

ODBORNÁ MATURITNÍ PRÁCE

Realizace 3D hry s rpg prvky s pohledem z první osoby



META MAYHEM

3d hra z první osoby s rpg prvky

Autoři: Lukáš Gubala, Tomáš Krejčík

Třída: C4a

Škola: Střední průmyslová škola elektrotechnická Ječná

Místo: Praha 2, Ječná 30, 120 00

Obor: Informační technologie

Konzultant: Mgr. Alena Reichlová

Anotace

Účelem této dokumentace je popsat postup tvorby, realizace a následné implementace hry v prostředí herního enginu Unity.

Prohlášení

Prohlašuji, že jsem na dlouhodobé práci pracoval ve dvojici se spolupracovníkem.

Všechny použité zdroje a prameny jsou uvedeny na konci dokumentace.

Obsah

Úvod

-Zadání

Využité aplikace a software

Herní objekty

-Dveře/brána

-Přístupová karta/Keycard

-Inventář

Skripty

-Inventory script

-Door script

-Item script

-UIMenuManager

-UIOptionsManager

-ItemAssigner

Zdroje

Závěr

Úvod

Zadání

Zadáním je vytvořit hru s rpg prvky v prostředí pro tvorbu videoher. Hlavním prvkem gameplaye bude souboj s vlnami nepřátel různých druhů řízené umělou inteligencí. Hráč má k dispozici arsenál zbraní, které se dají vylepšit společně s dalšími vylepšeními. Ve hře je pouze agresivní ai, takže nepřátelé budou útočit při prvním setkání s hráčem.

Využité aplikace a software

Unity

Na začátku tvorby dlouhodobé práce jsme se společně domlouvali jaké prostředí pro práci využít. Napadli nás Unreal Engine od Epic Games a Unity od Unity Technologies. Z důvodu, že s Unreal Enginem jsme neměli příliš mnoho zkušeností, zvolili jsme Unity. Konkrétní verze, kterou jsme zvolili je 2021.2.13f1. Verze totiž podporuje nové funkce na rozdíl od starších. Jediná nevýhoda byla spuštění projektu na školním počítači.

Visual Studio Community 2019

Pro samotné skripty pro herní objekty a logiku jsme využili plně integrované Visual Studio Community 2019 od společnosti Microsoft. Při psaní kódu je Visual Studio vysoce užitečné, pomáhají totiž funkcí IntelliSense, které napovídá i v kontextu Unity.

GIMP

GIMP je bezplatná alternativa pro Adobe Photoshop.

Používá se pro tvorbu, úpravu a formátování fotografií úpravou části fotografie nebo jednotlivých pixelů(bitmapový editor). Software jsem použil na úpravu některých textur/materiálů a tvorbu barevného gradientu pro herní ui(user interface).

Trello

Jedna z důležitých věcí při vývoji her je komunikace. Pro tento účel byl využit Trello od Fog Creek Software. Trello je online software, zaměřený na organizaci úkolů různých druhů. Umožňuje vizuálně zobrazit jednotlivé zadání a rozřadit je do různých kategorií.

Discord

Discord je komunikační platforma. Platformu jsme ve dvojici použili pro komunikaci společně s aplikací Trello.

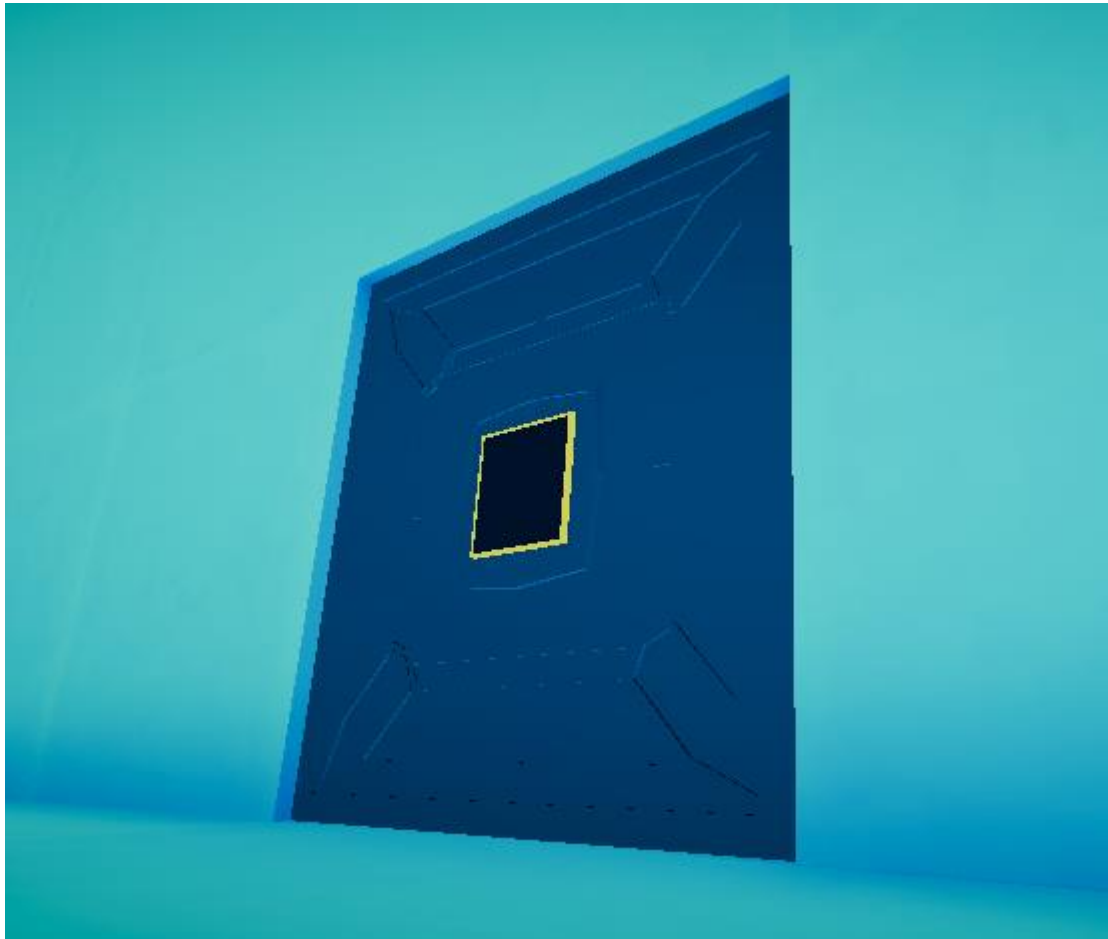
Audacity

Dalším důležitým aspektem video her je zvuk. Pouhý zvukový efekt může značně vylepšit pocit z hraní. Pro tuto oblast jsem využil aplikaci Audacity. Audacity je open source, multiplatformní editor digitálního zvuku.

Herní objekty

Dveře/brána

Ze stylistických důvodů a nižší náročnosti na vytvoření jsme zvolili minimalistický futuristický art-style. Dveře se skládají z pěti částí, což jsou čtyři části dveří a prostřední část. Část modelu uprostřed slouží pro rozlišení barvy přístupové karty(keycard), pomocí které může hráč dveře otevřít.



Dveře nefungují jen na klíč, dá se totiž nastavit jak se dveře budou chovat při interakci s hráčem.

Na nastavení způsobu, kterým budou dveře reagovat je v inspektoru objektů několik booleanů.

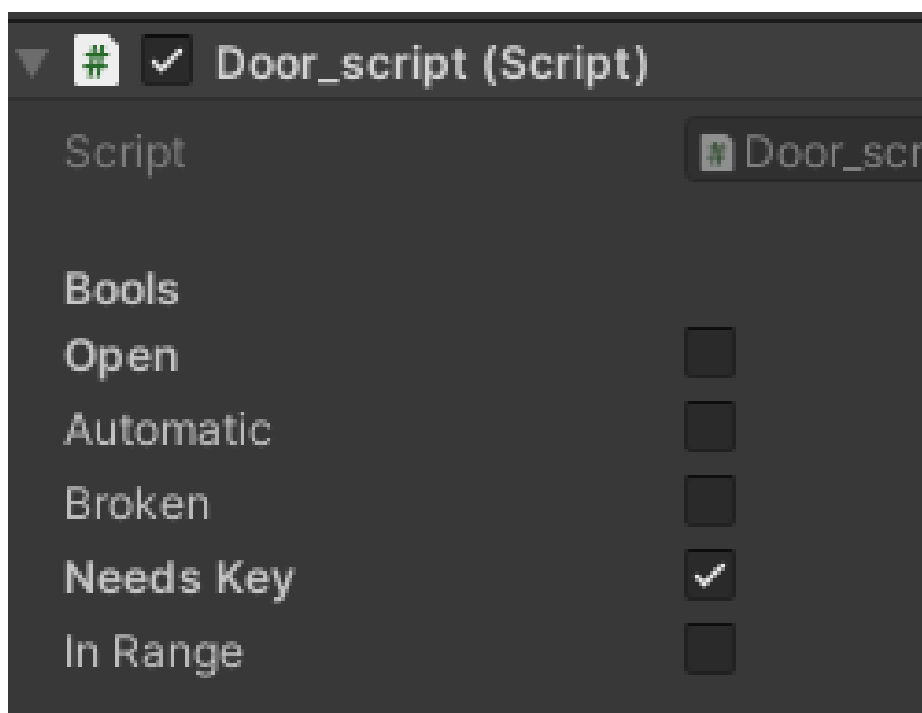
Boolean Open(otevřený) má hodnotu true, pokud hráč splňuje parametry pro otevření dveří.

Hodnota In Range ukazuje, jestli je uživatel dostatečně blízko u dveří, aby mohl s bránou operovat.

Je-li políčko pro proměnnou Automatic nastaveno na true, dveře se automaticky otevřou po příchodu hráče do BoxCollider.

Pro dveře nastavuje hodnota Needs Key, zda dveře potřebují klíč k otevření. V případě že je hodnota true, bude hráč potřebovat klíč stejné barvy, které odpovídá barva na dveřích(na obrázku je to žlutá barva).

Pokud je Broken(rozbítý) zaškrtnuté znamená to, že dveře při pokusu o otevření přehrají „rozbitou“ animaci a s dveřmi se pak dále nedá interagovat.



Přístupová karta/Keycard

Brány, které potřebují klíč musí mít stejný materiál jako samotný klíč.



Na objektu je Collider, který umožní přidat přístupovou kartu do inventáře.

Přístupová karta po sebrání zmizí a přidá se do inventáře.

Inventář

Inventář zobrazuje všechny předměty, které hráč stihl cestou posbírat.

Sebrané předměty se automaticky rozřazují do určených slotů.



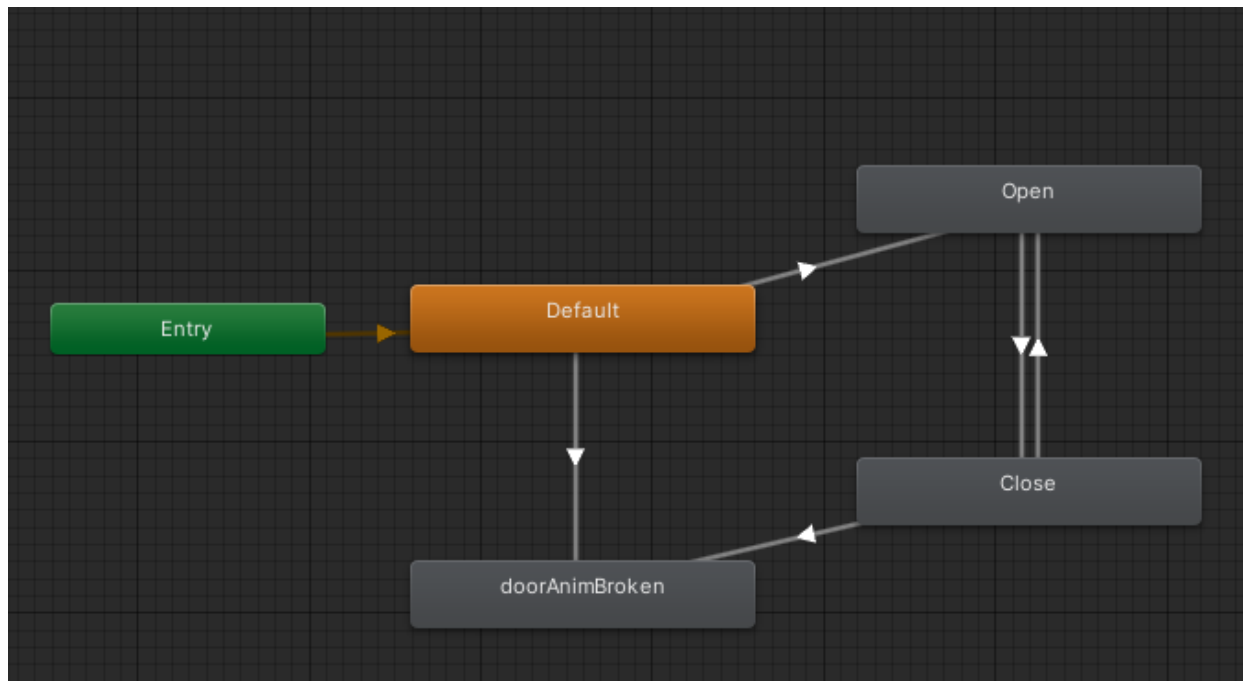
Skripty

Inventory script

Skript inventáře zaručuje, že se itemy opravdu dostanou do inventáře. Obsahuje různé metody pro přidání předmětů, jejich odebrání, získání počtu předmětů.

Door script

Kód spravuje animátor dveří.



Pokud jsou dveře Broken(rozbité) dostává se jejich stav do doorAnimBroken ze kterého se již nedá vrátit.

Je-li Broken vypnuto, dveře se dostávají do smyčky mezi Open a Close(animace otevírání a zavírání dveří).

V door skriptu se také odkazuje na instanci skriptu inventáře. Skrze něj se kontroluje, zda je v inventáři hráče přístupová karta se stejným materiálem jako mají dveře.

```
if (inventory.CheckForKey(displayColor))  
{  
  
    Open();  
}
```

```
1 reference  
public bool CheckForKey(Color color)  
{  
  
    int index = 0;  
  
    foreach((Item, Color) item in inventory_list)  
    {  
        if(item.Item1.GetType() == Item.Type.keycard && item.Item2 == color)  
        {  
            index++;  
        }  
    }  
  
    if (index > 0)  
    {  
        return true;  
    }  
    else  
    {  
        return false;  
    }  
}
```

Item Script

Z třídy Item se vyrábí instance předmětů. Instance se tvoří v *keycard scriptu*.

```
public void Start()
{
    keyRenderer = GetComponent<Renderer>();
    keyMats = keyRenderer.materials;
    keyColor = keyMats[1].color;

    keycard = (Item)ScriptableObject.CreateInstance("Item");
}
```

Item script nedědí od pro Unity obvyklého MonoBehaviour ale od ScriptableObject.

```
public class Item : ScriptableObject
{
}
```

Tímto způsobem vytvořený objekt nebude závislý na fyzickém objektu v prostředí Unity.

Třída obsahuje gettery a settery, přes které se lze dostat k private proměnným.

```
1 reference
public void SetType(Type type)
{
    this.type = type;
}

13 references
public new Type GetType()
{
    return this.type;
}

0 references
public void SetColor(Color color)
{
    this.color = color;
}

0 references
public Color GetColor()
{
    return this.color;
}
```

V konstruktoru se při vytvoření instance objektu definuje typ položky a její barvu.

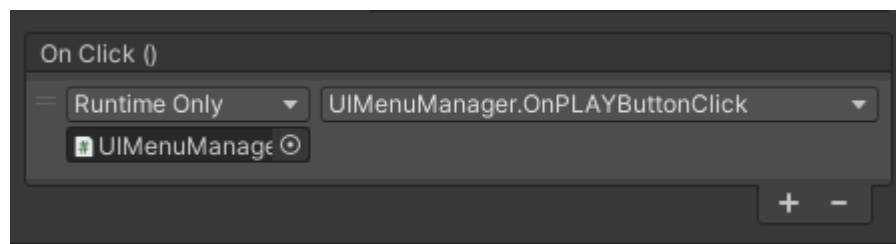
UIMenuManager script

Jednoduší skript pro definici metod, které se spustí po interakci s hlavním menu



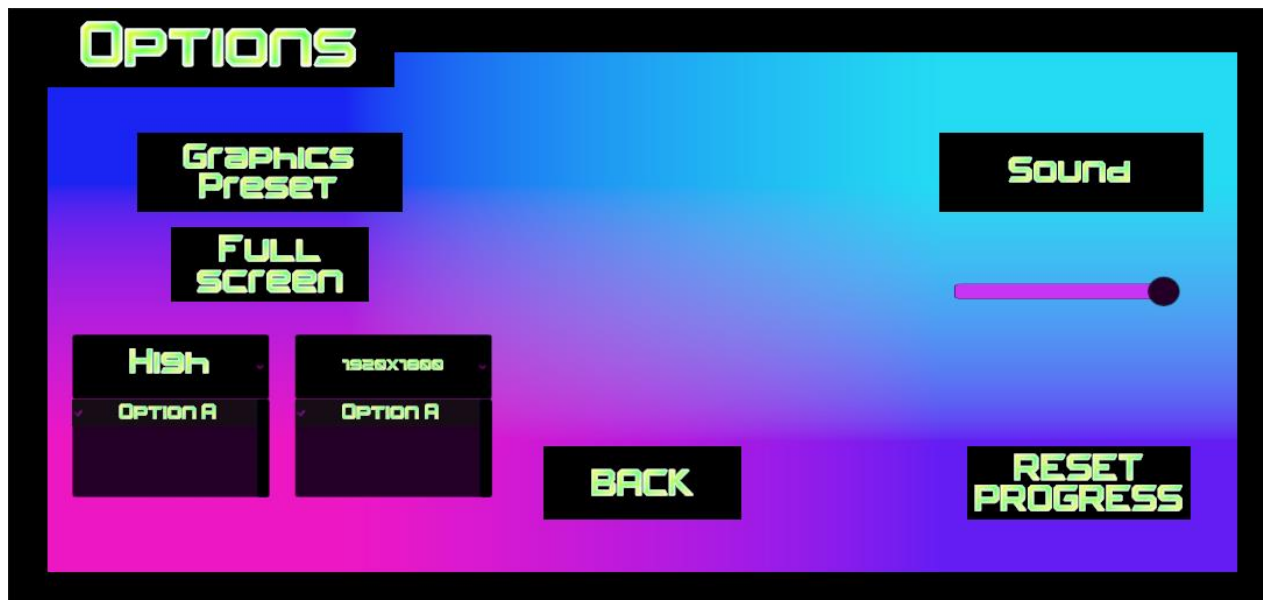
Metody se spouští přes Button.OnClick u daného elementu v UI.

Play button:



UIOptionsManager

Skript využívá stejně jako *UIMenuManager* `Button.OnClick`. Co se kódu týče je rozsáhlejší. Na možnostech jsme si dali záležet, aby se hra dala spustit skoro všude.



V levé spodní nabídce možností je přednastavení grafiky. Z rozevíracího seznamu lze zvolit low, medium či high grafické nastavení.

Tlačítkem s nadpisem Fullscreen se dá zvolit nastavení okna hry. Nabízejí se možnosti Fullscreen, Windowed, Maximized Window a Exclusive Fullscreen.

Vedle nastavení grafiky se nachází rozbalovací menu rozlišení obrazovky. Na výběr je 1920x1080, 1600x900 a 1280x720.

Na pravé straně ui je umístěn slider(posuvník), kterým se ovládá úroveň zvuku hry.

ItemAssigner script

ItemAssigner slouží jako úložiště a řadič pro názvy a ikony objektů v inventáři.

```
! reference  
public Sprite GetItemImage(Item item)  
{  
    Sprite sprite = null;  
  
    if(item.GetType() == Item.Type.ammo)  
    {  
        sprite = ammo_image;  
    }  
    else if (item.GetType() == Item.Type.exp)  
    {  
        sprite = exp_image;  
    }  
    else if (item.GetType() == Item.Type.health)  
    {  
        sprite = health_image;  
    }  
    else if (item.GetType() == Item.Type.keycard)  
    {  
        sprite = key_image;  
    }  
    else if (item.GetType() == Item.Type.currency)  
    {  
        sprite = currency_sprite;  
    }  
  
    return sprite;  
}
```

Podle toho, jaký typ předmětu byl přijat na vstupu se určí, jaký název daný předmět má a jaká ikona mu bude přiřazena.

Zdroje

Webové stránky

- <http://stackoverflow.com/>
- <https://forum.unity3d.com/>
- <https://docs.unity3d.com/Manual/index.html>

Závěr

Práce na tomto projektu mi přinesla spoustu nových zkušeností. Jako přední z nich bych uvedl tvoření zvukových efektů v Audacity. Dále jsem se přiučil, že obrazovka nastavení ve hrách není až tak komplikovaná, jak se na první pohled může zdát. Na závěr komunikace. Díky této společné práci vím více o pracování v týmu a věřím, že jednoho dne budu moci působit jako herní vývojář.