

Manuál k editoru

Jozef Skála

Úvod

Za účelom zjednodušenia a vlastne aj celkového umožnenia práce s editorom inštruktážnych zážitkov som vytvoril nasledujúci textovo/obrázkový návod. V tomto návode budem používať viacero termínov, ktoré by som vopred rád definoval.

- *Editor* - nástroj na tvorbu sférických inštruktážnych zážitkov
- *Inštruktážny zážitok* - Výstup editora a teda výsledná aplikácia
- *Unity* - vývojové prostredie na ktorom je postavený celý editor inštruktážnych videí
- *Tvorca* - osoba, ktorá vytvára pomocou editora inštruktážne videá
- *Klient/hráč* - osoba, ktorá bude využívať inštruktážny zážitok za cieľom si osvojiť pracovný proces

Potrebné nástroje

Na úvod je nutné uviesť, že na prácu s editorom je nutné disponovať štandardnými perifériami osobného počítača ako je myš a klávesnica. Myš by mala disponovať ľavým, pravým ako aj stredným tlačidlom (bežnejšie známe ako koliesko myši).



(1- ľavé, 2-pravé, 3-stredné tlačidlá)

Inštruktážne videá je možné vytvárať aj vo virtuálnej realite za pomoci technológie Oculus Air Link. Viac na:

<https://www.oculus.com/blog/introducing-oculus-air-link-a-wireless-way-to-play-pc-vr-games-on-oculus-quest-2-plus-infinite-office-updates-support-for-120-hz-on-quest-2-and-more/>

Ďalším predpokladom je aj nainštalované vývojové prostredie Unity. Doporučujem používať verziu 2020.1.17f1 v ktorej som samotný editor vytváral. S najväčšou pravdepodobnosťou však nebude problém s akoukoľvek vyššou verziou. V návode sa tiež budem snažiť čo najdôkladnejšie popísať aj prácu so samotným Unity, no na základe rozsahu tohto návodu budem v niektorých segmentoch uvažovať so základnou znalosťou tohto vývojového prostredia.

Ovládanie editora (PC):

- Periférie: myš a klávesnica
- Pravé tlačidlo myši (podrž) - rotácia kamery
- Stredné tlačidlo myši - tvorba tlačidiel (viac nižšie)
- Ľavé tlačidlo myši - Výber bodov v priestore a potvrdzovanie UI elementov

Ovládanie inštruktážneho zážitku (PC)::

- Periférie: myš a klávesnica
- Pravé tlačidlo myši (podrž) - rotácia kamery
- Ľavé tlačidlo myši - stiskom potvrdzujeme výber tlačidla, podržaním vykonávame presun

Ovládanie inštruktážneho zážitku (Oculus Quest 2):

- Periférie: Oculus okuliare a pravý a ľavý ovladač
- Pravý joystick - rotácia kamery (tzv. snap turn)
- Pravý trigger - stiskom potvrdzujeme výber tlačidla, podržaním vykonávame presun

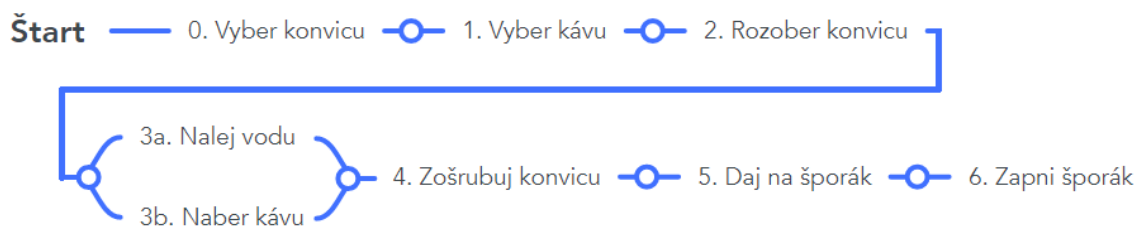
Importovanie projektu

Ako prvé je nutné vytvoriť samotný projekt unity.

1. V Unity Hub kliknete na New Project.
2. Vyberiete 3D možnosť
3. V položke Location vyberiete umiestnenie projektu
4. Otvoríte si adresár tohto novo vzniklého projektu a prekopírujete tam obsah InstructionalEditor.zip
5. V unity editor vyskočí varovná hláška Warning (ide len o zmenu input systému) - Kliknite na Yes
6. Editor sa reštartuje
7. V unity editore v položke Assets/Scenes otvoríte scénu ApplicationScene.

Plán pracovného procesu

Na samotnom začiatku stojí samotný pracovný úkon. Ten sa snažíme klientovi sprostretkovať čo najpodrobnejšie a zároveň jasnou formou. Preto je vhodné si ho jasne vopred celý definovať. Samotný editor počíta s tým, že celý pracovný proces je rozdeliteľný na čo najviac elementárne kroky. Doporučoval by som si vytvoriť akýsi grafický zjednodušený plán daného procesu. Na tomto grafickom alebo textovom pláne budú jasne uvedené jednotlivé elementárne kroky. Tieto kroky potom vo svojej následnosti vytvárajú celkový pracovný úkon.



(Príklad rozdelenia pracovného procesu prípravy kávy v Moka kanvici)

Je taktiež vhodné konštruovať dané kroky tak, aby sa u nich jasne dala určiť interakčná technika, ktorá ju definuje. Editor totiž disponuje tromi základnými možnosťami interakcie hráča s prostredím.

- V prípade, že ide o výber predmetu, jedná sa o techniku SELECT (výber).
- V momentoch, kedy ide o presunutie predmetu z jedného miesta na druhé, ide o techniku DRAG&DROP (ťahaj a pušť).
- V inšanciách, kedy je nutné zvoliť číselnú alebo textovú hodnotu, ide o techniku SETVALUE (nastavenie hodnoty).

Natáčanie

Pravdepodobne najzákladnejším a tiež najdôležitejším prvkom editora sú samotné sférické videá. Počas nahrávania pracovného procesu je dôležité myslieť na pár základných požiadaviek, ktoré editor očakáva od jednotlivých video záznamov.

V prípade, že chce tvorca vytvoriť inštruktážnu ako aj cvičnú časť aplikácie, je vhodné si uvedomiť, že bude potrebovať pre každý krok dva typy video záznamov. Jeden typ je inštruktážny záznam, kde dotýčná osoba predvádza vykonanie daného kroku. Druhý typ je cvičný záznam. V tomto prípade by sa v scéne nemala vyskytovať žiadna osoba ale len predmety, ktoré sú dôležité v zmysle interakcie s prostredím. U cvičných záznamov by tiež v scéne nemalo dochádzať k žiadnym výrazným zmenám a to z dôvodu, že stopa bude prehrávaná v režime smyčky (takzvaný režim loop).

Pri natáčaní je tiež vhodné využiť statív, na ktorý môžeme jednoducho upevniť kameru. Kameru môžeme týmto spôsobom ukotviť do výšky, ktorá zodpovedá priemernej ľudskej

výške. Vďaka tomuto môžeme zvýšiť pocit imerzie u klienta a celkovo tak zlepšiť celkový zážitok z inštruktaže. Prostredie natáčania by malo byť dobre osvetlené a s čo najmenším počtom rušivých predmetov, ktoré z pohľadu inštruktaže, nemajú žiadny účel. Uhol natáčania by mal byť čo najviac názorný a to v tom zmysle, aby na ňom používateľ inštruktažnej aplikácie vždy jasne videl ako zaučujúca osoba vykonáva daný proces. Doporučujem si vytvoriť akýsi plán natáčania, ktorý zefektívni natáčací proces. V tomto pláne jasne definujeme, kedy a v akej následnosti budeme natáčať jednotlivé typy záznamov. V rámci snahy lepšie vysvetliť, ako tento plán funguje, uvádzam časť toho, ktorý som zhotovil ja. Jednalo sa o inštruktaž receptu makaróny so syrom.

1. (Rozlož ingrediencie na stôl, pridaj pár zavádzajúcich predmetov)
2. C
3. T: Vyber predmety a premiestni ich na kuch. linku
4. C
5. T: Odkroj 150g syru
6. C
7. T: Presuň sýr k strúhadlu
8. C
9. T: Presuň
10. atď.

Inštruktažne časti označené písmenom T, su tie časti v ktorých bude vystupovať daný aktér a názorne vykonávať krok postupu.

Cvičné časti reprezentované písmenom C označujú zasa tie časti, v ktorých sa v scéne nachádzajú len predmety, s ktorými by mal hráč interagovať. V týchto častiach sa bude snažiť pracovný krok reprodukovať klient. Pri natáčaní cvičenia teda opustíme miestnosť. Kameru necháme scénu nahrávať v danom stave. Bohužiaľ preto, že sa jedná o sférické videá, je často nutné opustiť miestnosť a tým tak aj záber kamery. Pri tomto nahrávaní môžeme vyčkávať mimo záberu približne 10 sekúnd. To je dostatočne veľká doba na vytvorenie prirodzene vyzerajúcej video sľučky, ktorá sa bude hráčovi prehrávať pri cvičení.

Môžete si tiež všimnúť, že u inštruktažných segmentoch nám slova ako vyber a presuň implikujú interakčné techniky SELECT a DRAG&DROP. Inštruktažne časti, ktoré v tomto pláne obsahujú číselnú hodnotu implikujú, že bude dochádzať k interakciám nastavenia hodnoty, teda SETVALUE.

Domnievam sa, že tento návod natáčania zjednodušuje celý proces a slúži ako aj prevencia proti často nákladnému a zdĺhavému procesu dotáčania opomenutých segmentov.

Taktiež doporučujem celé natáčanie zaznamenať na pár veľkých záznamov. Strihu videa sa používateľ editora s najväčšou pravdepodobnosťou nevyhne. Mnohé sférické kamery sú však schopné zaznamenávať dostatočne dlhú dobu na jedno nabitie. Preto tiež doporučujem použiť dostatočne veľkú pamäťovú kartu, ktorá by taktiež mala umožniť nahrávať video v rozlíšení 4K a viac.

Editácia sférického videa

V mojom prípade som využil program s názvom ActionDirector. Pre toto rozhodnutie som sa rozhodol na základe toho, že ide o oficiálny nástroj pre editáciu videa kamery Samsung Gear360, ktorú som počas celej tvorby editora a aplikácie využíval. No editačné programy za použitia iných kamier sa môžu značne líšiť. Doporučujem si teda zistiť, či ku kamere existuje oficiálny program na editáciu.

Následne by mal tento program disponovať funkciami ako strih a zošitie. Zošitie je v tomto prípade myslené ako správne namapovanie sférického videa na sféru. Ďalšou požadovanou vlasnosťou by mohla byť aj možnosť efektov jednotlivých prechodov medzi segmentami. Jednotlivé prechody by tak napomáhali celkovej plynulosti herného zážitku.

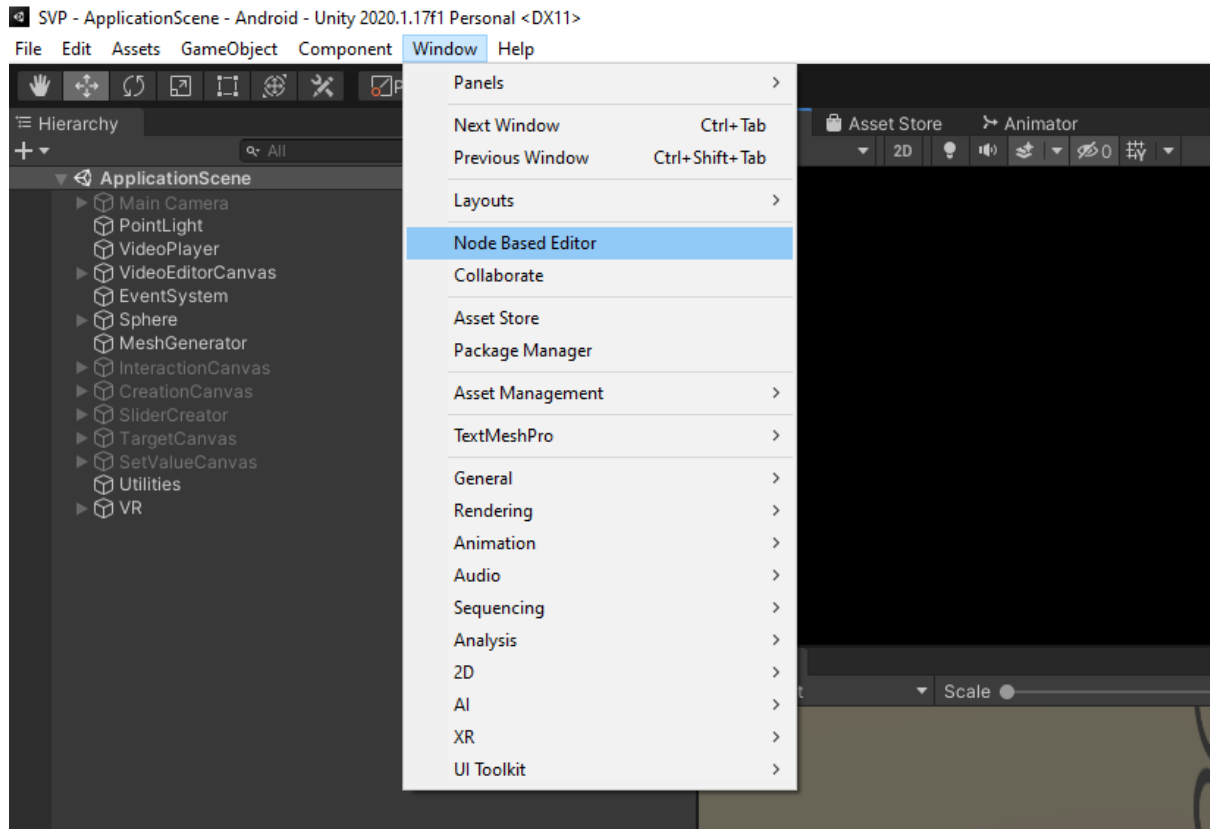
Ďalšie funkcie nad rámec týchto jednoduchých požiadaviek sú z hľadiska tvorby inštruktážnej aplikácie nadbytočné. V tomto editačnom programe si potom rozstriháme video do jednotlivých segmentov. Doporučujem si vytvoriť systém v pomenovaní jednotlivých elementárnych krokov. Jednotlivé názvy záznamov by sa mali odvíjať od poradového čísla, podľa ktorého sa určuje ich umiestnenie v pracovnom postupe. Pomenovanie by tiež malo implikovať, či sa jedná o inštruktážne alebo cvičné video. Uľahčuje to potom následné priradovanie v editore. V mojom prípade som často volil pomenovania ako napríklad *t0*, *t1*, *t2...*, *e1*, *e2*, *e3*. Tieto videá tiež doporučujem exportovať rovno do složky Videos, ktorá sa nachádza v Unity projekte. Celkové umiestnenie tejto složky je *Assets/Resources/Videos*.

Prečo je nutné umiestniť záznamy práve do tejto složky?

Dôvodom je hneď viacero. V skratke ide o prípad, kedy nám Unity umožňuje načítať assety na základe požiadavku zo skriptu. Ku všetkým Unity assetom, ktoré sa nachádzajú v priečinku s názvom „Resources“, môžeme pristupovať prostredníctvom funkcií *Resources.Load* ktorú využíva editor ako aj aplikácia na získanie prístupu k rôznym videám a to aj počas behu aplikácie. Túto techniku využívame v podstate vždy, keď je nevhodné alebo priam nemožné priradiť referenciu objektu v takzvanom Inspector Window. Je nutné tiež poznamenať, že všetky videá v složke Resources sa odzrkadlia na výslednej veľkosti vyexportovanej aplikácie. Je teda nutné si dávať pozor na to, aby sa v nej nenachádzalo množstvo videí, ktoré nepatria ku konkrétnemu pracovnému postupu.

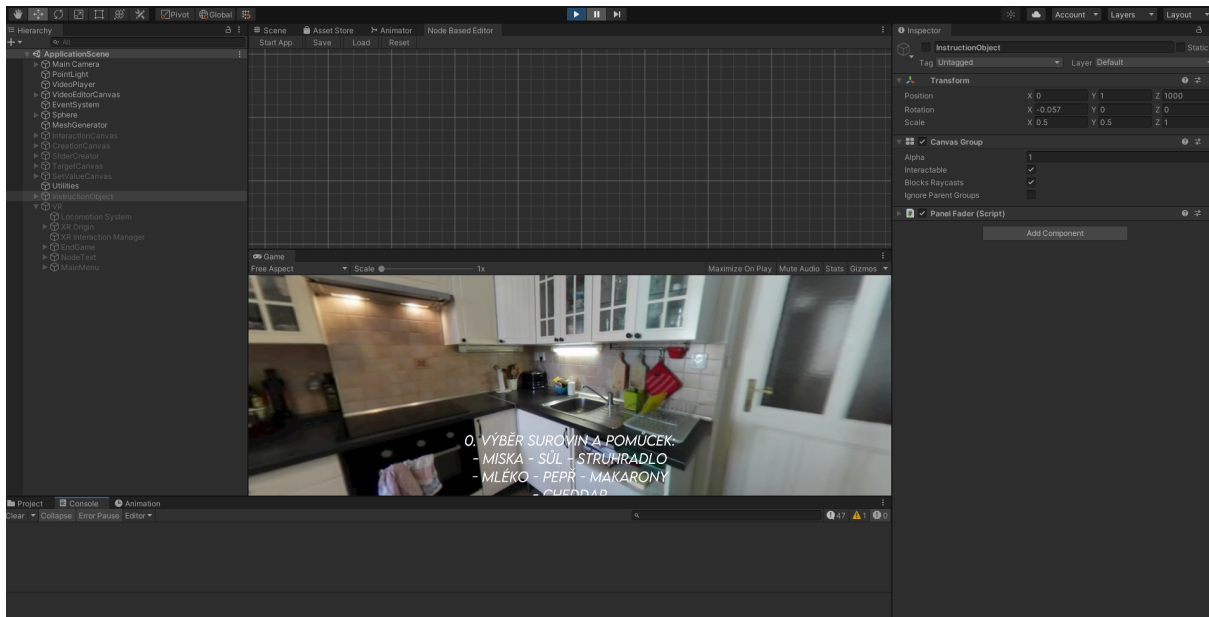
Editor a Unity

Dostávame sa do bodu, kedy dochádza k samotnej editácii v programe Unity. Po otvorení projektu sa otvorí jediná scéna v celom projekte s názvom ApplicationScene. V hornom paneli zvolíme možnosť Window -> NodeBaseEditor a spustíme tak mnou vyvinutý editor na vytváranie inštruktážnych a cvičných video uzlov.



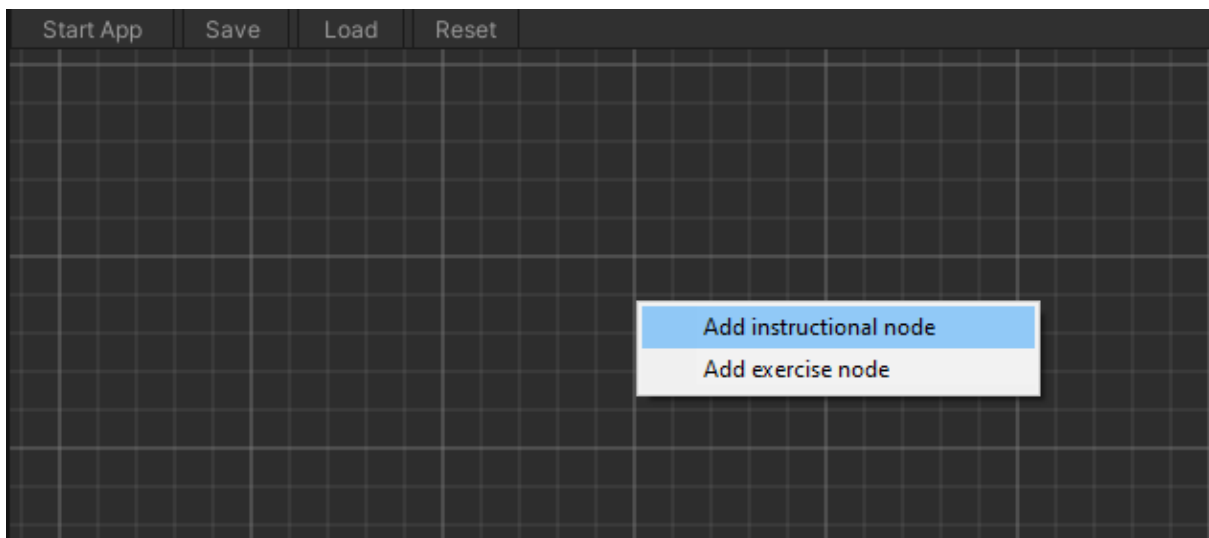
(Vyvolanie Node Based Editoru)

Po rozkliknutí sa nám objaví hlavné okno editora. Tento editor je akousi obdobou grafického programovania. Je založený na takzvaných grafických uzloch. Následné umiestnenie tohto pracovného okna je čisto závislé na používateľovi editora, no doporučujem si ho ukotviť tak, aby bolo možné zároveň pracovať aj s oknom, ktoré nesie názov Game. V okne Game sa totiž vykresľuje zobrazenie hry z hráčovej kamery. Teda v tomto okne bude prebiehať tvorba interakčných techník ako SELECT, DRAG&DROP a SETVALUE. V druhom okne NodeBasedEditor zas budeme vytvárať jednotlivé uzly, ktoré budú reprezentovať jednotlivé segmenty pracovného postupu.



(Príklad môjho rozloženia okien v Unity)

Nachádza fáza, kedy budeme vytvárať uzly v okne s názvom NodeBasedEditor. Potom, čo klikneme pravým tlačidlom myši na šachovnicovú plochu, dôjde k vytvoreniu generického menu. V tomto menu máme na výber z dvoch možností. Môžeme vytvoriť jednoduchý inštruktážny alebo cvičný uzol.

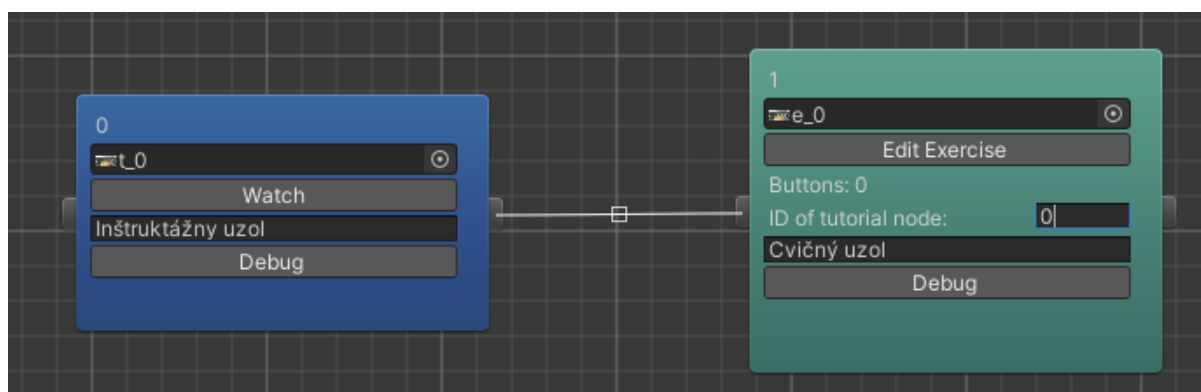


(Ukážka generického menu)

Celkovo sú teda dva typy uzlov, ktoré momentálne môžeme editorom vytvoriť. V prípade, že chceme vytvoriť jednoduchý typ inštruktážneho uzla. Zvolíme možnosť *Add instructional node*. V prípade cvičného uzla zasa volíme možnosť *Add exercise node*.

Uzly editora

Tieto uzly tvoria základný prvok celého inštruktážneho zážitku. Snažia sa intuitívne reprezentovať elementárne kroky pracovného postupu. Uchovávajú mnohé hodnoty, ktoré sa môžu ukládať ako aj načítavať zo súboru XML.



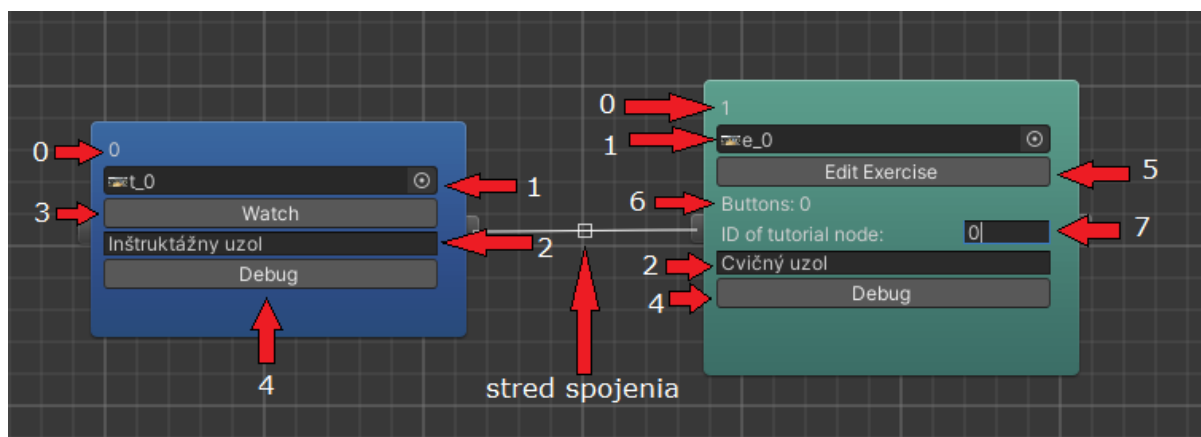
(Ukážka uzlov)

Typy uzlov

Inštruktážny uzol - Na modro sfarbený uzol. Značí typ inštruktážneho uzla. Tento typ slúži na jednoduché zobrazenie inštruktážnej časti práce. Ide teda len o prehriatie videá, na ktorom aktér vykonáva elementárnu časť práce.

Cvičný uzol - zobrazený zelenou farbou, slúži na vytvorenie cvičných častí aplikácie. To sú tie segmenty, v ktorých sa hráč pomocou interakčných techník snaží úspešne reprodukovať pracovný úkon.

V nasledujúcej kapitole budem opisovať jednotlivé vlastnosti daných uzlov. Vlastnosti, ktoré uzly zdieľajú bez ohľadu na ich typ, sú nasledovné.



(Popis vlasností uzlov)

0 - ID:

Každý uzol má svoje jedinečné číslo *ID*, ktoré je vyzobrazené ako prvá hodnota v ľavom hornom rohu. Uzol toto číslo obdrží pri jeho vytváraní. Pre používateľa Editoru má číslo *ID* významnú rolu len v tom zmysle, že pri spustení aplikácie sa začína na uzli s *ID* rovným nule.

1 - Video:

Druhou zložkou ktorá je nazávislá na type uzla je položka *video*. Používateľ si tak môže jednoducho ku uzlu priradiť aj konkrétnu príslušnú video stopu spomedzi všetkých video záznamov, ktoré sa nachádzajú v Unity projekte.

3 - Text:

Ďalšou spoločnou vlastnosťou týchto uzlov je aj textová komponenta. Tá slúži na zobrazenie textu, ktorý sa bude počas hry zobrazovať užívateľovi na takzvanom pomocnom paneli. Obsah textu je čisto na tvorcovi inštruktážneho zážitku. Tvorca môže teda sám usúdiť, do akej miery poskytne hráčovi dodatočné informácie a to či už v prípade inštruktážneho uzlu ako aj v prípade cvičného videa. Tiež môže text použiť ako dodatočnú výstrahu pred nebezpečím u rôznych pracovných úkonoch. Príklad takéhoto textu môže teda vyzeráť nasledovne: "Nakrájaj cibuľu na požadovanú hrúbku. Pozor! Neporež sa"
Dĺžka tohto textu je však obmedzená na základe veľkosti hráčovho pomocného panela. V prípade, že by ste chceli hráčovi zdeliť väčšie množstvo textu, doporučujem použiť objekt s názvom *InstructionObject*.

Tu končia vlastnosti ktoré zdieľajú tieto dva typy uzlov. Teraz sa bližšie pozrieme na každý samostatný typ.

Vlastnosti inštruktážneho uzla

3 - Watch:

Tlačítko *Watch* slúži na jednoduché prehratie priradeného videa. Tvorca inštruktážneho zážitku ho môže použiť na kontrolu priradenia správne videa k uzlu.

4 - Debug:

Tlačítko *Debug* slúži na jednoduchú kontrolu hodnôt uzla. Hodnoty sa zobrazia do konzoly Unity. Funkciu tohto tlačidla si tiež môže užívateľ jednoducho zmeniť v skripte *MyEditorNode*

Vlastnosti čvičného uzla

5 - *Edit Exercise*:

Tlačidlo *Edit Exercise* slúži na spustenie prostredia, v ktorom tvorca môže vytvárať jednotlivé interakčné tlačidlá. Po stlačení tohto tlačidla sa jeho význam, ako aj názov, mení na *Confirm Edit*. Opätovným stisknutím teda potvrdzujeme tvorbu tlačidiel, ktoré sa uložia do dát daného uzla.

6 - *Buttons*:

Pod týmto tlačidlom sa nachádza text *Buttons*, ktorý tvorcovi signalizuje celkový počet priradených k danému uzlu.

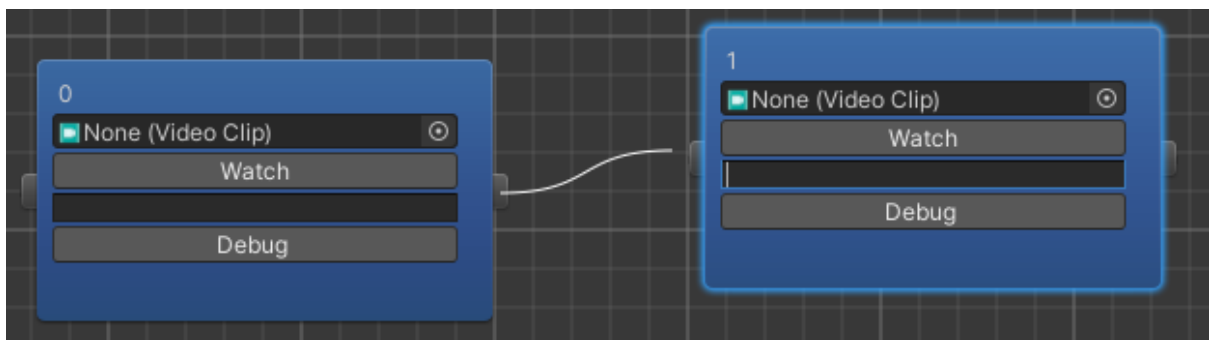
7 - *ID of tutorial node*:

Ďalší rozdiel oproti inštruktaznemu uzlu môžeme vidieť aj v kolónke *ID of tutorial node*. Ako už z názvu vyplýva, jedná sa o definovanie príslušného uzla, ktorý je inštruktazným ekvivalentom k uzlu cvičnému. Do tohto poľa by mal tvorca teda zadať zodpovedajúci inštruktazný uzol, ktorý sa prehráva v prípade hráčovho neúspechu pri plnení úkonu.

4 - *Debug*:

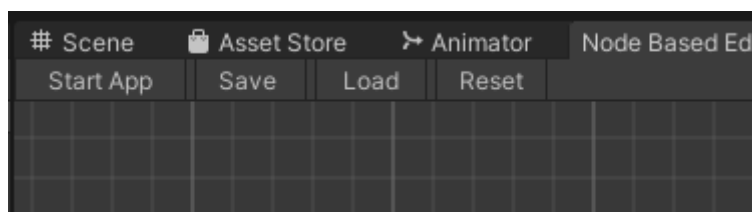
Tlačidlo *Debug* má rovnakú úlohu ako u inštruktážneho uzlu s tým rozdielom, že zobrazuje aj informácie špecifické pre cvičné uzly.

Uzly tiež majú z oboch strán napájacie konce. Tvorca teda môže ľubovoľne napájať jednotlivé uzly a tým tak definovať, v akom poradí sa budú prehrávať. Spojenie vytvoríme jednoducho. Klikneme ľavým tlačidlom myši na požadovaný koniec uzla a spojíme ho s opačným koncom nasledujúceho uzla pomocou techniky potiahnutia. Následne odstránenie spojenia je tiež riešené intuitívnym a jednoduchým spôsobom. Stačí kliknúť ľavým tlačidlom myši na malý štvorec ktorý sa nachádza v strede spojenia. Nezapomínajte na to, že inštruktážna aplikácia začína vždy v uzle *ID* rovným 0.



(Ukážka vytvárania spojenia)

Na samotnom vrchu okná NodeBasedEditor tiež môžeme nájsť ďalšie menu. Táto sadá štyroch tlačidiel predstavuje takzvanú menu bar položku.



(Menu bar)

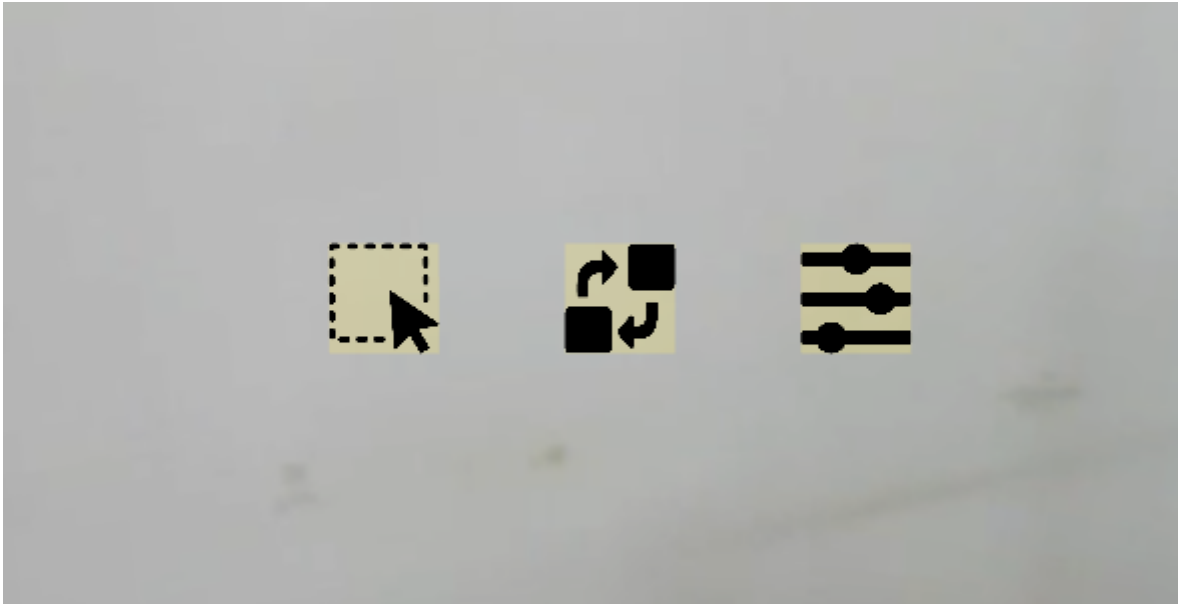
Tlačidlo *Start Application* predstavuje možnosť, ako si spustiť celý zážitok presne tak, ako bude prezentovaný hráčovi po spustení exportovanej verzie aplikácie. To znamená, že sa hra bude postupovať presne tak ako hráčovi vo finálnej verzii zážitku. Hráčovi sa teda v úvode zobrazí inštruktážny text, následne menu ako aj možnosť postupovať naprieč cvičeniami. V prípade, že by chcel tvorca kvôli debugovaniu cvičení preskočiť často zdĺhavú inštruktážne časti, môže tak urobiť zaskrtnutím hodnoty *Skip Tutorial* v komponente *StorylineManager*.

Tlačidlo *Save* zohráva v našom prípade skutočne významnú úlohu. Unity totiž štandardne neuloží naše zmeny v scéne, keď opustíte režim prehrávania. To by ale znamenalo, že všetky novo vytvorené objekty, ktoré predstavujú jednotlivé tlačidlá uzla budú nenávratne stratené po ukončení editácie. No a presne pre tento prípad existuje toto tlačidlo. Po stlačení sa totiž všetky dáta každého uzla uložia do XML dokumentu ktorý sa tiež nachádza v priečinku Resources. My tak neprídeme o žiadne dáta, ktoré sme počas behu aplikácie vytvorili.

Tlačidlo *Load* má zas funkciu načítania dát z XML dokumentu. Snažíme sa teda deserializovať dáta z XML dokumentu ale to len do takej miery, aké je to potrebné. Hlavnou výhodou tejto techniky je aj skutočnosť, že dáta môžeme bez problémov načítať počas behu aplikácie aj vo exportovanej verzii aplikácie.

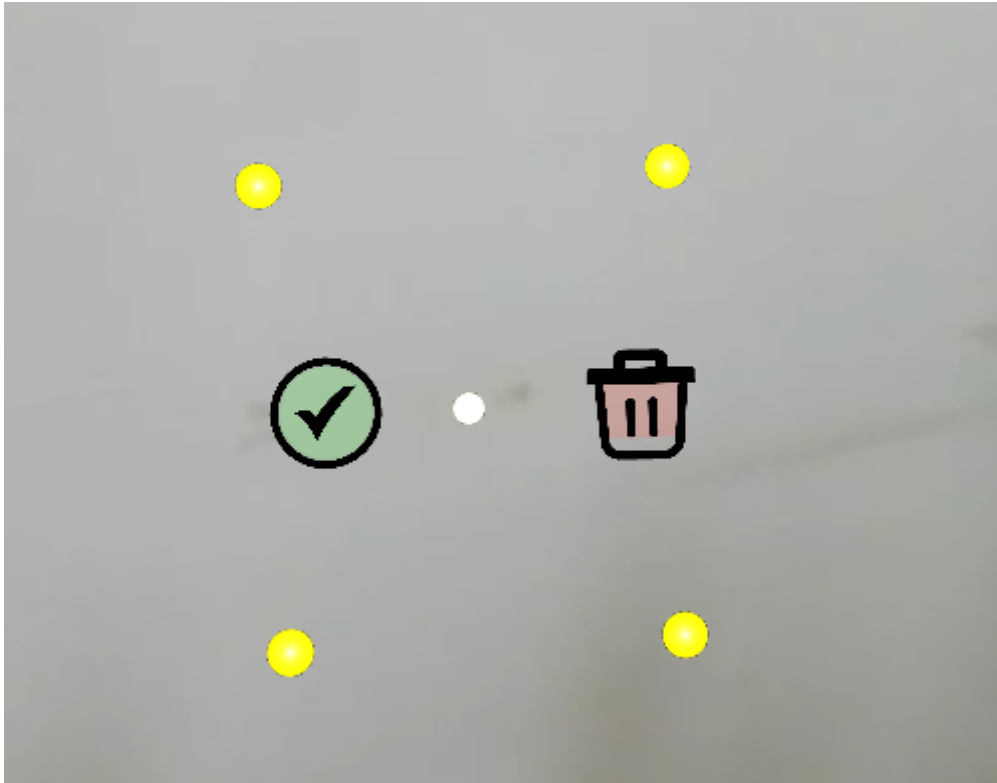
Tlačidlo *Reset* slúži na kompletne vymazanie všetkých stávajúcich dát. Túto funkciu môžeme teoreticky využiť v prípade, že z nejakého dôvodu došlo k chybe pri vytváraní uzlov. Ide o akúsi poistku toho, aby sme sa nedostali do nenávratného stavu.

V prípade, že chceme vytvoriť tlačidlá pre daný cvičný uzol postupujeme nasledovne. Spustíme editor, načítame si dáta z XML dokumentu pomocou tlačidla *Load*. Následne klikneme na tlačidlo daného uzla s názvom *Edit Button*. To nás preniesie do prostredia video slučky ktorá patrí danému uzlu. Po scéne sa môžeme rozhliadať za pomoci stisku pravého tlačidla myši a nasledným ťahaním do opačného smeru. Potom čo lokalizujeme priestor, v ktorom chceme vytvoriť tlačidlo, stlačíme koliesko (označované ako stredné tlačidlo) myši. Pred nami sa objaví jednoduché menu z ktorého si môžeme vybrať typ tlačidla, ktoré sa snažíme vytvoriť.



(Creation Menu)

Po zvolení typu tlačidla nám toto menu zmizne a hráč sa prepne do vytváracieho módu. Ľavým tlačidlom myši zvolí 4 body v priestore. Tieto body budú reprezentovať tvar tlačidla a sú znázornené žltou farbou. Body by mali byť umiestnené v poradí proti smeru hodinových ručičiek a to kvôli logike vytvárania konvexných objektov, ktoré sú k nám čelne orientované. Po zvolení štyroch bodov hráč znova stlačí koliesko myši. Tentokrát sa mu zobrazí potvrdzovacie menu. Toto menu slúži ako posledná výzva smerom k hráčovi. Ten má na výber dve možnosti, vytvoriť objekt (Mesh) alebo zrušiť tvorbu tlačidla.



(Confirmation menu - Creation Canvas)

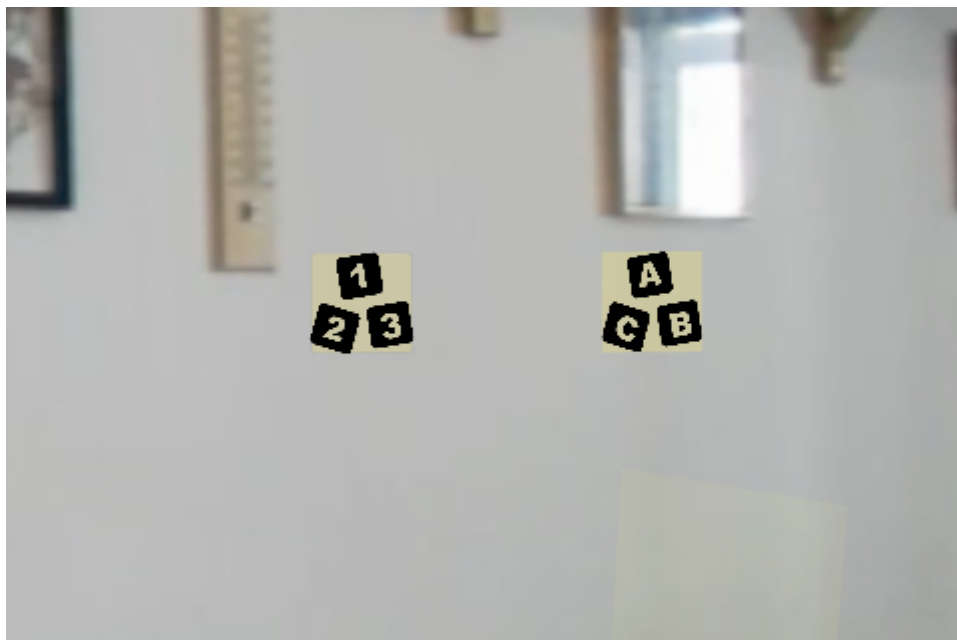
V prípade, že sme vytvárali tlačidlo typu select opúšťame vytvárací mód. V prípade, že chceme zmeniť hodnotu correct, vyhľadame novo vytvorený objekt v Unity hierarchii. Vyhľadanie tohto tlačidla by nemalo byť obtiažne pretože je pomenované ako Button a na základe toho, že sme ho momentálne vytvorili vieme sa nachádza na samotnom konci hierarchie scény. Nájdem jeho komponentu SelectButton a zmeníme hodnotu correct. Poznámka: východzie nastavenie tohto tlačidla je correct = true.

V prípade, že sme zvolili tlačidlo typu DragAndDrop dochádza k malej zmene oproti Select typu. Tlačidlo ktoré sme momentálne vytvorili je typu Drag. Teda jedná sa o tlačidlo, ktoré má hráč potiahnuť do požadovanej oblasti. Túto oblasť definujeme jednoducho. Po opätovnom stlačení stredného tlačidla myši sa nám objaví nové menu, ktoré nás vyzýva na tvorbu tejto spomínanej oblasti. Znova sa teda dostávame do situácie, kedy musíme v priestore korektne zvoliť 4 body. Po potvrdení bodov vzniká cieľová oblasť tlačidla a tým je proces vytvárania ukončený.



(Target menu)

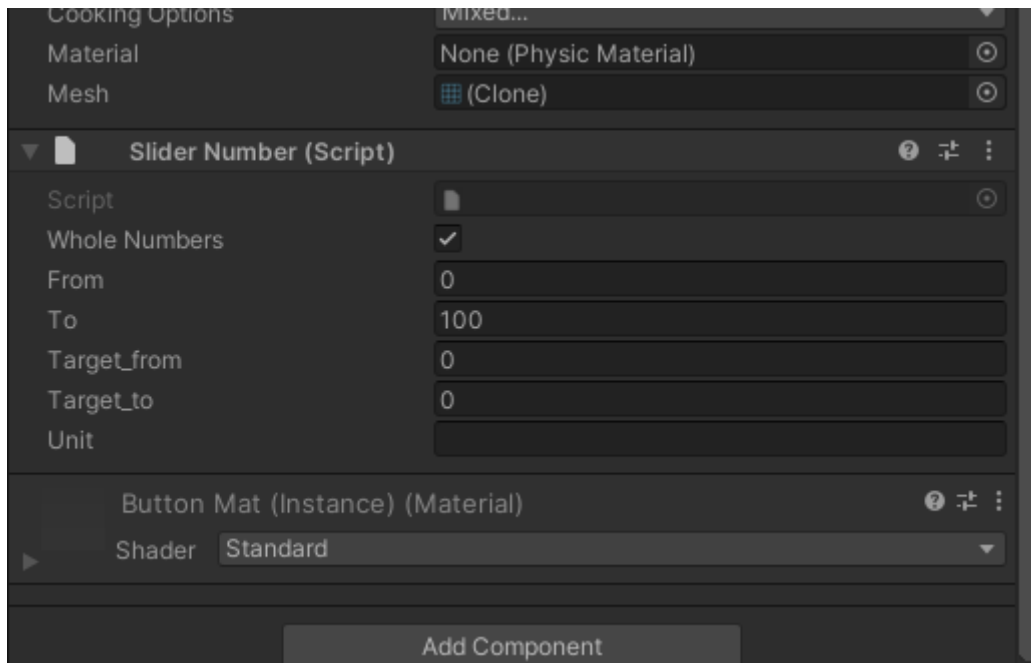
V poslednom prípade si ukážeme tvorbu SetValueButton. Znova plochu tlačidla definujeme štyrmi bodmi podobne ako u predošlých prípadoch. Tentokrát sa však po opätovnom stlačení stredného tlačidla myši zobrazí menu, v ktorom si vyberieme typ hodnoty ktorú budeme nastavovať. Môže sa jednať o číselné hodnoty (*int*, *float*) alebo hodnoty textové (*string*)



(Slider Creator menu)

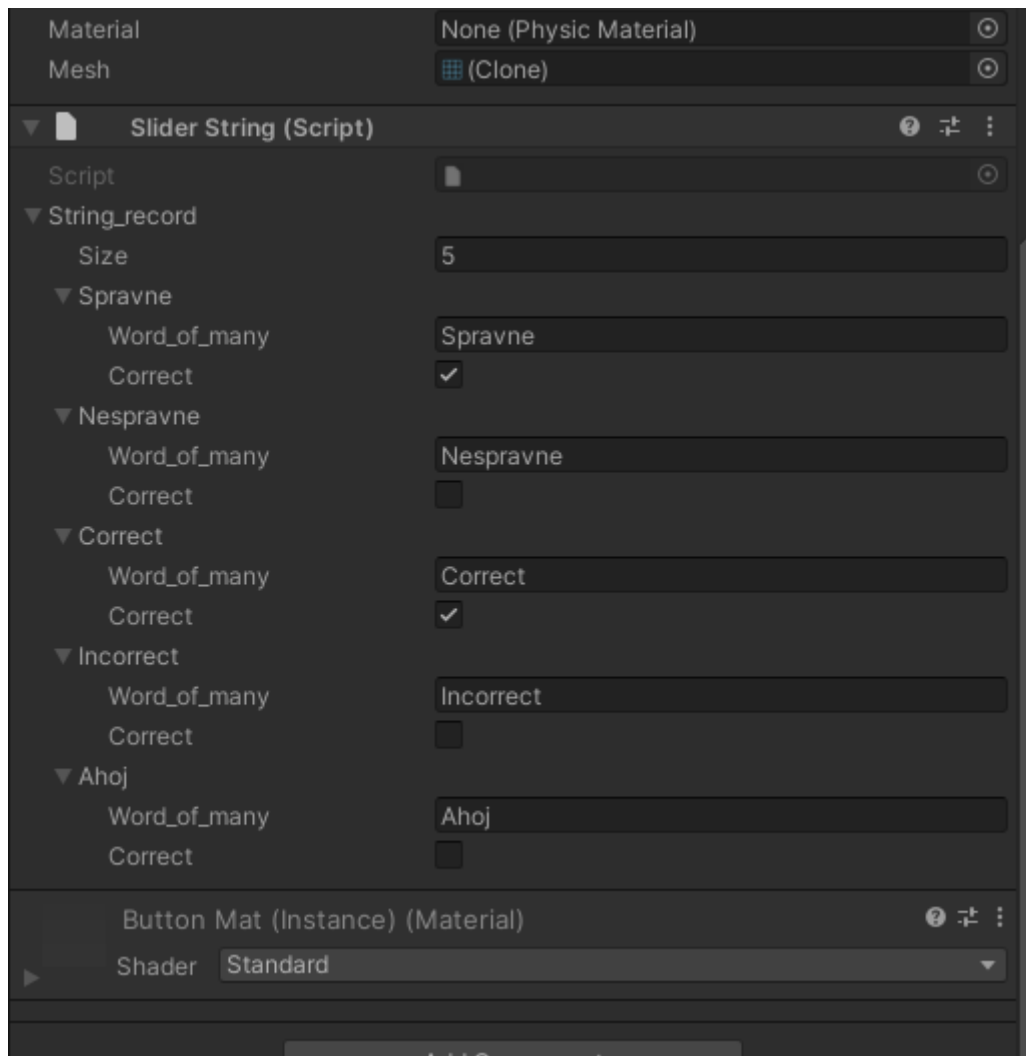
Po odkliknutí ktoréhokoľvek typu sa ukončuje mód tvorby. Jediným rozdielom je typ komponenty, ktorá bola novo vytvorenému tlačidlu priradená. Jedná sa o komponenty typu *SliderNumber* a *SliderString*. Po lokalizovaní tohto objektu v hierarchii môžeme nastavovať jednotlivé hodnoty jednej z týchto komponent.

V prípade komponenty *SliderNumber* môžeme nastaviť hodnoty ako *from* a *to*. Tieto slúžia na definovanie rozsahu hodnôt posuvníka. Ďalej môžeme definovať aj rozsah cieľového intervalu (*target_from*, *target_to*). V prípade, že je cieľová hodnota jedno požadované číslo, sú si tieto hranice intervalu rovné. Taktiež môžeme definovať aj určitú toleranciu v prípadoch, kedy nevyžadujeme úplne presné hodnoty od hráča. Pomocou *Whole Numbers* môžeme definovať, či budú hodnoty celočíselné alebo s desatinnou čiarkou. Do položky *Unit* zas môžeme zadať o akú jednotku hodnoty sa jedná.



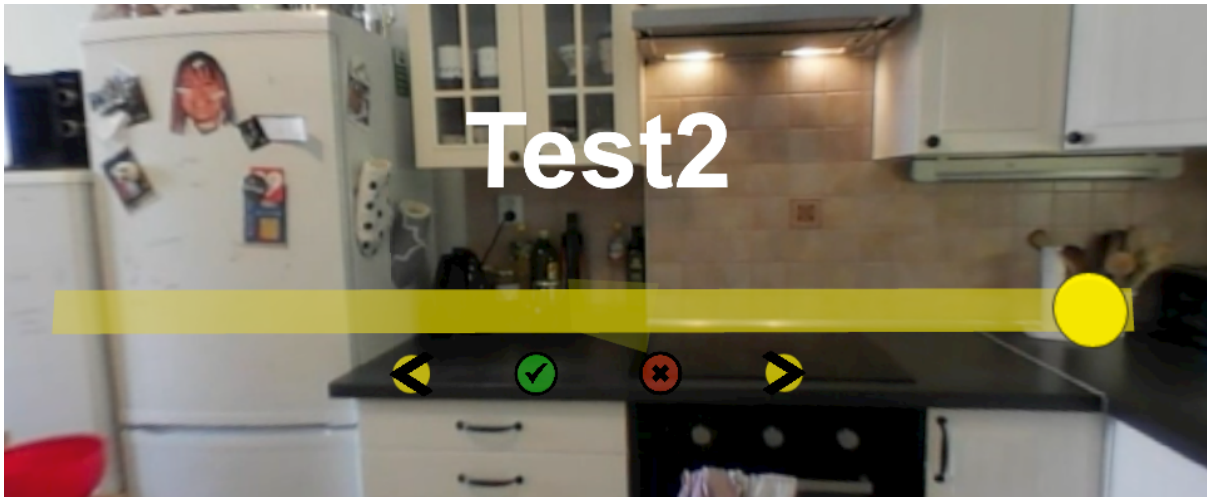
(Obrázok Slider Number komponenty)

Varianta typu *SliderString* je obdobná. V komponente musíme najprv určiť celkový počet možných elementov, ktoré bude mať hráč na výber. Neskôr môže u každého elementu nastavovať jeho správnosť. Je teda možné vytvoriť situáciu, kedy sú viaceré varianty správne.



(Obrázok Slider String komponenty)

Obe varianty zdieľajú totožný posuvník, rozdiel je iba v texte ktorý zobrazuje. Vzhľadom na to, že manipulácia s posuvníkom môže byť vo virtuálnej realite náročná, je možné hodnoty posuvníka ovládať pomocou dvojice šípok, ktoré sú umiestnené po bokoch posuvníka. Potom čo hráč stlačí tlačidlo a vyberie požadovanú hodnotu, má na výber či hodnotu potvrdí a zistí tak jej správnosť alebo zruší výber hodnoty.



(Obrázok SetValueCanvas)

Po vytvorení ľubovľného počtu interakčných tlačidiel môžeme kliknúť na potvrdzovacie tlačidlo uzla *Confirm Edit*. Pred ukončením behu je ešte nutné uložiť zmeny do XML dokumenta pomocou tlačidla *Save*.

Ako som už spomínal, v úvode inštruktážneho zážitku sa objaví pred hráčom textové zhrnutie celého pracovného postupu. Tento text môžete vložiť do herného objektu s názvom *InstructionObject*. V ňom potom lokalizujeme textovú komponentu Unity nasledujúcim spôsobom. *InstructionObject -> Instructions -> Text*.

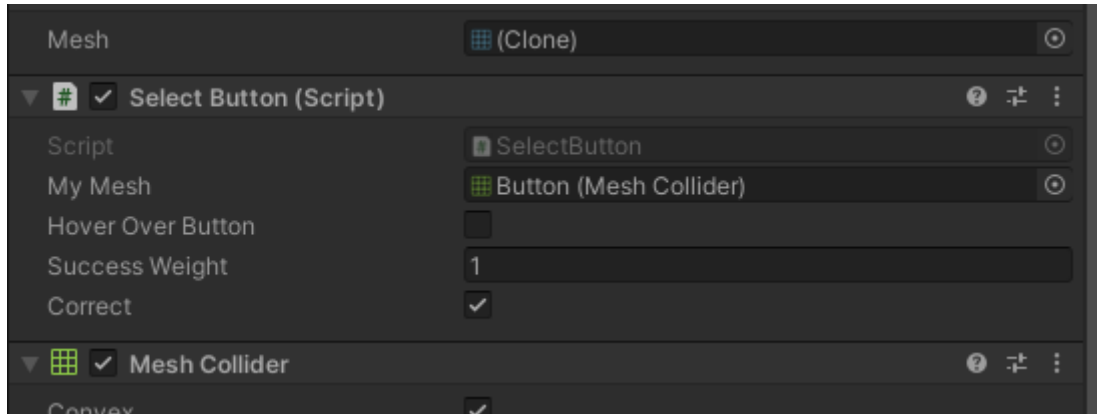
Pred vyexportovaním (build) inštruktážneho zážitku je nutné zvoliť, akým spôsobom chceme výslednú aplikáciu ovládať. Na základe toho aktivujeme alebo deaktivujeme príslušný herný objekt v hierarchii scény. V prípade, že chceme ovládať aplikáciu na PC tak sa jedná o objekt *PC_Editor_Player* a deaktivujeme objekt *VR_Player*. Keď už plánujeme aplikáciu vyexportovať do headsetu Oculus Quest tak deaktivujeme *PC_Editor_Player* a aktivujeme herný objekt *VR_Player*.

V prípade, že má tvorca skúsenosti s vývojovým prostredím, je miera úpravy inštruktážneho zážitku, ako aj editora, omnoho rozsiahlejšia. Môže napríklad meniť zvukové stopy ktoré sa prehrávajú po správnom, či nesprávnom vykonaní pracovného úkonu. Je schopný taktiež

Hodnoty úspechu

Každý cvičný uzol si tiež drží svoju takzvanú momentálnu hodnotu úspechu a taktiež celkovú hodnotu úspechu. Počiatočná hodnota úspechu je každého cvičného uzla nulová. Udáva totiž momentálny progress hráča v danom uzle. Celková hodnota úspechu uzla je vypočítaná na základe počtu tlačidiel, ktoré sa nachádzajú v uzle. (V prípade DRAG&DROP sa hodnota inkrementuje len u tlačidla drag) Hráč teda postupuje do nasledovného uzla až po splnení podmienky kedy sa momentálna hodnota úspechu rovná celkovej hodnote úspechu.

Celkovo môžeme u každého interakčného tlačidla meniť aj túto hodnotu *successWeight*. Tá nám symbolizuje váhu úspechu pri správnom vykonaní tlačidla. Môže sa jednať napríklad o situácie, kedy nám nezáleží na ktoré z dvoch správnych tlačidiel hráč klikol. Obe sú si rovnocenné v zmysle váhy správnosti.



(Success weight modifikácia)