

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

Abstrakt je stručný výtah z maturitní práce. Charakterizuje obsah dokumentu bez rozlišování autorství abstraktu, bez doplňkových informací, bez vlastní interpretace a hodnocení dokumentu (tj. nikoliv "v práci velmi dobře hodnotím podle mne zajímavý systém...", ale "práce hodnotí systém..."). Základními vlastnostmi abstraktu jsou výstižnost, přehlednost, jasnost, stručnost, přesnost, objektivnost a čtivost. Slouží především jako pomoc čtenáři rychle se zorientovat v dané práci, získat přehled o obsahu a výsledcích práce. Je psán v českém jazyce a jeho rozsah je 200 až 500, maximálně jedna strana A4.

Stavba abstraktu je následující:

1. Na začátku abstraktu byste měli čtenáře jednou či dvěma větami uvést do tématu, kterým jste se zabývali. Je možné zmínit také to, proč je dané téma aktuální.

2. Dalším bodem by měl být cíl práce, který jste si stanovili v zadání, respektive v úvodu práce.

3. Třetím bodem abstraktu by měly být použité metody, tedy způsob, jakým bylo cíle dosaženo. Metodu stačí pouze uvést, nepopisujte ji. Pokud máte práci rozdělenou na teoretickou a praktickou část, můžete u každé krátce konstatovat, čím se zabývá.

4. Nejdůležitějším bodem abstraktu je závěr, kde uvedete svá nejdůležitější zjištění, tedy co je přínosem vaší práce. Nakonec je možné nastínit, jaká je využitelnost vašich zjištění, eventuálně další možnou problematiku zkoumání.

Obsah

Obsah se vkládá až po napsání celé práce pomocí menu Odkazy / Obsah. Při jakékoliv změně nadpisů v práci je třeba obsah aktualizovat – pravé tlačítko / Aktualizovat pole / celá tabulka.

[Obsah 5](#_Toc525402557)

[Úvod 6](#_Toc525402558)

[1. Kapitola první 6](#_Toc525402559)

[1.1 Ilustrace 6](#_Toc525402560)

[1.2 Tabulky 6](#_Toc525402561)

[1.3 Vzorce 6](#_Toc525402562)

[1.4 Citace 6](#_Toc525402563)

[1.4.1 Příklad citace z knihy 6](#_Toc525402564)

[1.4.2 Příklad citace z www stránky 6](#_Toc525402565)

[2. Pokyny pro psaní textu 6](#_Toc525402566)

[2.1 Nejčastější nedostatky 6](#_Toc525402567)

[2.2 Druhá podkapitola 6](#_Toc525402568)

[2.3 Třetí podkapitola 6](#_Toc525402569)

[Závěr 6](#_Toc525402570)

[Seznam použité literatury 6](#_Toc525402571)

[Seznam obrázků a tabulek 6](#_Toc525402572)

Pozor do obsahu nezahrnujte abstrakt.

Úvod

Cílem je naprogramovat 2D hru pro 2 hráče za použití jazyka python a příslušné knihovny pygame. Hra bude fungovat takto: na levé a pravé straně obrazovky budou dvě vesmírné lodě, každou z nich bude ovládat příslušný hráč klávesami WSAD a nebo šipkami. Úkolem obou hráčů je střílet na nepřátelskou loď, dokud její životy nedosáhnou nuly. Ten hráč který tohoto cíle dosáhne jako první vyhrál. Hra bude dále mít i menu výběru vesmírné lodě. Každý hráč bude mít tři příslušné lodě mezi kterými si může vybrat.

Další část bude tvorba grafického rozhraní jak pro menu, tak pro hru samotnou. Vše bude ve stylu retro videoher tzv. pixel art. Pro tvorbu grafiky jsem zvolil webovou aplikaci Pixilart.

V úvodu práce autor seznamuje čtenáře s obsahem práce a její strukturou. Při psaní se vychází ze zadání práce. Úvod by měl vždy obsahovat cíl práce. Zdůrazní se co práce obsahuje, možno uvést i strukturu práce (tj. stavbu kapitol a co každá kapitola obsahuje). Může se uvést důvody, proč zvolil dané téma.

V úvodu by neměly být použity citace, obrázky, grafy apod. Úvod je vždy psán před započetím tvorby práce tj. v budoucím čase.

# 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 programátorům 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

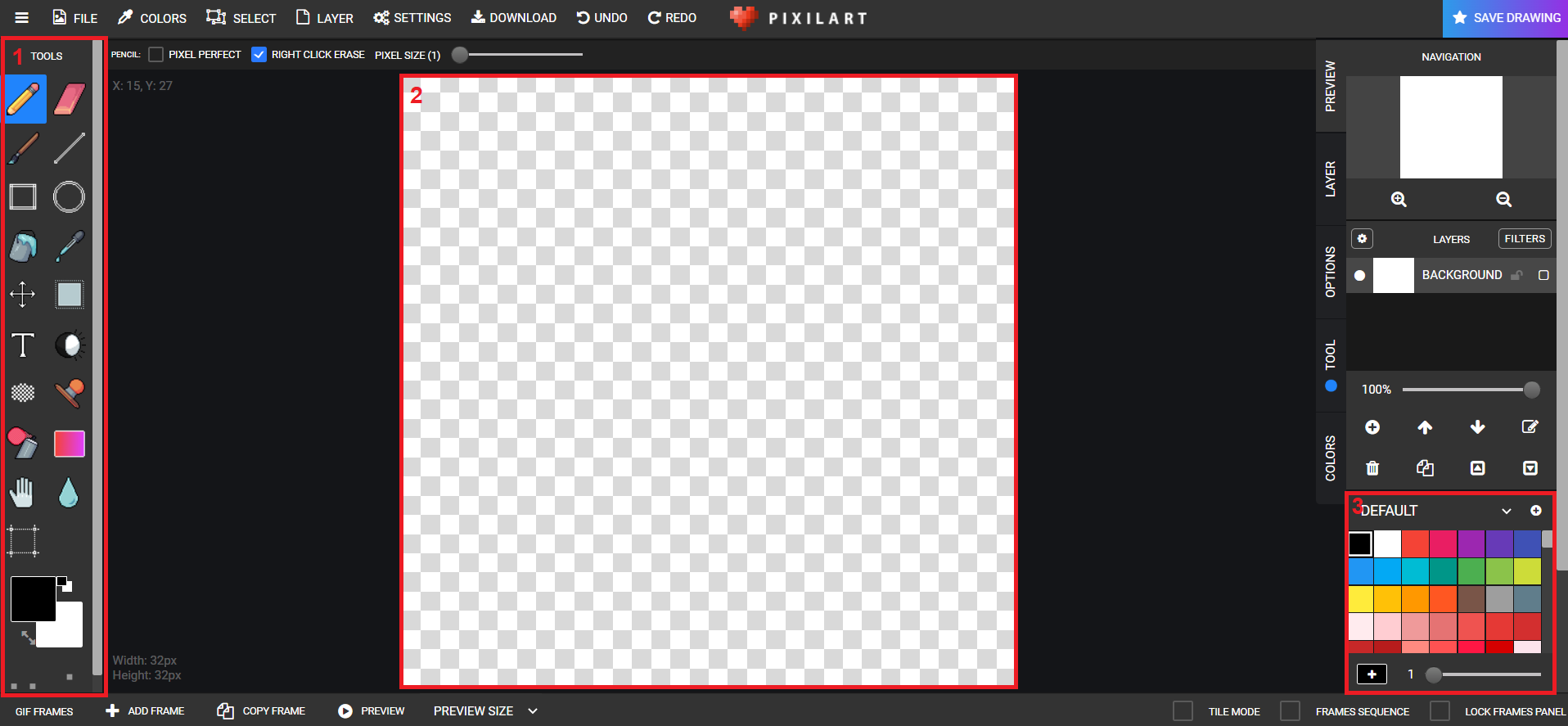
Os neboli operační systém. Knihovna, která umožňuje programu práci s funkcemi operačního systému. Důležitá pro nás bude manipulace s obrázky kterou tento modul zastane. Pygame umí zastat i tuto funkci, ale pouze pro některé typy souborů.Proto je důležité použít při importu obrázku tuto knihovnu. Ta nám 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 podobným nástrojům z malování, Photoshopu nebo jiného grafického editoru. [2] Vprostřed je samotný canvas na který jde kreslit. [3] Vlevo je vyber barev moznost 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 zaceňme 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, a tedy modely lodí budeme vytvářet za pomoci fotek ze samotných filmu. 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, tedy vzít předlohu a nakreslit ji méně detailní, pak vzít méně detailní kopii a překreslit ji s ještě méně detaily, a takhle opakovat, než dosáhneme kýženého výsledku
2. Kreslit přímo zjednodušený obrázek, a ten samotný upravovat, než vypadá dostatečně dobře. To, co vypadá dobře ponechat, co ne změnit.

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

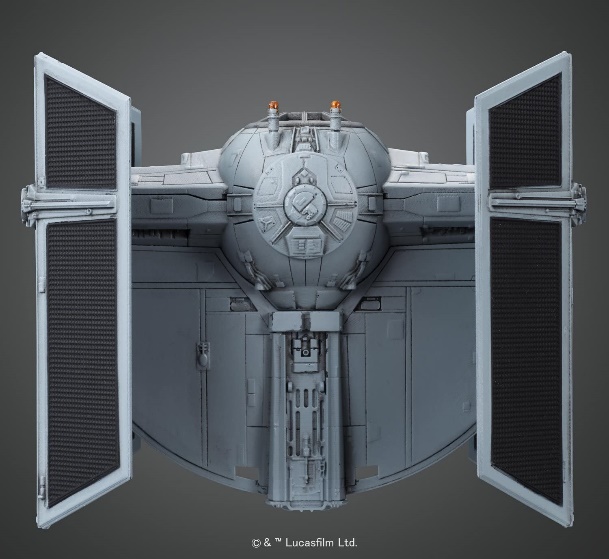
## Vesmírné Lodě

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 grafiky do hry.

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

1. Zde máme nezjednodušený model.

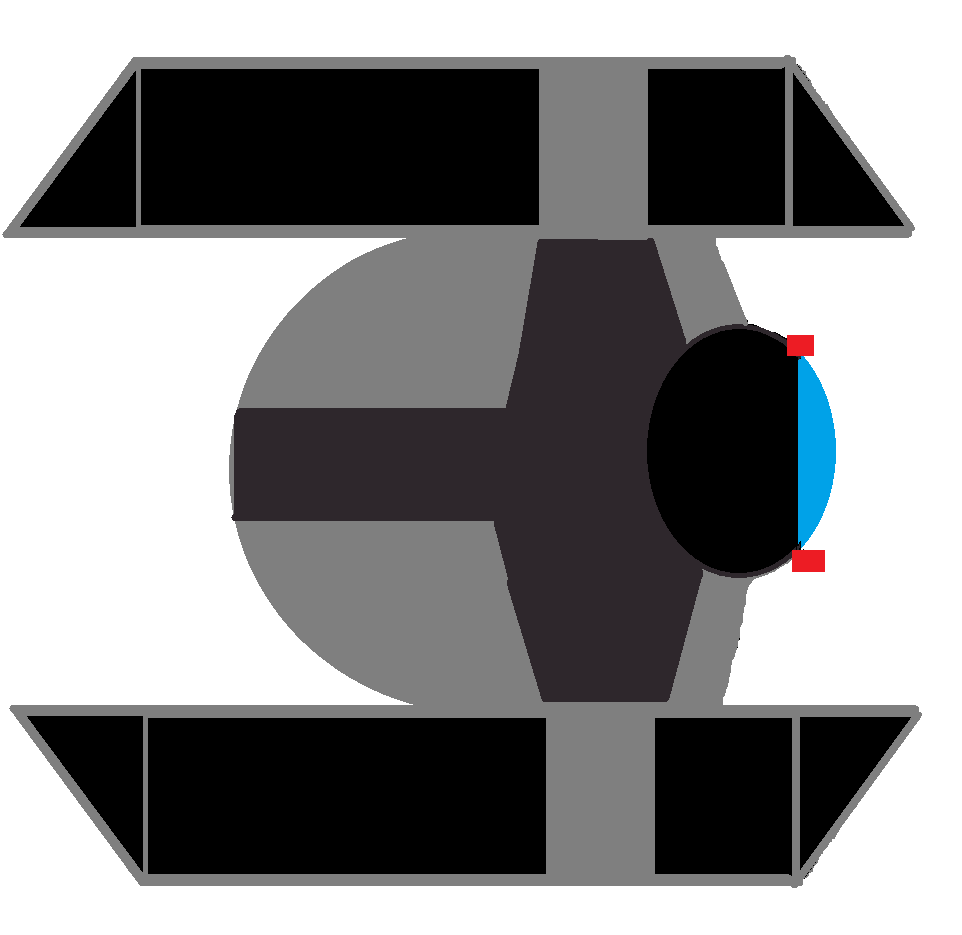
Obrázek : Tie Advanced, pohled shora



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

1. V druhém bodě se zbavíme všech přebytečných detailu, které nepůjdou vidět v menším formátu. Ještě nepřecházíme k pixel art. K tomuto kroku byl použit program malování.

Obrázek : Zjednodušení

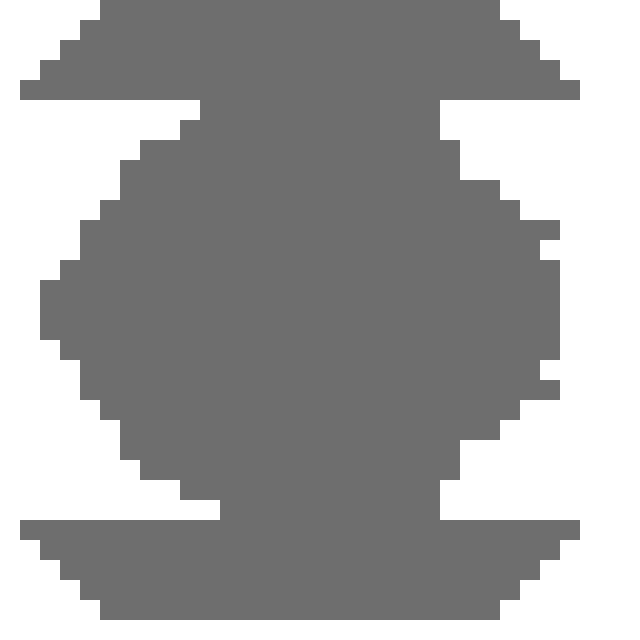


Zdroj: Vlastní zpracování

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.

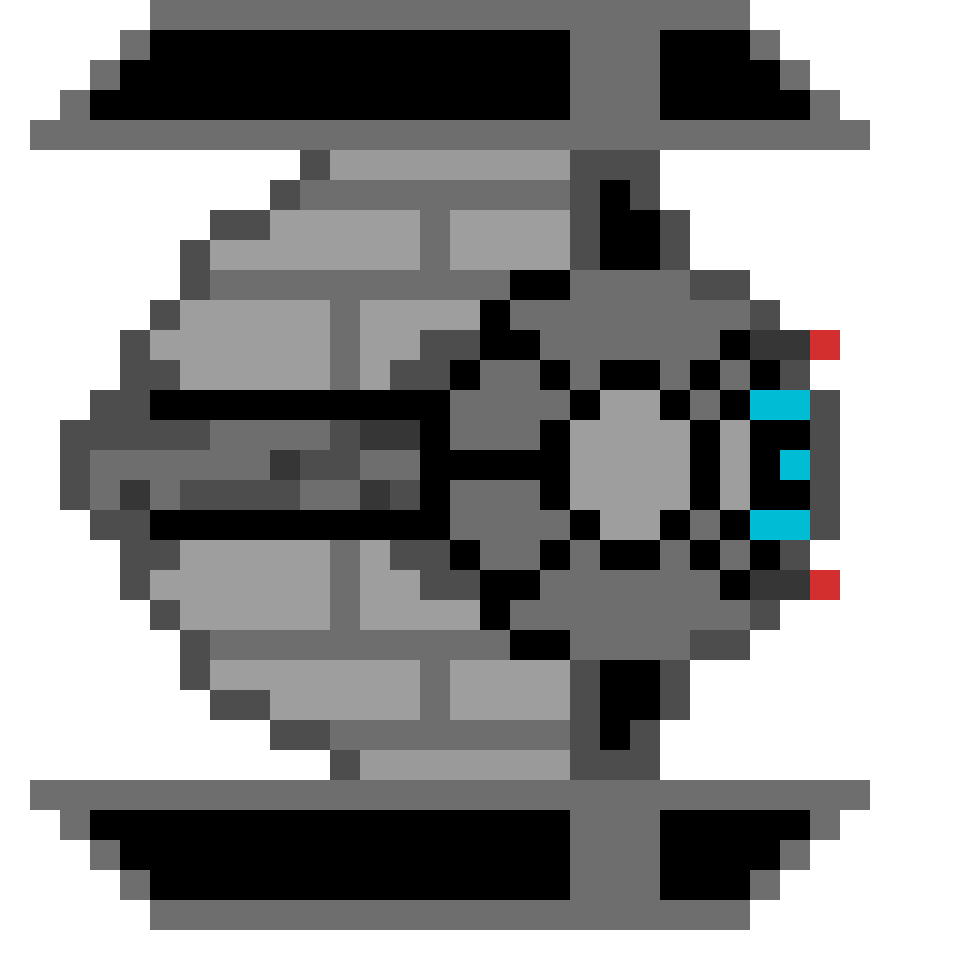
Obrázek : Obrys



Zdroj: Vlastní zpracování

1. Doplníme detaily a přidáme více barev.

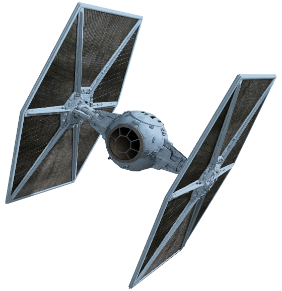
Obrázek :Tie Advanced Finální verze textury



Zdroj: Vlastní zpracování

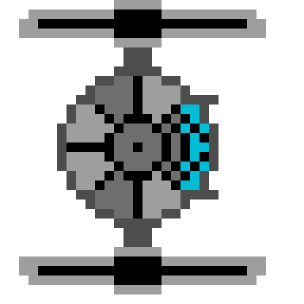
Po překreslení do programu Pixilart vypadá obrázek nějak takto. Touto metodou byly vytvořeny všechny textury pro lodě. Tady je jejich porovnání předloha/výsledek:

Obrázek : Tie Fighter - Předloha



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

Obrázek : Tie Fighter- Textura



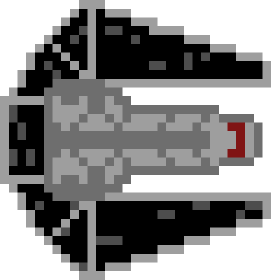
Zdroj: Vlastní zpracování

Obrázek : Tie Reaper - Textura



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

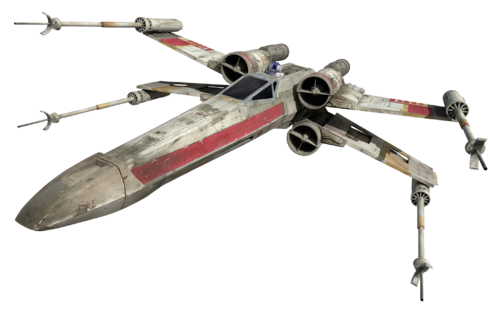
Obrázek : Tie Reaper - Textura



Zdroj: Vlastní zpracování

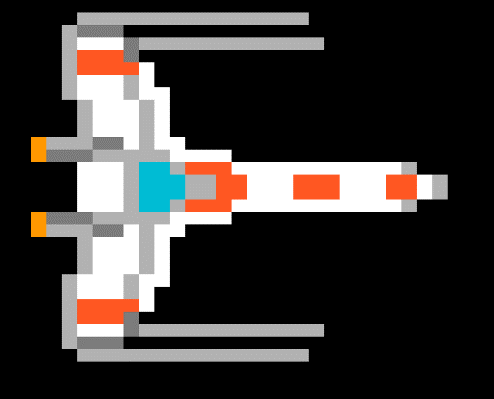
Toto jsou lodě pro prvního hráče. Lodě pro druhého hráče jsou následující:

Obrázek : Xwing - Předloha



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

Obrázek : Xwing - Textura



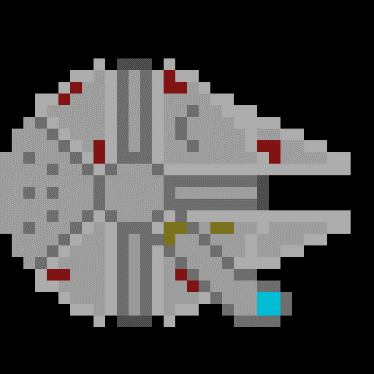
Zdroj: Vlastní zpracování

Obrázek : Milenium Falcon - Předloha



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

Obrázek : Milenium Falcon - Textura

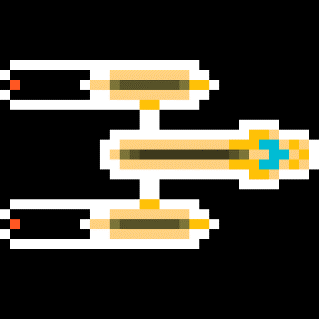


Obrázek : Ywing - Předloha



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:

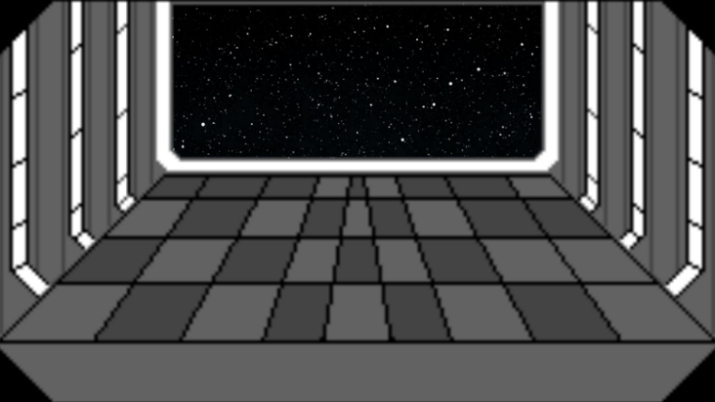
Obrázek : SPACE



Zdroj: Vlastní zpracování

Hangár potom vypadá takto:

Obrázek : HANGAR



Zdroj: Vlastní zpracování

V jeho středu byl použit znovu 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 něm 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 volat funkci game 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 začátek, 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, jeho 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 na druhý, viz obrázek č.19.

### Pohyb obrázků, vstupy od uživatele

Jsou 2 hlavní způsoby, jak zaznamenat vstupy uživatele. První je 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 ni 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()

Druhý 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 proměnných.

## Třídy a funkce

### Třídy

1. **Tie, Xwing**

Tyto třídy se starají o chování lodí Tie a Xwing. Jsou si v mnoha ohledech podobné, 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 ve 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()

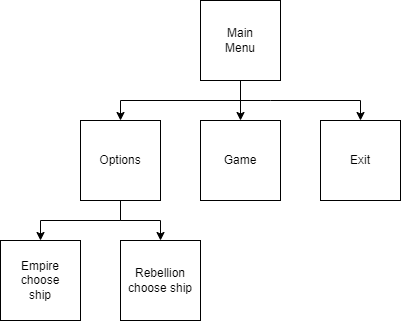
    sys.exit()

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 je 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 budeme 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 a výšce a pořadníku. V samotné funkci jsou definovaný 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/rebellion a podle toho se spustí funkce choose\_ship. Dále je tu možnost back. Ta program vrátí do hlavního menu.

Obrázek : Options

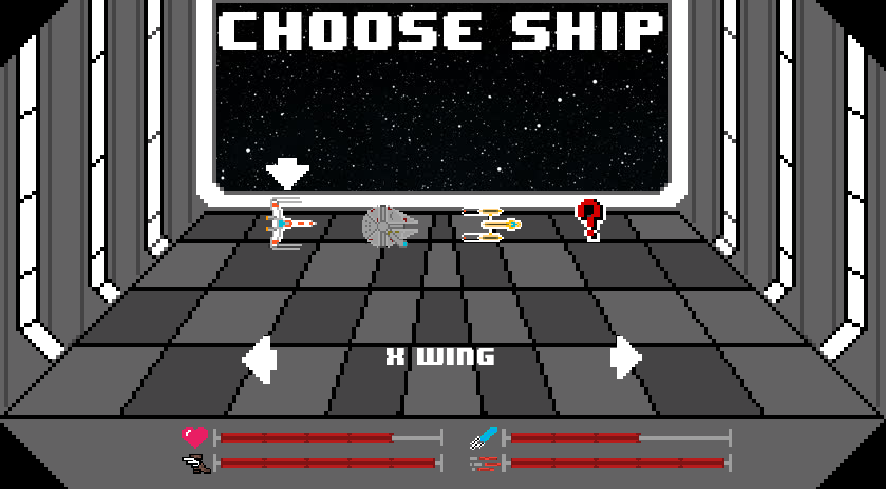


Zdroj: Vlastní zpracování

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 rebellion vypadá takto:

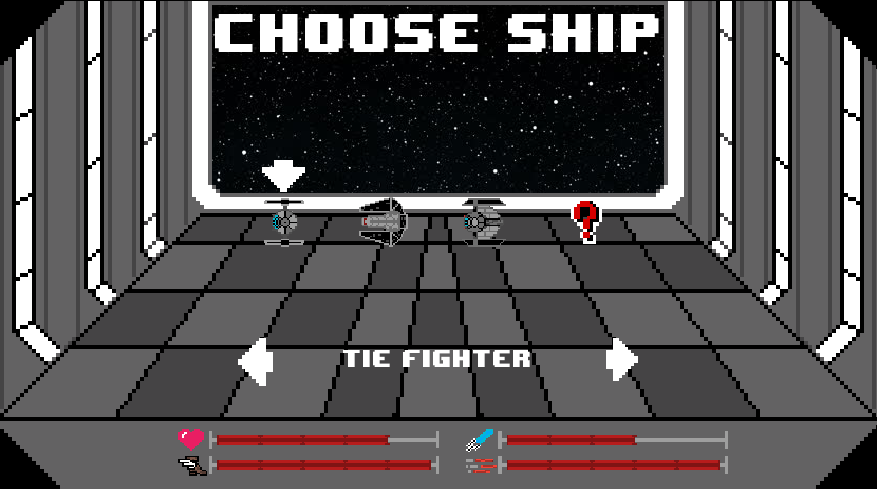
Obrázek : Rebellion výběr lodí



Zdroj: Vlastní zpracování

V případě empire vypadá takto:

Obrázek : Empire výběr lodí



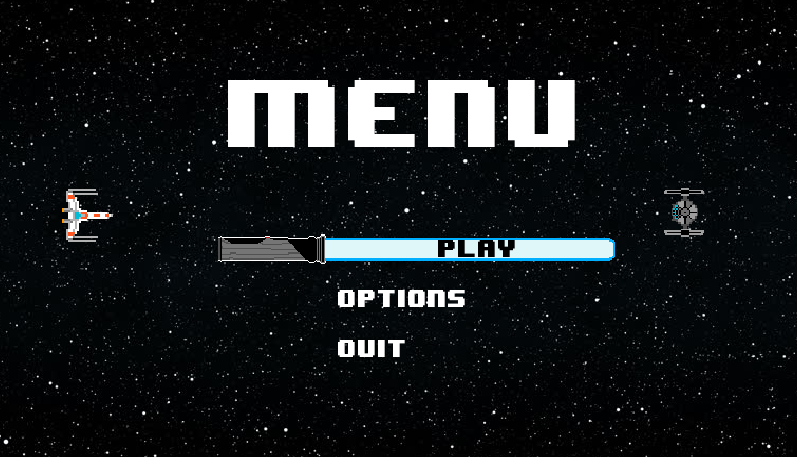
Zdroj: Vlastní zpracování

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 vybrat 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.

Obrázek : hlavní menu



Zdroj: Vlastní zpracování

# Uživatelská dokumentace