



**VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ**

BRNO UNIVERSITY OF TECHNOLOGY

**FAKULTA INFORMAČNÍCH TECHNOLOGIÍ**

FACULTY OF INFORMATION TECHNOLOGY

**ÚSTAV POČÍTAČOVÉ GRAFIKY A MULTIMÉDIÍ**

DEPARTMENT OF COMPUTER GRAPHICS AND MULTIMEDIA

**ONBOARDING PROCES PRO APLIKACI VIZUÁLNÍHO  
PROGRAMOVÁNÍ V ROZŠÍŘENÉ REALITĚ**

ONBOARDING PROCESS IN APPLICATION FOR VISUAL PROGRAMMING IN AUGMENTED

REALITY

**BAKALÁŘSKÁ PRÁCE**

BACHELOR'S THESIS

**AUTOR PRÁCE**

AUTHOR

**MICHAL MIKUŠ**

**VEDOUCÍ PRÁCE**

SUPERVISOR

**Ing. ZDENĚK MATERNA, Ph.D.**

BRNO 2021

## Zadání bakalářské práce



Student: **Mikuš Michal**  
Program: Informační technologie  
Název: **Onboarding proces pro aplikaci vizuálního programování v rozšířené realitě**  
**Onboarding Process in Application for Visual Programming in Augmented Reality**

Kategorie: Uživatelská rozhraní

### Zadání:

1. Seznamte se s problematikou mobilní rozšířené reality, existujícími doporučeními pro proces onboardingu a metodami pro ověřování uživatelské zkušenosti.
2. Seznamte se s aplikací AREditor, umožňující vizuální programování v rozšířené realitě.
3. Proveďte test, vyhodnoťte zkušenost pro nové uživatele a navrhňte opatření k jejímu zlepšení.
4. Implementujte navržené změny.
5. Opakujte testování a vyhodnoťte přínos provedených změn.
6. Vytvořte video prezentující vaši práci, její cíle a výsledky.

### Literatura:

- Kim, Un, Yaxi Wang, and Wenhao Yuan. "Study on user-centered usability elements of user interface designs in an augmented reality environment." *International conference on human-computer interaction*. Springer, Cham, 2020.
- Camacho Herrero, Sonia. "Gamified Learning of Software Tool Functionality: Design and implementation of an interactive in-app learning interface for complex software." (2019).

Pro udělení zápočtu za první semestr je požadováno:

- Body 1 až 3.

Podrobné závazné pokyny pro vypracování práce viz <https://www.fit.vut.cz/study/theses/>

Vedoucí práce: **Materna Zdeněk, Ing., Ph.D.**

Vedoucí ústavu: Černocký Jan, doc. Dr. Ing.

Datum zadání: 1. listopadu 2021

Datum odevzdání: 11. května 2022

Datum schválení: 1. listopadu 2021

## Abstrakt

Práca sa zaoberá vytvorením onboardingového procesu pre prvotných užívateľov aplikácie AREditor, ktorá má umožňovať ne-programátorom programovať robotické pracoviská a ich komponenty v rozšírenej realite. Nakoľko je práca s aplikáciou špecificky zameraná, je potreba do nej užívateľov zaučiť. Práca pozostáva z analýzy problematiky používania rozšírenej reality, analýzy informácií potrebných k efektívnemu zaučeniu užívateľov a analýza problémov v aplikácii, vytvorenie návrhu riešenia, implementácia tohoto riešenia a jeho následné otestovanie a vylepšenie. Onboarding je tvorený systémom návodov, ktoré užívateľa prevedú a zoznámia so základnou funkcionalitou aplikácie. Po absolvovaní onboardingu by užívateľ nemal mať problém ďalej objavovať funkcionality aplikácie sám za pomoci už implementovaných nástrojových tipov.

## Abstract

This thesis focuses on the implementation of the onboarding process in AREditor, which is an application developed to enable non-programmers to program robotic arms and its components in augmented reality. Because the app's workflow is so specific, it requires integration for new users. My work contains analysis of current issues with the use of augmented reality, analysis of issues found in AREditor, analysis of learning principles and existing onboarding methods. With that knowledge, I could create a draft, which was then implemented and tested. The tests helped me improve the existing implementations. Onboarding is there to provide information on the basic features of the application. Users that pass onboarding should be able to operate this app and be able to find other functionalities by their own.

## Klíčové slová

používateľská skúsenosť, onboarding, testovanie, rozšírená realita, Unity, vizuálne programovanie

## Keywords

user experience, onboarding, testing, augmented reality, Unity, visual programming

## Citácia

MIKUŠ, Michal. *Onboarding proces pro aplikaci vizuálního programování v rozšířené realitě*. Brno, 2021. Bakalářská práce. Vysoké učení technické v Brně, Fakulta informačních technologií. Vedoucí práce Ing. Zdeněk Materna, Ph.D.

# Onboarding proces pro aplikaci vizuálního programování v rozšířené realitě

## Prehlásenie

Prehlasujem, že som tuto bakalársku prácu vypracoval samostatne pod vedením pána Ing. Zdeňka Materny, Ph.D. Ďalšie informácie mi poskytol Ing. Michal Kapinus. Uviedol som všetky literárne pramene, publikácie a ďalšie zdroje, z ktorých som čerpal.

.....  
Michal Mikuš

9. mája 2022

## Podakovanie

Rád by som sa na začiatok poďakoval môjmu vedúcemu bakalárskej práce Ing. Zdeňkovi Maternovi, Ph.D., ktorý mi bol v každom kroku práce ochotný pomôcť, za jeho rady nápady a vedenie, ktoré ma vždy naviedlo na správnu cestu k vyriešeniu problému. Ďalej by som rád poďakoval Ing. Michalovi Kapinusovi za jeho rady a pomoc s orientáciou v zdrojových kódach aplikácie AREditor. Taktiež aj mojej mame a bratom, ktorý ma celé štúdium podporovali a dali mi veľa cenných rád, ktoré mi počas štúdia na vysokej škole nesmierne pomohli. V neposlednom rade mojej priateľke Zuzane Peckovej, ktorá mi nielen pomohla s gramatickou kontrolou výslednej práce, ale aj za jej podporu, ktorú mi pri vypracovávaní poskytovala. A na záver by som rád poďakoval všetkým kamarátom a známym, ktorý mi neváhali pomôcť či už v osobnom živote, štúdiu alebo experimentoch, menovite Richard Seipel, Matúš Bičanovský, Júlia Skalická, Teodor Žatko, Maximilián Mayer, Richard Hrmo, Sára Šujanová.

# Obsah

<b>1</b>	<b>Úvod</b>	<b>2</b>
<b>2</b>	<b>Rozbor problematiky</b>	<b>3</b>
2.1	Overovanie užívateľskej skúsenosti . . . . .	4
2.2	Pamäť a učenie . . . . .	5
2.3	Onboardingové procesy v aplikáciách . . . . .	8
2.4	Rozšírená realita . . . . .	11
<b>3</b>	<b>Návrh riešenia</b>	<b>14</b>
3.1	Aplikácia AREditor . . . . .	14
3.2	Analýza predošlých užívateľských experimentov . . . . .	16
3.3	Vykonanie vlastného používateľského experimentu . . . . .	17
3.4	Zhrnutie zistení vykonaného používateľského experimentu . . . . .	21
3.5	Návrh riešenia v aplikácii . . . . .	22
<b>4</b>	<b>Technické riešenie</b>	<b>30</b>
4.1	Implementácia návrhu UI . . . . .	30
4.2	Riadenie onboardingu . . . . .	36
<b>5</b>	<b>Testovanie</b>	<b>40</b>
5.1	Pilotné testovanie . . . . .	40
5.2	Vyhodnotenie a navrhnuté zmeny . . . . .	44
5.3	Opätovný test . . . . .	45
5.4	Zhrnuté výsledky . . . . .	47
5.5	Návrh budúcich vylepšení . . . . .	48
<b>6</b>	<b>Záver</b>	<b>49</b>
	<b>Literatúra</b>	<b>50</b>

# Kapitola 1

## Úvod

Každému užívateľovi či už mobilnej aplikácie alebo webovej stránky sa z času na čas stane, že je mu poskytnutých toľko možností, že nevie odkiaľ začať. Obzvlášť tak pri veľmi špecializovaných aplikáciach ako je AREditor vyvíjaná tímom Robo@FIT <sup>1</sup>, ktorá prináša možnosť ne-programátorom programovať manipulačných robotov a ich komponenty pomocou rozšírenej reality. Tento problém sa snaží riešiť dizajn užívateľskej skúsenosti (anglicky user experience design ďalej už iba skrátene UX dizajn). Niekedy je však potrebné mať onboardingový proces <sup>2</sup> súčasťou dobrého UX dizajnu.

V súčasnej aplikácii, okrem nápovede na jednotlivých tlačidlách, neexistuje iná forma uľahčenia prvotného použitia. A je preto potreba pri jej prvotnom používaní zaučenie niektorým z vývojárov, ktorý užívateľa zoznámí s hlavnými funkcionalitami a kľúčovými slovami v aplikácii, čo v konečnom dôsledku stojí čas a peniaze. Cieľom mojej práce je využiť prvky UX dizajnu, metódy overovania užívateľskej skúsenosti a zaužívané metódy onboardingu v kombinácii s psychológiou ľudského učenia, a navrhnúť vhodnú metódu onboardingového procesu, tak aby nebolo nutné jednotlivo školiť užívateľa v používaní aplikácie. Aplikácia môže obsahovať rozšírenie, ktoré užívateľa prevedie hlavnými funkcionalitami aplikácie. Čím by využitie AREditoru mohlo byť oveľa jednoduchšie a atraktívnejšie, nakoľko by prestalo byť nutné preškolenie každého užívateľa, ktorý by chcel aplikáciu použiť.

Moja práca zahŕňa analýzu existujúcich metód onboardingu spolu s analýzou psychológie skrytej za úspešným učením nových znalostí. Neskôr samotný návrh riešenia s popisom užívateľského experimentu, ktorý mi poskytol informácie na jeho tvorbu. A nakoniec samotnú implementáciu riešenia do AREditoru, jej následné otestovanie a vyhodnotenie výsledkov mojej práce. Rozšírenie obsahujúce kroky, ktoré opíšu a zvýrazia jednotlivé funkcionality, pohodlne prevedie nového užívateľa cez najpodstatnejšie aspekty aplikácie a uľahčí nasadzovanie aplikácie do praxe.

---

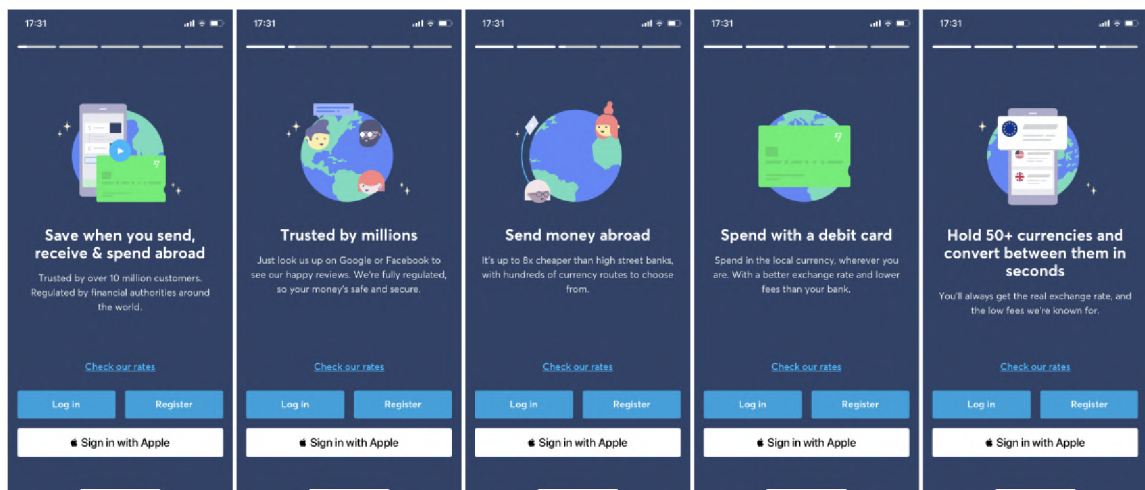
<sup>1</sup>[https://github.com/robofit/arcor2\\_areditor](https://github.com/robofit/arcor2_areditor)

<sup>2</sup>Proces uvedenia užívateľa do danej problematiky.

## Kapitola 2

# Rozbor problematiky

Stále viac a viac služieb sa presúva do online priestoru a s tým prichádza aj stále častejšia otázka: Ako zaručíme, že užívateľ sa dozvie všetko čo potreba, ale pri tom nestratíme jeho pozornosť? Onboarding procesy sa dnes nachádzajú takmer v každej aplikácii, či už len pár obrázkov, ktoré vás privítajú alebo komplexnejší proces, ktorý vás naučí s aplikáciou pracovať a predvedie jej využitie (obrázok 2.1).



Obr. 2.1: Onboarding aplikácie Wise na online bankovníctvo prevedie užívateľa prihlásením. Obrázky prevzaté z aplikácie Wise dostupnej na <https://apps.apple.com/us/app/wise-ex-transferwise/id612261027>

V tejto kapitole sú zhrnuté poznatky z oblasti psychológie, UX dizajnu a overených metódik používaných pri onboardingových procesoch, ktoré sú nevyhnutné pri snahe naučiť používateľa pracovať s aplikáciou.

## 2.1 Overovanie užívateľskej skúsenosti

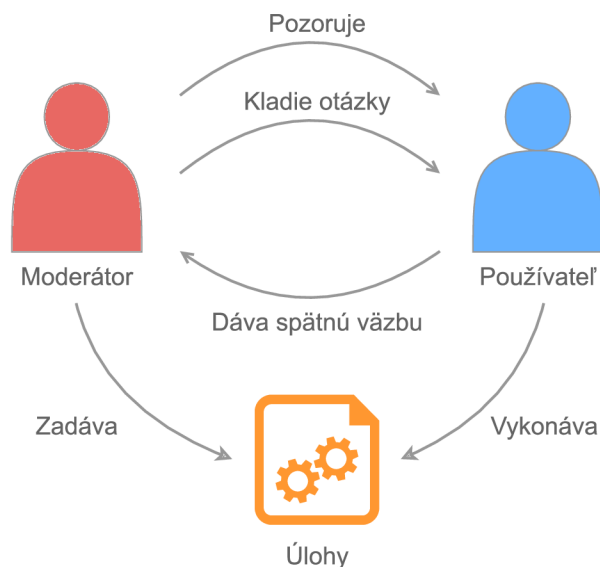
Overovanie užívateľskej skúsenosti je veľmi dôležitá časť vývoja produktu. Nakoľko vývojár, ktorý pracuje na vývoji má oproti bežnému užívateľovi neporovnateľne väčšiu technickú znalosť a expertízu v obore, a ťažko teda vie zhodnotiť či bude jeho výsledný produkt pre prvotného užívateľa dostatočne jednoduchý na použitie. Overovanie užívateľskej skúsenosti často vyžaduje len malé percento zdrojov, ktoré by sa na projekt vynaložili [15]. Na overovanie užívateľskej skúsenosti sa často používa populárna metodológia zvaná testovanie použiteľnosti. Pri testovaní použiteľnosti výskumník zadá úlohu, ktorú by mal byť užívateľ schopný v testovanej aplikácii splniť. Popri plnení úlohy výskumník sleduje používateľa a zaznamenáva spätnú väzbu. Taktiež sa používa nahrávanie celého procesu pre následnú analýzu [14].

V knihe *Handbook of human factors and ergonomics, Fifth Edition* v kapitole *Usability and user experience: Design and evaluation* rozlišujú dva koncepty používateľského testovania, formatívny a súhrnný. Formatívny cieľ je cieľom, ktorý od používateľského testovania očakáva identifikáciu a napravenie nedostatkov v použiteľnosti ešte pred vydaním výsledného produktu na trh. Sumatívny cieľ je zase cieľ, kedy sa v testovaní použiteľnosti nastaví špecifický cieľ na začiatku vývojovej fázy produktu, a vývojári spolu s dizajnérmi sa tento cieľ snažia naplniť. Fundamentálnym cieľom testovania použiteľnosti je ale v oboch prípadoch identifikovanie problémov v dizajne aplikácie alebo výrobku, odhaľovanie možností ďalšieho zlepšenia a zisťovanie užívateľských preferencií [11]. V knihe od Steva Kruga *Don't Make Me Think, Revisited* [8] hovorí o výsledkoch jeho pozorovaní, ktoré ukázali veľmi malú časť prvotných používateľov, ktorí pri zakúpení softvéru alebo výrobku prečítali jeho návod na použitie. Namiesto toho radšej naslepo skúšajú priamo použitie jednotlivých funkcií, čo vedie k postupnému objaveniu hlavných funkcionality metódou pokus-omyl. Z tohoto dôvodu je podstatné zisťovať ako používatelia reagujú na rozhranie, s ktorým majú pracovať.

Článok *Usability Testing 101* z portálu Norman Nielsen Group popisuje ako hlavné prvky testovania použiteľnosti moderátora, úlohy pre používateľa a samotný účastníci testovania (obrázok 2.2).

Moderátor môže byť, ako už vyššie spomínaný, výskumník alebo študent, ktorý prevádza užívateľa cez jednotlivé úlohy, odpovedá na užívateľom kladené otázky a kladie doplňujúce otázky. Moderátor musí zabezpečiť, že dáta budú smerodajné a nesmie nevedome ovplyvniť výsledok testovania. Úlohy môžu predstavovať jednotlivé funkcionality aplikácie, ktoré sú základom jej používania. Môžu byť buď otvorené (nechať užívateľovi priestor pri riešení) alebo veľmi špecifikované. Dôležitá je ich formulácia a vždy by mali byť predom pripravené z dôvodu konzistentnosti testovania (aj malá zmena vo formulácii či chyba vo frázovaní môže spôsobiť nepochopenie cieľa úlohy alebo ovplyvniť ako užívateľ danú úlohu vykoná) [14]. Účastníci by mali predstavovať realistickú vzorku užívateľov študovaného softvéru alebo produktu a mali by byť vybraný tak, aby ich predošlá skúsenosť neovplyvňovala výsledok testu [19]. Ďalej rozdelujeme užívateľské testovanie na kvalitatívne a kvantitatívne, kde kvalitatívne testovanie je zamerané na odhaľovanie problémov v užívateľskej skúsenosti a kvantitatívne je používané na zbieranie informácií o úspešnosti a rýchlosti vykonania jednotlivých úloh [14].





Obr. 2.2: Schéma toku informácií pri testovaní použiteľnosti. Podľa predlohy dostupnej na portály Norman Nielsen Group [14]

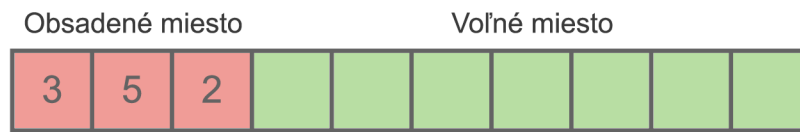
## 2.2 Pamäť a učenie

Učenie predstavuje neoddeliteľnú časť onboardingového procesu pre aplikáciu. Jedna z úloh tohoto procesu je naučiť nového užívateľa pracovať v novom prostredí alebo rozhraní. A fundamentálnym cieľom onboardingového procesu je užívateľovi pre neho čo najschodnejšou a najúčinnnejšou cestou dodať všetky potrebné informácie na obsluhu novej aplikácie a zoznámenie užívateľa s jej prínosom.

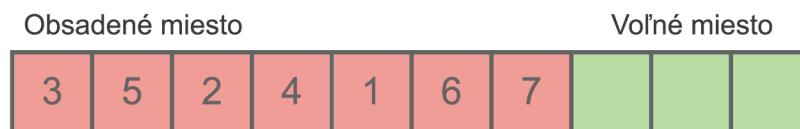
### 2.2.1 Pamäť

V článku Working Memory and External Memory [5], ktorý vychádza z knihy The psychology of learning and motivation: Advances in research and theory [2] je venovaná hlavným problémom, ktoré sa vyskytujú pri tvorbe užívateľských rozhraní a onboardingových procesov z pohľadu ľudskej pamäte a jej vyťažovania. Pozorovaniami bolo zistené, že častým problémom pri práci s rozhraniami je zabúdanie informácií, ktoré sú potrebné pre ďalšiu prácu z dôvodu, že rozhranie požaduje držať v pamäti viac informácií ako je naša pracovná pamäť schopná udržať. Pracovná pamäť funguje podobne ako zásobník s obmedzeným množstvom prvkov, ktoré sú relevantné k zadanej úlohe (obrázok 2.3). Ak pri plnení úlohy je potrebné viac informácií ako je naša pracovná pamäť schopná uchovať, má to za následok vypustenie niektorých informácií získaných skôr. Ak vypustené informácie ale sú potrebné na dokončenie úlohy užívateľ musí vynaložiť väčšie úsilie na obnovenie týchto informácií, čo má za následok dlhší čas na vykonanie úlohy alebo jej zvýšenú chybovosť. Koncept pracovnej pamäte bol prvýkrát ilustrovaný v sérii experimentov vykonaných psychológmi Alan Baddeley and Graham Hitch z University of Stirling, Scotland [2]. Kde v experimentoch boli účastníkom dané čísla od 1 po 6 aby ich uchovali v pamäti a následne im bola zadaná úloha ktorú mali plniť. Čím viac čísel si účastníci museli zapamätať, tým horšie výsledky dosahovali v zadanej úlohe. Čo naznačovalo, že čím viac priestoru v svojej pracovnej pamäti účastníci zabrali číslami, tým menej priestoru zostávalo na zadanú úlohu.

## Nevyťažená pamäť



## Vyťažená pamäť



Obr. 2.3: Ilustrácia zásobníka pracovnej pamäte pri experimente v práci Baddeley and Hitcha, ktorý naznačoval, že pracovná pamäť je systém s obmedzenou kapacitou a prostriedky sú rozdelené medzi uchovávanie informácií a ich spracovávanie. (Podľa predlohy dostupnej na portály Norman Nielsen Group [5])

Pojem pracovná pamäť a krátkodobá pamäť býva často zamieňaný. Pri čom ale koncept pracovnej pamäte je orientovaný na úlohu a krátkodobá pamäť je proces v našom mozgu, ktorý dovoľuje ukladanie informácií na krátky časový interval [5]. Krátkodobú pamäť skúmal aj Georg Miller, ktorý rozlišoval medzi bitmi a zhlukmi informácií (anglicky **chunks**) [20].

Zhlukovanie (ang. Chunking) informácií je proces, kedy skupiny atomických informácií vytvárajú skupiny podobných informácií, čo výrazne uľahčuje jedincovi zapamätať si túto skupinu informácií nakoľko ide o jeden informačný celok, ktorý má pre neho využitie a je pripravený na udržanie v pamäti [20]. Ľudská pamäť vie takýchto zhlukov uchovať viac. V pozorovaní George Miller prišiel na fakt, že priemerný počet takýchto zhlukov informácií, ktoré si dokáže jedinec zapamätať je 7. Číže z hľadiska onboardingového procesu nie je vhodné od používateľa požadovať uchovávať v pamäti viacej ako 7 informačných celkov na splnenie úlohy. Vyžadovať od užívateľa pri práci veľké množstvo predom podaných informácií zvyšuje kognitívnu záťaž, ktorá je na neho kladená [4]. Pri komplexnejších úlohách, ktoré vyžadujú viac informácií ako je priemerný užívateľ schopný uchovať je vhodné využiť istú formu externej pamäte. Externá pamäť označuje nástroj alebo funkcionality, ktorá užívateľovi dovoľí uchovať potrebné informácie pre prácu, ktoré by inak musel uchovávať v svojej pracovnej pamäti [5].

### 2.2.2 Metódy učenia

S učením blízko súvisí aj vyššie spomínané zhlukovanie informácií do zmysluplných skupín 2.4. Informácie podané v jednom súvislom bloku sú horšie interpretovateľné a ťažšie zapamätateľné a dokonca môžu užívateľa odradiť od čítania (obrázok 2.4). Steve Krug vo svojej knihe *Don't make me think*, Revisited taktiež hovorí o skladaní informácií do zmysluplných blokov, ktoré užívatelia ľahšie pre-skenujú <sup>1</sup>, čo zvyšuje pravdepodobnosť, že nájdu to, čo

<sup>1</sup>Prieskumy ukázali, že návštevníci stránok nečítajú obsah, ale skenujú stránku s cieľom nájsť pre nich relevantný obsah [8].

na stránke hľadajú [8]. **Zhlukovanie**<sup>2</sup> je často využívané ako technika na memorovanie [13]. V článku publikovanom v časopise Science popisujú autori Jeffrey D. Karpicke a Janell R. Blunt [7] ako experimentom zistili, že študenti, ktorí po prečítaní textu boli podrobení testu z jeho obsahu mali objem zapamätaných informácií o 145% väčší ako študenti, ktorí text iba prečítali. Vďaka práci s textom v pamäti kedy študenti museli text nie len prečítať ale aj informácie využiť bola ich pamäť schopná uchovať viacej informácií ako len pri jednoduchom prečítaní. Z týchto zistení vyplýva záver, že samotný akt využitia informácie zakóduje informáciu pevnejšie v mozgu.

Informácia v jednom bloku:

**+421912467985**

Informácia v rozdelená do skupiniek:

**+421 | 912 | 467 | 985**

Obr. 2.4: Ilustrácia rozdelenia informácie na menšie skupiny zľahčujúce jej spracovanie. (podľa predlohy dostupnej na [NNgroup](#))

---

<sup>2</sup>Proces zoskupovania informácií do zmysluplných skupín

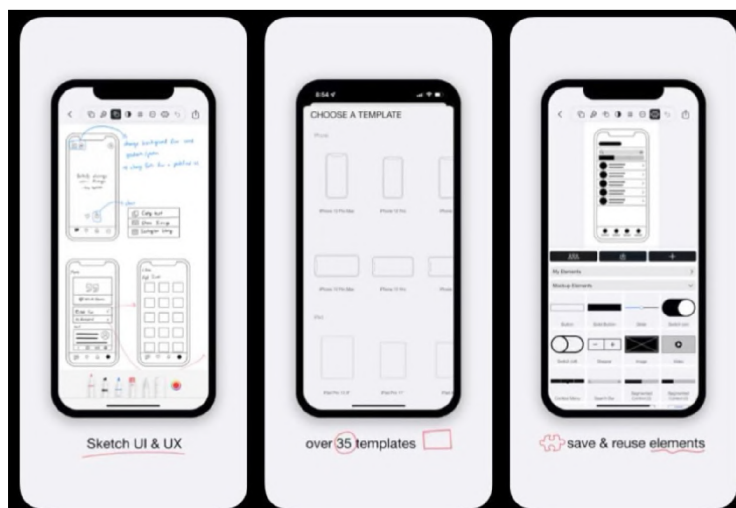
## 2.3 Onboardingové procesy v aplikáciách

Onboardingové procesy v aplikáciách slúžia na minimalizovanie kognitívnej záťaže na užívateľa pri prvotnom používaní [18]. Sú rozdelené primárne na pasívne a aktívne [16]. Medzi pasívne sú zaradené kartičky informujúce o možnostiach aplikácie a medzi aktívne patrí interaktívne prevedenie užívateľa cez ovládacie prvky hry alebo aplikácie. Taktiež bývajú využívané gamifikujúce prvky ako ukazovateľ postupu v procese alebo odmeny za dosiahnutý postup [12].

Na základe analýzy súčasných onboardingových stratégií, boli identifikované dve hlavné kategórie onboardingových procesov [18]:

- Úvodný obsah (zodpovedá k čomu aplikácia slúži a čo dokáže)
- Tutoriály<sup>3</sup> (zodpovedá ako aplikácia funguje a ako ju používať)

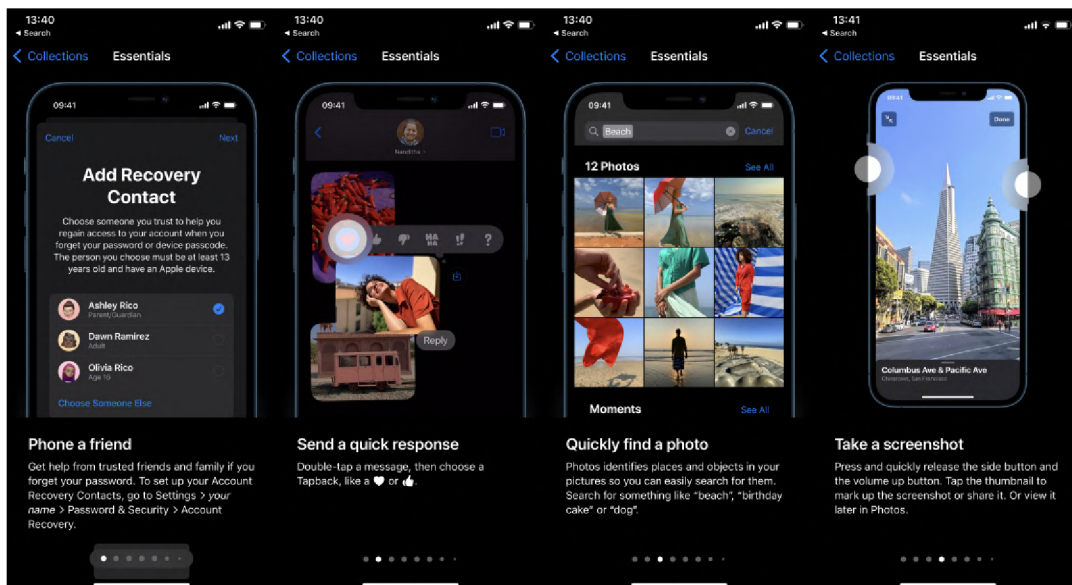
Do úvodného obsahu je zaradená napr. prezentácia funkcionality. Prezentácia funkcionality slúži na zoznámenie užívateľa s funkcionalitami, ktoré mu aplikácia ponúka. Môže ísť o úvodné zoznámenie alebo o zvýraznenie jednotlivých funkcií v aplikácii ako pomoc v kontexte (obrázok 2.5)[6].



Obr. 2.5: Funkcionálne orientovaný onboardingový proces aplikácie Mockup kde užívateľ je oboznámený s hlavnými funkcionalitami, ktoré mu aplikácia ponúka. Obrázok prevzatý z článku [12]

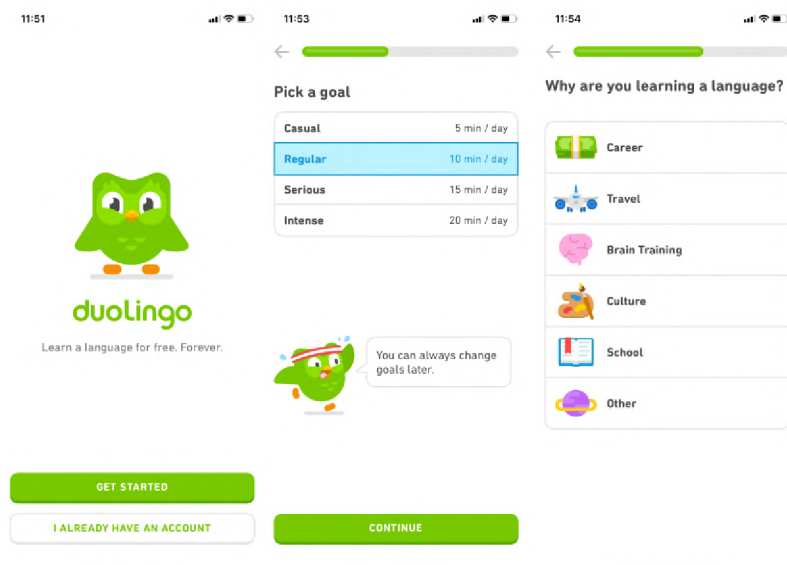
Využitie onboardingového procesu zameraného na akcie uľahčuje novému užívateľovi udržanie pozornosti pri procese učenia. Učiaci sa používatelia cítia potrebu spraviť postup v procese učenia a dosiahnuť cieľ [17]. Vyššie spomínané odmeny za dosiahnutie postupu spojené so správnymi inštrukciami výrazne zvyšujú úspešnosť onboardingového procesu. Inštrukcie by mali byť stručné, nepovinné a mali by oboznámiť užívateľa z minimálnym objemom informácií, ktoré užívateľ k použitiu potrebuje vedieť [6]. Existuje viacero štýlov podania informácií, často ide o sadu kartičiek ktorá zoznamuje užívateľa s rozmiestnením jednotlivých ovládacích prvkov a ich funkciami (obrázok 2.8). Pri špecializovanejších aplikáciách býva zvolená forma interaktívneho tutoriálu, ktorý dovoľuje priamu interakciu a adaptuje prístup učenia pri práci.

<sup>3</sup>slovenský preklad príručky, nakoľko ale ide o ťažko doslovne preložiteľné slovo ponechané v anglicizme



Obr. 2.6: Sada kartičiek v aplikácii Tips, ktorá je v zariadeniach Apple predinštalovaná a sprevádza používateľa od samotného predstavenia funkcionality cez ich použitie až po ich umiestnenie. Obrázky prevzaté z aplikácie <https://apps.apple.com/us/app/tips/id1069509450>.

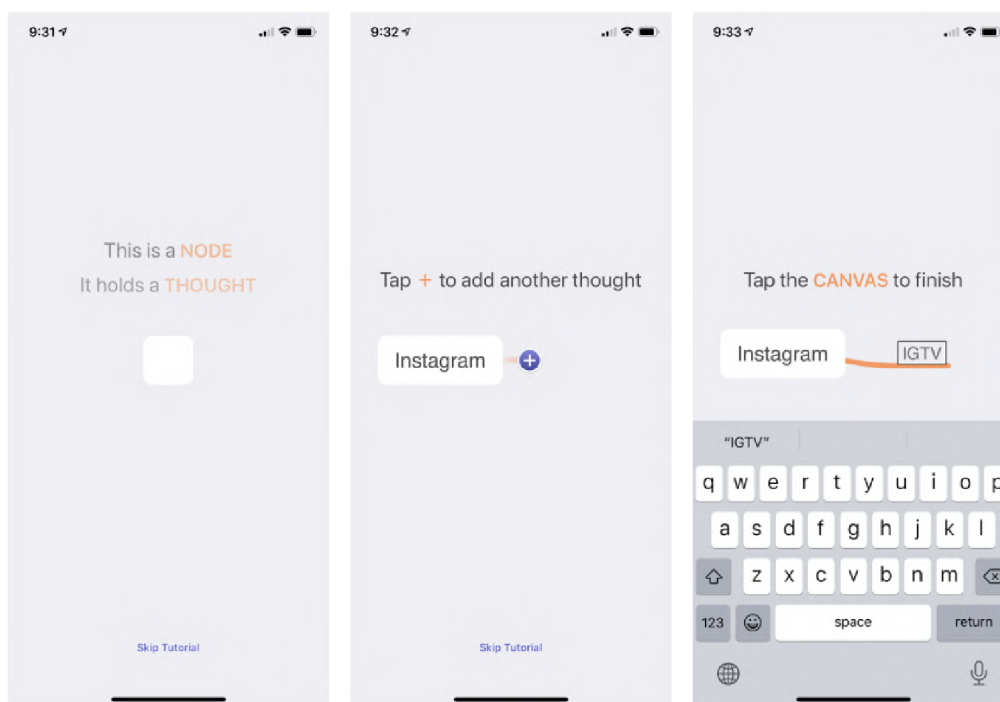
Tento prístup so sebou prináša isté nedostatky pozostávajúce z kladenia záťaže na pamäť užívateľa 2.4. Ako aj nedostatočná možnosť vyskúšania predstavovanej funkcionality, čo opäť môže záporne vplyvať na schopnosť užívateľa uchovať komunikované informácie v pamäti.



Obr. 2.7: Prispôsobenie aplikácie duolingo, ktorá je určená pre ľudí, ktorí sa chcú učiť cudzí jazyk pomocou aplikácie. Na začiatku užívateľ zadá aký jazyk sa má v pláne naučiť, aký čas je ochotný denne učeniu venovať a aký je cieľ jeho učenia. Bez týchto informácií by aplikácia fungovať nemohla. Obrázky prevzaté z portálu <https://www.reallygoodux.io/blog/duolingo-user-onboarding>.

Prispôsobenie užívateľskej skúsenosti býva súčasťou interaktívneho tutoriálu a je pri niektorých aplikáciách nevyhnutným prvkom, nakoľko na ich správne fungovanie sú potrebné dáta od užívateľa (obrázok 2.7)[6]. Formou interaktívneho tutoriálu je aj interaktívna walktrough<sup>4</sup>. Interaktívna walktrough býva využitá na prevedenie užívateľa úlohami, ktoré sú základom aplikácie, v ktorej sú implementované (obrázok 2.8). Hlavnou výhodou tohoto prístupu je zapojenie užívateľa do procesu učenia popri využívaní aplikácie k naplneniu jeho cieľa. Interaktívna walktrough by mala obsahovať krátke popisy, ktoré prevádzajú užívateľa prvotnou úlohou ako aj krátke popisy funkcionalít, ktoré sú pri plnení úlohy využívané [18].

<sup>4</sup>Prevedenie užívateľa základnou funkcionalitou aplikácie alebo hry.



Obr. 2.8: Aplikácia MindNode, ktorá je používaná na mapovanie nápadov a brainstorming<sup>6</sup>. Využíva walkthrough na zoznámenie užívateľov s ich menej známym pracovným postupom. Užívateľia vytvorili podľa inštrukcií jednoduchú mapu mysle a boli zoznámení s ovládacími prvkami a terminológiou používanou v aplikácii. Obrázok a popis prevzatý z článku <https://www.nngroup.com/articles/mobile-app-onboarding/>.

## 2.4 Rozšírená realita

V súčasnosti existujú na trhu rôzne druhy rozšírenej reality. Dvoma najpoužívanejšími typmi sú priehľadné displeje nosené na hlave a mobilné zariadenia ako tablety a inteligentné telefóny. Priehľadné náhlavné displeje poskytujúce užívateľovi možnosť širšieho zorného poľa a voľné ruky pre prácu s digitálnymi objektami.



Obr. 2.9: Zariadenie HoloLens 2 od firmy Microsoft poskytujúce možnosť práce v rozšírenej realite a manipuláciou s digitálnymi objektmi pomocou gestikulácie rúk. Obrázok prevzatý z <https://www.microsoft.com/en-us/hololens/buy>.

Ďalšia rozšírená platforma sú mobilné zariadenia ako inteligentné telefóny a tablety disponujúce kamerou, kde užívateľ môže pomocou týchto zariadení pracovať s digitálnymi objektami v priestore. Stačí mu na to aplikácia a zariadenie, ktoré je cenovo dostupnejšie ako vyššie uvedený produkt 2.9. Ale nevýhodou je potreba držania zariadenia pred sebou a únava rúk spojená s dlhšou prácou na takomto zariadení. Taktiež ako aj veľmi obmedzené využitie rúk pri práci s takýmto typom rozšírenej reality (obrázok 2.10).



Obr. 2.10: Aplikácia v tablete využívajúca rozšírenú realitu v priemyselnom prostredí pre programovanie robotického pracoviska [19].



V článku *Current issues in handheld augmented reality* rozoberajú problematiku zorného poľa ľudského oka verzus zorného poľa kamery sprostredkávajúcej obraz, ktorý užívateľ vidí na tablete [10]. Aj keď kamery v súčasných inteligentných telefónoch a tabletoch poskytujú vysoké rozlíšenie obrazu, dokážu pokryť zorné pole ľudského oka len z časti. Ľudské oko má zorné pole  $180^\circ$  oproti displeju používanému pre rozšírenú realitu, ktorý priestorovo zaberá minimálnu časť. A preto je potrebné užívateľovi pomocou šípok pripomínať jeho relatívnu pozíciu k objektom umiestneným v rozšírenej realite. Nakoľko by práca s viacerými objektmi v scéne značne vyťažovala krátkodobú pamäť a zvyšovala kognitívnu záťaž na užívateľa ako opisuje sekcia 2.2.1, v ktorej je rozoberaný aj význam jednoduchého dizajnu užívateľských rozhraní vzhľadom na kognitívnu záťaž kladenú na užívateľa.

Téme skreslenej hĺbky v rozšírenej realite je venovaný článok *Perceptual issues in augmented reality revisited*, kde popisujú tento problém ako jeden z najčastejších problémov rozšírenej reality [9]. Zaoberá sa celkovou problematikou ľudského vnímania rozšírenej reality od vnímania veľkostí, polohy, hĺbky až po usporiadania objektov na scéne .

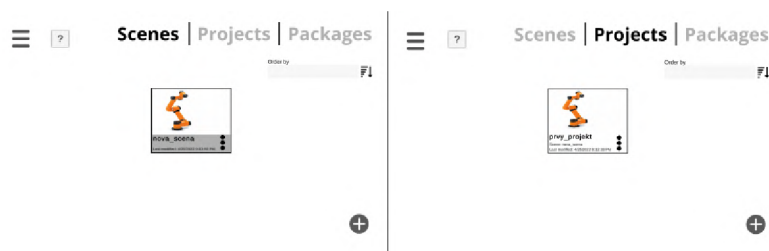
## Kapitola 3

# Návrh riešenia

V tejto kapitole sú popísané všetky aspekty, ktoré viedli k finálnej implementácii riešenia ako aj zoznámenie s prostredím, v ktorom bude moja práca implementovaná. Cieľom kapitoly je oboznámiť čitateľa s postupom, ktorý predchádzal výberu vhodnej metódy onboardingu. Kapitola zahŕňa zoznámenie s aplikáciou **ARCOR2 AREditor**<sup>1</sup>, analýzu predošlých používateľských experimentov, vykonanie vlastného experimentu so zameraním na problémy zistené v predošlých užívateľských experimentoch, analýza výsledkov mojich užívateľských experimentov a zvolenie vhodnej metódy onboardingového procesu. Na konci tejto kapitoly bude čitateľ oboznámený s návrhom riešenia, ktoré bude implementované v aplikácii AREditor.

### 3.1 Aplikácia AREditor

Aplikácia AREditor je klientská aplikácia ponúkajúca užívateľské rozhranie komunikujúce so serverom ARCOR-Augmented Reality Collaborative Robot vyvíjaná výskumnou skupinou robotiky Robo@FIT (ROBO)<sup>2</sup>. Systém ARCOR slúži k zjednodušenému programovaniu spolupracujúcich robotov založených na rozšírenej realite<sup>3</sup>. Systém bol vyvíjaný so zámerom zjednodušenia programovania spolupracujúcich robotov a možnosti vytvárania aj komplexnejších liniek pozostávajúcich z viacerých robotických zariadení priemerne kvalifikovaným užívateľom. A tým sprístupniť využitie takýchto robotických pracovísk širšiemu spektru užívateľov. Samotná aplikácia AREditor je implementovaná v engine Unity [1] a využíva jeho rozhranie pre rozšírenú realitu (obrázok 3.1).



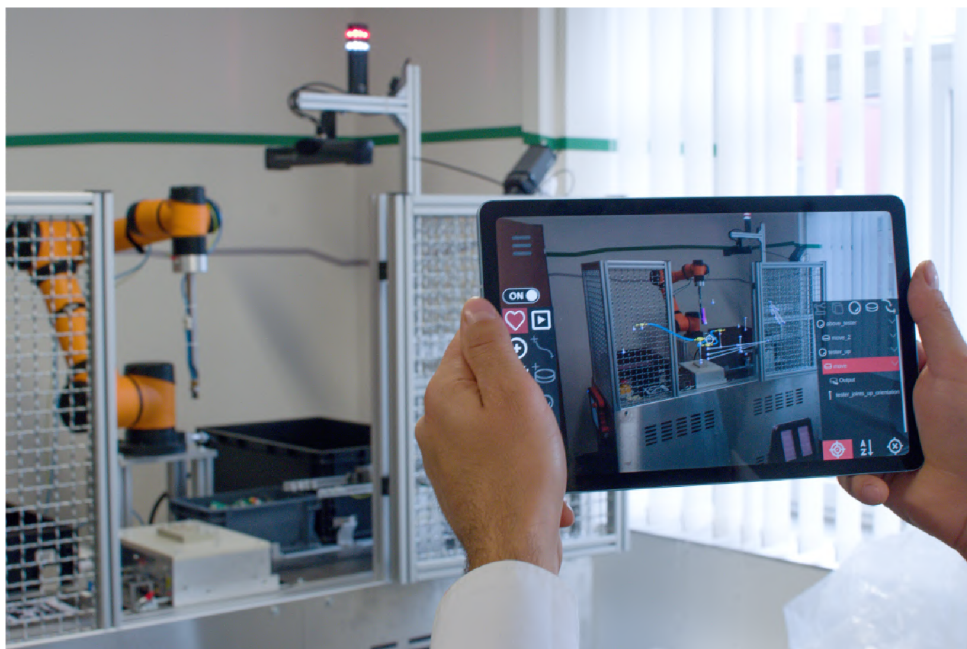
Obr. 3.1: Obrazovka hlavného menu aplikácie AREditor zobrazujúca vytvorené scény a projekty.

<sup>1</sup>Ďalej iba AREditor, dostupné na: [https://github.com/robofit/arcor2\\_areditor](https://github.com/robofit/arcor2_areditor)

<sup>2</sup>Webstránka výskumnej skupiny: <https://www.fit.vut.cz/research/group/robo/.cs>

<sup>3</sup>Dostupné na: <https://github.com/robofit/arcor2>

Aplikácia AREditor ponúka možnosť vytvárania scén, ktoré definujú prvky pracoviska, ich roloženie a konfiguráciu, v ktorom budú programy vykonávané (napr. robotické pracovisko, výrobná linka). Podporuje možnosť pridania viacerých spolupracujúcich robotov a pridania virtuálnych objektov reprezentujúce reálne ohraničenia a prekážky na pracovisku (klietka okolo robotického pracoviska 3.2)).

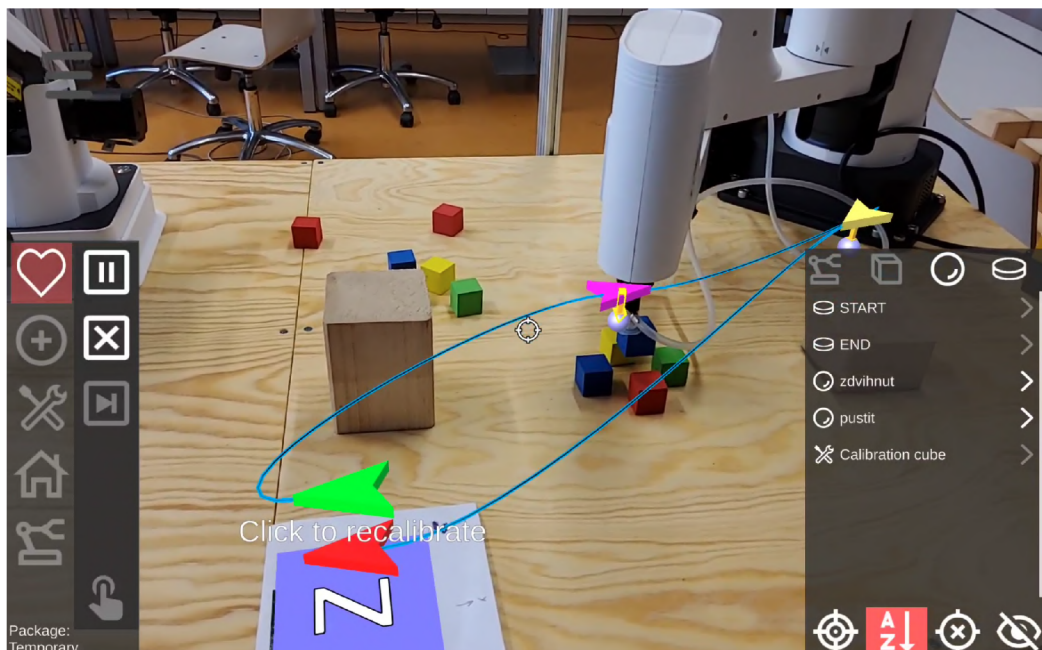


Obr. 3.2: Robotické pracovisko ohraničené klietkou [3].

Ďalej môže užívateľ v týchto scénach vytvárať projekty, ktoré už obsahujú samotné postupnosti akcií, ktoré majú byť vykonané (programy). Základným prvkom týchto programov sú **akčné body** <sup>4</sup>, ku ktorým sú naviazané **akcie**. <sup>5</sup> Akčné body je možné vytvárať buď manuálne, kedy je bodu neskôr pridaná orientácia a správna pozícia, alebo pomocou umiestnenia nástroja robota na miesto určenia, a vytvorenia akčného bodu v pozícii robotického hlavy. Pri vytváraní programov má užívateľ možnosť k vytvoreným akčným bodom pridávať akcie, ktoré sú v scéne pridané zariadenia schopné vykonať (vývojári môžu pridať alebo upraviť akcie, ktoré sú implementované). Tieto pridané akcie môžu byť pospájané spojeniami do programov. Spájanie akcií do programov vždy začína v akcii **START** a posledné spojenie smeruje do akcie **STOP** (obrázok 3.3). Tieto špeciálne akcie ohraničujú začiatok a koniec programu, a práve akcie pospájané medzi týmito špeciálnymi akciami budú vykonávané v slučke, až kým vykonávanie nebude manuálne ukončené.

<sup>4</sup>Virtuálny objekt ukotvený v priestore.

<sup>5</sup>Jednotlivé funkcie, ktoré je robot schopný vykonať.



Obr. 3.3: Náhľad na vytvorený projekt v scéne.

Taktiež aplikácia dáva možnosť vývojárom exportovať zdrojový kód programu vytvoreného v aplikácii pre ďalšiu úpravu, ako aj priamo spustiť vytvorený program pre overenie jeho funkčnosti. Základným ovládacím prvkom aplikácie je zameriavač, ktorý slúži na výber objektov v scéne. Pohybovaním tabletu (zamierením) možno vybrať jednotlivé objekty, ktoré sú v scénach umiestnené a budú zobrazené v pravom paneli aplikácie. V základe je pravý panel nastavený na usporiadanie zobrazovania objektov podľa najmenšej vzdialenosti k zameriavaču. V ľavom paneli je umiestnená funkcionálna AREditoru (pridávanie objektov a akcií, posúvanie objektov, ukladanie projektu, nastavovanie robota) a v pravom paneli sú umiestnené objekty v scéne a rozšírené možnosti niektorých funkcionalít (posuvník pre pohybovanie a rotovanie objektov, detailné informácie pri nastavovaní orientácie). Rotácie a posúvanie objektov sú vykonávané pomocou zobrazených osí, na ktoré užívateľ zamieri zameriavačom a môže pomocou rotujúceho posuvníka meniť uhol alebo pozíciu na vybranej osi.

## 3.2 Analýza predošlých užívateľských experimentov

Jednou z prvých úloh, ktorú som pri riešení vykonal bola analýza užívateľských experimentov, v ktorých boli zúčastnení experti hodnotiaci navrhnuté užívateľské rozhranie AREditoru (ďalej iba účastníci). Experimenty prebiehali aj s expertmi na užívateľské rozhrania v laboratóriu na fakulte, ako aj so zamestnancami vo firme, s ktorou výskumná skupina spolupracuje. Tieto experimenty vykonal môj vedúci bakalárskej práce Ing. Zdeněk Materna, Ph.D (ďalej iba moderátor). Účastníci mali s aplikáciou AREditor prvú skúsenosť a mali za úlohu vykonať jednoduché úlohy pod vedením moderátora, ktorý sledoval ich postup a zisťoval ich spätnú väzbu k jednotlivým funkcionalitám, s ktorými práve pracovali. Priebeh testu spočíval v zoznámení účastníkov so základnými funkcionalitami ponúkanými AREditorom. Prevedením cez základy pracovného postupu a to posúvaním a rotáciou objektu, pridávaním akčných bodov a akcií, spájanie akcií do uceleného programu. Výstupom expe-

rimentov bola spracovaná tabuľka jednotlivých problémov, ich závažnosti, doporučeného riešenia expertom, a komentár vývojára. Z výsledkov testovania je ďalšia časť tejto sekcie zameraná na problémy riešiteľné formou onboardingového procesu.

Výsledky experimentov ukázali potrebu implementácie onboardingového procesu nakoľko viacerí účastníci zhodnotili rozhranie aplikácie ako ťažko použiteľné pre prvotného užívateľa. Častým problémom bola ne-familiárnosť s pracovným prostredím aplikácie, nakoľko je aplikácia špecializovaná a pracuje sa v rozšírenej realite. Aplikácia neponúka konvenčné rozloženie ovládacích prvkov čo spôsobovalo účastníkom problém s nájdením umiestnenia niektorých ovládacích prvkov ako napríklad zobrazenie nápovedy k tlačidlu, nástroje na posun a rotáciu objektov. Ďalší častý problém mali užívatelia s úplne novým pracovným postupom, ktorý vyžadoval vysvetlenie. Tento problém bol spojený aj s faktom, že v súčasnom stave aplikácie by užívateľ musel prechádzať každé tlačidlo aby mohol prečítať jeho nástrojový tip, čo by bolo pre nového užívateľa pokúšajúceho sa naučiť sa prácu s aplikáciou značne neefektívne. Následne boli účastníkmi spomenuté na prvý pohľad nejasná funkcionálna rotácia a posunu (ďalej iba transformácia). Transformačná funkcionálna má v AREditore z hľadiska dlhodobej použiteľnosti veľmi funkčné ale dosť špecifické riešenie, ktoré účastníkom nebolo na prvý pohľad zrejmé.

Väčšina z vyššie spomínaných problémov by bola riešiteľná vyskúšaním jednotlivých funkcionálnych a vytvorením základného programu s do-pomocou vývojára, ktorý je už skúseným používateľom aplikácie. Analýza hlavných problémov, s ktorými sa v aplikácii stretli aj experti a spracovanie ich rád ma viedlo k prípraveniu vlastného experimentu, ktorý by mal prebiehať ako komentované prevedenie prvotného užívateľa cez program a zoznámenie ho s hlavnými funkcionálnymi a pracovným postupom.

### 3.3 Vykonanie vlastného používateľského experimentu

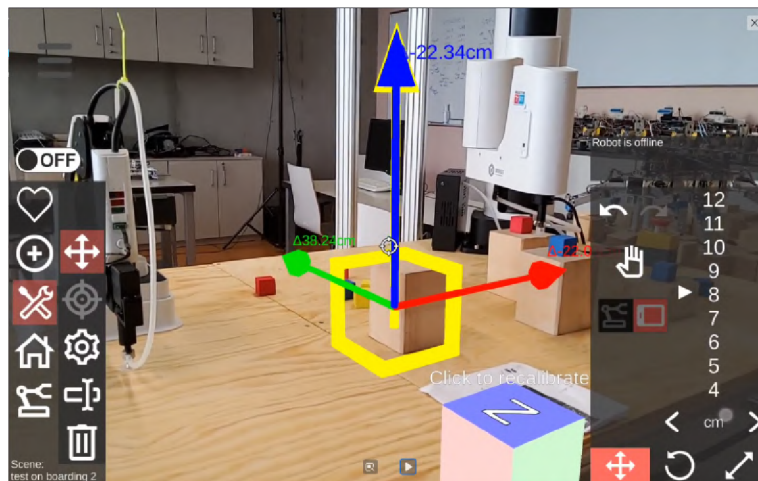
Po analýze užívateľským experimentom, na ktorých boli zúčastnení experti v obore a spracovaní nájdených problémov a ich doporučených riešení som vytvoril návrh a scenár pre svoj užívateľský experiment. V tejto sekcii som využíval informácie nadobudnuté v sekcii 2.1, kde sú popísané metódy užívateľského testovania. Experiment by mal zodpovedať otázky: Bude užívateľ schopný po vysvetlení a prevedení zadanou úlohou splniť podobnú ale zložitejšiu úlohu aj bez pomoci? Je takýto typ zaučenia pre používateľa prijateľný (nevyžadovala veľkú kognitívnu záťaž, udržala účastníka zaujatého 2.2.1)?

Zameranie experimentu bolo na overenie predpokladu, že vysvetlenie jednotlivých funkcionálnych pri používaní aplikácie by mohol stačiť užívateľovi na to, aby bol schopný po jeho absolvovaní s aplikáciou pracovať na úrovni, kedy už bude schopný zvyšok funkcionálnych objaviť metódou pokus-omyl 2.1. Nakoľko jednou s veľmi kritizovaných aspektov aplikácie bolo množstvo funkcionálnych, ktoré nie je možné slepým skúšaním objaviť a začať program používať bez vysvetlenia. Postup práce v aplikácii vyžaduje isté „*know-how*“. Ďalej by mal pomôcť odhaliť, ktoré funkcionality robia prvotným užívateľom najväčšie problémy, a tieto by boli v procese onboardingu podrobnejšie popísané. Experiment prebiehal na troch účastníkoch. Dvaja študenti Fakulty informačných technológií a jedna študentka humanitných vied. Výber účastníkov bol naplánovaný tak, aby bola v experimente vzorka účastníka, ktorý síce má skúsenosť s mobilnými zariadeniami ale nie je študentom informačných technológií. Dôvodom bola možnosť overenia, či vysvetlenie niektorých úloh nie je moc odborné pre ne-programátora, nakoľko práve ne-programátor by najskôr finálny onboarding využil. Celý experiment som zaznamenával pomocou nahrávania obrazovky tabletu pre neskoršiu analýzu.

Plán experimentu spočíval v troch fázach:

- Sprevádzanie účastníka jednoduchou úlohou s vysvetlením potrebnej funkcionality.
- Zadanie samostatnej práce účastníkovi pri ktorej som zapisoval problematické funkcionality.
- Diskusia s účastníkom a zodpovedanie pripravených otázok.

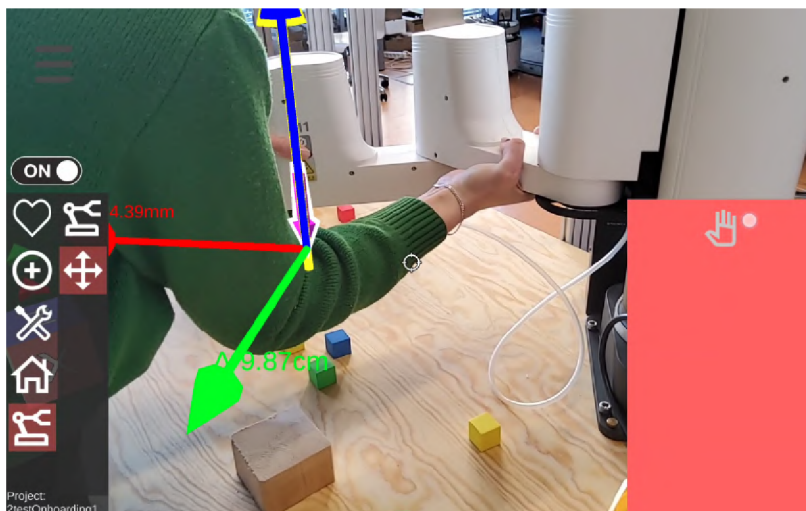
V **prvej fáze** prebehlo stručné zoznámenie účastníkov s priebehom testu. Následne som predal tablet užívateľovi a začal som vysvetlením na čo je aplikácia používaná, a ako to súvisí s vytváraním scén a programov. Ako prvú úlohu mali účastníci vytvoriť scénu, a následne v nej pridať používaného robota. Po vytvorení scény som im vysvetlil, kde hľadať funkcionality pridávania objektov, aký je rozdiel medzi dvoma možnosťami, ktoré sú umiestnené v menu na pridávanie, a kde nájdú pridanie robota do scény. Po pridaní robota do scény som účastníkov oboznámil s tým, že ich hlavný pracovný nástroj je zameriavač, a interagovať sa dá len s dvoma bočnými panelmi. Následne som účastníkom ukázal, kde nájdú transformačný nástroj (dovoľujúci rotáciu a posun objektu v scéne). Bolo nutné taktiež vysvetliť ako sa s týmto transformačným nástrojom pracuje a vysvetlil som im princíp vyberania osí zameriavačom, a následné posúvanie pomocou rolovacieho posuvníka v pravom paneli. Pri výklade som zdôraznil, že tento princíp posunu je v celej aplikácii rovnaký, taktiež aj potrebu pohybovať sa po priestore a kontrolovať pozíciu virtuálneho objektu z viacerých perspektív. Po umiestnení virtuálneho objektu robota na pozíciu reálneho bol ďalší krok pridať do scény objekt, ktorý bol reprezentovaný drevenou kockou na stole. Aby boli účastníci zoznámení aj s možnosťou zmeny veľkosti objektu v transformačnom paneli (obrázok 3.4).



Obr. 3.4: Umiestňovanie virtuálneho objektu na reálny objekt v pracovnom priestore (snímka obrazovky zo záznamu).

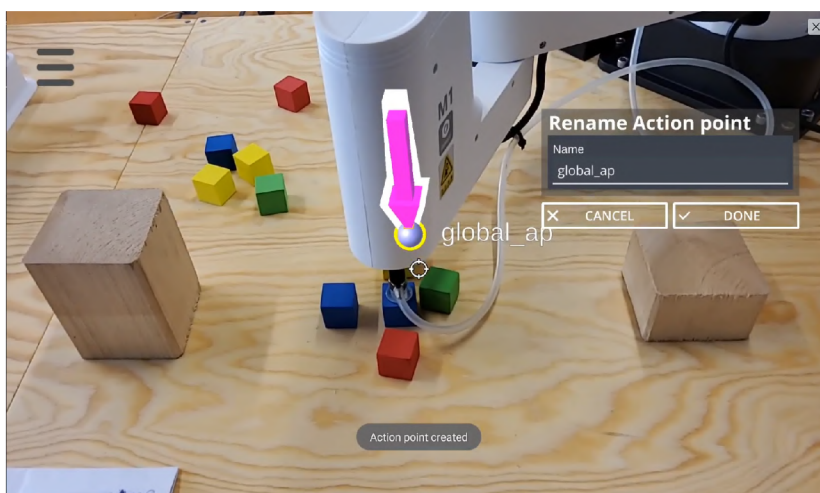
Po správnom umiestnení virtuálneho objektu nasledovalo zoznámenie účastníkov s tlačidlom domov, kde sa nachádza uloženie scény, opustenie scény a vytvorenie nového projektu. Ďalším krokom bolo vytvorenie nového projektu vo vytvorenej scéne. Po vytvorení projektu som účastníkom vysvetlil prečo sa už v projekte nachádzajú dve akcie **START** a **END** (viz sekcia 3.1), a ako to súvisí s vytváraním programu. Pokračoval som ďalej vo

vysvetľovanie samotnej úlohy ktorej logiku budú účastníci v projekte vytvárať. Šlo o presunutie jednej penovej kocky zo stola na vrchnú stranu drevenej kocky umiestnenej na stole. Ako prvé bolo nutné vysvetliť ako fungujú **akčné body**, a aké sú možnosti ich pridávania. V tomto experimente som zvolil vytváranie akčných bodov pomocou robota, nakoľko pri kalibračnej chybe by mohlo dôjsť k odchýlkam zobrazených akčných bodov a reálnej kocky a presúvanie by nefungovalo. Na vytváranie akčného bodu robotom bolo nutné účastníkom vysvetliť ako meniť pozíciu robotického ramena.



Obr. 3.5: Účastníčka posúvajúca robotické rameno na potrebné miesto (snímka obrazovky zo záznamu).

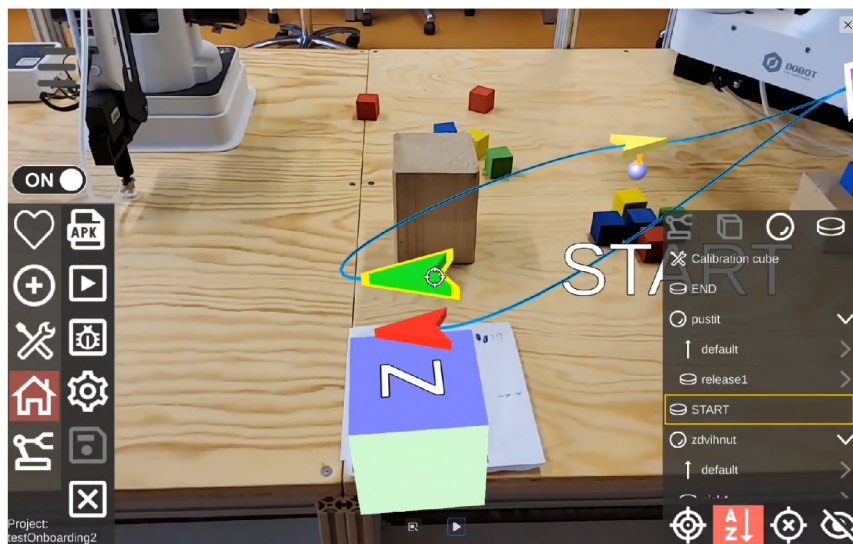
Pre účely experimentu som zvolil posúvanie pomocou voľného pohybu<sup>6</sup>. Po správnom umiestnení nástroja robota na penovú kocku, ktorú je za úlohu presunúť, nasledovalo pridanie **akčného bodu** na predtým nastavenú pozíciu robotického ramena.



Obr. 3.6: Pridanie akčného bodu na pozíciu robotického ramena (snímka obrazovky zo záznamu).

<sup>6</sup>Jednoduchší princíp ako pridanie akčného bodu, posunutie na správne miesto a následne pridanie orientácie.

Ďalej nasledovalo pridanie akcie predtým vytvorenému akčnému bodu. Na zdvihnutie kocky som účastníkom doporučil využiť funkciu `pick`, ktorá obsahuje aj presunutie robota na miesto akčného bodu aj prisatie kocky prísavkou. Následne som opäť s účastníkmi prešiel procesom vytvárania akčného bodu za pomoci robotického ramena a vytvorili sme druhý akčný bod, umiestnený na drevenej kocke kam bolo za úlohu penovú kocku presunúť. Tomuto akčnému bodu ale bola pridaná akcia `release`. V poslednej časti prvej úlohy som účastníkom ukázal kde sa nachádza tlačidlo na pridanie spojenia a ako tieto spojenia vytvárať. Pri vytváraní spojení som účastníkom taktiež poradil ako efektívne pracovať v kombinácii so zameriavačom a pravým panelom obsahujúcim objekty v scéne, nakoľko nie je veľmi zrejmé, že na vytvorenie spojenia je potrebné kliknúť na objekt, v ktorom má spojenie končiť. Apeloval som pri tom aj na potrebu správneho usporiadania akcií za sebou, tak ako budú vykonávané robotom a podľa toho ich správne spájať. Po úspešnom pospájaní akcií do programu som účastníkom zopakoval, že ukladanie nájdú pod tlačidlom domov a v projektovom menu je táto ponuka rozšírená aj o spustenie programu, pri ktorom treba zameriavačom mieriť na štart.



Obr. 3.7: Ukončené spájanie akcií do programu (snímka obrazovky zo záznamu).

V **druhej fáze** experimentu som účastníkov zoznámil s náplňou druhej úlohy, ktorá pozostávala z umiestnenia dvoch penových kociek rozložených v pracovnom priestore na jednu pozíciu. Úloha obsahovala kroky, ktoré boli v prvej fáze experimentu vysvetlené. Zadanie bolo rozšírené o jednu akciu čím som plánoval overiť porozumenie princípu vytvárania programu (spájanie jednotlivých akcií v správnom poradí). Účastníkom bol ponechaný priestor na samostatné vypracovanie úlohy a ja ako moderátor som ich sledoval a odpovedal na otázky ohľadom funkcionality.

V **tretej fáze**, ktorá nasledovala po splnení druhej úlohy nadviazala krátka diskusia, v ktorej sme sa s účastníkmi venovali pocitom z používania aplikácie, miere dosiahnutého porozumenia a pocitom z formy zoznamovania. Z tejto diskusie som vytvoril krátke poznámky, ktoré neskôr slúžili k lepšiemu zameraniu pri návrhu implementácie.

Experiment prebehol podľa plánu a žiadny z účastníkov nemal výrazný problém s jeho priebehom. Dokonca všetkých účastníkov účasť na experimente bavila a boli radi za možnosť zoznámiť sa s takýmto druhom technológií.



### 3.4 Zhrnutie zistení vykonaného používateľského experimentu

Používateľský experiment mi priniesol dôležité zistenia, ktoré som neskôr využil v návrhu implementácie onboardingového procesu. Tiež mi pomohol overiť typ použitej metódy na zoznámenie užívateľa s aplikáciou AREditor a pomohol nájsť funkcionality, ktoré sú najviac problematické pre prvotného užívateľa. Na tie som sa následne pri vytváraní systému rád zameriaval dôkladnejšie.

Zistenia z experimentu mali dva primárne zdroje:

- Diskusia a polozenie otázok na konci experimentu.
- Analýza záznamu obrazovky vytvoreného pri experimente.

Po diskusii s účastníkmi a analyzovaní ich odpovedí na otázky ohľadom testovania bolo zrejme, že účastníci nemali veľký problém chápať jednotlivým krokom pracovného postupu v aplikácii. Práca s robotom v rozšírenej realite účastníkov bavila, a bolo zrejme, že sa snažia dosiahnuť cieľ, ktorým bolo finálne vyskúšanie programu 2.2.1. Problémy na, ktoré účastníci narazili a vnímaná zložitosť na stupnici od 1 po 10 pred začiatkom práce a po absolvovaní školenia v AREditore boli takéto:

- **Prvá účastníčka** zhodnotila vnímanú zložitosť na začiatku práce hodnotou **9/10**. Kľúčové slová v aplikácii jej prišli mierne mäťúce, ako aj spájanie akcií do programov. Po absolvovaní školenia ale bola schopná len s pár otázkami dokončiť aj zložitejšiu úlohu a jej hodnotenie vnímanej zložitosti na konci bolo na hodnote **3/10**.
- **Druhý účastník** zhodnotil vnímanú zložitosť na začiatku práce hodnotou **7/10**, nakoľko ale išlo o študenta informatiky veľmi rýchlo po vysvetlení základných funkcií na prvej úlohe bol schopný spraviť druhú zložitejšiu úlohu, a na konci zhodnotil vnímanú zložitosť na použitia hodnotou **2/10**.
- **Tretí účastník** zhodnotil vnímanú zložitosť na začiatku práce rovnako ako druhý hodnotou **7/10**. Išlo opäť o študenta informatiky, ktorý nemal problém so spájaním dielčích častí programu do finálneho programu. A po vysvetlení nemal problém splniť aj zložitejšiu úlohu. Po skončení práce zhodnotil vnímanú zložitosť na použitie hodnotou **3/10**.

Tieto pozorovania ukazovali výrazné zlepšenie použiteľnosti aplikácie pre zaškolených užívateľov a potvrdzovali to aj pozorovania užívateľov pri samostatnej práci na druhej úlohe. Niektoré problémy ale aj po vysvetlení funkcionalít a postupov práce stále pretrvávali. Vykonaním experimentu som na účastníkoch testovania pozoroval ich problém s uvedomením faktu, že rozšírená realita pridáva do reálneho priestoru virtuálne objekty. Čo viedlo k nepresnému umiestneniu objektov v rozšírenej realite, nakoľko vnímanie vzdialenosti objektu v rozšírenej realite zobrazenej na displeji, dávalo užívateľom falošný pocit správne odhadnutej vzdialenosti. Z dôvodu, že displej tabletu je monoskopický na rozdiel od HoloLens, ktoré ponúkajú stereoskopický obraz čiže je možné vnímať hĺbku (viz sekcia 2.4). Tiež aj zanedbanie faktu, že pracujú v trojrozmernom priestore. Na efektívnu prácu s AREditorom je potrebné sa pohybovať okolo objektov, ktoré sú do rozšírenej reality umiestnené, čím vlastne zabezpečuje užívateľ pohľady na objekty z rôznej perspektívy a výrazne to uľahčuje presné umiestňovanie objektov. Pre účastníka bez predošlej skúsenosti s programovaním bol väčší problém pochopiť princíp spájania objektov v správnom poradí do finálneho programu.

Nakolko ale aj pri ostatných účastníkoch bolo treba podrobne popísať postup vytvárania spojení, by sa dal tento problém pripísať aj menej intuitívnemu postupu práce. Ďalej som pozoroval problém užívateľov rušiť alebo mazať vytvorené spojenia nakolko nie je jasné, že pri vytvorení nového spojenia sa to staré spojenie zruší.

Žiaden z účastníkov nemal problém po vysvetlení prvej úlohy vykonať úlohu druhú, len s minimálnou pomocou. Z experimentu bolo zrejmé, že jedno z najlepších riešení by bola **interaktívna walktrough**<sup>7</sup> aplikáciou s panelmi obsahujúce informácie 2.4, ktoré by užívateľovi pomohli splniť základné úlohy ako vytvoriť pracovnú plochu, v nej projekt a v ňom program 2.3. Ďalej oboznámiť užívateľa s princípom nástrojových typov a ich použitia, a zoznámiť ho s krokmi a princípmi pracovného postupu. Tieto nápovedy by mohli byť ako okienka, ktoré budú obsahovať vysvetlenie k funkcionalite, ktorú budú mať za úlohu použiť. A sprevádzať užívateľa jednotlivými krokmi rovnako, ako by ich sprevádzal skúsený používateľ. Tým by mal používateľ možnosť vyskúšať si aplikáciu a jej funkcie a bude sa snažiť dosiahnuť cieľ, ktorým by bolo spustenie prvého programu. Dodatočnú pomoc po absolvovaní interaktívnej walktrough môžu pri používaní poskytnúť nástrojové tipy, ktoré sú už v aplikácii implementované.

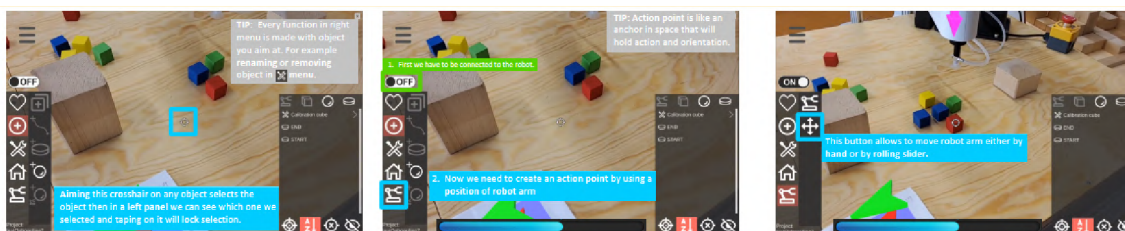
## 3.5 Návrh riešenia v aplikácii

V tejto sekcii sa budem venovať samotnému návrhu riešenia, ktoré bude implementované. Rozhodol som sa využiť pre svoje riešenie metódu interaktívneho walktrough, nakolko vyššie uvedené zistenia a pozorovania ukazovali vhodnosť tejto metódy na zaučenie používateľa do použitia aplikácie 3.4. Navrhol som systém nápovedí sprevádzajúci užívateľa jednoduchou tvorbou programu v AREditore. Systém nápovedí bude obsahovať jednotlivé kroky s popisom postupu práce s AREditorom, a poskytne užívateľovi dodatočné tipy k ďalšiemu použitiu aplikácie. Systém nápovedí bude fungovať ako vrchná vrstva obsahujúca panely s informáciami, zvýraznenie nasledujúcich funkcií a tlačítka na posúvanie späť, dopredu v krokoch onboardingu taktiež aj tlačítka na jeho uzatvorenie. Táto vrstva bude spustená z dialógového okna, ktoré užívateľa oboznámi s používaním nástrojových tipov a umožní začať walktrough. Rozhodol som sa využiť tlačítka v hlavnom menu AREditoru, s ikonou otázniku.

### 3.5.1 Koncept užívateľského rozhrania

Pri návrhu užívateľského rozhrania som sa inšpiroval rozhraním využívaným v hrách, kedy sú užívateľovi popri prechádzaní jednoduchého levelu postupne vysvetľované ovládacie prvky hry. Rozhodol som sa pre tento postup, nakolko veľmi dobre sedel ku komplexnosti ovládacích prvkov v AREditore. Inšpiráciu som čerpal už existujúcich riešení popísaných v sekcii 2.3. Vytvorenie konceptu spočívalo vo využití videa vytvoreného pri experimente na vytvorenie návrhu. Pri prechádzaní videa som jednotlivé vysvetlenia podané užívateľovi lepšie pre-formuloval a následne skrátil tak, aby výpovedná hodnota nebola zmenená, ale mohli byť zapísané do jednotlivých panelov. S vytvorením konceptu teda bolo spojené aj rozpísanie obsahu jednotlivých krokov, ktoré budú potom do aplikácie implementované.

<sup>7</sup>Prevedenie užívateľa základnou funkcionalitou aplikácie alebo hry.

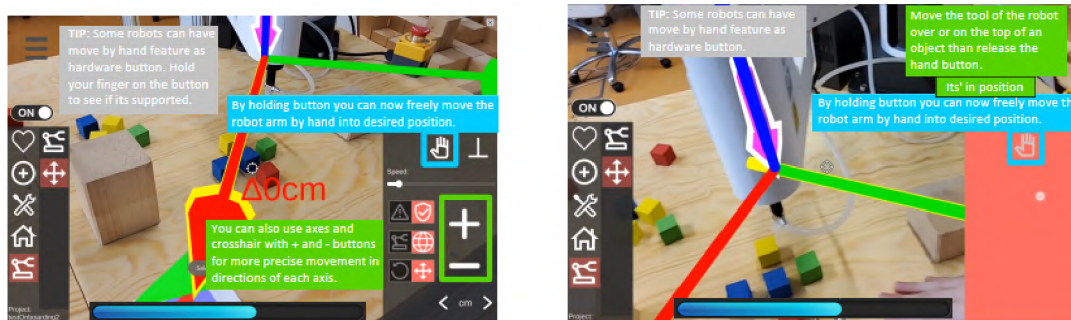


Obr. 3.8: Koncept návrhu užívateľského rozhrania vytvorený v Microsoft OneNote slúžiaci na overenie vybranej metódy a vizualizáciu riešenia pred jeho implementáciou. Koncept taktiež slúžil ako základné informácie, ktoré by mali byť užívateľovi komunikované v jednotlivých krokoch.

### 3.5.2 Využitie nadobudnutých poznatkov pri návrhu

Metódu interaktívnej walktrough som navrhol na základe zistení popísaných v sekcii 2.3. S následným overením tejto metódy pomocou experimentu. Pomocou experimentu a dodatočne položených otázok som aj zistil, ktorým funkcionalitám pri prechádzaní pracovným postupom je potrebné podať podrobnejšie vysvetlenie 2.1. Aj to, ktoré funkcie nerobili prvotným používateľom žiaden problém. Ďalej táto metóda pomôže užívateľom ľahšie vstrebávať objem informácií, ktoré sú potrebné k ovládaniu AREditoru, využitím učenia pri práci. Na navrhnutie samotných krokov v systéme nápoved som využil poznatky z metód učenia 2.2.2, kedy som navrhol jednotlivé informačné celky ako tri farebne rozlíšiteľné panely obsahujúce časť informácie potrebnej k úlohe alebo funkcionalite. Takto užívateľ pri prechádzaní walktrough bude mať jednotlivé informácie potrebné v kroku rozdelené do menších skupiniek informácií s kontextom 2.4. Farebné odlíšenie potom bude využité na odlíšenie rôznych typov informácií v priebehu walktrough. Rozvrhol som informácie do troch typov:

- Modrý panel s primárnymi informáciami o funkcionalite používanej v kroku.
- Zelený panel so sekundárnymi informáciami o ďalšom pracovnom postupe.
- Šedý panel obsahujúci tip k danej funkcionalite alebo úlohe, ktorá má byť vykonaná.



Obr. 3.9: Kroky z konceptu využívajúce jednotlivé panely na komunikáciu informácií.

**Modrý panel** s primárnymi informáciami k funkcionalite bude taktiež obsahovať ukazovateľ postupu v onboardingovom procese. Rozhodol som sa využiť tento prvok, nakoľko

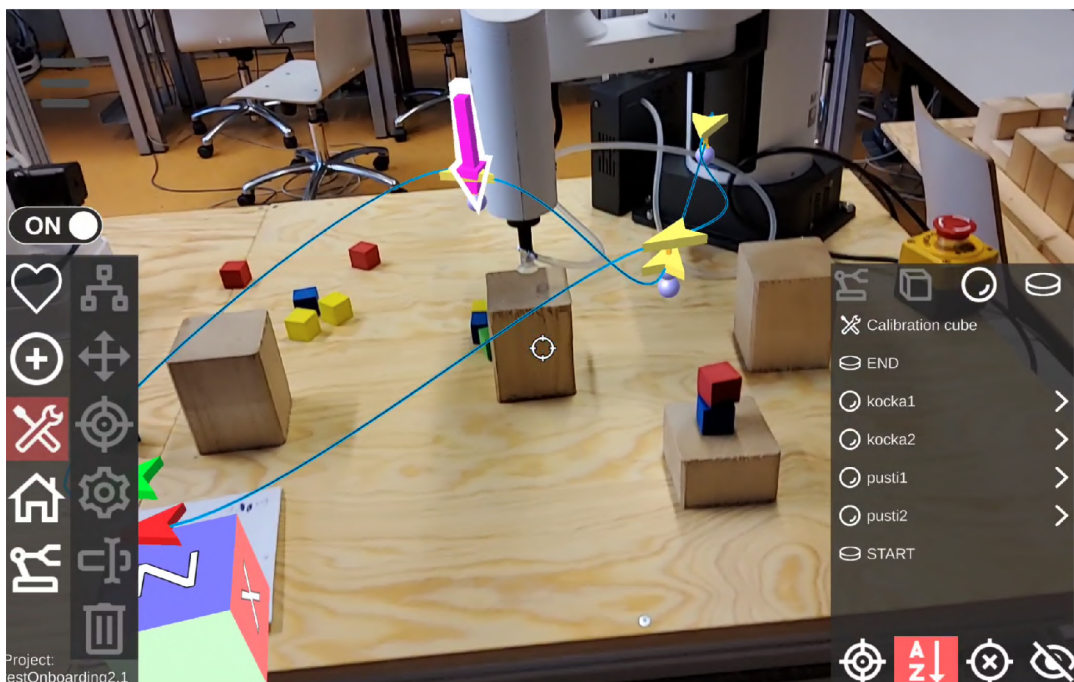
užívateľia budú oboznámení so svojím postupom 2.1. Taktiež to dáva novým užívateľom motiváciu dosiahnuť cieľ, keďže vidia ako ďaleko postúpili v onboardingu.

**Zelený panel** bude obsahovať informácie týkajúce sa pracovného postupu v aplikácii, a tým bude sprevádzať užívateľa úlohami zadanými v onboardingu. Farebné odlíšenie týchto informácií by malo ubrať kognitívnu záťaž na užívateľa.

**Šedý panel** bude posledným panelom umiestneným na obrazovke, a bude obsahovať tip ohľadne používania funkcionality (napr. aké funkcionality dané menu ešte obsahuje alebo na čo je možné funkcionality ešte využiť) 2.4. Na zvýraznenie funkcionality som navrhol využitie zvýrazňovacieho rámčeka, ktorý zvýrazní tlačítko, ktoré bude ďalších krokov v pracovnom postupe pri vykonávaní úlohy využité. Opäť ide o snahu znížiť kognitívnu záťaž na užívateľa, ktorý by pri jeho absencii musel hľadať o ktoré tlačítko ide, aj keby vložím do textu jeho ikonku 2.2.1. Posledný prvok je tlačítko **SKIP**, ktoré slúži na preskočenie niektorých krokov onboardingu nakoľko funkcionality, ktorá je tam opísaná sa nenachádza na všetkých robotických zariadeniach a môžu sa pri jej skúšaní vyskytnúť problémy. Kombinácia týchto prvkov by mala byť schopná poskytnúť prevedenie užívateľa aplikáciou podobným spôsobom ako by ho previedol jeden z vývojárov. Priebežný návrh bol vytvorený pomocou Microsoft One-Note, kde po navrhnutí tohoto riešenia boli jednotlivé kroky rozpísané podrobnejšie. Tento návrh bol konzultovaný s vývojármi AREditoru Ing. Michalom Kapinusom a s mojím vedúcim bakalárskej práce Ing. Zdeněkom Maternom, Ph.D, ktorí zhodnotili návrh ako vhodný na implementáciu.

### 3.5.3 Priebeh walktrough.

Náplň jednotlivých krokov vo walktrough pozostáva z podobných úloh, aké sa nachádzali v mnou vykonanom experimente. Jednotlivé kroky vychádzajú z vytvoreného konceptu. Pri tvorení konceptu boli spísané myšlienky a hrubý obsah jednotlivých krokov. Finálne využitie týchto krokov onboardingu si vyžadovalo zovšeobecniť úlohy tak, aby mohli byť vykonané na čo najväčšom množstve robotických pracovísk. Nie je žiaduce aby užívateľ nemal možnosť absolvovať onboarding kvôli zložitým akciám, ktoré by jeho robotické zariadenie nepodporovalo. Do samotných krokov som sa snažil zahrnúť čo najväčšie množstvo zaujímavých funkcionalít, ktoré sa v AREditore nachádzajú. Snažil som sa ich zakomponovať do úlohy vytvorenia jednoduchého programu. Preto aj z pôvodného konceptu boli niektoré časti vynechané a nahradené využitím iných funkcionalít AREditoru na dosiahnutie rovnakého cieľa.



Obr. 3.10: Finálna úloha vytvorená účastníkmi experimentu, z ktorej som bral inšpiráciu na tvorbu obsahu onboardingových krokov.

V tabuľke 3.1 môžete vidieť finálnu verziu jednotlivých krokov. Obsah jednotlivých krokov je zhrnutý ako celok informácií, ktorý si má užívateľ z kroku odniesť a pri implementácii budú tieto informácie rozdelené do jednotlivých farebných okien ako je popísané vyššie.

Číslo kroku	Obsah kroku
Krok 1	Modrý panel vás zoznámi so zvýrazneným objektom. Zelený panel vás bude prevádzať jednotlivými krokmi onboardingu a šedý panel obsahuje tipy a triky, ktoré môžete neskôr v AREditore využiť.
Krok 2	Zvýrazneným tlačítkom vytvoríte novú scénu. Scény sú využívané na vytvorenie pracovného priestoru.
Krok 3	Pomenujte scénu, a následne stlačte tlačítko <b>Create</b> pre jej vytvorenie.
Krok 4	Výborne! Zameriavač je váš hlavný nástroj na výber objektov, s ktorými chcete pracovať. Pre postup stlačte tlačítko <b>Next</b> .
Krok 5	Tlačítkom <b>Add</b> je možné pridávanie objektov. Náš cieľ je pridať robota do scény.
Krok 6	Zvýraznené tlačítko dovoľuje pridávanie objektov, druhé tlačítko v menu slúži na vytváranie kolíznych objektov. Poďme pridať robota do scény.
Krok 7	Pravé menu obsahuje objekty v scéne alebo rozšírené možnosti funkcionalít. Teraz je možné pridať robota alebo zariadenie ktoré máme k dispozícii.
Krok 8	Vyplňte informácie o vybranom robotovi z predošlého kroku. A stlačte tlačítko <b>Create</b> .
Krok 9	Výborne! <b>Utility Menu</b> obsahuje možnosti ako sú transformácia <sup>8</sup> , mazanie a premenovávanie. Ďalej je potreba presunúť virtuálny objekt robota tak, aby poloha sedela s reálnym robotom.
Krok 10	Vyberte transformačný nástroj. Pre prácu s objektom je potreba zameriavať na daný virtuálny objekt.
Krok 11	Pri mierení na jednu zo zobrazených osí je možné s touto osou pracovať. Pri práci s transformačným nástrojom si môžete v spodnej časti pravého panelu vyberať, či chcete využiť nástroj na rotáciu alebo posun.
Krok 12	Pre správne umiestnenie objektov do scény je nutnosť pohybovať sa okolo objektov, s ktorými pracujete aby bolo umiestnenie správne.
Krok 13	Výborne. Teraz ste už pripravený vytvoriť nový objekt v scéne.
Krok 14	Kliknite na zvýraznené tlačítko <b>AddProject</b> .
Krok 15	Treba pomenovať váš prvý projekt. Úlohou bude vytvoriť jednoduchý pohyb robotického ramena. Programy slúžia na vytváranie akcií spojených do programov spojeniami.
Krok 16	Dobrá práca! Teraz ste už pripravený vytvoriť váš prvý program. Rozhliadnite sa okolo, uvidíte zelenú šípku <b>START</b> a červenú šípku <b>END</b> , ktoré značia začiatok a koniec programu.
Preskočiteľné kroky	
Krok 17	Pre prácu s robotom bude potrebné pripojenie AREditoru k robotickému ramenu. Túto časť onboardingu je možné preskočiť v prípade problému alebo nepodporovaného zariadenia.
Krok 18	Robot sa dá pohybovať buď rukou alebo pomocou tlačidiel v pravom paneli. Na vytvorenie jednoduchého programu potrebujeme akčný bod, ktorý predstavuje ukotvenie v priestore s orientáciou. Akcia potrebuje mať svoj akčný bod, ku ktorému sa viaže.

<sup>8</sup>Posúvanie alebo rotovanie objektu.

Číslo kroku	Text kroku
Krok 19	Teraz si vyskúšajte presunúť robota na dostupnú pozíciu v pracovisku. Stlačením tlačítka sa robotické rameno uvoľní a dá sa s ním pohybovať.
Krok 20	Pri presúvaní je potrebné držať tlačítko stlačené a popri držaní umiestniť hlavu robota na ľubovoľnú pozíciu. Nie každý robot túto funkciu podporuje, ale môže mať túto funkciu ako hardwarové tlačítko na robotovi.
Krok 21	Výborne! Teraz ste pripravený na pridanie akčného bodu, na miesto kam ste robota umiestnili. Stlačte tlačítko <b>Add</b> .
Krok 22	Nástroje v <b>AddMenu</b> obsahujú pridanie akčných bodov, akcií a spojení. Vytvorte akčný bod na predtým nastavenej pozícii. V scéne potom môžete vidieť ukazovateľ pozície a orientácie robota (fialová šípka).
Krok 23	Vyberte meno pre akčný bod, alebo môžete ponechať základné.
Krok 24	Výborne! Teraz môžete kliknúť na tlačítko <b>Add</b> a vytvoriť akčný bod bez orientácie a predom stanovenej polohy. Na vytvorenie jednoduchého programu potrebujeme akčný bod, ktorý predstavuje ukotvenie v priestore s orientáciou. Akcia potrebuje mať svoj akčný bod, ku ktorému sa viaže.
Krok 25	Kliknutím na tlačítko <b>Add global action point</b> vytvoríte prázdny akčný bod, ktorému musíte pridať orientáciu.
Krok 26	Vyberte meno pre akčný bod, alebo môžete ponechať aj základné.
Krok 27	Orientácia je potrebná pre akčný bod, aby robotické rameno vedelo odkiaľ má k danému objektu pristúpiť. Pridanie orientácie nájdete pod <b>Utility</b> tlačítkom.
Krok 28	Teraz vyberte zvýraznené <b>Aiming</b> tlačítko.
Krok 29	Na spodnej strane pravého panelu nájdete guľatú šípku reprezentujúcu orientácie. Tieto sa dajú pridávať aj pomocou robota.
Krok 30	Teraz si skúste pridať orientáciu manuálne.
Krok 31	Doporučujeme nastaviť novú orientáciu na $-90.0$ stupňov, aby orientácia smerovala zhora dole.
Krok 32	Dobrá práca! Teraz ideme tlačítkom <b>Add</b> vytvoriť novú akciu.
Krok 33	V tlačítku <b>Add</b> sa nachádza vytváranie akčných bodov, akcií a spojení.
Krok 34	Rozkliknutím pravého menu uvidíte akcie, ktoré jednotlivé pridané zariadenia dokážu vykonať.
Krok 35	Vyberte akciu <b>move</b> z otvoreného menu.
Krok 36	Pomenujte vašu prvú akciu. Najlepšie <b>first_move</b> .
Krok 37	Ďalším krokom bude skopírovanie vytvoreného akčného bodu, aby boli akčné body dva a dalo sa medzi nimi pohybovať.
Krok 38	Je potreba zamieriť na akčný bod, ktorý chcete skopírovať. V prípade, že objekt v pravom paneli svieti na červeno je uzamknutý výber. Kliknutím uzamknutie zrušíte.
Krok 39	Vytvorte meno skopírovaného akčného bodu. Kopírovanie akčného bodu skopíruje aj jeho orientáciu a akciu.
Krok 40	Stlačte <b>Utility</b> , ktoré obsahuje nástroje všetkých úprav a zmien. Skopírovaný objekt treba posunúť na iné miesto ako originálny.
Krok 41	Vyberte <b>Transformation tool</b> , ktorá dovoľuje posun a rotáciu.
Krok 42	Zameranie osi, s ktorou chcete pracovať dovoľí jej posun pomocou rotujúceho bežca v pravom paneli. Taktiež môže byť použité ručné posúvanie v pravom paneli na hrubé umiestnenie. Posuňte objekt mimo pôvodného.

Číslo kroku	Text kroku
Krok 43	Výborne, teraz ste pripravený na pospájanie akcií do programu od <b>START</b> po <b>END</b> .
Krok 44	Zamierte zameriavač na <b>START</b> a potom stlačte zvýraznené tlačítko. Spojenia sa používajú na špecifikáciu poradia výkonu akcií. Pri potiahnutí spojenia od prvej akcie k druhej.
Krok 45	Môžete vidieť začaté spojenie z akcie <b>START</b> . Dotiahnite toto spojenie k akcii <b>first move</b> a kliknite na ňu v pravom paneli. Dotiahnutie nie je nutné pokiaľ prvú akciu v ľavom paneli vidíte.
Krok 46	Teraz musíte proces zopakovať a vytvoriť spojenie z <b>first move</b> do <b>second move</b> . Zamierte zameriavačom na <b>first move</b> a stlačte zvýraznené tlačítko. Potom ťuknutím na akciu <b>second move</b> v pravom paneli dokončíte spojenie.
Krok 47	Na koniec zamierte na akciu <b>second move</b> a v pravom paneli ťuknite na akciu <b>END</b> .
Krok 48	Dobrá práca, teraz sme pripravení uložiť program a následne ho spustiť. Projekt musí byť uložený pred jeho spustením.
Krok 49	Stlačte vyznačené tlačítko <b>Save</b> .
Krok 50	Stlačte <b>Home</b> .
Krok 51	Gratulujem! Teraz si môžete vychutnať vašu odvedenú prácu. Ak budete mať problém s nájdením funkcionality vždy môžete podržať prst na tlačítku a zobrazí sa vám nástrojový tip.

Tabuľka 3.1: Popis jednotlivých krokov navrhnutých pre onboardingový proces AREditoru.

Texty v podobnej podobe boli využité v jednotlivých krokoch. Tieto texty boli overené pilotným testovaním návrhu kde užívateľ prechádzal návrhom konceptu vytvoreným v Microsoft OneNote a jednotlivé kroky sa snažil plniť. Toto testovanie, aj keď len na jednom užívateľovi, mi pomohlo lepšie formulovať dané kroky. Zložité časti aplikácie, ktoré užívateľovi neboli jasné som popísal dôkladnejšie. Návrh textu po implementácii prešiel ešte niekoľkými zmenami riadenými doporučením vedúceho práce Ing. Zdeněkom Maternom, Ph.D. Ten doporučil niektoré texty skrátiť a rozložiť zložitejšie úlohy na viacero krokov, nakoľko v niektorých krokoch bolo priveľa textu.

### 3.5.4 Zaver

Pri vytváraní krokov som sa rozhodol využiť čo najpriamejšie vysvetlenie daných funkcionalít, aby užívateľ nebol zbytočne zatažený snahou pochopiť napísanému textu. Taktiež som využíval odmeňovanie užívateľa pochvalou po ukončení niektorej z podúloh. Kroky sú navrhnuté tak, aby si užívatelia vyskúšali čo najviac funkcionalít, ktoré im AREditor ponúka ale zároveň s dôrazom na jednoduchosť jednotlivých úloh. Text jednotlivých krokov je napísaný tak, aby mal užívateľ pocit neformálnej komunikácie s tým, že je kladený dôraz na to aby jednotlivé kroky boli zamerané na užívateľa, a nie len imperatívne prikazované kroky, ktoré musia byť splnené, aj keď sa bez takých krokov vysvetľovanie aplikácie nezaobíde. V tom prípade sú navrhnuté kroky tak, aby užívateľ vedel čo je jeho cieľom, ku ktorému mu onboarding pomáha. Pre potreby textu sú niektoré kroky skrátené o vysvetlenie jednotlivých funkcionalít, nakoľko presný popis jednotlivých funkcionalít AREditoru sa



nachádza v sekcii 3.1. Kroky sú v aplikácii implementované po anglicky nakoľko AREditor zatiaľ neponúka ponuku pre český a slovenský jazyk. S tým sú aj spojené menšie odlišnosti v texte krokov popísaných v tejto sekcii od tých, ktoré sú v aplikácii implementované.

## Kapitola 4

# Technické riešenie

Táto kapitola je venovaná implementácii pripravených onboardingových krokov do aplikácie 3.5.3. Technológiám, komponentom a postupom v engine **Unity**<sup>1</sup>, ktoré boli pri implementácii využité. Podrobnejšie v nej bude popísaná implementácia navrhnutého užívateľského rozhrania a zdôvodnenie rozhodnutí pri implementácii. Na konci bude čitateľ zoznámený so spôsobom riadenia onboardingu.

### 4.1 Implementácia návrhu UI

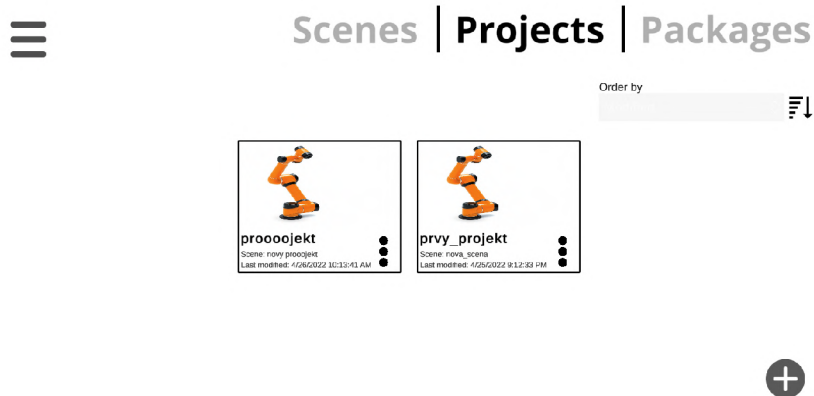
Samotná implementácia onboardingu zahŕňala vytvorenie dialógového okna slúžiaceho na oboznámenie užívateľa s použitím nástrojových tipov. Toto dialógové okno je možné otvoriť z hlavného menu aplikácie, do ktorého sa užívateľ dostane po pripojení na server. Dialógové okno privíta užívateľa do aplikácie, dovoľí mu vyskúšať si zobrazenie nástrojového tipu a následne mu dá možnosť absolvovať onboarding. Dialógové okno po kliknutí mimo neho zmizne a užívateľ môže pokračovať v práci.

#### 4.1.1 Dialógové okno

Každé užívateľské rozhranie, ktoré umožňuje užívateľom využiť implementovaný systém musí mať možnosť vstupu do tohoto systému. Či už ide o menu v hrách alebo o jednoduché prihlasovacie dialógové okno. Tak isto aj vstup do systému nápovedí musí byť pre užívateľa na prvý pohľad zrejmý. Preto nie je výhodné skrývať ho do rôznych menu, ako to je napríklad pri niektorých Editoroch (Visual Studio, Unity Editor). Ťažko prístupné menu, ktoré má pomáhať užívateľovi stráca zmysel ak si užívateľ musí vyhľadať kde sa nachádza na internete. Preto som sa rozhodol umiestniť toto menu priamo do hlavnej obrazovky AREditoru, kde aj prvotný užívateľ hneď pochopí, že tu nájde pomoc v použití. Nakoľko väčšina aplikácií, ktoré využívajú formu nápovedí, onboardingu alebo pomoci užívateľom označujú tlačítka umožňujúce vstup do rozhrania nápisom **Help** alebo ikonou **?** rozhodol som sa pre familiárnosť využiť ikonu **?** nakoľko jej umiestnenie som už mal z časti vyriešené. Bolo treba ho umiestniť tak aby zapadalo do návrhu úvodnej obrazovky AREditoru.

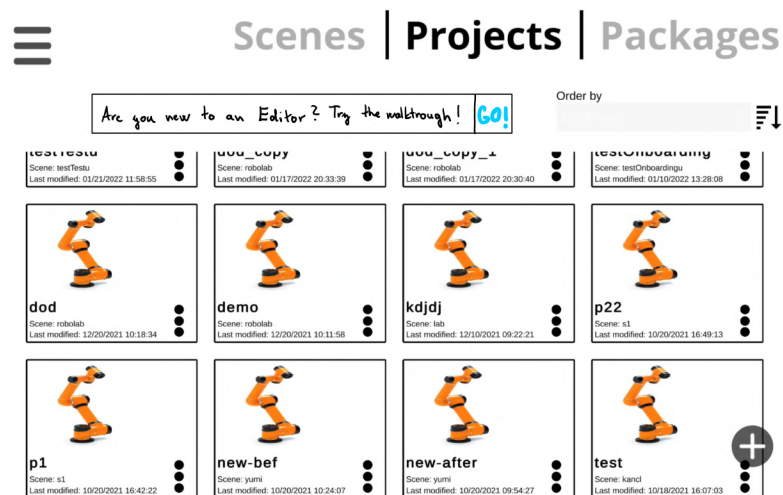
---

<sup>1</sup>Multiplatformový herný engine voľne dostupný na: <https://unity.com/>



Obr. 4.1: Pôvodné menu bez možnosti vstupu do onboardingu. Nakoľko sú prvky špecificky rozložené nezostávalo mi veľa možností ako toto tlačítko do menu umiestniť.

V pôvodnom návrhu 3.5.1, ktorý vznikol zároveň s konceptom UI<sup>2</sup> som navrhol veľké tlačítko s textom, ktoré neprehliadne žiadny nový užívateľ (obrázok 4.2).

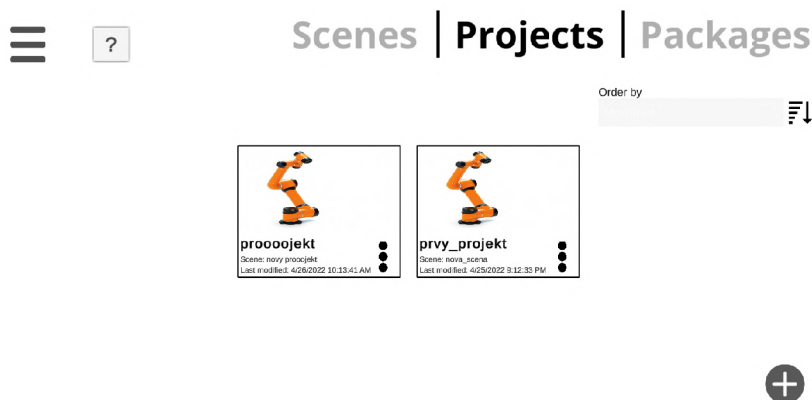


Obr. 4.2: Pôvodný návrh umiestnenia tlačítka. Využitie bolo miesto pod výberom scén a projektov. S popisom pýtajúcim sa, či je užívateľ v aplikácii nový, a ponúka mu využiť walkthrough.

Pri samotnej implementácii som sa rozhodol pre oveľa minimalistickejšie riešenie. A to hlavne z dôvodu, že väčšina užívateľov onboarding prejdú jeden alebo dvakrát, kým sa naučia pracovať s aplikáciou, a potom po zvyšok jej používania nebude využité. Preto nie je najvhodnejšie riešenie veľké tlačítko. Takéto by užívateľa mohlo vyrušovať pri práci a pri prepínaní scén a projektov by omylom mohol vstúpiť do onboardingu čo by ho vyrušilo pri práci ešte viac. Ďalšia možnosť bola ľavý dolný roh aplikácie obr. 4.1 umiestnenie tlačítka na pridanie scény +. Táto možnosť mala rovnaký problém ako prvé umiestnenie. Používateľ nebude tak často používať onboarding ako výber jednotlivých scén a projektov. Preto

<sup>2</sup>User Interface - používateľské rozhranie

zostala posledná možnosť, a to priestor medzi hamburger menu a výberom scén, projektov a balíčkov (obrázok 4.3). Toto umiestnenie vyhovuje svojou polohou nakoľko je umiestnené pri možnostiach aplikácie ale mimo bežného pracovného postupu užívateľa.



Obr. 4.3: Finálne umiestnenie tlačítka na vyvolanie dialógového okna pre vstup do onboardingu v menu AREditora.

Zvolil som farebnú schému a tvar tlačítka tak aby pasoval vedľajšiemu tlačítku. Toto tlačítko po stlačení vyvolá dialógové okno, cez ktoré je možné zapnúť onboarding. Pri implementácii bol pridaný do okna priehľadný herný objekt<sup>3</sup> cez celú šírku obrazovky, ktorý je ukotvený na vrchol obrazovky s centrovaním na stred. Ten bol použitý ako kontajner pre menší herný objekt, ktorý bol zase ukotvený na ľavý okraj obrazovky a tlačítko bolo vložené a ukotvené na druhý koniec tohoto menšieho herného objektu, kde bolo ukotvené a centrované na stred. Ďalej je ponúknuté užívateľovi začať onboarding do aplikácie.

Diagram herných objektov z ktorých sa skladá úvodné tlačítko:

**Content** - Celá plocha obrazovky, v ktorej sú umiestnené zvyšné prvky hlavného menu.

- └─ **HelpButton** - Kontajner obsahujúci ďalšie rozdelenia obrazovky.
  - └─ **UpperPanel** - Panel ukotvený na vrchol obrazovky.
    - └─ **LeftCenteredPanel** - Panel čiastočnej šírky ukotvený v ľavo.
      - └─ **OnboardingButton** - Tlačítko s ikonou ukotvené na pravú stranu rodičovského objektu.

Takéto umiestnenie zabezpečí, že pri zmenách rozlíšenia obrazovky tlačítko bude stále na správnom mieste zarovnané vedľa hamburger menu. Celý tento herný objekt bol pridaný do objektu, ktorý je zobrazený v scéne hlavného menu.

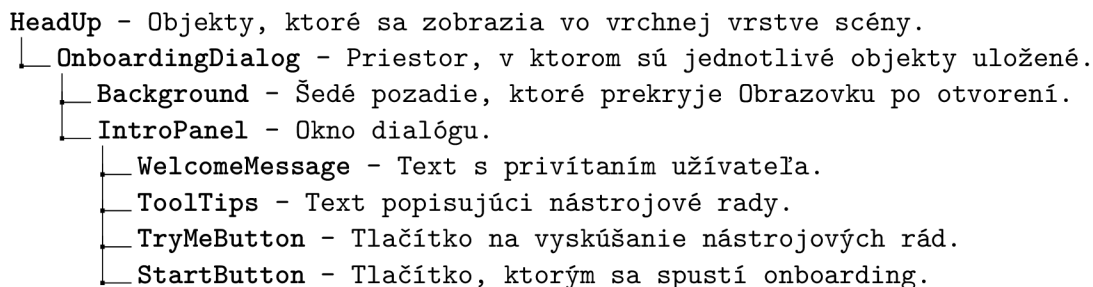
<sup>3</sup>**GameObject** sú fundamentálne objekty, ktoré reprezentujú postavy scény a ďalšie prvky využívané v Unity. Bližšie popísané v manuáloch Unity dostupných <https://docs.unity3d.com/560/Documentation/Manual/class-GameObject.html>.



Obr. 4.4: Dialógové okno pre vstup do onboardingu. Obsahujúce vstup do onboardingu ako aj zoznámenie užívateľa s použitím nástrojových tipov.

Na otváranie **dialógového okna** (obrázok 4.4) som využil animáciu, ktorá postupne okno zviditeľnila pri jeho otvorení a zneviditeľnila pri jeho uzatvorení. Táto animácia bola vytvorená pomocou komponenty `CanvasGroup`<sup>4</sup>, ktorá dovoľuje ovládanie viditeľnosti skupiny prvkov, aj ďalšie možnosti, ako zošednutie herného objektu, s ktorým sa nedá interagovať. Viditeľnosť objektu je ovládaná parametrom **alfa**, ktorý je pre potrebu animácie zvyšovaný postupne v časovom intervale. Obsah dialógového okna oproti pôvodnému návrhu, ktorý obsahoval len uvítaciu správu a zoznámenie užívateľa s ovládaním nástrojových rád, bol po doporučení vedúceho bakalárskej práce rozšírený aj o tlačítko, na ktorom si užívateľ môže túto funkcionality skúsiť. Čo môže mať za následok lepšie zapamätanie a častejšie použitie funkcionality nástrojových rád v používaní aplikácie. Taktiež po jeho doporučení som pridal zošednuté pozadie obrazovky aby nebolo možné klikať na objekty v scéne, kým je dialógové okno otvorené. Okno sa dá vypnúť kliknutím na zošednutú časť nakoľko som hernému objektu pridal komponentu `Button`<sup>5</sup>, ktorá po kliknutí využila vyššie spomínanú funkcionality. Toto riešenie býva využité v mobilných a webových aplikáciách celkom bežne. Keď ide o aplikáciu využívanú na dotykovom zariadení, nemusí byť užívateľovi pohodlné snažiť sa trafiť prstom na krížik ktorý by toto okno zrušil.

Priložený diagram popisuje štruktúru herných objektov, s ktorých sa toto menu skladá:

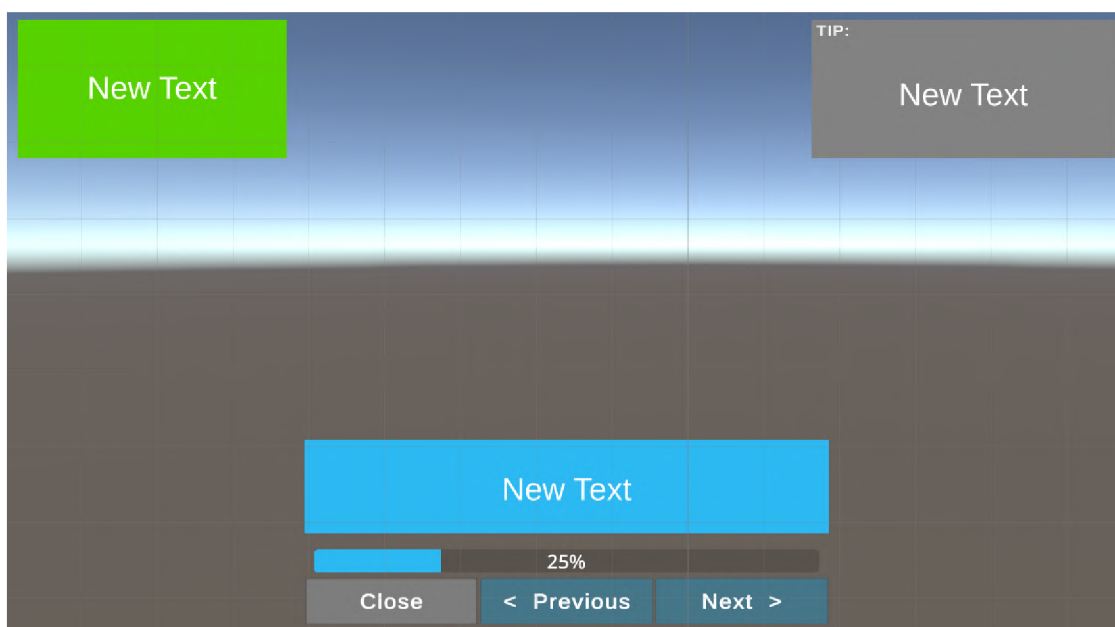


<sup>4</sup>Bližší popis komponenty `CanvasGroup` je dostupný na: <https://docs.unity3d.com/Packages/com.unity.ugui@1.0/manual/CanvasGroup.html>.

<sup>5</sup>Komponenta dovoľuje pridať cez Unity inspector metódy, ktoré sa spustia po kliknutí. Bližší popis komponenty `Button` je dostupný na: <https://docs.unity3d.com/2018.2/Documentation/ScriptReference/UI.Button.html>.

### 4.1.2 Onbardingová vrstva

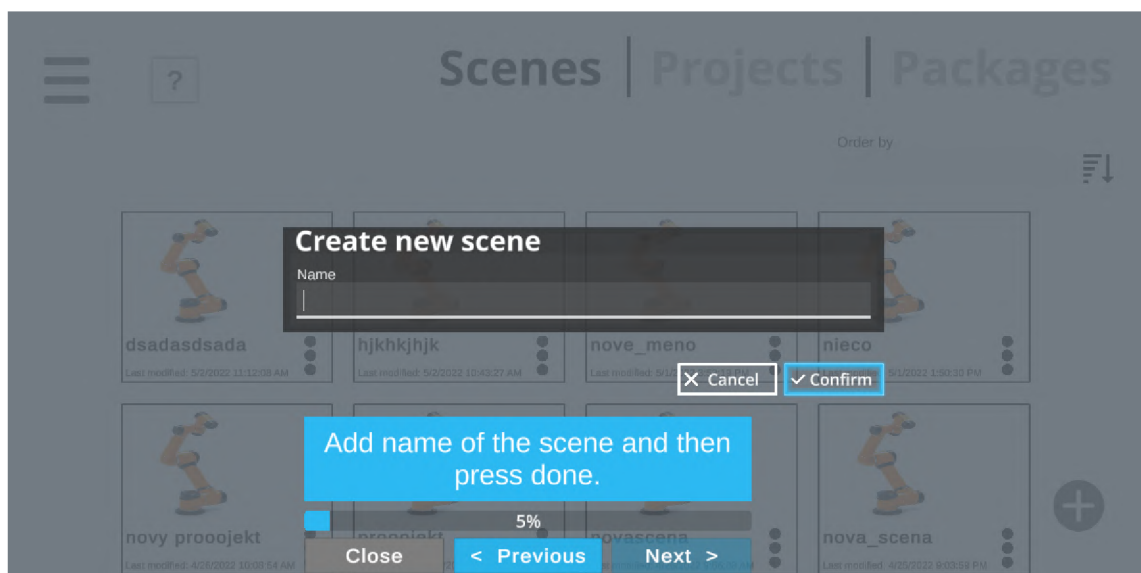
Vrstva obsahujúca jednotlivé kroky je najdôležitejšou súčasťou onboardingu. Je potrebné teda rozmiestniť informácie potrebné k jednotlivým krokom tak, aby užívateľ nemal problém s rýchlym pochopením náplne kroku. Tiež je potrebné mať možnosť vypnúť tento onboarding pokiaľ ním užívateľ prechádza len kvôli jednej funkcionalite, ktorej neporozumel prvýkrát. Užívateľ musí zostať zaujatý daniím, inak bude postupne strácať pozornosť. Taktiež je podstatné informovať užívateľa o postupe, ktorý urobil na to, aby mal prehľad o svojom postupe k cieľu, ktorý sa snaží dosiahnuť. Toto zabezpečuje implementácia vrstvy, ktorá pridá na obrazovku popri používaní vrstvy editoru jednotlivé nápovedy a tlačítka na jeho ovládanie. Tieto informácie sú umiestnené vo farebne odlíšených paneloch nachádzajúcich sa v prázdnych miestach obrazovky editoru. Panely budú plnené informáciami podľa ich typu. Nie každý krok obsahuje všetky panely. V niektorých krokoch ako vyberanie mena **akčného bodu** plne postačí jeden panel, ktorý informuje užívateľa o potrebe zadať názov. Jednotlivé panely sú ovládané riadiacim skriptom, ktorý ich zobrazí alebo schová podľa potreby s využitím komponenty `CanvasGroup` spomínanej v predošlej sekcii.



Obr. 4.5: Rozloženie jednotlivých farebne odlíšených panelov v onboardingovej vrstvy. S tlačítkami na jeho ovládanie a ukazovateľom postupu v onboarding.

Modrý panel bude obsahovať informácie o nástroji a funkcionalite. Zelený bude užívateľa sprevádzať onboardingom a šedý bude obsahovať tipy k jednotlivým funkcionalitám 3.5.2. Farebné odlíšenie bolo zvolené hlavne kvôli asociácii jednotlivých farieb s typom informácie ktorú nesú. Nakoľko AREditor vo vrstve editoru obsahuje farebnú schému zloženú z šedej, čiernej, bielej a červenej zvolil som farby ktoré budú vynikať zo samotného používateľského rozhrania editoru (obrázok 4.5). Nakoľko cieľom je, aby onboarding pomohol užívateľovi vstrebať čo najviac informácií je treba tieto informácie rozdeliť do skupín, ktoré sa ľahšie zapamätajú 2.2.1. Tlačítka **Previous**, **Next** a **Close** slúžia na ovládanie onboarding a k nim sú naviazané metódy z riadiacich skriptov, ktoré sa starajú o postupe v onboarding. Tlačítka taktiež využívajú komponentu `CanvasGroup`, ktorú využívam na znemožnenie in-

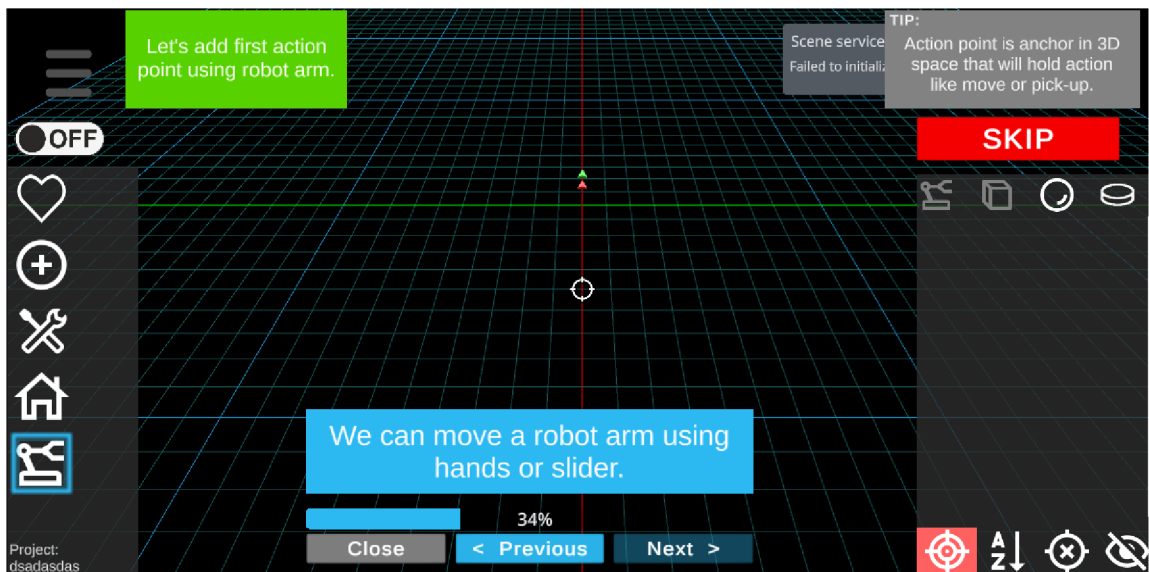
terakcie s jednotlivými tlačítkami pokiaľ to nie je vhodné. V prípade prvého kroku nie je možné vrátiť sa o jeden naspäť. Dôležitou súčasťou rozhrania onboardingu je aj zvýrazňovací rámik (obrázok 4.6), ktorý jednotlivé tlačítka a panely vo vrstve editoru zvýrazňuje. Tieto zvýraznené tlačítka a panely sú popísané v modrom paneli onboardingu. Pre zvýšenie použiteľnosti onboardingu som implementoval funkcionality, ktorá dovoľuje pridať metódu posúvania v krokoch onboardingu na zvýraznené tlačítka. Jedine v prípade, že ide o objekt ako napr. pravý panel obsahujúci posuvné kolečko, ktorým sa ovláda posun, na ktorý nie je možné takú metódu pridať, zostane aktívne tlačítka **Next**.



Obr. 4.6: Po vyplnení mena novej scény a stlačení zvýrazneného tlačítka bude onboarding pokračovať ďalším krokom bez potreby ďalšieho klikania.

Toto umožní užívateľom lepšie sústredenie na funkcionality a tlačítka, ktoré používajú a nebudú vyrušovaný neustálou potrebou prepínať jednotlivé kroky. Pretože použiteľnosť rozhrania sa odvíja aj od množstva krokov, ktoré musí užívateľ spraviť na dosiahnutie svojho cieľa, s ktorým používanie započal[8]. Pri plnení cieľu ich podporuje aj ukazovateľ postupu v onboardingu, ktorý je implementovaný pomocou rozšírenia **Modern UI Pack**<sup>6</sup>. Komponenta sa volá **ProgressBar**, kde jednoduchým nastavením parametra **CurrentPercent** je možné vizualizovať percentuálnu hodnotu postupu. Táto hodnota sa vypočítava v každom kroku metódou implementovanou v riadiacom skripte onboardingovej vrstvy. Posledným herným objektom, ktorý sa využíva v onboardingovej vrstve je tlačítka **Skip** (obrázok 4.7), ktoré je prístupné len z časti, v ktorej je možné, že robot užívateľa nebude podporovať používané funkcionality (nepodari sa pripojiť k robotickému ramenu). Toto tlačítka je primárne nastavené ako neaktívne a jeho aktivita sa nastaví len v krokoch, v ktorých existuje možnosť, že užívateľ by rád niektoré kroky preskočil.

<sup>6</sup>Rozšírenie do obsahujúce rôzne pred-implementované objekty ako ukazatele postupu, tlačítka s nápo- vedou. Dostupné na <https://assetstore.unity.com/packages/tools/gui/modern-ui-pack-201717>



Obr. 4.7: Tlačítka na preskočenie využité v prípade, kedy užívateľ má nekompatibilného robota s funkcionalitou alebo sa vyskytol problém, ktorý zabránil dokončeniu úlohy, ktorá pre celkový výsledok nie je nevyhnutná.

Priložený diagram popisuje štruktúru herných objektov, z ktorých sa skladá onboardingová vrstva:

- HeadUp - Objekty, ktoré sa zobrazia vo vrchnej vrstve scény.
  - └ WalkthroughOverlay - Kontajner obsahujúci nasledujúce herné objekty.
    - └ PrimaryPanel - Primárny panel s informáciami o používanej funkcionalite.
      - └ PrimaryTaskText - Text primárneho panelu.
      - └ NextStep - Tlačítka umožňujúce posúvanie v onboardingu vpred.
      - └ BackStep - Tlačítka umožňujúce posúvanie v onboardingu vzad.
      - └ Close - Tlačítka umožňujúce zatvorenie onboardingovej vrstvy.
      - └ ProgressBar - Komponenta umožňujúca zobrazenie postupu užívateľa v onboardingu.
    - └ SecondaryPanel - Sekundárny panel s informáciami o ďalšom postupe.
      - └ SecondTaskText - Text sekundárneho panelu.
    - └ TipPanel - Panel obsahujúci rozšírené rady k použitým funkcionalitám.
      - └ TipPanelText - Text panelu s radami.
    - └ HighlightingFrame - Rámik zvýrazňujúci jednotlivé tlačítka a panely.
    - └ SkipButton - Tlačítka umožňujúce preskočenie niektorých krokov.

## 4.2 Riadenie onboardingu

V tejto sekcii sú v krátkosti popísané základné riadiace mechanizmy onboardingu. Čitateľ bude oboznámený s princípom ukladania krokov, ako aj s riadiacim mechanizmom, ktorý sa stará o funkčnosť UI.



### 4.2.1 Onboardingový krok

Pri implementácii walktrough sa bolo treba vysporiadať s viacerými problémami. Jedným z prvých problémov, s ktorým sa implementácia musela vysporiadať spočíval v spôsobe zapísania krokov. Jednotlivé kroky museli obsahovať náplň jedného až troch textových polí, pravdivostnú hodnotu, ktorá slúži na označenie preskočiteľných krokov, a s tým spojenú hodnotu, na ktorú v prípade preskočenia nastavíme aktuálny krok. A ako posledné bolo v kroku potrebné mať uložené tlačítko, ktoré bude pri danom kroku zvýraznené. Na tento problém som sa rozhodol využiť triedu, ktorej inštancie budú tvoriť kroky onboardingu. Samotná trieda implementovaná v jazyku C# teda vyzerala takto:

```
public class WalktroughStep
{
    public int Order { get; set; }
    public int Progress { get; set; }
    public bool Skippable { get; set; }
    public string Tip { get; set; }
    public string PrimaryText { get; set; }
    public string SecondaryText { get; set; }
    public GameObject HighlitedButton { get; set; }
}
```

- `Order` reprezentuje poradie kroku v onboardingovom procese, ale slúži ako pomôcka pri písaní jednotlivých krokov, nakoľko bez očíslovania by boli úpravy krokov veľmi mäťúce.
- `Progress` uchováva číslo poradia kroku, na ktorý sa v prípade potreby má preskočiť.
- `Skippable` je pravdivostná hodnota, ktorá dáva možnosť preskočenia časti onboardingu.  
Ďalej nasledujú tri atribúty obsahujúce samotný text onboardingu:
- `Tip` obsahuje text k tipom v jednotlivých krokoch.
- `PrimaryText` ďalej obsahuje všetky potrebné informácie o funkcionalite, ktorá je v kroku použitá.
- `SecondaryText` zase obsahuje text, ktorý sprevádza užívateľa onboardingom a úlohami v ňom.
- `HighlitedButton` obsahuje herné objekty, ktoré bude potrebné v kroku zvýrazniť a taktiež bude využité pri posúvaní jednotlivými krokmi, kedy využijem pridanie funkcie `StepOver()` kompatibilným objektom, ktorá po stlačení zvýrazneného objektu posunie onboarding o krok ďalej.

`HighlitedButton` je atribút triedy implementovanou triedou `MonoBehaviour`<sup>7</sup> z ktorej dedí každý skript v **Unity**.

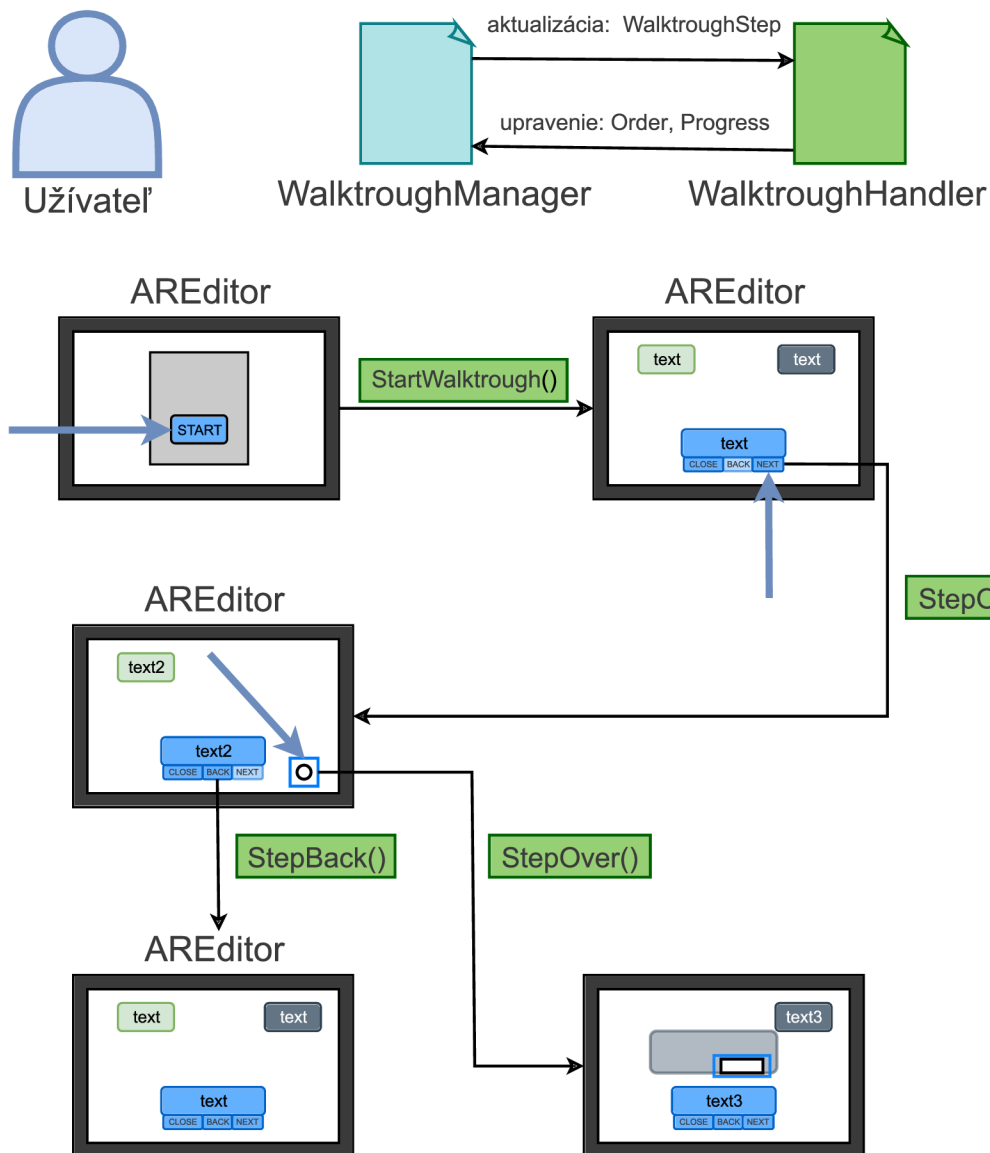
<sup>7</sup>Manuál k `MonoBehaviour` dostupný <https://docs.unity3d.com/ScriptReference/MonoBehaviour.html>.



Obr. 4.8: Rozloženie panelov na obrazovke.

#### 4.2.2 Riadiaci skript.

Trieda `WalkthroughHandler` je skript naviazaný na herný objekt `WalkthroughOverlay`, ktorý je pridaný v hernom objekte `HeadUp` a dovoľuje zobrazenie objektov vo vrchnej vrstve scény. Tento skript obsahuje všetky metódy potrebné na ovládanie onboardingu a funkcionality s tým spojené. Zmena krokov dopredu a dozadu prebieha pomocou metód `StepOver()` a `StepBack()` (obrázok 4.9). Tieto metódy v sebe obsahujú všetky ďalšie metódy, ktoré sa starajú o vymieňanie textov v paneloch, presúvanie a aktiváciu zvýrazňovacieho rámčeka, pridávanie a odoberanie metódy `StepOver()` zo zvýraznených herných objektov. Pri zmene krokov sa vykonáva kontrola, ktoré prvky majú byť viditeľné, a ktoré nie. Taktiež či jednotlivé tlačítka majú byť stlačiteľné alebo zošednuté. V prípade, že na zvýraznený herný objekt je možné pridať metódu `StepOver()` zošedne tlačítko **Next** a zmena kroku nastane po stlačení zvýrazneného objektu. V prípadoch kedy na zvýraznené objekty nie je možné pridať metódu spustenú po kliknutí, bude využitie tlačítka **Next**.



Obr. 4.9: Diagram ilustrujúci využitie metód zo skriptu WalktroughHandler na prechádzanie onboardingovým procesom.

Walktrough sa zapne metódou `StartWalktrough()`, v ktorej sa nastaví vrstva obsahujúca panely na aktívnu. Panely pridané vo `WalktroughOverlay` sú neustále umiestnené v scéne s nastavenou priehľadnosťou. Na ich zobrazenie je použitý komponent `CanvasGroup`, ktorý dovoľuje zviditeľňovať alebo zneviditeľňovať herné objekty použitím skriptu.

# Kapitola 5

## Testovanie

Cieľom experimentu zameraného na implementovaný onboardingový proces bude poskytnúť spätnú väzbu k priebehu onboardingového procesu, ako aj nájsť nedostatky, ktoré budú vylepšené. Tak isto pomôže overiť splnenie cieľu zoznámenia užívateľa s prostredím aplikácie. A pomôže odhaliť aj riešenia, ktoré budú do onboardingového procesu doplnené alebo zostanú ako návrh na rozšírenie. Na konci kapitoly bude zhodnotenie celkového výsledku, ktorý dosiahlo mnou implementované riešenie onboardingového procesu a porovnanie schopností užívateľa splniť danú úlohu po absolvovaní onboardingového procesu verzus po absolvovaní vysvetlenia od vývojára.

### 5.1 Pilotné testovanie

Ide o prvý test, ktorý bol po implementácii vykonaný. Jeho hlavným cieľom je zistiť kde má súčasná implementácia nedostatky a ako by ich bolo možné riešiť. V tejto sekcii bude popísaný návrh užívateľského testu, priebeh testovania, a na záver budú vyhodnotené zistenia z testu a zhrnuté návrhy na zlepšenie nájdených nedostatkov.

#### 5.1.1 Návrh užívateľského testu

Testovanie bude prebiehať na účastníkoch, ktorí majú značné skúsenosti s používaním rôzneho softvéru. Boli vybraní 2 účastníci, kde jeden reprezentoval laického užívateľa, nakoľko neštuduje odbor, ktorý by sa zameriaval na informačné technológie, a druhý účastník reprezentoval viac realistického používateľa z oboru. Išlo o študenta 3. ročníka VUT FIT. Obidvaja účastníci nemali predošlú skúsenosť s prácou v rozšírenej realite. Plán testovania spočíval v štyroch častiach:

1. Zoznámenie s využitím aplikácie AREditor.
2. Dokončenie onboardingového procesu.
3. Splnenie zadanej úlohy.
4. Odpovedanie na otázky a diskusia.

Na získanie informácií o účinnosti procesu som zvolil sadu otázok, ktoré boli položené na konci onboardingového procesu. Znenie otázok:

- Ako by ste na stupnici od 1 po 10 zhodnotili vnímanú zložitnosť používania aplikácie na začiatku jej používania.

- Ako by ste na stupnici od 1 po 10 zhodnotili vnímanú zložitosť používania po absolvovaní onboardingového procesu?
- Boli by ste schopný objavovať s nadobudnutými znalosťami ďalšie funkcionality aplikácie?
- Zhodnotili by ste onboardingový proces ako účinný pri zoznámení so základmi použitia aplikácie ?
- Narazili ste na časti onboarding, ktoré vám prišli nejasné?

Ďalej bude nasledovať diskusia na tému: Ktoré časti onboardingu vám robili problém. Pôjde o rozvedenie poslednej otázky. Cieľom tejto diskusie bude zameranie na problematické časti onboarding, ktoré by sa dali zlepšiť. Prípadne si vypočúť návrhy účastníkov čo by im dalo väčší zmysel ako súčasné riešenie.

### 5.1.2 Pribeh testovania

Na začiatku boli užívatelia oboznámení s využitím aplikácie AREditor. Bolo nutné užívateľov uviesť do kontextu využitia AREditoru a toho čo idú robiť. Nakoľko budúci užívatelia, ktorí využijú onboardingový proces budú mať istú znalosť prostredia, v ktorom sa aplikácia využíva a motiváciu naučiť sa ju ovládať.

Užívateľ dostal za úlohu otvoriť aplikáciu AREditor a bez žiadneho ďalšieho vysvetlenia prejsť onboardingovým procesom. Po dokončení bude nasledovať rovnako ako v prvom experimente jednoduchá úloha, v ktorej bolo potrebné využiť rovnaké funkcionality ako boli využité v onboardingovom procese. Rozhodol som sa pre rovnakú formu experimentu nakoľko výsledok bude môcť byť porovnaný s výsledkami prvého experimentu.



Obr. 5.1: Prvý účastník prechádzajúci onboardingom.

**Prvý účastník** (obrázok 5.1) bol študent venujúci sa hudobnej tvorbe. Nakoľko ale s používaním mobilných zariadení ako tabletou mal dobrú skúsenosť, bol to vhodný účastník reprezentujúci laického užívateľa. Pri prvotnom spustení aplikácie chvíľu hľadal tlačítko, ktoré by mu ponúklo nápovede čo ma viedlo k záveru, že by možno bolo lepšie pre tlačítko spúšťajúce dialógové okno využiť namiesto ikony otáznika slovo „Help“. Pri vytváraní scény som pozoroval na účastníkovi mierne zmätenie pri úlohe napasovať virtuálny objekt na reálneho robota. Pokračovanie zvyšných krokov išlo plynule až po presúvanie akčného bodu, kde účastníkovi nebolo jasné na čo má pri posune zamieriť, nakoľko síce v inštrukciách bolo zamierť na akčný bod, účastník nemal asociovaný názov akčného bodu z virtuálnym objektom ktorý ho reprezentoval. Pridávanie akcie aj akčného bodu prebehlo bez problémov. Ďalší problém pre účastníka nastal pri spájaní jednotlivých akcií do programu pomocou spojení. Opäť konštatoval, že síce mu inštrukcie hovoria čo má robiť, on ale nevie čo tieto názvy reprezentujú v scéne. Nakoniec bol schopný podľa inštrukcii onboarding dokončiť a bol pripravený splniť zadanú úlohu. Pri prvej úlohe mal účastník chvíľu problém pri zorientovaní sa v aplikácii bez nápovedí. Nakoniec po mojej rade využiť nástrojové nápovedy na jednotlivých tlačítkach, bol schopný splniť úlohu s malým počtom otázok ohľadne náplne úlohy. Bolo potrebné zopakovať, že náplňou úlohy je vytvoriť dva akčné body s akciami a tie pospájať do programu.

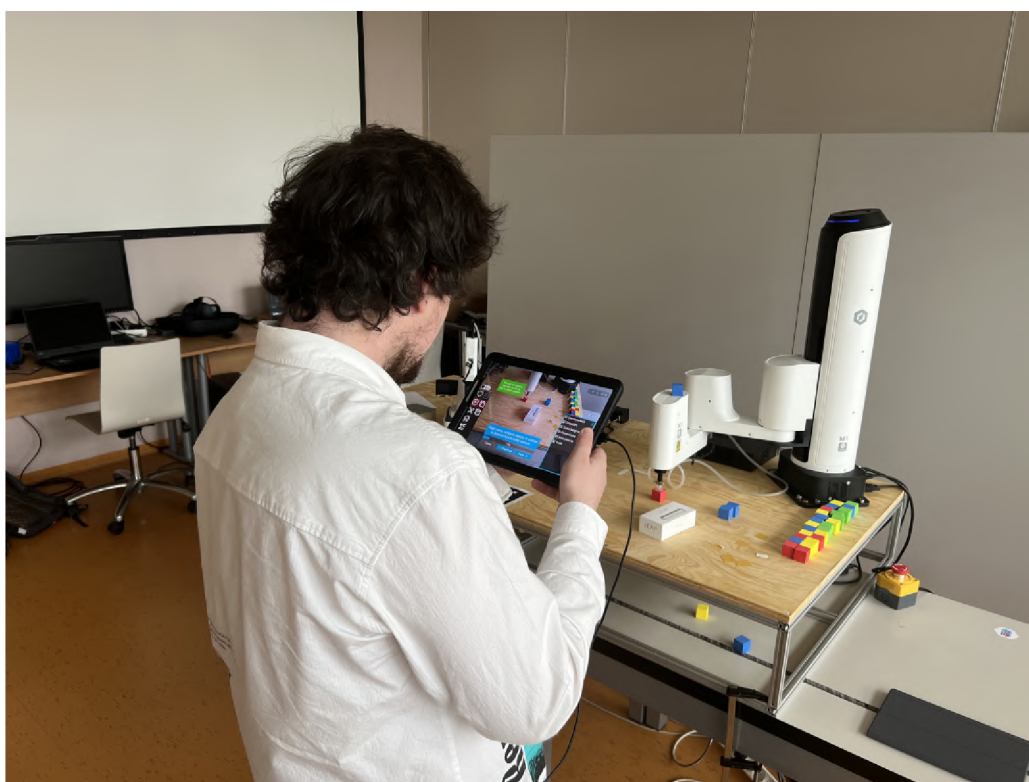
Na otázky odpovedal nasledovne:

- Na začiatku používania som sa nevedel moc orientovať, čiže ohodnotil by som zložitost na **8,5/10**.
- Po absolvovaní onboardingu som mal už isté porozumenie aplikácii, vedel som zhruba kde čo hľadať čiže **4,5/10**.
- Keďže som už mal istú predstavu kde čo hľadať s využitím nástrojových nápovedí by som dokázal objavovať aplikáciu ďalej.
- Myslím si, že mi to dalo základné oboznámenie s funkcionalitami a procesom vytvárania programu.
- Miestami som mal problém s použitou terminológiou ako akčný bod, akcia. Nevedel som čo pod tým názvom mám hľadať v scéne.

Pri diskusii opísal účastník mierny problém s umiestňovaním robota do scény. Prišla mu celkom jednoduchá úloha, dosť zložito popísaná. Ďalej spomenul, že v dvoch krokoch sa mu stalo, že bolo vysvetlené niečo v predošlom kroku a mal to využiť v ďalšom čo spôsobovalo, že sa musel vrátiť a prečítať nápovedu znova. Konkrétne šlo o prepínanie medzi rotáciou a posunom a spájanie programu. Taktiež zhodnotil že by možno privítal trochu hrubší zvýrazňovací rámček.

**Druhý účastník** (obrázok 5.2) bol študent VUT FIT, ktorý študuje tretí ročník. Rovnako ako prvý účastník testovania má dostatočnú skúsenosť v ovládaní mobilných zariadení. Bol vybraný ako účastník z dôvodu jeho študijného zamerania. Ako študent informačných technológií má skúsenosť so zložitejšími systémami, a tiež aj rozumie princípu skladania akcií za sebou, ktoré vytvoria jeden finálny program. Taktiež má skúsenosť s vývojom užívateľských rozhraní. Bol schopný mi podať jeho nápady na zlepšenie a pomôcť nájsť úskalía, ktoré pre mňa ako vývojára neboli zrejmé. Pri započatí práce v AREditore spočiatku

taktiež nevedel nájsť tlačítko, ktorým sa do onboardingu dostane. Po zoznámení s aplikáciou nasledovalo spustenie onboardingu. Pri prvých úlohách som na účastníkovi nesledoval žiaden problém ani zaváhanie. Mierne zaváhanie bolo vidno pri umiestňovaní obrysu digitálneho objektu robota na skutočné robotické rameno. Ďalej nasledovalo vytvorenie objektu akčného bodu bez väčších problémov. Taktiež som si všimol chybné využitie kopírovacej funkcie, kedy účastník skopíroval namiesto celého akčného bodu len akciu na ňom. Po mojej dodatočnej otázke skonštatoval, že nevedel na čo má v scéne mieriť aj keď inštrukcie hovorili o mierení na akčný bod. Pri spájaní jednotlivých krokov mal mierny problém s koncentrovanosťou informácií. Zhodnotil, že by asi bolo prívetivejšie rozložiť spájanie akcií do viacerých bodov. Následne po dokončení onboardingu bola účastníkovi zadaná úloha vytvoriť jednoduchý program, ktorý bude pohybovať ramenom. Tento program dokázal vytvoriť s využitím nástrojových nápovedí a jeho otázky boli smerované hlavne na názvy jednotlivých objektov v scéne.



Obr. 5.2: Druhý účastník prechádzajúci onboardingom.

Na otázky odpovedal nasledovne:

- Pri prvom vstupe do aplikácie mi nebolo moc jasné kde začať. Editor bol v prvých pár minútach práce dosť zložitý len s minimálnym popisom. zhodnotil by som úvodnú zložitosť ako **10/10**.
- Po absolvovaní onboardingu som mal oveľa lepšie porozumenie toho ako aplikácia funguje a ako sa s ňou pracuje. Zložitosť po absolvovaní onboardingu by som zhodnotil ako **3/10**.

- Onboarding mi pomohol zorientovať sa v základných funkcionalitách aplikácie a taktiež mi uľahčil objavovanie funkcionalít, ktoré aplikácia ponúka, pretože som vedel kde ich zhruba hľadať.
- Pri prvom použití to UI nie je úplne jasné, a po onboardingu mi nerobilo až taký problém nájsť jednotlivé funkcionality potrebné k dokončeniu úlohy, čiže to vnímam ako účinné.
- Terminológia aplikácie by mohla byť lepšie vysvetlená vzhľadom na to, že užívateľ to vidí po prvýkrát. V niektorých krokoch mi nebolo jasné čo spraviť len na základe toho, že som nevedel či „gulička“ v priestore je akcia alebo akčný bod.

Diskusia bola venovaná prvkom v onboardingu, ktoré účastníkovi prišli ako zbytočne zložité v jednotlivých krokoch, ako napr. vysvetlenie robota alebo spomínané vysvetľovanie spájania v málo krokoch. Zdôraznil, že si všimol ako boli niektoré kroky písané človekom, ktorý s AREditorom už dlhšie pracuje a nenapadlo ho, že nový užívateľ nebude mať možnosť spojiť si terminológiu s digitálnou reprezentáciou jednotlivých prvkov. Taktiež zhodnotil ako možné vylepšenie, sformulovať informácie v niektorých krokoch tak, aby boli viacej priamočiare. Pri diskusii o krokoch tiež dodal, že keď bolo v troch paneloch rozdelených veľa informácií, človek nevedel kde začať. A preto by to možno bolo lepšie spracované ako tri za sebou idúce kroky rozpísané v rozbalovacom okne, ktoré by boli rozbaliteľné z primárneho panelu. Takto by si mohol prečítať aj viac textu a po prečítaní panel zavrieť, nech ho neruší veľké množstvo textu v zornom poli a môže plne pracovať na úlohe. Tiež aj navrhol využiť v inštrukciách okrem názvov prvkov ako akčný, aj to ako ich v scéne nájsť (napr akčný bod - guľička).

## 5.2 Vyhodnotenie a navrhnuté zmeny

Priebeh testu ukázal, že proces onboardingu bol pri zoznamovaní s aplikáciou úspešný. Po absolvovaní onboardingu sa pri obidvoch účastníkoch výrazne znížila vnímaná zložitosť systému. Konštatovali dostatočnú orientáciu na to, aby ďalej dokázali objavovať aplikáciu sami.

Z testovania boli patrné problémy, ktoré by sa dali vylepšiť:

1. Účastníci si nevedeli asociovať kľúčové slová s prvkami v scéne.
2. Niektoré kroky obsahovali až moc informácií naraz.
3. Použitie funkcionality vysvetlenej v predchádzajúcom kroku nebolo veľmi úspešné.
4. Zložité vysvetlenie niektorých jednoduchých funkcionalít.

K týmto problémom som navrhol vylepšenia, ktoré som implementoval pred ďalším užívateľským testovaním.

Navrhnuté vylepšenia:

- Problematické kroky boli prepísané tak, aby jednotlivé inštrukcie boli jasné aj laikovi.
- Pridanie očíslovania jednotlivých panelov aby užívateľ vnímal súslednosť informácií.



- Pri využití funkcionality vo viacerých krokoch zostane jej popis pri každom kroku ako **TIP**.
- Zvýraznenie kľúčových slov v inštrukciách s popisom ich reprezentácie v scéne (akcia - žltá šípka, štart - zelená šípka).
- Možnosť uzavretia panelu **TIP**.

Po upravení textov jednotlivých úloh a implementovaní navrhnutých zmien som uskutočnil ďalší užívateľský experiment na overenie navrhnutých vylepšení.

### 5.3 Opätovný test

Opätovný test prebiehal na dvoch účastníkoch. Prvým bol môj vedúci bakalárskej práce Ing. Zdeněk Materna, Ph.D ako expert. Nakoľko ale má skúsenosť v práci so systémom neboli mu položené rovnaké otázky ani neplnil úlohu po prejdení onboardingom, ako ostatní účastníci. Išlo skorej o zhodnotenie úplne onboardingu a posúdenie prevedenia s cieľom získať nápady na možné vylepšenia. Ďalším účastníkom bol študent MUNI ECON, ktorý reprezentoval laického užívateľa. Pre tohoto účastníka bol priebeh testovania rovnaký ako v pilotnom testovaní.



Obr. 5.3: Testovanie vylepšeného onboardingového rozšírenia vývojárom samotného systému.

**Prvý účastník** (obrázok 5.3): Pri prechádzaní onboardingom sme sa zhodli na viacerých možných vylepšeniach, ktoré by rozšírenie mohlo v budúcnosti obsahovať. Jedným z navrhovaných vylepšení by bolo hlbšie prepojenie onboardingu s funkcionalitou aplikácie,

napr. zobrazovať nápovedy k jednotlivým objektom v scéne po tom, čo na ne užívateľ zameria zameriavač. Taktiež ako veľmi dobrý návrh na vylepšenie bolo spýtať sa na meno užívateľa na začiatku testovania, ktoré by sa potom počas onboardingu používalo v jednotlivých krokoch čo by dalo účastníkom pocit osobnejšieho prístupu. Taktiež aj možnosť zadať typ robota, ktorý bude použitý a priamo ho použiť v inštrukciách napr. pri výbere pridaného zariadenia alebo akcie, ktorú chceme do scény pridať. Celkovo ale zhodnotil, že náplň jednotlivých krokov je dobrá a obsahuje väčšinu základných funkcionalít, s ktorými by sa mal užívateľ zoznámiť.



Obr. 5.4: Druhý účastník opätovného testu prechádzajúci onboardingom.

**Druhý účastník** (obrázok 5.4) po implementácii navrhnutých úprav nemal problém vytvoriť scénu a umiestniť do nej robota. Dokonca umiestnenie robota bolo s veľmi dobrou presnosťou. Vytváranie akčných bodov a pridávanie orientácii sa zaobišlo tiež bez zaváhania. Prvé náznaky zaváhania nastali až pri kopírovaní pripraveného akčného bodu. Tam si účastník zle prečítal nápovedu a mieril na akciu miesto akčného bodu. Čo prebehlo po mojom navrhnutí vrátiť sa a skúsiť krok odznova bez problémov. Spájanie vytvorených akcií s upravenými popismi jednotlivých prvkov v scéne prebiehalo plynulejšie. Nakoľko užívateľ vedel čo znamená, ktorý digitálny bod v scéne bol schopný prejsť spájaním jednoduchšie ako užívateľa pred ním. Celkový prechod onboardingom bol bez otázok a taktiež rýchlejší ako laik pri pilotných testoch. Pri plnení zadanej úlohy potom užívateľ bez problému pridal akčné body a tým orientácie. Zasekol sa pri jedinej veci a to pri pridávaní akcie, nakoľko akcia má reprezentáciu „šípky“ v editore ale „puku“ v ovládacom paneli. Čo ale po mojom návrhu využiť nástrojové nápovedy prebehlo ďalej bez problémov. A bol schopný vytvoriť aj spojenia bez ďalších rád.

Odpovede na pripravené otázky.

- Na začiatku som moc nevedel čo mám robiť, alebo čo vôbec znamená scéna a projekt. Po prvom otvorení editoru by som začať používať aplikáciu asi nevedel, preto by som zhodnotil úvodnú zložitosť na použitie 9/10.
- Po absolvovaní onboardingom som mal celkom dobré pochopenie o čo v aplikácii ide, a ako toho dosiahnem. Ešte si nie som niektorými funkcionalitami istý, ale zhodnotil by som teraz zložitosť na použitie 3/10.

- Myslím si, že keby prejedem onboardingom ešte raz nemal by som problém vytvoriť aj zložitejší program nakoľko som pochopil ako princíp vytvárania programov funguje.
- Určite áno. Dokonca ma pri plnení úlohy bavilo objavovať čo sa ešte v aplikácii nachádza.
- Možno by som len vytkol dĺžku onboardingu, že človek je prevedený celým pracovným postupom v aplikácii naraz a preto som zabudol na niektoré funkcionality, ktoré tam boli spomenuté.

V diskusii ďalej dodal, že očíslovanie jednotlivých panelov, aj keď si ich nevšimol hneď, mu pomohlo práve pri zložitejších úlohách, ktoré vyžadovali pochopiť čo sa deje. Ďalej veľmi kladne zhodnotil zvýraznené kľúčové slová s popisom ich reprezentácie v scénach.

### 5.3.1 Výsledky opätovného testu

Z testovania bolo zrejmé, že samotný onboardingový proces výrazne znižoval vnímanú zložitost' systému účastníkom. Po úprave problematických krokov a chýb, ktoré boli objavené v pilotnom testovaní, užívateľ nemal skoro žiadny problém prejsť sám onboardingom a neskôr splniť zadanú úlohu. Pri diskusii taktiež bolo vidno, že už po prvom prechode onboardingom mal dobré pochopenie princípu vytvárania programov a znalost' terminológie spojenej s použitím AREditoru.

## 5.4 Zhrnuté výsledky

Cieľom mojej práce bolo nahradiť vývojára, ktorý vysvetlí použitie aplikácie novým užívateľom rozšírením, ktoré takého užívateľa uvedie do problematiky a zoznámí ho so základnými funkcionalitami aplikácie. A toto zoznámenie dá užívateľovi dostatočnú znalost' aplikácie, na ktorej bude môcť neskôr stavať ďalším používaním AREditoru. Nakoľko všetci prvotní účastníci, ktorí sa podieľali na testovaní boli schopní dostať sa z nulovej znalosti aplikácie až k samostatnému vytvoreniu jednoduchého programu, znamená to, že onboarding uspel v zoznamovaní užívateľa s AREditorom. Na základe toho pokladám tento proces zoznámenia užívateľa za účinný. Nakoľko v prvom užívateľskom experimente zhodnotili užívatelia podobnú vnímanú zložitost' po absolvovaní školenia s vývojárom, ako užívatelia, ktorí prešli onboardingom. Pokladám preto tento proces v zmysle zoznámenia so základnými funkcionalitami aplikácie ako porovnateľne účinný.

## 5.5 Návrh budúcich vylepšení

Ako pri každej práci v priebehu jej vytvárania a testovania vývojár príde na veľa riešení, ktoré by mohli prácu vylepšiť. Pri práci naberá vývojár stále viac informácií o problematike a jej možných riešeniach.

Pri samotnom testovaní bolo navrhnutých alebo spomenutých veľa riešení, ktoré by mohlo do budúcnosti onboardingové rozšírenie obsahovať. Možné rozšírenia:

- **Rozdeliť onboarding na časti.** Nakoľko momentálne je jediná možnosť pri prechode onboardingom absolvovať celý onboarding. Bolo by vhodné mať možnosť absolvovať onboarding samostatne pre skupiny funkcionalít v prípade, že užívateľ so zvyškom nemá žiadny problém. Toto rozšírenie by mohlo fungovať ako úrovne, ktorými užívateľ prejde pri úvodnom použití aplikácie alebo si ich bude môcť jednotlivito zopakovať v prípade potreby. Napríklad využitie robota na prácu s akčnými bodmi, tvorenie programov alebo tvorenie scén. Tieto jednotlivé úrovne by taktiež mohli mať postupnosť, po ktorej splnení by užívateľ bol zoznamovaný s väčšinou funkcionalít. Postupne by sa mu odomykali nové úrovne čo by učeniu pridalo aj herný aspekt.
- **Integrácia rád do virtuálneho priestoru.** Užívateľ by mohol dostávať jednotlivé nápovedy k herným objektom a akciám pri spájaní priamo vo vytvorených objektoch v scéne. Toto by mohlo výrazne zlepšiť účinnosť onboardingu pri spájaní akcií do programov, aj pri vytváraní akčných bodov s akciami.
- **Možnosť si jednotlivé panely rozkliknúť pre viac informácií.** Nakoľko užívatelia zvykli pokladať otázky týkajúce sa podrobnejších informácií o funkcionalite a jej využití. Mohla by byť implementovaná možnosť prístupu k rozšírenému popisu funkcionality z hlavného panelu obsahujúceho jednoduché vysvetlenie.
- **Personalizácia onboardingu.** Možnosť pridať do onboardingu informácie o sebe a o svojej organizácii či pracovisku, a tým by mohol byť onboarding prispôsobenejší danému užívateľovi. To by mohlo užívateľom dodať pocit osobnejšieho prístupu, lepšej používateľskej skúsenosti a zjednodušiť niektoré kroky v onboardingu.

Vybral som možné rozšírenia, ktoré podľa mojich zistení testovaním, mohli užívateľom zlepšiť používateľskú skúsenosť a spraviť onboarding ako nástroj na učenie účinnejším. Tieto vylepšenia vychádzali hlavne z analýzy spätnej väzby od užívateľov a návrhov, ktoré boli spomenuté mojím vedúcim bakalárskej práce ako aj účastníkmi z oboru.

## Kapitola 6

### Záver

Náplňou mojej práce bolo vytvoriť onboarding (systém nápoved) pre aplikáciu ARCOR2 AReDitor, ktorá prináša možnosť ne-programátorom programovať manipulačných robotov a ich komponent pomocou rozšírenej reality. Cieľom onboardingu v aplikácii AReDitor bolo zoznámiť užívateľa so základnými ovládacími prvkami aplikácie, kedy po tomto zoznámení bude užívateľ schopný zvyšok funkcionality objaviť aj sám len s pomocou nápoved už dostupných v aplikácii. Nakoľko túto úlohu museli doposiaľ plniť vývojári aplikácie. Pri návrhu takéhoto systému bolo potrebné naštudovať používané metódy onboardingu a vybrať najvhodnejšiu metódu pre AReDitor. To zahŕňalo štúdium v oblasti tvorby užívateľských rozhraní, užívateľského testovania, psychológie a v neposlednom rade dôkladné zoznámenie sa s aplikáciou AReDitor. Po naštudovaní problematiky nasledoval návrh samotného systému a jeho implementácia do aplikácie, po ktorej nasledovalo užívateľské testovanie a doladovanie nájdených chýb. Výsledkom tejto práce je onboardingový proces pre aplikáciu, ktorý je pripravený na nasadenie do produkčnej aplikácie. Testovaním sa mi podarilo overiť splnenie cieľa, kedy užívatelia, ktorí sami absolvovali onboarding boli schopní naprogramovať jednoduchú úlohu rovnako ako užívatelia, ktorí boli do systému zaučení vývojárom. Taktiež všetci účastníci testovania konštatovali, že zložitosť aplikácie, ktorú vnímali na začiatku jej používania po absolvovaní onboardingu klesla minimálne o polovicu, podobne ako to bolo pri osobnom zaučení. Pri diskusiách s účastníkmi na konci testovania bolo zrejmé že im onboarding dal istú formu pochopenia základných stavebných kameňov pre prácu s aplikáciou. V priebehu práce na onboardingovom procese som vykonal štyri experimenty, v ktorých sa celkovo zúčastnilo deväť užívateľov. Ako to už býva pri dokončovaní práce, človek príde na veľa vecí, ktoré by s nadobudnutými znalosťami spravil inak. Čím dlhšie som sa touto tématikou zaoberal, tým viac nápadov na zlepšenie mi napadalo. V budúcnosti by mohlo byť pre onboarding veľmi prínosné rozdelenie celého procesu zoznamovania aplikácie na menšie celky, ktoré užívatelia zoznámia s jednotlivými skupinami funkcionalít podrobnejšie. Takto by mal užívateľ možnosť rad za radom postupne prejsť všetky celky a vracat sa len k častiam, s ktorými mal problém. Pre užívatelia by bolo tiež prínosné lepšie prepojenie onboardingu so zvyškom systému ako aj forma personalizácie pomocou konfigurácie onboardingu pred jeho začatím. Práca mi umožnila rozšíriť si obzory pri návrhu užívateľských rozhraní a hlavne v oblasti testovania užívateľskej skúsenosti. Som veľmi rád že som sa v rámci tejto práce mohol podieľať na inovatívnom projekte, ktorý sa pozerá do budúcnosti, kde robotické zariadenia nebudú len vo fabrikách.

# Literatúra

- [1] Unity. 2005. Dostupné z: <https://unity.com/>.
- [2] BADDELEY, A. D. a HITCH, G. *Working Memory*. Academic Press, 1974. 47-89 s. Psychology of Learning and Motivation. Dostupné z: <https://www.sciencedirect.com/science/article/pii/S0079742108604521>.
- [3] BAMBUŠEK, D., MATERNA, Z., KAPINUS, M., BERAN, V. a SMRŽ, P. Handheld Augmented Reality: Overcoming Reachability Limitations by Enabling Temporal Switching to Virtual Reality. In: *Proceedings of the 2022 ACM/IEEE International Conference on Human-Robot Interaction*. IEEE Press, 2022, s. 698–702. HRI '22.
- [4] BARROUILLET, P., BERNARDIN, S., PORTRAT, S., VERGAUWE, E. a CAMOS, V. Time and Cognitive Load in Working Memory. *Journal of experimental psychology. Learning, memory, and cognition*. Jún 2007, zv. 33, s. 570–85. DOI: 10.1037/0278-7393.33.3.570.
- [5] BUDIU, R. Working Memory and External Memory. 2018. Dostupné z: <https://www.nngroup.com/articles/working-memory-external-memory/>.
- [6] JOYCE, A. Mobile-App Onboarding: An Analysis of Components and Techniques. 2019. Dostupné z: <https://www.nngroup.com/articles/mobile-app-onboarding/>.
- [7] KARPICKE, J. D. a BLUNT, J. R. Retrieval Practice Produces More Learning than Elaborative Studying with Concept Mapping. *Science*. 2011, zv. 331, č. 6018, s. 772–775. DOI: 10.1126/science.1199327. Dostupné z: <https://www.science.org/doi/abs/10.1126/science.1199327>.
- [8] KRUG, S. *Don't Make Me Think, Revisited: A Common Sense Approach to Web Usability*. 3rd. USA: New Riders Publishing, 2014. ISBN 0321965515.
- [9] KRUIJFF, E., SWAN, J. E. a FEINER, S. Perceptual issues in augmented reality revisited. In: *2010 IEEE International Symposium on Mixed and Augmented Reality*. 2010, s. 3–12. DOI: 10.1109/ISMAR.2010.5643530.
- [10] KURKOVSKY, S., KOSHY, R., NOVAK, V. a SZUL, P. Current issues in handheld augmented reality. 2012, s. 68–72. DOI: 10.1109/ICCITechnol.2012.6285844.
- [11] LEWIS, J. R. a SAURO, J. USABILITY AND USER EXPERIENCE: DESIGN AND EVALUATION. In: *HANDBOOK OF HUMAN FACTORS AND ERGONOMICS*. John Wiley Sons, Ltd, 2021, kap. 38, s. 972–1015. DOI: <https://doi.org/10.1002/9781119636113.ch38>. ISBN 9781119636113. Dostupné z: <https://onlinelibrary.wiley.com/doi/abs/10.1002/9781119636113.ch38>.

- [12] MOR SAMUELS, E. Mobile-App Onboarding: An Analysis of Components and Techniques. 2021. Dostupné z: <https://www.appsflyer.com/blog/tips-strategy/app-onboarding/>.
- [13] MORAN, K. How Chunking Helps Content Processing. 2016. Dostupné z: <https://www.nngroup.com/articles/chunking/>.
- [14] MORAN, K. Usability Testing 101). 2019. Dostupné z: <https://www.nngroup.com/articles/usability-testing-101/>.
- [15] NIELSEN, J. User Testing: Why How (Jakob Nielsen). 2018. Dostupné z: <https://www.nngroup.com/videos/user-testing-jakob-nielsen/>.
- [16] RENZ, J., STAUBITZ, T., POLLACK, J. a MEINEL, C. Improving the onboarding user experience in MOOCs. *Proceedings EduLearn*. 2014, s. 3931–3941.
- [17] STRAHM, B., GRAY, C. a VORVOREANU, M. Generating Mobile Application Onboarding Insights Through Minimalist Instruction. In: . Jún 2018. DOI: 10.1145/3196709.3196727.
- [18] TOKÁROVÁ, L. a WEIDEMAN, M. Onboarding Strategies for mobile applications. [0037]. In: . September 2013.
- [19] TUMLER, J., DOIL, F., MECKE, R., PAUL, G., SCHENK, M. et al. Mobile Augmented Reality in industrial applications: Approaches for solution of user-related issues. 2008, s. 87–90. DOI: 10.1109/ISMAR.2008.4637330.
- [20] XU, F. Short-term working memory and chunking in SLA. *Theory and Practice in Language Studies*. Academy Publication Co., Ltd. 2016, zv. 6, č. 1, s. 119.