl;
    public List<QuestionData> questions = new List<QuestionData>();

    public TextMeshProUGUI fullTextDisplay;

    void Start()
    {
        StartCoroutine(LoadQuestionsFromGoogleSheet());
    }

    IEnumerator LoadQuestionsFromGoogleSheet()
    {
        UnityWebRequest www = UnityWebRequest.Get(sheetUrl);
        yield return www.SendWebRequest();

        if (www.result == UnityWebRequest.Result.Success)
        {
            string data = www.downloadHandler.text;
            ParseCSV(data);
            DisplayQuestion(0);
        }
        else
        {
            Debug.LogError("Napaka pri nalaganju CSV: " + www.error);
        }
    }

    void ParseCSV(string csv)
    {
        questions = new List<QuestionData>();

        string[] rows = csv.Split('\n');
        for (int i = 1; i < rows.Length; i++) // Skip header
        {
            string line = rows[i].Trim();
            if (string.IsNullOrEmpty(line)) continue;

```

```

// 🚧 Pametno razbijanje vrstice - podpira vejice znotraj
narekovajejev
string[] cols = Regex.Split(line,
", (?=(?:[^\"]*" * "[^\"]*" * ") * [^\"]*$)");

for (int c = 0; c < cols.Length; c++)
{
    cols[c] = cols[c].Trim().Trim(' ');
}

if (cols.Length < 7)
{
    Debug.LogWarning($"! Preskočena vrstica {i}, ni dovolj
stolpcev ({cols.Length}): {line}");
    continue;
}

if (!int.TryParse(cols[0], out int id))
{
    Debug.LogWarning($"! Napaka pri ID v vrstici {i}:
'{cols[0]}');
    continue;
}

if (!int.TryParse(cols[5], out int pravilni))
{
    Debug.LogWarning($"! Napaka pri Pravilni v vrstici {i}:
'{cols[5]}');
    continue;
}

QuestionData q = new QuestionData
{
    ID = id,
    Vprasanje = cols[1],
    Odgovori = new List<string> { cols[2], cols[3], cols[4] },
    Pravilni = pravilni,
    Crka = cols[6]
};

questions.Add(q);
}

Debug.Log($"✔ Naloženih {questions.Count} vprašanj.");
}

void DisplayQuestion(int index)
{
    if (index < questions.Count)
    {
        QuestionData q = questions[index];

        string output = "TABLA\n\n"; // ☐ Naslov
        output += q.Vprasanje + "\n\n"; // !? Vprašanje

        // 📄 Dodaj odgovore s črkami
        char[] crke = { 'R', 'S', 'T' };
    }
}

```

```

        for (int i = 0; i < q.Odgovori.Count; i++)
        {
            output += $"{crke[i]} {q.Odgovori[i]}\n";
        }

        if (fullTextDisplay != null)
        {
            fullTextDisplay.text = output;
        }

        Debug.Log("☞ Prikazano besedilo:\n" + output);
    }
}

```

Izveček kode 12: Koda za nalaganje in prikaz vprašanj (ChatGPT, 2025)

Na koncu pa smo še dodali »QuestionDisplay«, ki poskrbi za prikaz in omogoča tudi razvrščanje po indeksu (ID) iz tabele. (Izveček kode 13) (ChatGPT, 2025)

```

using UnityEngine;
using TMPro;

public class QuestionDisplay : MonoBehaviour
{
    public TextMeshProUGUI questionText;
    private QuestionManager questionManager;

    void Start()
    {
        questionManager = FindObjectOfType<QuestionManager>();

        if (questionText != null && questionManager != null &&
            questionManager.questions.Count > 0)
        {
            UpdateQuestion(0);
        }
    }

    public void UpdateQuestion(int index)
    {
        if (questionManager != null && index <
            questionManager.questions.Count)
        {
            QuestionData q = questionManager.questions[index];

            string output = "TABLA\n\n";
            output += q.Vprasanje + "\n\n";

            char[] crke = { 'R', 'S', 'T' };
            for (int i = 0; i < q.Odgovori.Count; i++)
            {
                output += $"{crke[i]} {q.Odgovori[i]}\n";
            }

            questionText.text = output;
        }
    }
}

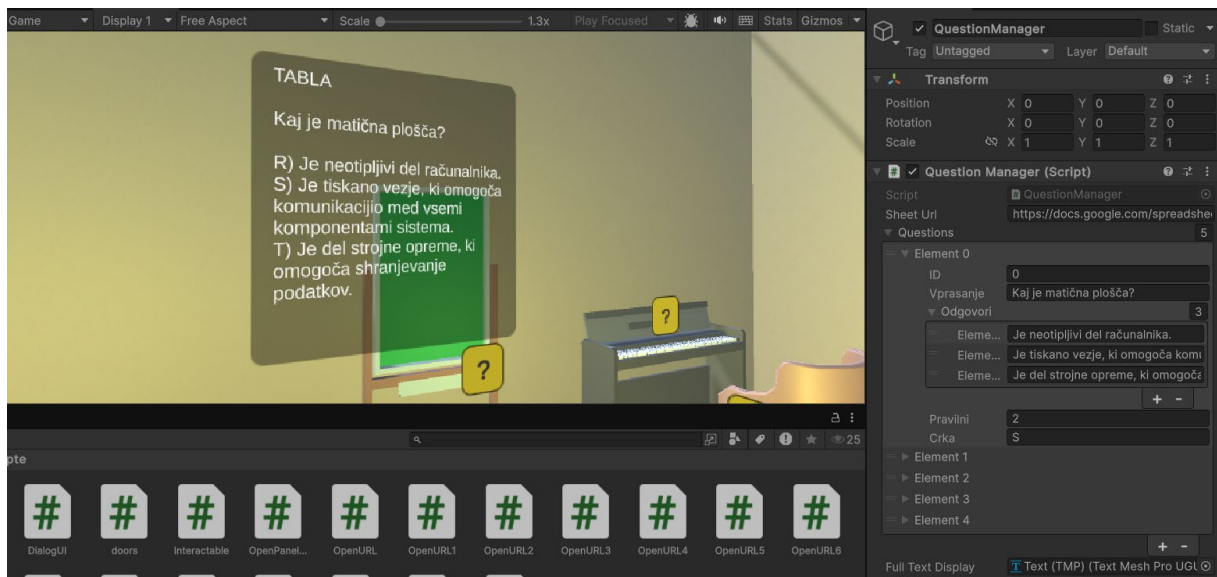
```

}

}

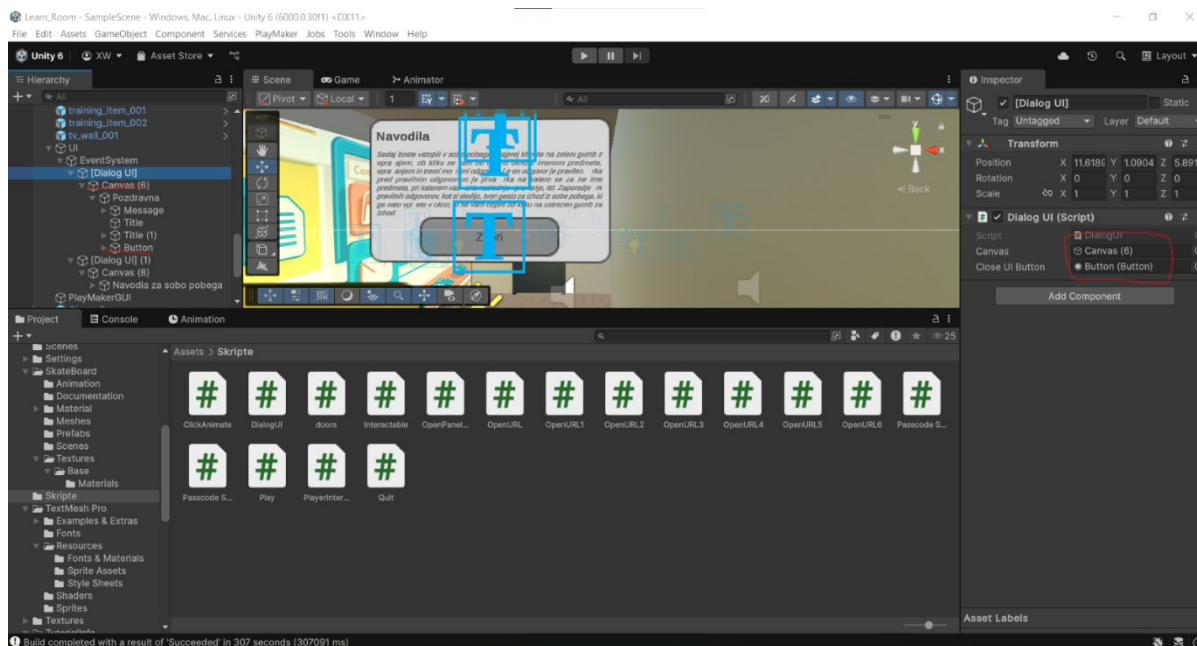
Izveček kode 13: Koda za prikaz in razvrščanje podatkov po indeksu (ChatGPT, 2025)

Nato smo poskrbeli še samo za to, da smo kodo pravilno povezali v Unity - ju, in sicer smo v drevesni strukturi datotek ustvarili nov prazen igralni objekt, ki smo ga poimenovali »QuestionManager«. Vanj smo prilepili isto imenovano kodo. »QuestionData« smo pustili med sredstvi v mapi s programskimi kodami, »QuestionDisplay«, pa smo prilepili na ploščo, ki vsebuje besedilo, ki ga bomo spreminjali. Povezave v pregledniku smo uredili tako, da smo v polje, ki se je prikazalo, ko smo dodali »QuestionDisplay« povlekli besedilo pod tem panelom (tisto, ki ga bomo spreminjali). Pri »QuestionManagerju« pa smo morali prilepiti povezavo od Google tabele, ki smo jo dobili tako, da smo v Google preglednicah kliknili na »Datoteke« > »Deljenje« > »Objavi v splet«, nato smo pri okencu v katerem je pisalo »Spletna stran« spremenili na »Vrednosti, ločene z vejico (csv).« in stisnili »Objavi«. Nato se nam je prikazal naslov CSV datoteke, ki smo ga skopirali v to polje v Unity - u. Po tem smo nastavili 3 odgovore in povlekli tekst, ki ga bomo spreminjali v prazno polje, kot lahko vidite na sliki. (Slika 10)



Slika 10: Slika scene s prikazanimi podatki iz tabele in preglednik z ustreznimi povezavami in naloženimi podatki iz Google tabele (lasten vir)

2.4.17 Odpravljanje napak

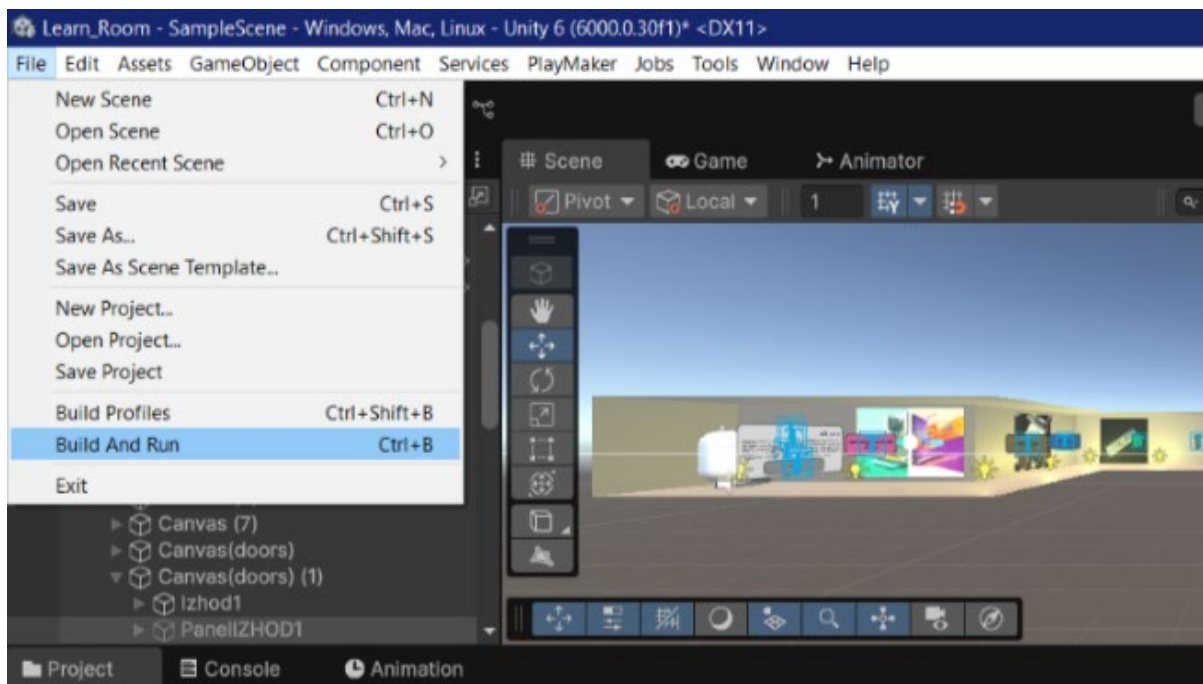


Slika 11: Na sliki so označeni popravki, ki smo jih naredili (lasten vir)

Kliknili smo na napako v konzoli. Označilo nam je, da so v platnu napake. V platnu smo popravili ustrezne povezave, kot je prikazano na sliki zgoraj. (Slika 11).

2.4.18 Izvoz igre

V Unity - ju smo igre izvažali tako, da smo najprej odprli meni »Datoteka« in izbrali možnost »Zgradi in zaženi« (Slika 12). Nato se je prikazalo okno za izbiro mape, kamor smo shranili izvožene datoteke. Po potrditvi shranjevanja je Unity začel postopek gradnje igre, ki je trajal nekaj časa. Ko je bil izvoz končan, se je igra samodejno zagnala za preizkus.



Slika 12: Kako izvoziti igro (lasten vir)

2.5 OPOMBE

V projektu smo uporabili še naslednje pripomočke za pohištvo: »Nizkopolični Predal Animiran«, (Studio EK, 2023) »Metulj« (INNOWELL GmbH, 2016) in »Spalnica / Notranjost - Nizkopolični pripomočki«. (Fries and Seagull, 2025)

Na začetku smo imeli stene bolj svetle barve, vendar smo jih po dodajanju slik spremenili na temnejšo barvo, da so slike prišle do večjega izraza.

Kasneje smo v projekt vgradili še eno sredstvo, ki nam je po prenosu odstranilo snovi iz nekaterih predmetov in spremenilo osvetljavo, itd. Projekt se je takrat v veliki meri sesul, zato smo morali odstraniti sredstvo in reševati težave. Zaradi tega se lahko končni izdelek nekoliko razlikuje od slik, ki so nastale med razvojem igre. Prav tako je celotna igra še v razvoju, se izboljšuje, zato si pridružujemo pravico do sprememb.

2.6 POVZETEK METODOLOGIJE DELA

Naša metodologija dela je sestavljena iz izbire razvijalskega orodja, načrtovanja postavitve učnega gradiva v hodniku modrosti in načrtovanja sistema za izhod iz sobe pobega. Sledilo je modeliranje 3D prostorov v Unity - ju, urejanje osvetlitve in dodajanje

tekstur. Nato smo dodali prvoosebni krmilnik za realistično doživljanje. V hodnik modrosti smo na stene namestili slike, dodali vrata med prostoroma in poskrbeli za odpravo napak pri prehajanju prvoosebnega krmilnika skozi vrata. Sledilo je urejanje sobe pobega v katero smo dodali pohištvo. V Canvi smo izdelali vsa učna gradiva za učence, ki smo jih delili kot povezavo. To povezavo smo vstavili vsako posebej v svojo programsko kodo, ki je omogočala ob kliku preusmeritev na spletno stran na povezavi. Nato smo morali poskrbeti za interakcijo prvoosebnega krmilnika, da je lahko klikal na gumb. Sledilo je dodajanje dveh pozdravnih oken z navodili, eno na začetek hodnika, drugo pa pred vstopom v sobo pobega. Potem smo se lotili nadaljnjega razvijanja sobe pobega, in sicer smo dodali gumb na predmete v sobi, ob kliku na katerega se ti je odprlo okno z imenom predmeta, vprašanjem in tremi možnimi odgovori. Pred koncem smo v sobo pobega dodali tri vrata, vsaka s svojim gumbom, ki odpre panel za vnos gesla. Le pri pravih vratih (pravilen odgovor pri zadnjem vprašanju nam pove, katera vrata so pravilna) se ob pravilnem vnosu gesla (ki ga dobimo iz pravih zaporedja črk pred pravih odgovori v sobi pobega) pojavi gumb za izhod s katerim lahko zapustimo program. S pomočjo CSV – ja smo uredili povezavo med Unity in Google preglednicami ter s tem omogočili spreminjanje besedil tudi po izvozu igre. Čisto na koncu smo še odpravili napake in igro izvozili.

Pri delu smo izbrali razvijalsko orodje Unity, saj nudi veliko podporo skupnosti in omogoča izvoz na različne platforme. Sledilo je načrtovanje, ki smo ga opravili v grafičnem orodju Canva. Po načrtovanju smo s pomočjo video vodičev, pisali kode in se seznanjali z delom v Unity - u. Za pomoč pri prilagajanju programskih kod in za manjše nasvete pri izvedbi našega izdelka smo uporabili tudi umetni inteligenci ChatGPT in Gemini. Podatki, ki sta jih zagotovila orodji, so bili vselej pregledani, prilagojeni in verificirani z naše strani. Za analizo razvijalskih orodij, interesov ljudi po 2D in 3D igrah ter za splošna dejstva pa smo uporabljali spletne članke. Vsi zgoraj navedeni viri, so navedeni za besedili v katerih smo jih uporabili.

3 REZULTATI IN INTERPRETACIJA REZULTATOV

Razvoj igre v programskem okolju Unity je pokazal, da je mogoče ustvariti delujoč 3D hodnih modrosti in sobo pobega, ki učencem omogoča interaktivno raziskovanje učne snovi. V okviru projekta je bil izdelan in preizkušen inovativni model učnega okolja, ki združuje pridobivanje znanja in njegovo preverjanje skozi elemente igre.

Osrednji del igre predstavlja interaktivni hodnik modrosti, kjer učenci prek gumbov dostopajo do različnih virov, kot so videoposnetki, predstavitve in miselni vzorci. Ta del omogoča sistematično učenje ključnih konceptov, preden se učenci soočijo z izzivi v sobi pobega.

Soba pobega služi kot preverjanje znanja – učenci morajo z uporabo pridobljenih informacij pravilno odgovoriti na vprašanja in sestaviti geslo, ki jim omogoči pobeg iz sobe pobega, kar pomeni uspešno osvojeno novo znanje. Ta pristop spodbuja aktivno sodelovanje in poglobljeno razumevanje učne snovi.

Eden ključnih ciljev raziskave je bil tudi preizkus možnosti sprotne posodobitve gradiva, kar nam je uspelo, saj smo ustvarili univerzalno igro, ki jo je tudi po izvozu možno spreminjati s pomočjo Canve in Google preglednic. S tem smo omogočili njeno večkratno uporabo za različne učne snovi, saj lahko učiteljice ali učitelji sami spreminjajo vse vsebine tako hodnika modrosti, kot sobe pobega. Ta rešitev nam omogoča, da se vsebine v igri posodablja brez ponovnega izvoza programa, kar zagotavlja večjo prilagodljivost in uporabnost učnega okolja. Igra se lahko uporabi za katerokoli učno snov, ne samo za učno snov računalništva.

Podobno kot že v uvodu omenjeni raziskavi, je tudi ta projekt pokazal, da 3D okolja ponujajo poglobljeno in interaktivno učno izkušnjo, ki spodbuja aktivno učenje in večjo motiviranost učencev. Kar lahko potrdimo tudi z navedbami Akademskega centra za zobozdravstvo Amsterdam, ki so raziskali uspešnost in zadovoljstvo študentov pri delanju v 2D in 3D učnem okolju. 95% sodelujočih študentov je izpolnilo vprašalnik in od tega 90% izbralo 3D, kot boljše učno okolje. Same študije so dokazale, da so študentje, ki so delali s 3D učnim okoljem dosegli bistveno boljše rezultate, kot tisti, ki so delali v 2D okolju. (Akademski center za zobozdravstvo Amsterdam, 2016)

3D igro sobe pobega in hodnika modrosti smo preizkusili tudi sami na desetih ljudeh. Ljudje so zavzeto igrali igro in vsi uspeli pobegniti iz sobe pobega. S tem je dokazano, da si je s popolno koncentracijo možno zapomniti podatke iz hodnika modrosti in jih nato uporabiti v sobi pobega.

Med samim izdelovanjem sta se ChatGPT in Gemini iznašla kot zelo koristni orodji. Z navedeno umetno inteligenco mi ni uspelo napisati delujoče kode in sem ju uporabljala v namene svetovanja in prilagajanja programskih kod. Torej ugotovila sem, da prva rešitev, ki mi jo je dal ni bila pravilna in sva jo morala še zelo dolgo izboljševati, iskati vzroke zakaj ne deluje tako kot želimo in odpravljati napake. Tako je ta postopek trajal precej dolgo in bi ga človek s predhodnim znanjem opravil mnogo hitreje.

4 DRUŽBENA ODGOVORNOST IN TRAJNOST

Projekt prispeva k družbeni odgovornosti na več načinov. Z zagotavljanjem brezplačne in dostopne učne platforme spodbuja enake možnosti za učenje, ne glede na socialni ali ekonomski status učenca. Interaktivna narava igre povečuje motiviranost in omogoča bolj učinkovito učenje, kar lahko dolgoročno prispeva k boljši izobrazbi in večji računalniški pismenosti mladih.

Poleg tega projekt spodbuja trajnostni razvoj, saj zmanjšuje potrebo po tiskanju učnih materialov in omogoča sprotno posodabljanje vsebin brez dodatnih stroškov. S tem se zmanjšuje negativni okoljski vpliv izobraževalnega procesa.

Končno pa igra promovira sodelovanje in skupinsko reševanje problemov v primeru igranja učnih sob pobega v skupinah, kar krepi socialne veščine učencev ter spodbuja kreativno razmišljanje in inovativnost, saj v primeru da ne vedo odgovora, morajo iskati druge možnosti, še kako bi lahko prišli do izhoda. V prihodnosti bi bilo smiselno takšne računalniške igre uporabiti v namen izobraževanja, saj bi tako imele še širši družbeni vpliv.

5 PRISPEVEK K NAPREDKU

Inovacijski predlog, ki smo ga podali, se osredotoča na razvoj interaktivnega digitalnega okolja s pomočjo programskega orodja Unity. Ta projekt ne prinaša le novih znanj in veščin udeležencem, temveč ima tudi širši pomen za razvoj in napredek na več področjih.

Ena izmed ključnih prednosti inovacijskega predloga je spodbujanje tehnološkega izobraževanja in digitalne pismenosti. S projektom smo razvili interaktivno izobraževalno sobo pobega, ki omogoča uporabnikom, da skozi igro pridobijo nova znanja o računalništvu. Takšne digitalne rešitve bi lahko uporabili v šolah in izobraževalnih ustanovah, s čimer bi prispevali k sodobnejšim učnim metodam in večji motivaciji dijakov za učenje STEM-področij.

Poleg tega naš predlog spodbuja inovacije na področju turizma in kulture. Interaktivne sobe pobega in digitalni izobraževalni hodniki bi lahko postali del turistične ponudbe Slovenije, ki bi privabila tako domače kot tuje obiskovalce. Pri tem si predstavljamo, da bi imeli pri turistični znamenitosti tablo s QR – kodo, ki bi turiste preusmerila v sobo pobega s hodnikom modrosti. V hodniku modrosti bi spoznavali nove informacije o znamenitosti, ki bi jih nato potrebovali za pobeg iz sobe pobega. Ta ideja je zelo revolucionarna, saj so fizične sobe pobega vedno enake in se jih hitro naveličamo. V nasprotju s fizičnimi sobami pobega, virtualna soba pobega omogoča nenehno posodabljanje in spreminjanje. Tako bi lahko npr. vsak mesec spremenili vsebino, ki jo prikazuje QR – koda. Vsak mesec bi lahko bili novi podatki o znamenitosti z novo sobo pobega in se tako ne bi nikoli naveličali sobe pobega, saj bi bila vsake toliko časa čisto drugačna. S tem bi se širilo znanje na zabaven način, prebivalci Slovenije, mladostniki in tujci bi se lahko naučili mnogo novega, kar bi prispevalo k večji turistični prepoznavnosti Slovenije, še posebej v tem času hitre digitalizacije.

Nadalje, naloga prispeva k razvoju podjetniškega okolja v Sloveniji. Usposobljenost za delo z orodji, kot sta Unity in programski jezik C#, je v tehnološki industriji izjemno iskana. Takšni projekti spodbujajo razvoj novih tehnoloških podjetij in start-upov, ki bi lahko prispevali k rasti lokalnega gospodarstva. Slovenija bi s tem postal privlačnejša

za mlade talente in investitorje, ki iščejo inovativne projekte na področju informacijske tehnologije.

Nenazadnje naš inovacijski predlog spodbuja sodelovanje med študenti, učitelji, raziskovalci in tehnološkimi podjetji. S tem se krepi ekosistem znanja in inovacij, kar dolgoročno pripomore k napredku Slovenije in njeni preobrazbi v tehnološko napredno okolje.

Naša naloga tako ni zgolj akademski projekt, temveč ima potencial za širšo uporabo v izobraževanju, turizmu, gospodarstvu in tehnološkem razvoju. Sodobne digitalne rešitve so ključne za prihodnost, saj omogočajo večjo prepoznavnost, boljše izobraževanje in nove priložnosti za gospodarski razvoj.

6 ZAKLJUČEK / SKLEPI

Skozi celoten projekt sem prišla do številnih ugotovitev. Moja najpomembnejša ugotovitev je, da lahko kot začetnik narediš takšen projekt, saj se na spletu najde potrebno gradivo, vendar pa moraš imeti veliko vztrajnosti in motivacije za izdelavo, da ne obupaš že ob prvem neuspelem poizkusu, saj je neuspeh kar ogromno. Zavedati se moramo, da spletni viri ne nadomestijo potreb po šolanju, saj menim, da bi z že predhodno pridobljenim znanjem (iz šolanja) ta projekt opravila mnogo hitreje.

Z izdelavo inovacijskega predloga smo pokazali, da je mogoče s pomočjo 3D svetov ustvariti učinkovito in spodbudno učno okolje za učence. Projekt dokazuje, da lahko takšno obliko učenja vključimo tudi v učni proces in da lahko po potrebi tudi spreminjamo in posodabljammo učno gradivo v hodniku modrosti. Z uporabo izobraževalnega hodnika in sobe pobega dosežemo usvajanje novega znanja, povečujemo zanimanje za učno snov in spodbujamo kreativno mišljenje.

Za nadaljnje delo bi bilo smiselno razmisliti o dodajanju več interaktivnih ali animiranih elementov, dodajanju časovnika, ki bi v sobi pobega odšteval čas in morda še možnost zbiranja podatkov uporabnika v primeru točkovanja.

7 LITERATURA

- Akademski center za zobozdravstvo Amsterdam. (2016). *pubmed*. Pridobljeno iz Student performance and appreciation using 3D vs. 2D vision in a virtual learning environment: <https://pubmed.ncbi.nlm.nih.gov/26072997/>
- Biostart. (2024). *Free Wood Door Pack*. Pridobljeno iz Asset Store: <https://assetstore.unity.com/packages/3d/props/interior/free-wood-door-pack-280509>
- Canva. (2025). Pridobljeno iz Canva: <https://www.canva.com/>
- ChatGPT. (2025). Pridobljeno iz ChatGPT: <https://chatgpt.com>
- Chris Nolet. (2022). *Quick Outline*. Pridobljeno iz Asset Store: <https://assetstore.unity.com/packages/tools/particles-effects/quick-outline-115488>
- Clack, N. (2023). *Demographics of Players of Platformer Games (2D and 3D)*. Pridobljeno iz Gamedev.net: <https://www.gamedev.net/blogs/entry/2276048-demographics-of-players-of-platformer-games-2d-and-3d/>
- Clip Collection Vault. (2022). *Unity Open URL / Links with a button Quick and Easy*. Pridobljeno iz Youtube: https://youtu.be/3L8bDWLfo_8
- Egor, P. (2025). *2D vs 3D Games: Can You Tell The Difference?* Pridobljeno iz iLOGOS: <https://ilogos.biz/2d-vs-3d-games-5-biggest-differences-between-2d-and-3d-games/>
- Fakulteta za elektrotehniko računalništvo in informatiko, M. (2024). *Osnove programiranja v programskem jeziku C++ (za dekleta)*. Pridobljeno iz FERi: <https://feri.um.si/akademija-feri/pocitniske-sole/osnove-programiranja-v-programskem-jeziku-c/>
- Fakulteta za elektrotehniko, računalništvo in informatiko. (2024). *Python programerke*. Pridobljeno iz FERi: <https://feri.um.si/akademija-feri/pocitniske-sole/python-programerke/>
- Fries and Seagull. (2025). *Bedroom / Interior - Low Poly assets*. Pridobljeno iz Asset Store: <https://assetstore.unity.com/packages/3d/props/interior/bedroom-interior-low-poly-assets-295074>
- Gemini. (2025). Pridobljeno iz Gemini: <https://gemini.google.com/>
- Herbou, H. (2021). *Popup window in Unity [Part 2 : Coding]*. Pridobljeno iz Youtube: <https://youtu.be/vTMI6hb3QF0>
- INNOWELL GmbH. (2016). *Butterfly (Animated)*. Pridobljeno iz Asset Store: <https://assetstore.unity.com/packages/3d/characters/animals/insects/butterfly-animated-58355>

- ithappy. (2025). *Furniture FREE - Low Poly 3D Models Pack*. Pridobljeno iz Asset Store: <https://assetstore.unity.com/packages/3d/props/furniture/furniture-free-low-poly-3d-models-pack-260522>
- Jayanam. (2019). *Unity Tutorial: Open Panel on Button Click*. Pridobljeno iz Youtube: <https://youtu.be/LzillB2Kt4>
- Johns, R. (2025). *Unity vs Unreal: Which Game Engine? [2025 Update]*. Pridobljeno iz Hackr.io: <https://hackr.io/blog/unity-vs-unreal-engine>
- Kap Koder. (2021). *How to Make a Quit Button in 3 MINUTES!? Quick & Simple Unity Tutorial*. Pridobljeno iz Youtube: <https://youtu.be/WRolw3ktZTE>
- Mladi podjetnik. (2022). *Pomen digitalne pismenosti v sodobnem času*. Pridobljeno iz Mladi podjetnik: <https://mladipodjetnik.si/podjetniski-koticek/marketing/pomen-digitalne-pismenosti-v-sodobnem-casu>
- Poly Haven. (2022). *Beige Wall 001*. Pridobljeno iz Poly Haven: https://polyhaven.com/a/beige_wall_001
- Poly Haven. (2023). *Laminate Floor 03*. Pridobljeno iz Poly Haven: https://polyhaven.com/a/laminate_floor_03
- Poly Haven. (2025). Pridobljeno iz Poly Haven: <https://polyhaven.com/>
- Reydar. (2023). *Top 5 Benefits of 3D Immersive Learning Platforms*. Pridobljeno iz Reydar: <https://www.reydar.com/top-5-benefits-of-3d-immersive-learning-platforms/>
- Simon Serge Pasi. (2021). *Mini First Person Controller*. Pridobljeno iz Asset Store: <https://assetstore.unity.com/packages/tools/input-management/mini-first-person-controller-174710>
- Studio EK. (2023). *Low Poly Chest Animated*. Pridobljeno iz Asset Store: <https://assetstore.unity.com/packages/3d/props/low-poly-chest-animated-247127>
- The Online Learning Space. (2021). *Creating a Simple Virtual Gallery | Made with Unity Tutorial*. Pridobljeno iz Youtube: <https://youtu.be/rNwEFVs4B4Y>
- ThirteenNov. (2024). *Unity "Enter Correct Passcode" Tutorial - Part 2 : Displaying each number entered*. Pridobljeno iz Youtube: <https://youtu.be/oGb2DjA0I5g>
- ThirteenNov. (2024). *Unity "Enter Correct Passcode" Tutorial - Part 3 : The wrong and correct passcode*. Pridobljeno iz Youtube: <https://youtu.be/Gz-0HChpoD4>
- Unity. (2024). *Create with Unity in three steps*. Pridobljeno iz Unity: <https://unity.com/download>