

Vyšší odborná škola a Střední průmyslová škola elektrotechnická Olomouc, Božetěchova 3

**PRAKTICKÁ ZKOUŠKA Z ODBORNÝCH PŘEDMĚTŮ**

2D hra pro dva hráče

Autor **Lukáš Balog**

Obor **Elektrotechnika**

Vedoucí práce **Mgr. Radka Gébová**

Školní rok **2021/2022**

Na druhé stránce bude vloženo originální zadání.

*(včetně podpisů vedoucího práce a maturujícího žáka)*

Prohlašuji, že jsem praktickou zkoušku z odborných předmětů vypracoval samostatně a všechny prameny jsem uvedl v seznamu použité literatury.

……………………………  
 Lukáš Balog

Chtěl bych vyslovit poděkování panu Xxx. Xxxx Xxxxx za odborné konzultace a poskytnuté informace.

……………………………  
 Lukáš Balog

Prohlašuji, že nemám námitek proti půjčování nebo zveřejňování mé práce nebo její části se souhlasem školy.

……………………………  
 Lukáš Balog

Abstrakt

Práce se zabývá tvorbou 2D hry inspirovanou Star Wars. Hra je postavena na modulu pygame pro jazyk python. Tato knihovna a její funkce jsou v práci popsány. Dále je zde popsána celá cesta tvorby hry, od grafického prostředí, až po samotný program. Stručně popisuje techniky použité pro vývoj hry jako jsou: tvorba herního okna, kolize obrázků, nebo ovládání uživatelem.

Obsah

[Obsah 5](#_Toc94722460)

[Úvod 7](#_Toc94722461)

[1. Použitý software 8](#_Toc94722462)

[1.1 Programovací jazyk python 8](#_Toc94722463)

[1.2 Knihovny 8](#_Toc94722464)

[1.2.1 Pygame 8](#_Toc94722465)

[1.2.2 Random 8](#_Toc94722466)

[1.2.3 Sys 8](#_Toc94722467)

[1.2.4 OS 9](#_Toc94722468)

[1.3 Grafický program Pixilart 9](#_Toc94722469)

[2. Herní grafika, zvuky a hudba 10](#_Toc94722470)

[2.1 Vesmírné Lodě 10](#_Toc94722471)

[2.2 Pozadí 16](#_Toc94722472)

[3. Programování 17](#_Toc94722473)

[3.1 Principy vývoje her a herní mechaniky 17](#_Toc94722474)

[3.1.1 Vytvoření okna 17](#_Toc94722475)

[3.1.2 Surface, rect 18](#_Toc94722476)

[3.1.3 Pohyb obrázků, vstupy od uživatele 18](#_Toc94722477)

[3.1.4 Kolize 19](#_Toc94722478)

[3.2 Třídy a funkce 20](#_Toc94722479)

[3.2.1 Třídy 20](#_Toc94722480)

[3.2.2 Funkce 20](#_Toc94722481)

[3.3 Menu 22](#_Toc94722482)

[3.3.1 Funkce 22](#_Toc94722483)

[4. Uživatelská dokumentace 26](#_Toc94722484)

[4.1 První Spuštění 26](#_Toc94722485)

[4.2 Hlavní menu 26](#_Toc94722486)

[4.3 Options 27](#_Toc94722487)

[4.4 Play 28](#_Toc94722488)

[Závěr 29](#_Toc94722489)

[Seznam použité literatury 30](#_Toc94722490)

[Seznam obrázků 31](#_Toc94722491)

[Přílohy 33](#_Toc94722492)

Úvod

Díky modulu pygame je tvorba videoher pro operační systém Windows velice snadná. Poskytne nám vše potřebné pro vývoj her, jako například okno, ve kterém aplikace poběží, nebo ovládání vstupů z klávesnice. Cílem této práce bude za pomocí tohoto modulu vytvořit funkční 2D hru. Práce bude popisovat celý proces tvorby. Od kreslení grafiky a textur, až po samotné programování, a řešení problému s ním spojených.

# Použitý software

## Programovací jazyk python

Python je vysokoúrovňový programovací jazyk, který byl v roce 1991 navržen Guidem van Rossumem. Velkou výhodou tohoto jazyka jsou jeho vyjadřovací schopnosti. To oproti ostatním jazykům umožňuje psát krátký a čitelný kód. Pro tuto práci byl program zvolen zejména pro jeho široké možnosti, které nabízí programátorům.

## Knihovny

### Pygame

Nejdůležitější použitou knihovnou je Pygame. Pygame je souborem funkci a tříd určených pro vývoj jednoduchých 2D videoher v jazyce Python. Tato knihovna má svůj vlastní systém pro vizualizaci a GUI. Dále umí řídit vstupy z klávesnice, ale i myši. Tyto vlastnosti pro nás budou klíčové při vývoji a značně usnadní programování.

Důvod pro užití této knihovny je prostý: Pygame je nerozšířenější knihovna pro vývoj 2D her v jazyce Python. Má stálou podporu, širokou škálu využití, dokumentaci od vývojářů a mnoho materiálů, které vysvětlují problematiky a řeší je za použití pravě této knihovny.

### Random

Knihovna Random je velice jednoduchá. Má několik funkcí pro výběr pseudonáhodných čísel. Pro nás bude důležitá funkce, která umí ze seznamu vybrat náhodný prvek.

### Sys

Knihovna sys je velice rozsáhlá interní knihovna samotného Pythonu. Bude nám stačit vědět, že je používána k ukončování samotného programu po převodu na exe soubor. Pygame už sice tuto funkci zastat umí, ale i samotní autoři doporučují používat právě Sys jako pojistku.

### OS

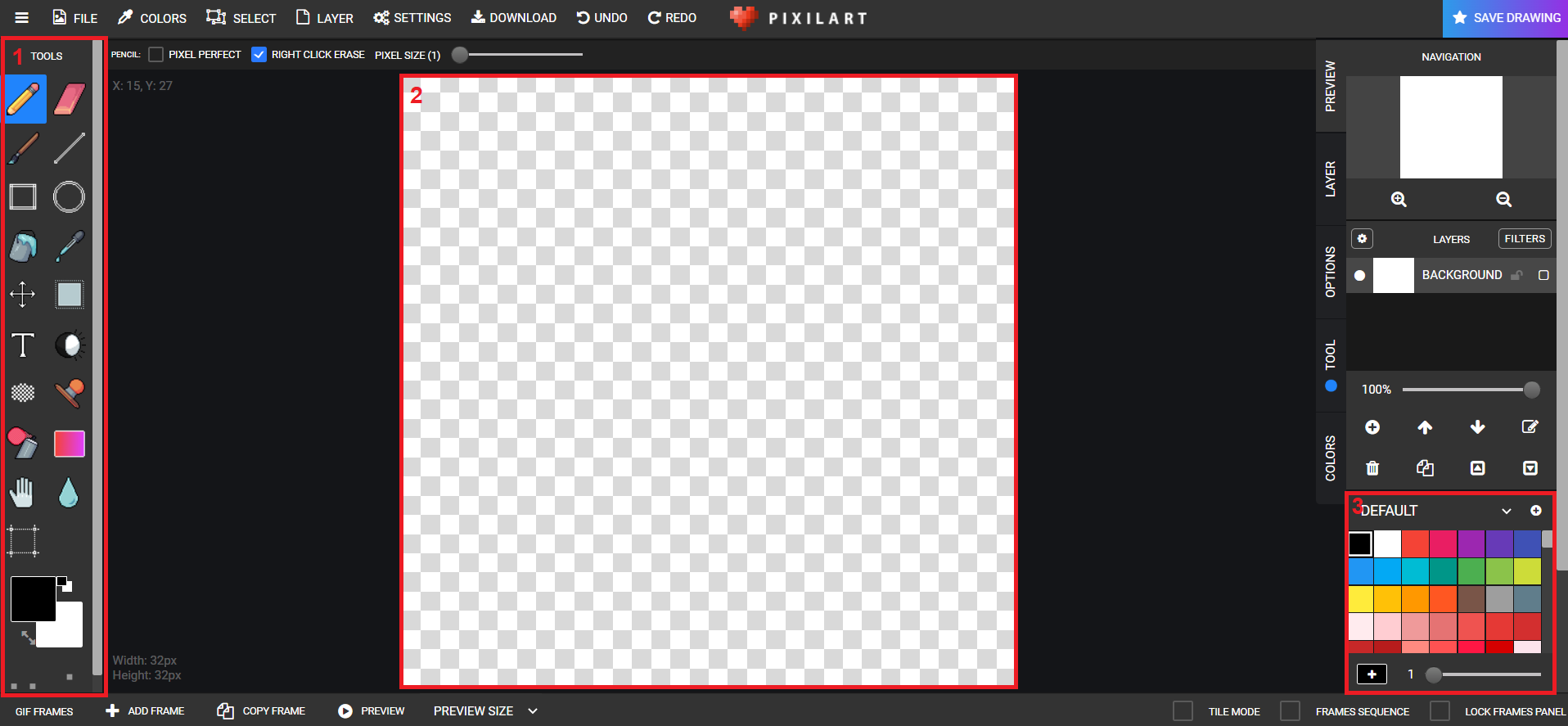
Os neboli operační systém je knihovna, která umožňuje programu práci s funkcemi operačního systému. Důležitá pro nás bude manipulace s obrázky, kterou tento modul zastane. Pygame zvládá i tuto funkci, ale pouze pro některé typy souborů. Proto je důležité použít při importu obrázku tuto knihovnu. Ta nám zajistí vetší volnost a variabilitu v souborech, které můžeme použít.

## Grafický program Pixilart

Pixilart je webová aplikace pro vytváření tzv. „pixel art“ grafik a obrázků. Tento styl byl používán v prvních počítačových hrách. Hlavní důvod, proč používat pixel art, bylo omezení tehdejšími parametry počítačů. Mohlo by se zdát, že pixel art už byl nahrazen propracovanými 3D grafikami, ale opak je pravdou. Pixel art je v dnešní době populárnější, než kdy jindy. Je používán v mnoha hrách jako například Undertale, Binding of Isaac nebo Terraria.

Samotný program je velice jednoduchý a nic nekomplikuje. [1] Na pravé straně je výběr nástrojů, podobných nástrojům z malování, Photoshopu nebo jiného grafického editoru. [2] Uprostřed je samotný canvas, na který jde kreslit. [3] Vlevo je výběr barev, možnost vytvořit si svou paletu barev a zadávaní, které umožňuje RGB HSB nebo hexadecimální kód.

Obrázek : Prostředí Pixilart



Zdroj: Vlastní zpracování

# Herní grafika, zvuky a hudba

Ještě než začneme programovat, musíme vytvořit obrázky a animace, které budeme pro tento program potřebovat. Už jsme se dříve seznámili s aplikaci, kterou pro tento účel využijeme.

Hra bude inspirovaná filmy Star Wars, takže modely lodí budeme vytvářet za pomoci fotek ze samotných filmů. Tvorba pixel art se může zdát velice jednoduchá, ale přetvořit detailní obrázek tak, aby v pixelové formě vypadal dobře, není tak snadné. Existuje na to ovšem několik používaných postupů:

1. Zjednodušovat postupně - vzít předlohu a nakreslit ji méně detailní, pak vzít tuto méně detailní kopii a překreslit ji s ještě méně detaily. Takhle opakovat, až dosáhneme kýženého výsledku.
2. Kreslit přímo zjednodušený obrázek – samotný obrázek upravovat tak dlouho. až vypadá dostatečně dobře.

Každá z metod je dobrá pro něco jiného, jak si ukážeme níže.

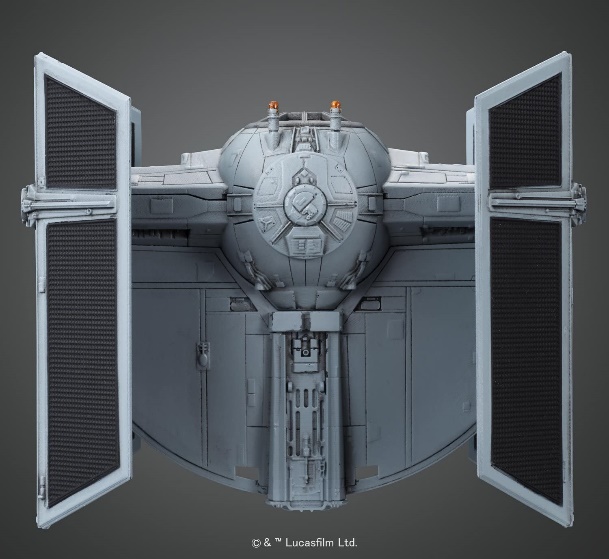
## Vesmírné lodě

Jak už jsme si řekli v úvodu, budou proti sobě soupeřit dvě strany – impérium a rebelové. Každá strana bude mít na výběr ze tří lodí, které může ovládat. Na jedné z nich si ukážeme proces tvorby herní grafiky.

K tvorbě všech lodí byl použit postup č.1, to znamená postupné zjednodušování. Proces vypadá nějak takto:

1. Zde máme nezjednodušený model.

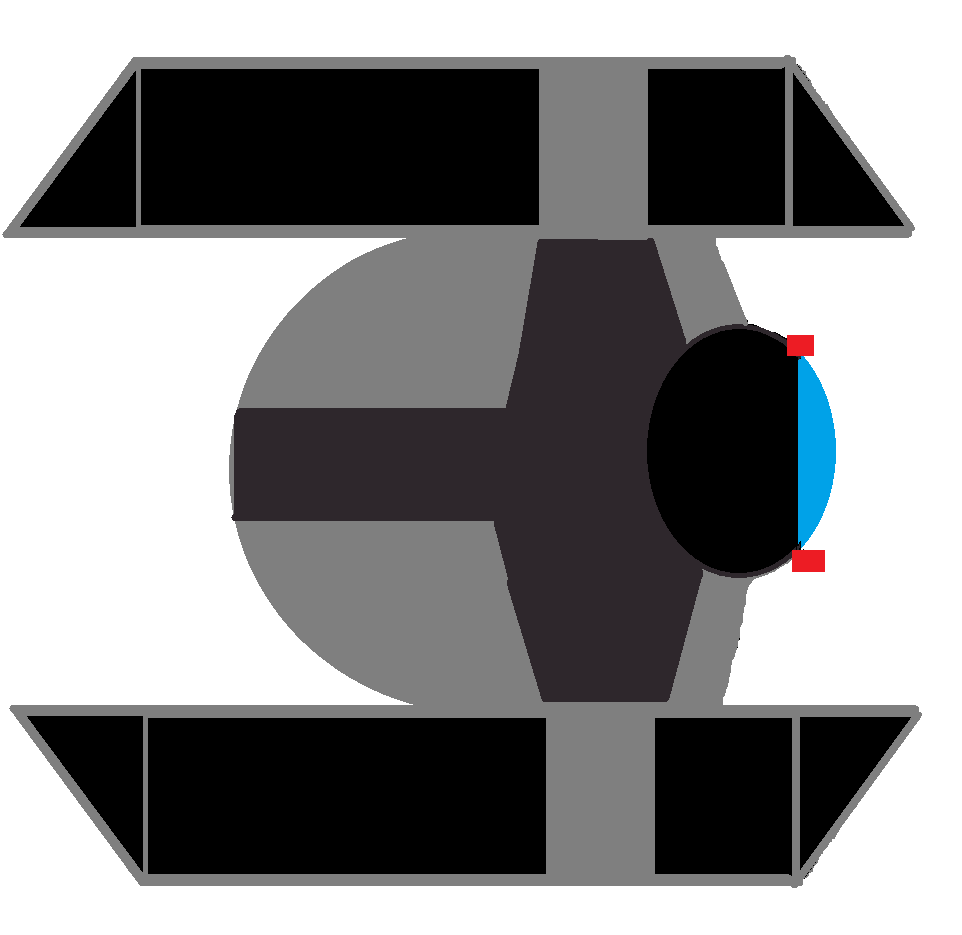
Obrázek : Tie Advanced, pohled shora



Zdroj:<https://www.amazon.com/Bandai-Star-Wars-TIE-Advanced/dp/B00O3RUSOO>

1. V druhém bodě se zbavíme všech přebytečných detailů, které nepůjdou vidět v menším formátu, ještě než přejdeme k pixel art. K tomuto kroku byl použit program malování.

Obrázek : Zjednodušení

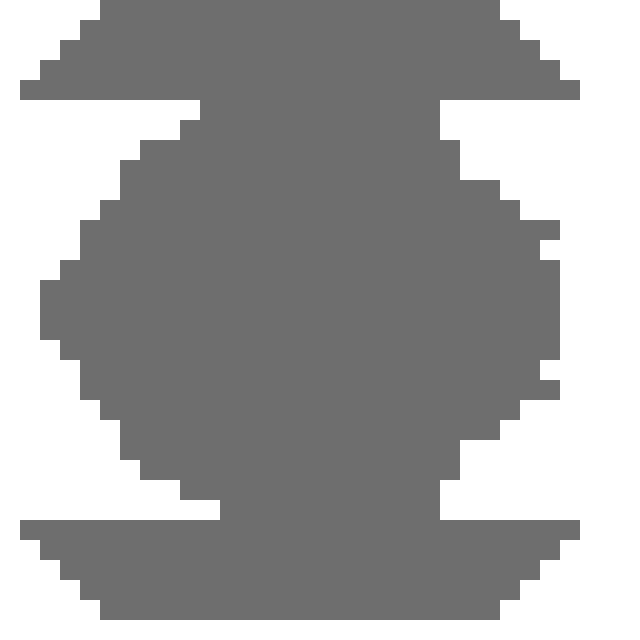


Zdroj: Vlastní zpracování

Nevadí, že obrázek je lehce nevzhledný, jako předloha pro další kroky nám postačí.

1. Přesuneme se do programu Pixilart a v něm nakreslíme hrubou kostru podle předem zjednodušeného obrázku.

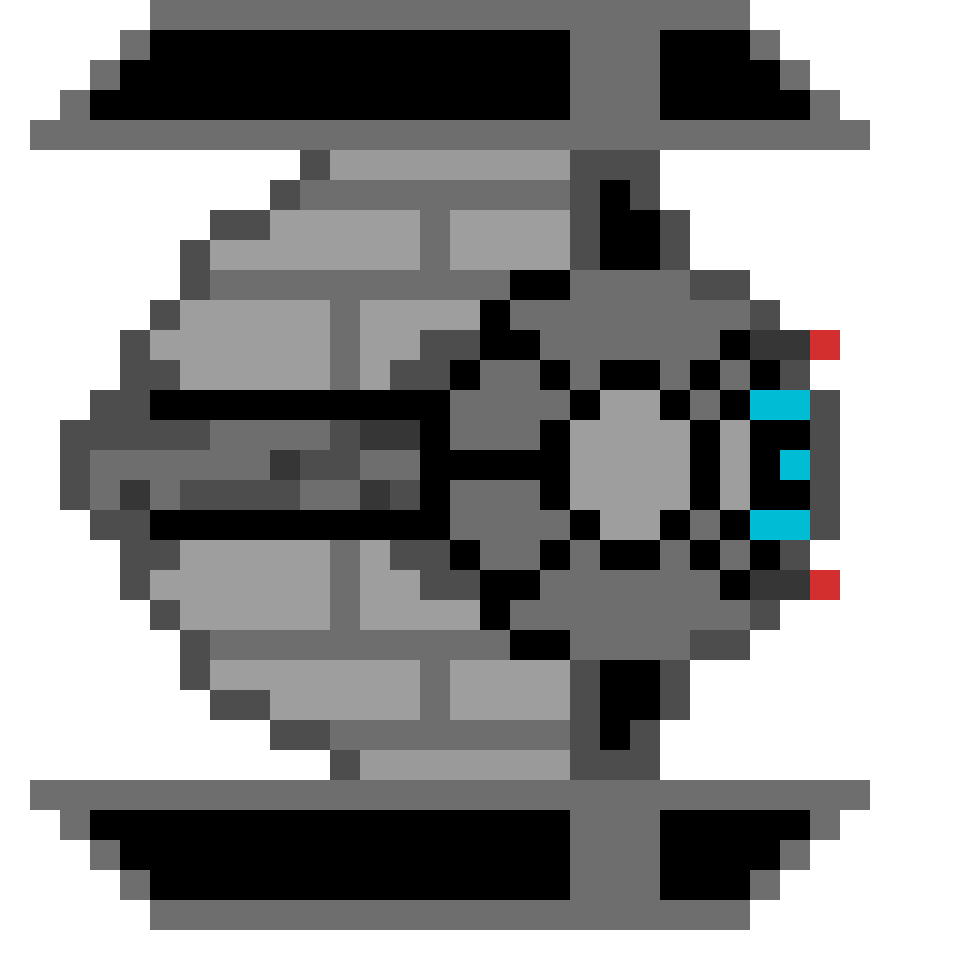
Obrázek : Obrys



Zdroj: Vlastní zpracování

1. Doplníme detaily a přidáme více barev.

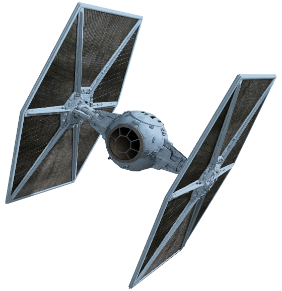
Obrázek :Tie Advanced Finální verze textury



Zdroj: Vlastní zpracování

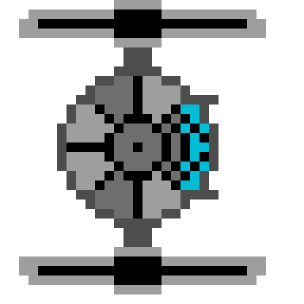
Po překreslení do programu Pixilart vypadá obrázek nějak takto. Touto metodou byly vytvořeny všechny textury pro lodě. Tady je jejich porovnání předloha/výsledek:

Obrázek : Tie Fighter - Předloha



Zdroj: <https://starwars.fandom.com/wiki/TIE_Advanced_v1>

Obrázek : Tie Fighter- Textura



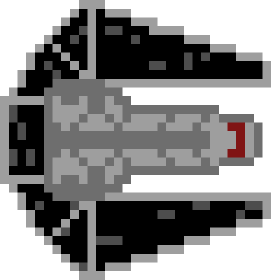
Zdroj: Vlastní zpracování

Obrázek : Tie Reaper - Textura



Zdroj: <https://starwars.fandom.com/wiki/Reaper>

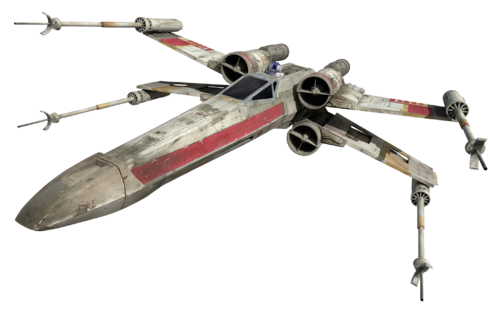
Obrázek : Tie Reaper - Textura



Zdroj: Vlastní zpracování

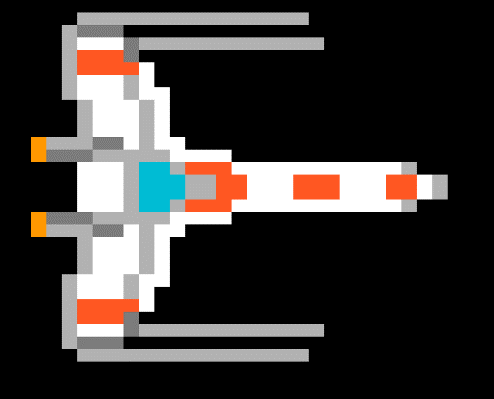
Toto jsou lodě pro prvního hráče. Lodě pro druhého hráče jsou následující:

Obrázek : Xwing - Předloha



Zdroj: <https://starwars.fandom.com/wiki/T-65_X-wing_starfighter/Legends>

Obrázek : Xwing - Textura



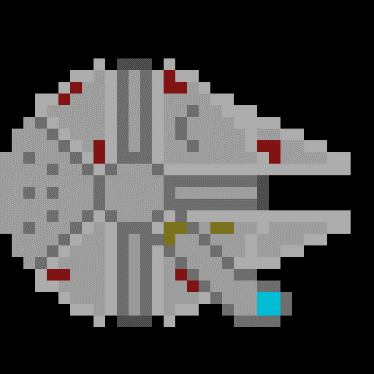
Zdroj: Vlastní zpracování

Obrázek : Milenium Falcon - Předloha



Zdroj: <https://starwars.fandom.com/wiki/Millennium_Falcon>

Obrázek : Milenium Falcon - Textura

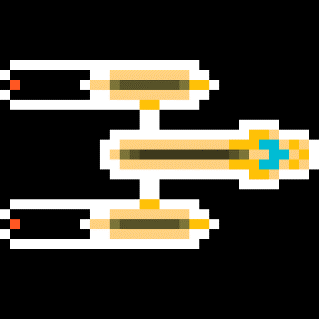


Obrázek : Ywing - Předloha



Zdroj: <https://starwars.fandom.com/wiki/BTL_Y-wing_starfighter>

Obrázek : Ywing - Textura



Zdroj: Vlastní zpracování

## 

## Pozadí

Pozadí bylo tvořeno druhou metodou, zmiňovanou výše. Pro hru bylo potřeba vytvořit dva druhy pozadí:

1. Vesmír, ve kterém probíhá samotná hra
2. Hangár pro menu výběru lodí

Tvorba vesmíru byla prostá, šlo jen o to, vytvořit černou plochu a na ní mnoho bílých hvězd. Výsledek pak vypadá nějak takto:

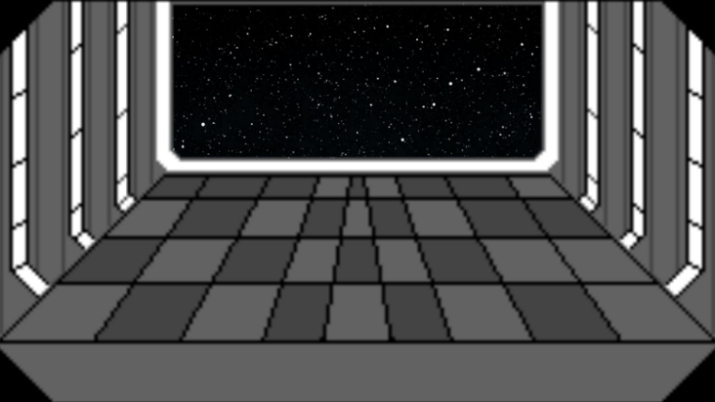
Obrázek : SPACE



Zdroj: Vlastní zpracování

Hangár potom vypadá takto:

Obrázek : HANGAR



Zdroj: Vlastní zpracování

V jeho středu byl opět použit obrázek vesmíru, aby hangár vypadal, že je otevřený.

# Programování

## Principy vývoje her a herní mechaniky

V této kapitole se seznámíme s teoretickými postupy používanými herními vývojáři. Hry nejsou vlastně nic jiného, než rychle se pohybující obrázky, které může uživatel do jisté míry ovlivňovat.

### Vytvoření okna

Ještě než začneme s tvorbou hry, musíme vytvořit funkční okno. Některé knihovny nebo enginy umí okno vytvořit automaticky, ale pygame tuto funkci nemá. Vše musíme udělat za použiti jeho funkcí. První, co musíme udělat pro vytvoření okna, je funkce pygame init():

pygame.init()

pygame.font.init()

pygame.mixer.init()

Tato funkce inicializuje samotnou knihovnu, a my ji tedy můžeme začít používat.

WIDTH, HEIGHT = 900, 500

WIN = pygame.display.set\_mode((WIDTH, HEIGHT))

Takto se vytvoří prázdné okno, které ale po spuštění není funkční. Musíme ještě přidat obnovovací frekvenci a možnost zavíraní okna, než bude vůbec možné s ním pracovat. Pro tyto dva požadavky ale musíme vytvořit cyklus While, v němž pak budou vloženy funkce a třídy, ovládající hru.

FPS = 60

def game(WIN):

    run = True

    while run:

        clock.tick(FPS)

        for event in pygame.event.get():

            if event.type == pygame.QUIT:

                run = False

        WIN.blit(SPACE, (0,0))

  pygame.display.update()

Do budoucna bude lepší tento cyklus dát do samotné funkce. Bude totiž potřeba funkci game volat z herního menu. WIN (jméno proměnné našeho okna).blit umístí na okno obrázek (v tomto případě SPACE – pozadí) na souřadnice 0,0.

Souřadnicový systém pygame je stejný, jako kartézský souřadnicový systém, s tou výjimkou, že počátek (tedy bod 0,0) je v levém horním rohu.

### Surface, rect

Surface (povrch) a rect(zkráceně rectangle) jsou datové typy knihovny pygame. Oba nám dovolují vytvořit v proměnné obrázek nebo tvar. Pro obrázek můžeme použít tuto funkci:

X\_WING\_IMG= pygame.image.load(os.path.join('textures','rebellion', 'Xwing.png'))

Tady používáme knihovnu pygame, ale i os. Do proměnné ukládáme cestu obrázku, jehož parametry jsou ale nevyhovující, takže použijeme funkci pro změnu velikosti a otočení.

X\_WING= pygame.transform.rotate(pygame.transform.scale(X\_WING\_IMG, (60,60)), 0)

Transformujeme obrázek na velikost 60x60px. Tato velikost odpovídá velikosti našeho okna. Rotace je nastavená na 0°, protože obrázek otáčet vůbec nechceme.

TIE\_ADVANCED= pygame.transform.rotate(pygame.transform.scale(TIE\_IMG, (50,50)), 180)

V tomto případe je otočení žádoucí, protože chceme, aby byly obrázky otočené jeden proti druhému, viz obrázek č.19.

### Pohyb obrázků, vstupy od uživatele

Jsou 2 hlavní způsoby, jak zaznamenat vstupy uživatele:

1. pomocí pygame.key.get\_pressed().

keys\_pressed = pygame.key.get\_pressed()

        if keys\_pressed[pygame.K\_a]:

           xwing\_rec.x -= SPEED

        if keys\_pressed[pygame.K\_w]:

           xwing\_rec.y -= SPEED

        if keys\_pressed[pygame.K\_d]:

           xwing\_rec.x += SPEED

        if keys\_pressed[pygame.K\_s]:

           xwing\_rec.y += SPEED

Tuto funkci uložíme do proměnné, aby se nám s ní lépe pracovalo. Tato proměnná teď nabývá hodnot True nebo False. Pokud je keys\_pressed True, změní se proměnná s polohou x nebo y souřadnice adekvátně ke zmáčknuté klávese, v tomto případě WASD. Výhoda této metody je, že zjišťuje, jestli je klávesa zmáčknuta, nebo ne. Díky tomu můžeme tlačítko držet a obrázek se bude stále pohybovat.

if event.type == pygame.KEYDOWN:

if event.key == pygame.K\_RCTRL:

    Tie.shoot\_laser\_tie()

2. Tento způsob vrací hodnotu True pouze po stisknutí klávesy. To znamená, že pokud je klávesa zmáčknuta a držena, vrátí hodnotu True pouze jednou a nevrací ji stále, jako tomu je u předchozího způsobu.

### Kolize

Kolize je další z funkci pygame, která provede nějakou část kódu pouze pokud se překryjí dva nebo více daných obrázků. Vytvořit event, který bude sledovat jestli se daná kolize stala můžeme takto.

TIE\_HIT = pygame.USEREVENT + 1

Musíme přidat číslo + 1 každá kolize má jiné číslo.

if Xwing.xwing\_rec.colliderect(laser):

pygame.event.post(pygame.event.Event(TIE\_HIT))

Pokud se splní tato podmínka, stav kolize se změní na True.

if event.type == TIE\_HIT:

Xwing.HP\_xwing -= 1

HIT\_SOUND.play()

## Třídy a funkce

### Třídy

1. **Tie, Xwing**

Tyto třídy se starají o chování lodí Tie a Xwing. Jsou si v mnoha ohledech podobné, ale liší se zejména funkcemi pro vstupy z klávesnice, texturami, zvuky a proměnnými, které jsou specifické pro každého hráče.

1. **Explosions**

Tato třída má na starost animaci exploze, která se aktivuje, pokud životy jedné z lodí dosáhnou hodnoty 0. V tomto případě bude třída dědit z třídy jménem sprites která je součástí knihovny pygame.

class Explosion(pygame.sprite.Sprite):

    def \_\_init\_\_(self, x, y):

        pygame.sprite.Sprite.\_\_init\_\_(self)

        self.images = []

V inicializační funkci musíme do parametrů uvést hodnoty x a y. Jaké tyto hodnoty budou se vyhodnotí podle toho, který hráč má 0 životů. Souřadnice tohoto hráče se předají třídě při jejím vytvoření. Samotná animace funguje tak, že načte seznam obrázků, které pomocí for cyklu přehraje.

### Funkce

1. **Restart**

Tato funkce, jak už její název napovídá, po ukončeni hry resetuje všechny proměnné, jako např. životy lodí, tím, že zavolá init funkce tříd Tie a Xwing.

1. **Winner**

Funkce kontroluje, jestli mají lodě stále dostatek životů. Pokud některá z nich nemá, vytvoří objekt explosion a spustí se animace. Program chvíli počká, než se exploze dokončí a napíše na obrazovku velkým písmem, která strana vyhrála hru.

Obrázek : Výherní obrazovka



Zdroj: Vlastní zpracování

1. **Game**

Tato funkce zpracovává chod celé hry v jednom While cyklu.

while run:

Proměnná run může nabývat hodnot True nebo False. Volá funkce a ovládá vstupy z klávesnice.

V této funkci se kreslí na herní okno všechna grafika, která se netýká lodí, to znamená pozadí a text. Další důležitou částí je hudba na pozadí. Tu také ovládá tato funkce. Hlavní jsou ale kolize. Každý jeden frame funkce game kontroluje, jestli nekoliduje loď s vystřeleným laserem. Pokud ano, odečte příslušné lodi život.

pygame.display.update()

    pygame.quit()

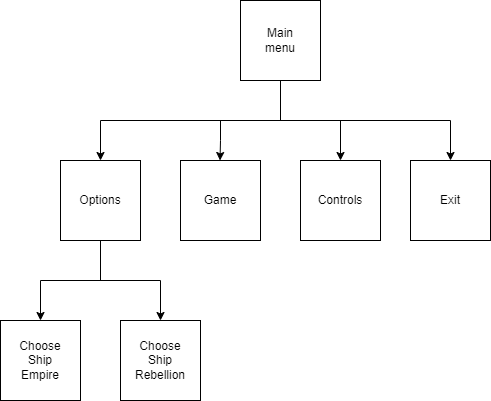
    sys.exit()

Na konci této funkce je metoda display.update. Ta 60x za sekundu obnoví display a aplikuje změny uživatele. Pokud uživatel pohne lodí, metoda update ihned zaregistruje, že se loď pohnula, a přemístí obrázek na jiné místo, odpovídající jeho x a y. Za update metodou už následuje jen pygame.quit. Tato metoda zavře okno a ukončí program, pokud je ukončen While cyklus, tedy pokud se run rovná False. Pro případ, že by se okno nezavřelo správně (dojde k nějaké chybě v počítači nebo v programu), je na dalším řádku ještě sys.exit, který ukonči okno za jakýchkoliv okolností.

## Menu

Okno pro menu vytvoříme stejným způsobem jako herní okno. Pygame má tu vlastnost, že nám dovoluje plynule přecházet z jednoho okna do druhého. Tento princip nám umožňuje vytvořit větvící se menu. Každý blok menu bude mít svůj vlastní While cyklus, při výběru nového bloku se ten předchozí ukonči a otevře se blok nový. Zde je jednoduchý diagram, demonstrující větvení menu:

Obrázek : blokový diagram menu



Zdroj: Vlastní zpracování

### Funkce

1. **Indicator**

Tato funkce načte parametry o textu, výšce a pořadníku. V samotné funkci jsou definovány X souřadnice, ve kterých se indikátor při změně hodnoty pořadníku bude měnit.

if orderer == 1:

    indicator\_height = 247

    submenu = HEALTH\_FONT.render(text1, 1, BLACK)

if orderer == 2:

    indicator\_height = 297

    submenu = HEALTH\_FONT.render(text2, 1, BLACK)

if orderer == 3:

    indicator\_height = 347

    submenu = HEALTH\_FONT.render(text3, 1, BLACK)

if orderer == 4:

    indicator\_height = 397

    submenu = HEALTH\_FONT.render(text4, 1, BLACK)

Pokud je pořadník *a,* výška indikátoru = *a\_výška,*

pokud je pořadník *a,*  text= *a\_text .*

Tímto způsobem se indikátor mění. Funkce je napsána tak, aby pouze změnou jejích parametrů mohla pracovat pro všechny úrovně menu.

Obrázek : Indikátor v menu



Zdroj: Vlastní zpracování

1. **Options\_submenu**

Tato funkce je rozcestníkem. Zvolíme možnost empire nebo rebellion a podle toho se spustí funkce choose\_ship. Dále je tu možnost back, která program vrátí do hlavního menu.

Obrázek : Options

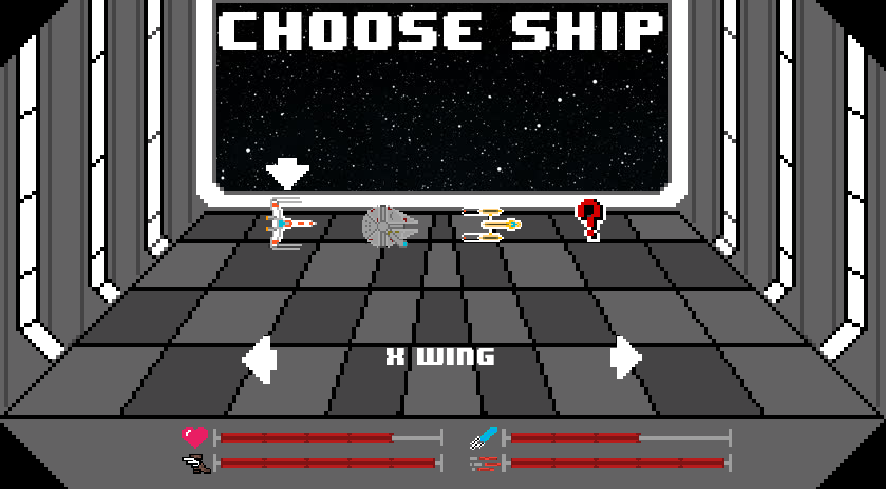


Zdroj: Vlastní zpracování

1. **choose\_ship**

Po zvolení empire/rebellion v options menu se v obou případech spustí tato funkce. Mění se pouze její vzhled a hodnoty, které vrací. V případě volby rebellion, vypadá takto:

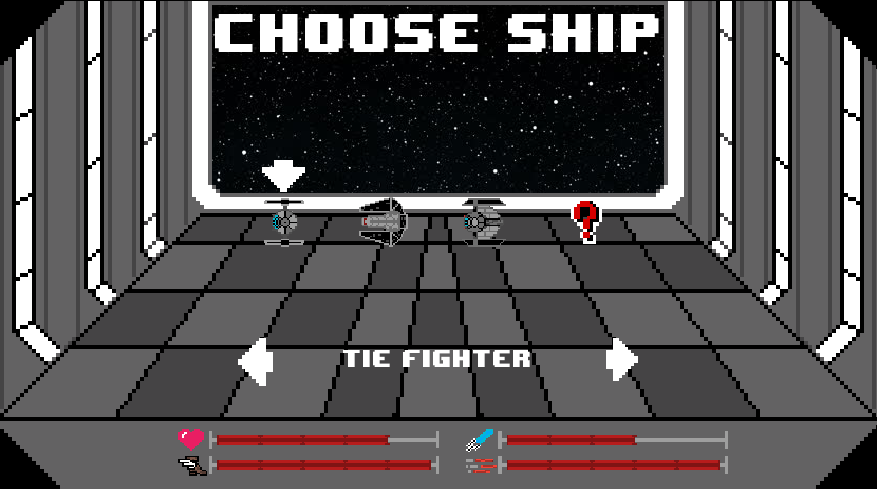
Obrázek : Rebellion výběr lodí



Zdroj: Vlastní zpracování

V případě empire vypadá takto:

Obrázek : Empire výběr lodí



Zdroj: Vlastní zpracování

Pohybem šipek se přemisťuje ukazatel nad loděmi. Pokud je ukazatel nad lodí a zároveň uživatel zmáčkne mezerník, tato loď bude na další hru zvolena a automaticky se program vrátí do options menu. Jde zvolit i červený otazník (random). V tomto případě funkce pomocí knihovny random vybere loď náhodně.

1. **Menu**

Hlavní menu je zapsáno ve funkci, která v sobě spojuje všechny ostatní menu a také samotnou hru.

# Uživatelská dokumentace

## První spuštění

Hru lze stáhnout jako archiv, po jehož rozbalení je možné hru ihned spustit. Spustitelný soubor se jmenuje Dlouhodobka.exe a nachází se v hlavní složce. Po otevření hry se spustí Menu.

Obrázek : Hlavní menu



Zdroj: Vlastní zpracování

## Hlavní menu

Hlavní menu se větví na čtyři části. Pohybovat se mezi nimi lze pomocí šipek nahoru a dolů, potvrdit volbu pak lze pomocí mezerníku. První možný výběr „Play“ spustí samotnou hru. Druhá možnost „options“ otevře další menu, ve kterém si hráči mohou zvolit svou loď. Třetí možnost „controls“ otevře další okno s ovládáním hry.

Obrázek : Controls



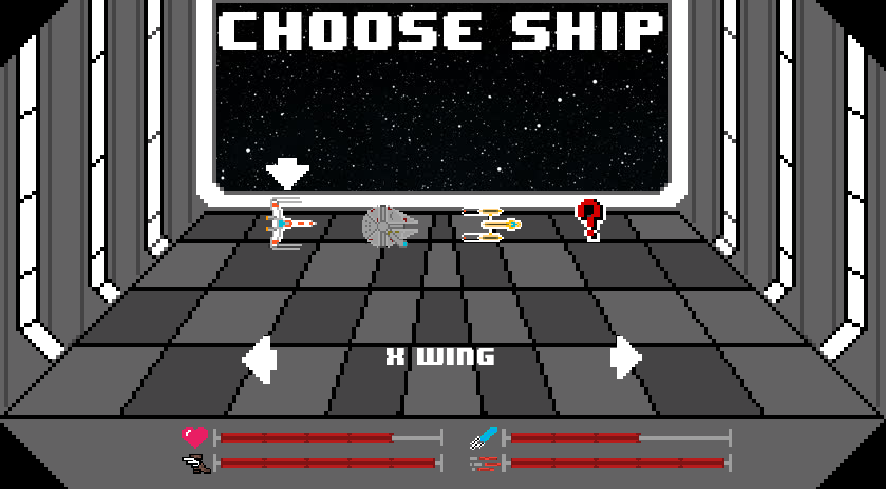
Zdroj: Vlastní zpracování

Poslední možnost „Quit“ program vypne.

## Options

V tomto menu jsou dvě možnosti, kam dále jít - „rebellion“, nebo „empire“. Po výběru jedné z možností se otevře menu pro výběr lodě (v rebellion budeme volit loď pro hráče jedna, v empire pro hráče dva). Mezi jednotlivými loděmi se lze pohybovat pomocí šipek doprava a doleva. Pod právě zvolenou lodí se zobrazí tabulka s hodnotami, která nám ukazuje, jak se lodě budou chovat. Každá loď má totiž jiné vlastnosti. Některé lodě jsou rychlé, ale jejich střely nejsou tak silné. Jiné lodě jsou naopak velmi pomalé, zato májí více životů.

Obrázek : výběr lodí



Zdroj: Vlastní zpracování

## Play

Hra je určena pro dva hráče. První hráč ovládající loď nalevo (rebellion), používá k jejímu pohybu klávesy WASD a pro výstřel levou klávesu CTRL. Druhý hráč, ovládající loď napravo (empire), k pohybu používá šipky a pro výstřel pravý CTRL. Cílem hry je zničit nepřátelskou loď, čehož lze dosáhnout tím, že protihráčovy životy snížím na 0. Pokud se tohle jednomu z hráčů povede, poražená loď exploduje a objeví se text oznamující vítěze. Hra se po chvíli vrátí zpět do hlavního menu.

# Závěr

Použít pygame se ukázalo jako dobrá volba. Modul poskytl pro programování hry hodnotné funkce. Mechaniky a logika hry byli realizovány v plném rozsahu. Při tvorbě menu se ale pygame příliš neosvědčil. Pygame není určen na tvorbu menu do her, možná by byl vhodnější jiný modul jako je například Tkinter. Tvorba grafického prostředí ve stylu pixel art se ukázala jako limitující, ovšem to není nutně špatné, protože bylo potřeba řešit problémy textur více kreativně.

# Seznam použité literatury

[1] Nauč se python. *Nauč se python* [online]. [cit. 2022-02-02].

Dostupné z: <https://naucse.python.cz/course/pyladies/>

[2] Python documentation. *Python documentation* [online]. [cit. 2022-02-02]. Dostupné z: <https://docs.python.org/3.9/tutorial/index.html>

[3] Pygame documentation. *Pygame documentation* [online]. [cit. 2022-02-02]. Dostupné z: <https://www.pygame.org/docs/>

# Seznam obrázků

[Obrázek 1: Prostředí Pixilart 8](#_Toc94722289)

[Obrázek 2: Tie Advanced, pohled shora 10](#_Toc94722290)

[Obrázek 3: Zjednodušení 10](#_Toc94722291)

[Obrázek 4: Obrys 11](#_Toc94722292)

[Obrázek 5:Tie Advanced Finální verze textury 11](#_Toc94722293)

[Obrázek 6: Tie Fighter - Předloha 12](#_Toc94722294)

[Obrázek 7: Tie Fighter- Textura 12](#_Toc94722295)

[Obrázek 8: Tie Reaper - Textura 12](#_Toc94722296)

[Obrázek 9: Tie Reaper - Textura 12](#_Toc94722297)

[Obrázek 10: Xwing - Předloha 13](#_Toc94722298)

[Obrázek 11: Xwing - Textura 13](#_Toc94722299)

[Obrázek 12: Milenium Falcon - Předloha 13](#_Toc94722300)

[Obrázek 13: Milenium Falcon - Textura 14](#_Toc94722301)

[Obrázek 14: Ywing - Předloha 14](#_Toc94722302)

[Obrázek 15: Ywing - Textura 14](#_Toc94722303)

[Obrázek 16: SPACE 15](#_Toc94722304)

[Obrázek 17: HANGAR 15](#_Toc94722305)

[Obrázek 18: Výherní obrazovka 20](#_Toc94722306)

[Obrázek 19: blokový diagram menu 21](#_Toc94722307)

[Obrázek 20: Indikátor v menu 22](#_Toc94722308)

[Obrázek 21: Options 23](#_Toc94722309)

[Obrázek 22: Rebellion výběr lodí 23](#_Toc94722310)

[Obrázek 23: Empire výběr lodí 24](#_Toc94722311)

[Obrázek 24: Hlavní menu 25](#_Toc94722312)

[Obrázek 25: Controls 26](#_Toc94722313)

[Obrázek 26: výběr lodí 27](#_Toc94722314)

## Přílohy

VLOŽIT POSTER