Střední průmyslová škola elektrotechnická  
a Vyšší odborná škola Pardubice

**STŘEDNÍ PRŮMYSLOVÁ ŠKOLA ELEKTROTECHNICKÁ**

**MATURITNÍ PRÁCE – PROGRAMOVÁNÍ**

**AR adventure**

březen rok 2022 Antonín Pecka 4.E

*„Prohlašuji, že jsem maturitní práci vypracoval(a) samostatně a použil(a) jsem literárních pramenů, informací a obrázků, které cituji a uvádím v seznamu použité literatury a zdrojů informací a v seznamu použitých obrázků a neporušil jsem autorská práva.*

*Souhlasím s umístěním kompletní maturitní práce nebo její části na školní internetové stránky a s použitím jejích ukázek pro výuku.“*

*V Pardubicích dne ........................... ................................................*

*podpis*

Text, letter

Description automatically generated

Text, letter

Description automatically generated*Anotace*

*Aplikace, díky které můžeme pohybovat s helikoptéru v reálném světě a sbírat s ní objekty.*

*Annotation*

*An application that allows us to move with the helicopter in the real world and collect objects with it.***Obsah**

[Úvod 9](#_Toc99968547)

[1 Analýza obdobných aplikací 10](#_Toc99968548)

[1.1 Angry Birds AR: Isle of Pigs 10](#_Toc99968549)

[1.1.1 Kladné stránky 11](#_Toc99968550)

[1.1.2 Záporné stránky 11](#_Toc99968551)

[1.2 Stack AR 11](#_Toc99968552)

[1.2.1 Kladné stránky 12](#_Toc99968553)

[1.2.2 Záporné stránky 12](#_Toc99968554)

[1.3 AR Desktop Helicopter 12](#_Toc99968555)

[1.3.1 Kladné stránky 13](#_Toc99968556)

[1.3.2 Záporné stránky 13](#_Toc99968557)

[1.4 Shrnutí 14](#_Toc99968558)

[2 Návrh projektu 15](#_Toc99968559)

[2.1 Cílové skupiny 15](#_Toc99968560)

[2.2 Aktualizace aplikace 15](#_Toc99968561)

[2.3 Kompatibilita 15](#_Toc99968562)

[3 Popis projektu 16](#_Toc99968563)

[3.1 3D model helikoptéry 16](#_Toc99968564)

[3.1.1 Modelování 16](#_Toc99968565)

[3.1.2 Otexturování 16](#_Toc99968566)

[3.1.3 Nastavení pro export modelu 16](#_Toc99968567)

[3.2 Vytvoření a nastavení Unity projektu 17](#_Toc99968568)

[3.2.1 Nastavení build settings projektu 17](#_Toc99968569)

[3.2.2 Instalování pluginů 17](#_Toc99968570)

[3.3 Přidané XR GameObjecty v Unity scéně 18](#_Toc99968571)

[3.3.1 AR Session 18](#_Toc99968572)

[3.3.2 AR Session Origin 18](#_Toc99968573)

[3.4 Vytvoření Input Actions assetu 19](#_Toc99968574)

[3.5 Tvorba Helicopter prefab 20](#_Toc99968575)

[3.5.1 RigidBody 20](#_Toc99968576)

[3.5.2 Box Collider 20](#_Toc99968577)

[3.6 Tvorba Collectible prefab 20](#_Toc99968578)

[3.6.1 Vytvoření objektu 21](#_Toc99968579)

[3.6.2 Vytvoření a přidání animace vrtule helikoptéry 22](#_Toc99968580)

[3.7 Vytváření User Interface aplikace 22](#_Toc99968581)

[3.7.1 Módy aplikace 23](#_Toc99968582)

[3.7.2 User Interface panely 27](#_Toc99968583)

[3.8 Build a kompilace aplikace na zařízení 29](#_Toc99968584)

[Závěr 30](#_Toc99968585)

[Seznam použité literatury a zdrojů obrázků 31](#_Toc99968586)

[Seznam obrázků 32](#_Toc99968587)

[Přílohy 33](#_Toc99968588)

# Úvod

Cílem projektu je si vyzkoušet a naučit nové věci v oblasti technologie augmentované reality. Obsahem projektu je helikoptéra, kterou uvidíme díky AR technologii v zařízení na displeji jako kdyby se helikoptéra pohybovala v reálném světě. S helikoptérou budeme moci pohybovat. Věřím, že AR technologie je věc, která se bude v budoucnu běžně vyskytovat v našich životech. Z tohoto důvodu si rád novou technologii vyzkouším. Pro vývoj projektu typu AR použiji program Blender pro vymodelování a otexturování modelu a Unity s XR pluginem, abych mohl využít ARkit a znalost jazyka C#.

Projekt obsahuje helikoptéru, která se promítá do našeho světa. S helikoptérou můžeme pohybovat pomocí dotyku na displej a sbírat s ní balíčky. Na displeji zobrazuje rozhraní navigující uživatele, co má dělat.

# Analýza obdobných aplikací

Analýzu obdobných aplikací provádíme kvůli inspirování se, nebo i objevení funkcí, o kterých jsme dosud nemuseli vědět. Dále můžeme vidět nějaké nedostatky, kterých se při vytváření vlastní aplikace můžeme vyvarovat.

## Angry Birds AR: Isle of Pigs

Adresa: [play.google.com/store/apps/details?id=com.rovio.abar&hl=cs&gl=US](../../../../Working%20files/Pecka%20Antonín/Analýza%20-%20rešerše/play.google.com/store/apps/details?id=com.rovio.abar&hl=cs&gl=US)

Hra, jejímž cílem je zničení zelených prasátek, které se brání za objekty různých tvarů. Aplikace se skládá z různých levelů, které jsou jinak obtížné.



Obrázek 1 <https://youtu.be/4oXZRAdIMeI?t=3>

Nejdříve musíme naskenovat prostor kde hru budeme chtít hrát.



Obrázek 2 https://youtu.be/4oXZRAdIMeI?t=21

Na obrázku můžeme vidět stíny, modelů a levitující budovy, kvůli špatné detekci horizontální plochy.

### Kladné stránky

Dobře vymodelované a otexturované modely

Stíny objektů

Hra umožňuje naskenovat rovnou plochu, která se po naskenování dále nerozšiřuje

### Záporné stránky

Hra dovolí špatné umístění budov (například na roh stolu)

## Stack AR

Adresa: [apps.apple.com/us/app/stack-ar/id1269638287](../../../../Working%20files/Pecka%20Antonín/Analýza%20-%20rešerše/apps.apple.com/us/app/stack-ar/id1269638287)

Cílem hry je postavit co nejvíce kostiček na sebe, pokud kostičku postavíme nepřesně na tu předchozí. Část, která přečnívá, se ořízne. Konec hry nastane, když už nemáme kostičku, kterou bychom postavili na vršek konstrukce objektů.

Obsah obrázku patro, budova, dřevěné, dřevo

Popis byl vytvořen automaticky

Obrázek 3 <https://youtu.be/RNtYdlP1hpQ?t=96>

Hra má velice barevně výrazné kostičky.

### Kladné stránky

Pěkné použité pestrobarevné materiály pro otexturování objektů

Při přesném položení kostičky vás hra odmění uspokojivým zvukem

### Záporné stránky

Hra je monotónní, časem omrzí

Do hry bych přidal jiné tvary než jen objekty typu Cube

## AR Desktop Helicopter

Adresa: [apps.apple.com/us/app/ar-desktop-helicopter/id1550385309](../../../../Working%20files/Pecka%20Antonín/Analýza%20-%20rešerše/apps.apple.com/us/app/ar-desktop-helicopter/id1550385309)

Velikým přínosem, co jsem u aplikací viděl, bylo „skenování“ prostoru pro položení modelů na rovný povrch. O téhle funkci jsem nevěděl a určitě ji budu chtít v projektu použít.

Náplní aplikace je lítání s helikoptérou pomocí 2 režimů. V 1. režimu se helikoptéra drží před zařízením kamkoliv, kam se se zařízením podíváme. V 2. režimu s helikoptérou můžeme lítat po 2 osách pomocí joysticku.

A drone on a carpet

Description automatically generated with low confidence

Obrázek 4 apps.apple.com/us/app/ar-desktop-helicopter/id1550385309

Na obrázku 4 můžeme spatřit helikoptéru a joystick, díky kterému můžeme s helikoptérou pohybovat.

### Kladné stránky

Pěkný model helikoptéry

### Záporné stránky

Helikoptéra snáze vylítne z obrazovky a poté se těžce hledá v prostředí

Levitující stromy a kameny v prostředí

## Shrnutí

Aplikace z mého pohledu jsou velmi zajímavé. Dokážou zaujmout uživatele, kteří brouzdají App Storem, aby si aplikaci nainstalovali a vyzkoušeli.

# Návrh projektu

## Cílové skupiny

Aplikace je pro všechny iOS uživatele, kteří chtějí prozkoumávat nové věci a třeba se i chvíli zabavit.

## Aktualizace aplikace

U aplikace by probíhali aktualizace, pokud by se objevila chyba nebo by byla možná optimalizace hry. Augmentovaná realita je pořád nová technologie a každých pár měsíců se programování AR aplikací mění.

## Kompatibilita

ARKit vyžaduje iOS 11.0 nebo vyšší a iOS zařízení s A9 nebo lepším procesorem.

# Popis projektu

Projekt obsahuje vymodelovaný a otexturovaný 3D model helikoptéry, nastavení Unity scény. UI panel a Interaction controller pro správné zobrazení jednotlivých prvků na displeji. Animaci pohybující se vrtule helikoptéry. Script pro plynulý pohyb a rotaci helikoptéry na základě stisku uživatele. Script na detekování velikosti vygenerovaných AR plane pro následné spawnování objektů na náhodných lokacích v AR plane. Panel zobrazující počet sebraných objektů helikoptérou.

## 3D model helikoptéry

### Modelování

Model helikoptéry byl polygonově vymodelován v 3D grafickém softwaru Blender pomocí základních nástrojů jako například extrude nebo loop cut.

### Otexturování

Pro otexturování modelu byly využity materiály, které Blender nabízí. U všech materiálů byla upravena barva, roughness a metalic hodnota materiálu.

### Nastavení pro export modelu

Úprava nastavení modelu je velice důležitý krok, pokud tento krok nebyl proveden, mohli by v budoucnu nastat problémy při vytváření animací v Unity. Proto je vždycky dobré provést tyto změny:

* Vymazat původní Blender kameru a osvícení
* Nastavení v panelu Transform vlastnosti Rotation a Scale u všech objektů na 0
* Nastavení Originu ideálně do středu části modelu, která bude animována v budoucnu
* Nastavení Location modelu v panelu Transform na 0

A screenshot of a video game

Description automatically generated

Obrázek 5 Blender scéna s modelem helikoptéry

## Vytvoření a nastavení Unity projektu

Při vytváření unity projektu, ve kterém chceme vytvářet aplikaci s augmentovanou realitou, je nejlepší při vytváření scény zvolit Universal Render Pipeline template. Universal Render Pipeline template oproti ostatním templatům není tak hardwarově náročný a zároveň disponuje rychlým renderováním scény.

### Nastavení build settings projektu

Po vytvoření projektu si musíme zkontrolovat, zda máme správně zvolenou cílovou platformu, pro kterou chceme aplikaci vyvíjet. Po zvolení správné platformy se celý projekt přenastaví pro danou platformu. Jestli jsme nemohli zvolit platformu, pro kterou chceme aplikaci vyvíjet, je třeba doinstalovat build modul. Build modul můžeme doinstalovat v Unity hub, přes který je projekt spuštěn.

### Instalování pluginů

Jestliže chceme vytvářet aplikaci, která obsahuje augmentovanou realitu, budeme muset doinstalovat potřebné balíčky. V package manageru je potřeba naistalovat balíček AR Foundation. Díky nainstalovanému balíčku AR Foundation lze do scény přidávat prvky augmentované reality, se kterými jsme schopni pracovat a využívat je pro získání potřebných dat. Dále je třeba nainstalovat XR plugin. Pokud cílová platforma je android, musí se nainstalovat ARCore XR Plugin, který najdeme v Package manageru. Jestli cílovou platformou je iOS, je třeba naistalovat ARKit XR Plugin.

## Přidané XR GameObjecty v Unity scéně

Do vytvořené scény jsou přidány XR objekty, které lze přidat díky nainstalovanému balíčku AR Foundation.

### AR Session

AR Session se pokusí o nainstalování potřebného AR softwaru v zařízení, pokud na zařízení chybí. Toto není vyžadováno pro všechna zařízení. Například iOS zařízení nevyžadují žádné další aktualizace, pokud zařízení podporuje AR. V případě potřeby při spuštění aplikace bude zjištěna chybějící podpora nebo potřebná aktualizaci, AR Session se pokusí nainstalovat potřebný software přes služby Google Play pro AR. Pokud požadovaný software není nainstalován, AR nebude na zařízení k dispozici.

### AR Session Origin

AR Session Origin je root objektem všech sledovatelných objektů. Tím, že AR Session Origin je root objekt, dokáže zachovat AR kameru a všechny sledovatelné objekty ve stejném prostoru na stejných pozicích vzájemně vůči sobě. Bez AR Session Origin by se AR objekty nezobrazovali na správných místech.

#### AR Raycast Manager

AR Raycat Manager je komponenta, kterou musíme přidat v AR Session Origin. Komponenta AR Raycast Manager umožňuje zjistit, zda se uživatel dotknul displeje. Komponenta převede dotek uživatele na hodnoty, se kterými kód může dále pracovat. Například díky těmito zaznamenanými hodnotami je možné zjistit, jestli se na stisku nenacházela vygenerovaná AR plane.

#### AR Plane Manager

AR Plane Manager je komponenta, kterou můžeme přidat v AR Session Origin. Díky téhle komponentě lze nastavit, jak bude probíhat generování AR Plane prefab. AR Plane prefab lze pomocí AR Plane Manageru generovat na rovných horizontálních i vertikálních plochách. Dále je možné v komponentě vybrat, jaká prefab se bude generovat na detekované rovné ploše.

#### AR Camera

AR Camera je child AR Session Origin. K AR Camera je přiřazen tag jménem MainCamera. Díky přiřazení tagu je možné se lépe dotazovat na kameru při vytváření scriptů.

##### Light Estimation script

Light Estimation script je přidán jako komponenta do AR Camera , která se ve scéně nachází. Script upraví zdroj světla aplikace v závislosti snímaného okolí kamerou. Script byl importován jako asset z jiného projektu.

## Vytvoření Input Actions assetu

S použitím Unity Input System balíčku je možné rozpoznávat, zda se uživatel dotknul displeje pouze jednou či opakovaně, nebo jestli uživatel táhnul přes displej jedním či více prsty. Gesto, které uživatel na displeji provede, je zaznamenáno a porovnáváním gesta se zjistí, zda se neshoduje s vytvořenou akcí. V Input Actions assetu je vytvořená akce, která se jmenuje PlaceObject. Akce PlaceObject má nastavenou detekci na dvojité kliknutí. Pokud se uživatel provede vícenásobné kliknutí na displej, aktivuje se metoda OnPlaceObject, ve které je napsán kód, který se při aktivaci této akce provede. Pro použití Input Action assetu je nutné, aby byl v Player settings nastaven novější Active Input Handling. Systém s názvem Input System Package. Můžeme také nastavit, aby se používal i starší Active Input Handing, který se jmenuje Input Manager. To lze provést vybráním z nabídky možnost hodnoty both.

## Graphical user interface, text, application Description automatically generatedTvorba Helicopter prefab

Obrázek 6 Nastavení Active Input Handling v Player settings

Do Helicopter prefab je přidán model helikoptéry jako GameObject. Model helikoptéry byl vytvořen a exportován z 3D grafického softwaru Blender. Helikoptéra v Inspector window má nastavené původní hodnoty. Do helikoptéry byly přidány dvě komponenty RigidBody a Box Collider.

### RigidBody

Tato komponenta přidává objektu možnost gravitace, kolize objektů a další prvky fyziky. U objektu helikoptéry v komponentě RigidBody je vypnutá možnost použití gravitace. Nastavení Is Kinematic je zapnuté. Colission Detection je nastaven na hodnotu Discrete. V záložce Constraints se nachází nastavení zmražení pozice a rotace v osách x, y, z. Všechny checkboxy jsou nezaškrtnuty kvůli potřebě animování pozice a rotace helikoptéry.

### Box Collider

Komponenta Box Collider obsahuje collider, který slouží jako maska objektu. Pokud se masky odlišných objektů dotknou, komponenta Box Collider pošle signál prostřednictvím zavolané metody například OnTriggerEnter(). Collider je nastaven tak, aby uvnitř collideru byly všechny části helikoptéry. Checkbox s nastavením Is Trigger je zaškrtnut. Díku tomu dokáže objekt při doteku ostatních objektů zavolat Trigger metodu.

## Tvorba Collectible prefab

Vytvořená Collectible prefab musí kvůli fyzice objektů obsahovat komponenty RigidBody a Box Collider. Díky těmto komponentám se může zjistit, zda model spawnuté helikoptéry se dotknul Collectible prefab.

### Vytvoření objektu

V editoru Collectible prefab je vytvořen 3D GameObject Cube, který má nastaven v inspector window nulovou position a rotation. Pouze scale má nastaven hodnoty x=0.03, y=0.04 a z=0.03. Díky nastavení scale vznikne lehce protáhlá krychle do výšky, tudíž výsledným objektem je kvádr. Pro GameObject Cube byl vytvořen materiál. Materiál je hladký, metalický a Base Map materiálu je nastaven na oranžovou barvu. Vytvořený GameObject obsahuje komponenty RigidBody a Box Collider.

#### RigidBody

Komponenta Rigidbody přidává jistou fyziku objektu. V nastavení RigidBody u Cube objektu je vypnuto použití gravitace. Dále je také vypnuta kinematika objektu. Vlastnost Collision Detection je nastavena na hodnotu Discrete. V záložce Constraints nalezneme vlastnosti Freeze Position a Freeze Roration, kde je nastaveno úplné zmražení objektu díky zaškrtnutí všech checkboxů v osách x, y, z.

#### Box Collider

V komponentě Box Collider je collider nastaven přesně na tvar kvádru. Box Collider díky svému collideru, který kopíruje obrysy GameObjectu detekuje objekt, který se collideru dotkne. Při dotyku collideru se spustí metoda OnTriggerEnter(). U Box Collideru musí být checkbox jménem Is Trigger nezaškrtnutý kvůli tomu, že je sbírán helikoptérou. Pokud by to tak nebylo znamenalo by to, že metoda OnTriggerEnter() nebude nikdy spuštěna.

#### CollectiblePrefab script

GameObject Cube má přiřazen script jménem CollectiblePrefab, který udává, co se stane po dotyku collideru objektu. Script také připočítává hodnotu jedna do ScoreManager, který se stará o zobrazení čísla celkového počtu sebraných objektů helikoptérou. Na prvních řádcích scriptu CollectiblePrefab je inicializován objekt scoreManager, který je potřeba pro zvýšení hodnoty na displeji o jedna. Dále se ve scriptu nachází metoda OnTriggerEnter(), která se zavolá v případě, že se objekt například model helikoptéry dotkne collideru objektu. Pokud se tato metoda zavolá, v objektu scoreManager se zavolá funkce, která navýší hodnotu sebraných objektů o jedna a poté se objekt zničí díky metodě Destroy().

### Vytvoření a přidání animace vrtule helikoptéry

V GameObjectu vrtule u Helicopter prefab je přidaná komponenta Animator. Do slotu v komponentě Animator je přidán vytvořený Animator Controller. Animator Controller umožňuje kombinovat a cyklit jednotlivé animace. U vytvořené animace v Animation window je nastavena lineární změna property Rotation.z z hodnoty 360 na hodnotu 0. Tato vytvořená animace má v Inspector window zaškrtnutou property jménem Loop Time, díky tomu se animace bude neustále opakovat dokola. Animace je přidána do Animator Controlleru, kde je zacyklena.

## Vytváření User Interface aplikace

User Interface neboli také UI, které je ve scéně vytvořeno, je používáno pro informování uživatele. Ve scéně se nachází čtyři UI panely Startup UI, NonAR UI, Scan UI a Main UI. Dále ve scéně najdeme čtyři empty GameObjecty Startup Mode, NonAR mode, Scan Mode a Main Mode. Tyto GameObjecty jsou child Interaction Controlleru. Interaction Controller je také empty GameObject. Každý z těchto empty GameObjectů má svůj vlastní script, který se stará o správné zobrazení jednotlivých UI panelů. Při běhu aplikace je zobrazováno pouze jeden ze čtyř UI panelů a je povolen pouze jeden mode.

Text

Description automatically generated

Obrázek 7 Hierarchie jednotlivých objektů

### Módy aplikace

Aplikace se skládá z jednotlivých módů, které jsou postupně povolovány a zakazovány. Všechny módy mají rodiče Interaction Controller, který při startu aplikace povolí Startup mode, který vykreslí Startup UI. Ostatní módy jsou zakázány.

#### Interaction Controller

Interaction Controller je Empty GameObject, který má mezi komponenty přidaný script jménem InteractionController a komponentu Player Input. InteractionController script používá unity package Serialized Dictionary Lite, který pomáhá řadit jednotlivé UI panely do jednotlivých slotů, které najdeme v Unity Inspector window. Díky slotům se vytvoří pole, na které se lze ve scriptu snadno dotazovat. Serialized Dictionary Lite lze bezplatně stáhnout na stránce Unity Asset Store. Ve scriptu InteractionController se nachází kód, který při spuštění aplikace zakáže všechny dostupné módy, které jsou zadány v Serialized Dictionary. Módy se zakáží proběhnutím foreach cyklu. Při spuštění aplikace se ve scriptu InteractionController v metodě Start() povolí Startup mode. V komponentě Player Input ve vlastnosti Actions je nastaven AR Input Actions, který byl vytvořen na začátku projektu.

#### Startup mode

Tento mód probíhá při spuštění aplikace. Mód také spustí rozhraní Startup UI. Dále kontroluje, zda mobil disponuje potřebnými funkcemi, ke spuštění AR aplikace. Kontrolu provádí XR Gameobject AR Session. Pokud zařízení nemá potřebné funkce ke spuštění aplikace, program se pokusí potřebné věci doinstalovat. Jestliže instalace proběhne úspěšně, dostaneme se do skenovacího módu aplikace. V opačném případě se přesměrujeme do NonAR módu.

#### NonAR mode

Tento mód se spustí, pokud u zařízení proběhlo neúspěšné nainstalování potřebných funkcí pro chod AR aplikace nebo zařízení, na kterém je aplikace spuštěna není podporováno. Na displeji se objeví hláška „Augmented reality not supported on this device.“ Dále na černém displeji zůstane pouze tato hláška a z NonAR módu se do dalších módů nelze přejít.

#### Scan mode

Pokud zařízení disponuje potřebnými funkcemi pro spuštění AR aplikace a zároveň je zařízení, na kterém je aplikace spuštěna podporováno, dostaneme se do Scan módu. Scan mode povolí rozhraní Scan UI, který na displeji zobrazí pokyny, které navigují uživatele, co má dělat. Navigující pokyny se skládají z různých animací, které byly importovány jako asset jménem OnboardingUX z jiného projektu. AR Plane Manager, komponenta, která je přidána v AR Session Origin se pokusí detekovat rovné horizontální plochy. Pokud AR Plane Manager rovnou horizontální plochu detekuje, plocha se zobrazí na displeji s předem definovaným materiálem, který je  jako property přidán v AR Plane Manageru v podobě prefab. Scan mode skryje Scan UI rozhraní a povolí Main mode, který nám zobrazí rozhraní Main UI.

#### Main mode

Rozhraní navigující uživatele zmizí. Objeví se Main UI. Main UI obsahuje counter, který zobrazuje počet sebraných objektů helikoptérou. Main mode také obsahuje script, ve kterém je vytvořena animace rotace helikoptéry a posun na místo, které se nachází ve vygenerované AR plane prefab. Dále se ve scriptu nachází generování collectible objektů na náhodných místech v AR plane prefab.

##### Moving Script

Moving Script řeší veškerou problematiku, která se týká detekce AR Plane a následné provedení animace rotace a pohybu helikoptéry. Script dále obsahuje náhodné spawnování collectible objektů ve vygenerovaném objektu AR Plane, u které se rozměry během chodu aplikace můžou měnit.

###### Animace rotace a pohybu helikoptéry

Na začátku scriptu nalezneme metodu OnPlaceObject(), která se zavolá na základě provedení akce, co uživatel na displeji provedl za gesto. Vlastností metody je input doteku. První podmínkou, kterou v metodě uvidíme nám zajišťuje, že metoda nebude dále pokračovat, pokud se zrovna provádí animace helikoptéry. Je to prevence vůči uživateli, který by se neustále dotýkal displeje a helikoptéra by se nestíhala otáčet a přesouvat na cílové body, které byly stisknuty. Pokud se žádná animace neprovádí metoda si uloží input doteku do Vector2 touchPosition. Po uložení pozice doteku následuje další podmínka, ve které zjišťujeme, zda dotek byl proveden ve vygenerované AR Plane. Pokud byl dotek opravdu proveden v AR Plane, dostaneme se k dalším dvěma podmínkám, kde se zjišťuje, jestli se jedná o první dotek na AR Plane v aplikaci nebo ne. Jestliže ano, na pozici doteku bude vytvořena instance objektu helikoptéry a následně se uloží pozice, na které byla helikoptéra vytvořena. Pokud bude zjištěno, že se nejedná o první dotek, bude zavolána metoda SetPositionAndRotation(). Vlastností této metody bude pozice dotyku. V metodě SetPositionAndRotation() je nejprve uložena původní pozice helikoptéry, předchozí rotace helikoptéry a finální pozice helikoptéry (pozice stisku uživatele). Dále je v metodě vypočítán rozdíl mezi předchozí a finální lokací helikoptéry, aby se dále mohl vypočítat finální úhel, o který se má helikoptéra otočit. Úhel je vypočítán díky goniometrické funkci tangens. Po provedení výpočtu je výsledný úhel uložen do proměnné finalRotation. Dále je vypočítána výsledná rotace rozdílem předchozí a finální rotace. Na konci metody se nachází podmínka, která ošetřuje neefektivní rotaci helikoptéry. Poslední řádek pouze spustí animace v metodě Update() s nastavenými hodnotami. Metoda Update() je předem vygenerovaná metoda ve scriptu společně s metodou Start(). Tato metoda je volána s každým snímkem aplikace. V metodě jsou definovány dvě proměnné typu float jménem ratio a ratio2, které udávají rychlost animace rotace a pohybu. Dále se v metodě nachází postupné změnění hodnot rotace a pozice helikoptéry díky funkci Vector3.Lerp() v závislosti na ratio. Až animace budou provedeny, nastaví se proměnné ratio na hodnotu 0 a bool runAnim na hodnotu false.

###### Náhodné spawnování collectible objektů

Moving script také řeší spwanování collectible objektů na náhodném místě v AR Plane. V metodě OnPlaceObject() nalezneme podmínku, která zjišťuje, zda dotyk uživatele byl proveden na AR Plane. Pokud bude zjištěno, že dotyk byl opravdu proveden v AR Plane, uložíme si komponentu AR Plane Manager, která se nachází v AR Session Origin do proměnné a přidáme AR Plane novou Action, jež se zavolá, pokud bude změněna rozloha vygenerované AR Plane. AR Plane se během chodu aplikace může neustále zvětšovat, pokud uživatel namíří fotoaparátem mobilu na novou rovnou horizontální plochu, která se bude nacházet v odlišné výšce než vygenerované AR Planes, vygeneruje se nová AR Plane na právě viditelné rovné ploše. Každá AR Plane má vlastní Action. Po dotyku na stisknuté AR Plane se aktualizují rozměry a uloží se do proměnných typu Vector2. Hodnoty obsahují informaci o rozsahu a výšce stisknuté plane. Rozsahy stisknuté AR Plane se vypočítají díky údajům středu dané AR plany a její velikosti. Další potřebný kód ke spawnování collectible objektů na náhodných místech se nachází v metodě SpawnCube(). Metoda SpawnCube() náhodně vypočítá lokaci v dané AR Plane, kam se objekt spawne. K vypočítání se využívá funkce Random.Range, kde jsou zadány hodnoty, které byly zjištěny a aktualizovány v metodě SetRange(). S vypočítanou náhodnou pozicí v AR Plane se vytvoří instance collectible objektu, který bude umístěn v AR Plane. Pokud se ve scéně nachází více vygenerovaných AR Plane, pozice objektu se vypočítává na poslední AR Plane, která byla stisknuta. Díky metodám SetRange() a SpawnCube() se může provádět pravidelné spawnování collectible objektů. V metodě Update(), která je volána každým snímkem se nachází proměnná typu int jménem counter. Každým voláním metody Update() se counter navýší o hodnotu jedna. V metodě je podmínka, která hlídá hodnotu proměnné counter. Až hodnota dosáhne čísla 180, podmínka nastaví hodnotu counteru na 0 a zavolá metodu SpawnCube(). Kdyby například aplikace byla spuštěna na zařízení, které má frekvenci displeje 60 Hz, metoda SpawnCube() bude provedena každou třetí sekundou, co je aplikace spuštěna.

### User Interface panely

UI panely mají rodiče UI Canvas, který zobrazuje jednotlivé User Interface panely. UI Canvas má přidanou jako komponentu script jménem UIController. UIController script obsahuje metodu jménem ChangeUI(), která mění jednotlivé UI panely podle parametru, který do metody byl zadán. Před změnou UI panelu se nejdříve provede metoda FadeIn() a metoda FadeOut(), které jsou volány v metodě ChangeUI(). Metody FadeIn() a FadeOut() zajišťují plynulejší přechod mezi jednotlivými UI panely. V těchto metodách se používá rozšíření DOTween, díky kterému můžeme dosáhnout již zmíněné plynulosti přechodu UI panelů. DOTween obsahuje metodu DOFade(), která nám dokáže vytvořit plynulý přechod mezi panely. DOTween extension je zdarma a lze ho nainstalovat v Package manageru.

#### App Title Panel

App Title Panel je součástí UI Canvas. Panel za celý běh aplikace nezmizí. Zobrazuje název aplikace. V komponentě Rect Transform je nastaven Anchor Presets na Stretch Top. Díky Stretch top nastavení se panel zobrazí na vrchu displeje a bude roztažen přes celou šířku displeje. V komponentě Image je nastavena bílá barva s hodnotou alpha 100. Díky tomu je obrázek bílé barvy a je průhledný. V App Tittle panelu také najdeme Title Text s nastaveným Anchor Presets Stretch Stretch. V panelu je nastaven text na „ARAdventure“ a Vertex Color je nastavena na růžovou.

#### Startup User Interface

Startup User Interface spadá pod UI Canvas. Startup UI se objeví pouze při startu aplikace díky Startup módu. Vlastnost Anchor Presets v komponentě Rect Transform jsou nastaveny na Stretch Stretch. S nastavením tohoto typu se panel roztáhne do obou směrů po celém displeji. Startup UI obsahuje Text, který má stejně nastavené Anchor Presets jako u Startup UI. Text zobrazí na displeji „Initializing….“ v bílé barvě.

#### NonAR User Interface

NonAR User Interface je součástí UI Canvas. Panel se zobrazí po Startup UI v případě, že zařízení nemá nainstalované potřebné AR funkce, nebo zařízení není podporováno. O zobrazení NonAR UI se stará NonAR mode. Panel obsahuje Text „Augmented reality not supported on this device.“ bílé barvy. NonAR UI a Text mají nastavené Anchor Presets na hodnoty Stretch Stretch, aby panel byl uprostřed displeji a mohl se roztahovat až do okrajů.

#### Scan User Interface

Scan User Interaface spadá pod UI Canvas. Panel se objeví hned po Startup UI, pokud zařízení disponuje všemi AR funkcemi, co aplikace vyžaduje. O povolení zobrazení Scan UI se stará Scan mode. Scan UI obsahuje objekt Animated Prompt, který využívá importovaných animací k navigaci uživatele, co má se zařízením dělat. Nastavení Anchor Presets je mezi objekty odlišné. Scan UI je nastaven hodnotami Stretch Strerch, zatím co Animated Prompt je nastaven Middle Center.

#### Main User Interface

Main User Interface je také potomkem objektu UI Canvas a je poslední UI, které se v Canvas UI nachází. Main UI se zobrazí hned poté, co zařízení detekuje rovnou horizontální plochu ve Scan mode. Rovná horizontální plocha se na displeji zařízení zobrazí jako AR Plane prefab. Po detekci a vykreslení rovné plochy na displej se na zařízení přestává zobrazovat Scan UI a Main mode povolí zobrazení Main UI. Toto UI zůstává aktivní až do ukončení aplikace. To znamená, že z Main UI nemůžeme přejít zpátky do předchozích již zmíněných UI. Main UI obsahuje ScoreManager a TextNumber. ScoreManager se stará o zobrazení počtu sebraných objektů helikoptérou. S každým novým sebraným objektem se TextNumber, které je zobrazen na displeji, navýší o hodnotu jedna. TextNumber najdeme na displeji hned pod App Title Panel, který je na vrchu displeje.

## Build a kompilace aplikace na zařízení

Aplikace je mířena na iOS zařízení. Pro kompilaci na iOS zařízení je potřeba vlastnit Apple Developer licenci. Dále se musí v programu XCode ověřit, zda jsou build settings správně nastavené. Dále musí být zařízení, na kterém bude aplikace kompilována přidáno mezi vývojářské zařízení v Apple Developer Account. Při kompilaci projektu je použito iOS a macOS zařízení. Pokud obě zařízení budou připojena k internetu, aplikace se úspěšně zkompiluje.

# Závěr

Během tvorby projektu jsem narazil na řadu problémů. Kvůli špatnému nastavení unity scény jsem byl donucen projekt refaktorovat vytvořením zcela novým projektem. Nejtěžší úlohou v projektu bylo vytvoření MovingScriptu, který nastavuje pozici a rotaci helikoptéry a generuje collectible objekty na náhodných místech v AR Plane. Do projektu plánuji ještě věnovat čas a vylepšit model helikoptéry a dále vytvořit komplexnější model, který bude helikoptérou sbírán. Rovněž je žádoucí přidání zvuku, který se vždy pustí při sebrání dalšího collectible objektu. Dále plánuji přidat timer, který bude uživatele motivovat k rychlému sbírání objektů. Výsledný projekt plánuji po aplikaci uvedených návrhů ke zlepšení publikovat v obchodě App Store, kde si budou moci hru vyzkoušet ostatní.

# Seznam použité literatury a zdrojů obrázků

Augmented Reality with Unity AR Foundation [online]. 1. Birmingham: Packt Publishing, 2021 [cit. 2022-04-06]. ISBN 978-1-83898-259-1. Dostupné z: <https://www.packtpub.com/product/augmented-reality-with-unity-ar-foundation/9781838982591>

arfoundation-demos: [online]. [cit. 14. 03. 2022]. Dostupný z: [*https://github.com/Unity-Technologies/arfoundation-demos*](https://github.com/Unity-Technologies/arfoundation-demos)

arfoundation-samples: [online]. [cit. 14. 03. 2022]. Dostupný z: <https://github.com/Unity-Technologies/arfoundation-samples>

# Seznam obrázků

[Obrázek 1 https://youtu.be/4oXZRAdIMeI?t=3 10](#_Toc100125208)

[Obrázek 2 https://youtu.be/4oXZRAdIMeI?t=21 11](#_Toc100125209)

[Obrázek 3 https://youtu.be/RNtYdlP1hpQ?t=96 12](#_Toc100125210)

[Obrázek 4 apps.apple.com/us/app/ar-desktop-helicopter/id1550385309 13](#_Toc100125211)

[Obrázek 5 Blender scéna s modelem helikoptéry 17](#_Toc100125212)

[Obrázek 6 Nastavení Active Input Handling v Player settings 20](file:////Users/pecka/work/School/ARAdventure/dokumentace/2022_4E_MP_PeckaAntonín_AR%20adventure.docx#_Toc100125213)

[Obrázek 7 Hierarchie jednotlivých objektů 22](#_Toc100125214)

# A screenshot of a computer Description automatically generated with medium confidencePřílohy

Obrázek 8 Unity scéna