

# Návrh aplikácie E-Zožit

**Úloha:** Našou úlohou je navrhnuť a vytvoriť aplikáciu e-zožit. Jej cieľom je pomôcť učiteľom pri tvorbe interaktívnych aktivít, ktoré pomôžu deťom rozvíjať svoje logické schopnosti a vedomosti. Vychádzame zo [zadania](#) definovanom na predmete Tvorba edukačného softvéru.

Aplikácia uvažuje dve role, na ktoré sa budú viazať prípady použitia. Prvou je **učiteľ**, pre ktorého vytvárame prostredie na definovanie interaktívnych úloh a druhým je **žiak**, pre ktorého sú úlohy určené. Každá z týchto rolí bude mať svoje možnosti ako s aplikáciou pracovať. Poďme si teda tieto možnosti poriadne definovať.

**Vývojové prostredie:** Pracovať budeme s nástrojom [Unity Engine](#) v jazyku C#. Výber nástroja súvisí hlavne s rozsiahlymi skúsenosťami autora s nástrojom. Ak by sme mali hodnotiť, v čom sú jeho výhody/nevýhody oproti iným možnostiam:

## Výhody:

1. **Multiplatformový** - umožňuje nám vyexportovať výslednú aplikáciu aj ako WebGL plugin pre web stránku, aj ako natívnu aplikáciu. Aplikácia by si nevyžadovala žiadnu náročnú inštaláciu.
2. **Canvas ako súčasť nástroja** - nástroj ma riadne definovaný Canvas, ktorý vieme použiť pre prácu s 2D grafikou v kombinácii s textovými poľami, tlačidlami a podobne.
3. **File manager systém** - v rámci iného predmetu sme vytvorili systém pre správu súborov, ktorý nám v nástroji umožňuje ukladať a čítať súbory na disku v rôznych formátoch. Vieme špecifikovať napríklad aby sa žiakovi ukázali iba súbory v žiadanom formáte počas prehľadávania a bolo pre neho tak jednoduchšie nájsť hľadanú úlohu.
4. **2D Shader** - z iných projektov máme prichystanú užitočnú knižnicu shader komponentov. Tie by napríklad mohli slúžiť na zvýraznenie hrán ľubovoľného objektu na ktorom má žiak práve myšku alebo na zvýšenie zážitku práce žiaka efektami ako klonovanie, rozpadnutie objektu....

## Nevýhody:

1. **Veľkosť aplikácie** - nástroj je pomerne optimalizovaný aj pre 2D aplikácie, ale stále bude mať výsledná aplikácia v sebe pár modulov, ktoré pre potreby našej aplikácie nie sú podstatné a pridávajú na jej veľkosť. Odhadovaná veľkosť finálnej aplikácie bude okolo **15-40mb**.
2. **Náhľad úloh** - v zadaní je uvedené, že ak si žiak pri výbere úlohy zvolí myškou súbor s úlohou tak by mal vidieť náhľad ako úloha asi vyzerá. Toto neviem zatiaľ zaručiť, pretože ten file manager system takú úlohu zatiaľ neposkytuje.
3. **Serializácia scény(úlohy)** - jednotlivé interakcie nad objektami typu obrázok text by pre nás nemali byť žiadnym problémom, však doteraz som neriešil problém ako množstvo týchto serializovať aby si ich žiak vedel načítať ako svoju úlohu. Verím však, že sa mi podarí vymyslieť elegantné riešenie napríklad prostredníctvom JSON alebo iných serializovaných objektov.

### Aplikácia z pohľadu žiaka:

1. **Načítanie úlohy** - Aplikácia pri spustení pozdraví žiaka a požiada ho aby zadal svoje meno a zvolil si úlohu, na ktorej chce pracovať. Žiak si otvorí úlohu pripravenú pre daný vyučovací deň.
2. **Prispôsobenie zobrazenia úlohy** - Vzhľadom na povahu úlohy by sa mali žiakovi zobrazovať tlačidlá, ktoré môže počas úlohy využívať. Tento krok je dôležitý, lebo ak by úloha pozostávala napríklad iba z klikania na objekty v mriežke, nechceme žiaka myliť tlačidlami ako zväčši/zmenši objekt.
3. **Plnenie úlohy** - Žiak môže, ale nemusí mať limitovaný čas, prípadne limitovaný počet pokusov na skontrolovanie na splnenia úlohy, túto možnosť chceme nechať na učiteľa. Počas úlohy chceme nechať žiakovi možnosť experimentovať so svojimi riešeniami. Preto by bolo dobré ak by mal k dispozícii tlačidlo, ktoré mu úlohu resetuje. Prípadne doplníme tlačidlo späť, ktoré vráti posledný modifikovaný objekt do svojej predchádzajúcej podoby. Ak je žiak s riešením úlohy spokojný, môže úlohu posunúť na schválenie.
4. **Schválenie úlohy** - Aplikácia skontroluje správnosť žiakovho riešenia. Ak je riešenie nesprávne, aplikácia motivuje žiaka aby skúsil najst' lepšie riešenie. Peknou motivačnou funkciou by mohlo byť ak by sme mu vysvietili časti riešenia, ktoré má správne. Takéto niečo by sa teoreticky dalo spraviť pre každý typ úlohy, ktorý som videl v špecifikácii.
5. **Opt. Notifikácia učiteľa** - Ak by to malo zmysel vedeli by sme notifikovať učiteľa o správnom splnení úlohy, napríklad emailom. Keďže v prvej časti sme sa spýtali žiaka na meno, vieme poslať email s menom žiaka a názvom úlohy. Takéto niečo považujeme za jednoduchšie riešenie akoby sami žiaci museli ukladať svoje riešenia a učiteľ ich manuálne kontroloval.

### Aplikácia z pohľadu učiteľa:

1. **Vytvorenie úlohy** - Učiteľ vytvorí nový projekt s prázdny oknom alebo si vyberie jeden zo šablónových projektov, ktoré mu ušetrí čas pri tvorbe úlohy. Šablónový projekt by sme vytvorili pre každý z typov úlohy prezentovaných v špecifikácii.
2. **Vytvorenie knižnice materiálov** - Učiteľ si načíta potrebné obrázky do takzvanej knižnice materiálov. Táto knižnica mu bude k dispozícii v samostatnom paneli. Vždy potom keď v úlohe bude chcieť definovať nejaký obrázok môže použiť ten uložený v knižnici, aby nemusel znova hľadať obrázok na disku. Taktiež mu môžeme poskytnúť našu knižnicu v ktorej by boli staticky uložené často používané obrázky vo výučbe. Teda napríklad by mohlo ísť o zvieratá, rastliny, základné objekty(kameň, drevo)...
3. **Definícia úlohy** - Teraz ma učiteľ priestor definovať úlohu, k dispozícii mu chceme dať všetky možnosti zo špecifikácie. Každý typ objektu môže byť buď geometrickým tvarom, obrázkom alebo textom. Bez ohľadu na to o aký typ ide tak objekt sa vie správať ako:
  - a. statický objekt - žiak s ním nevie žiadnym spôsobom interagovať.
  - b. pohyblivý objekt - žiak ho vie presúvať myšou.
  - c. klikací objekt - žiak ho nevie presúvať, ale keď na neho klikne tak obrázok zmení svoj tvar, učiteľ definuje množinu tvarov, na ktoré sa môže zmeniť.

- d. cieľový objekt - žiak do neho môže umiestniť pohyblivý objekt a ak sa ich tvar podobá tak pohyblivý objekt do cieľového zapadne. Do cieľového objektu sa bude dať vložiť iba jeden pohyblivý objekt.
4. **Definícia správneho riešenia** - Učiteľ umiestni/nastaví objekty tak aby tvorili správne riešenie danej úlohy. Tento stav si uloží kliknutím tlačidla v paneli nástrojov. Ideálne chceme učiteľovi dať možnosť správnych riešení vytvoriť viacero, keďže niektoré úlohy nemusia mať len jedno. Preto si učiteľ bude manažovať akoby taký malý zoznam správnych riešení, vie si ich zobrazíť a vie vybrané riešenie zahodiť.
  5. **Definícia počiatočného stavu** - Podobne ako pri definícii riešenia si vie učiteľ presúvať objekty tak aby boli v rozporení ako má úloha začínať. Takýchto počiatočných stavov by si tiež mal vedieť vytvoriť viacero a žiak potom dostane náhodný z nich.
  6. **Nastavenie parametrov úlohy** - V tomto kroku si vie učiteľ nastaviť spomínané parametre ako počet pokusov, čas na úlohu, názov úlohy, zadanie úlohy a podobne. Taktiež ak budeme implementovať email notifikáciu tak mu ponúkneme možnosť vložiť svoj mail.
  7. **Export úlohy** - Ak je učiteľ s úlohou spokojný, tak zvoliť možnosť exportovať úlohu, ktorá sa mu uloží do jedného súboru v type definovanom našou aplikáciou.

#### **Postup práce:**

1. **Interaktívne elementy** - začneme časťou s ktorou bude pracovať žiak, vyskúšame si spraviť všetky typy objektov a definujeme ich interakcie.
2. **Serializácia scény** - spravíme ukladanie a načítanie úloh, tak aby sme vedeli vhodným spôsobom vytvorené(zatiaľ natvrdo v Unity) úlohy zdieľať.
3. **Editor úloh** - spravíme časť pre učiteľa, aby vedel vytvoriť všetky typy úloh.
4. **Polish** - vyhladáme nedostatky aplikácie na základe feedbacku a obohatíme aplikáciu o jednoduché animácie/efekty pre lepší pocit z práce.

**Záver:** Na aplikácii budeme pracovať postupne počas semestra a progress bude uložený vo verejnom [github repozitári](#) aj s týmto dokumentom.

Vytvoril Samuel Kubala v roku 2021 na FMFI univerzity Komenského.