

Dokumentace k zápočtovému programu

Jakub Levý

červen/červenec 2018

Abstrakt

Tento program reimplementuje hru Space Invaders. Je napsán v jazyce C#, využívá knihovnu WinForms a GDI+. Hraní hry je doprovázeno původními zvukovými efekty a animacemi při eliminaci nepřátel. Součástí programu je několik cheatů, pomocí kterých je možné jednodušeji testovat hru.

Obsah

1	O Space Invaders	3
2	Využité technologie	3
2.1	C#	3
2.2	WinForms	3
2.3	Visual Studio 2017	3
2.4	GDI+	3
2.5	Adobe Photoshop	4
3	Program	4
3.1	Uživatelská část	4
3.1.1	Ovládání	4
3.1.2	Modely postav	5
3.1.3	Konec hry	7
3.1.4	Pochod invaze	7
3.1.5	Úrovně	7
3.1.6	Bariéry	8
3.1.7	Střelba	8
3.1.8	Autentičnost	8
3.1.9	Responzivita	9
3.1.10	Zvukové efekty	10
3.2	Programátorská část	11
3.2.1	Struktura projektu	11
3.2.2	Adresáře projektu	11
3.2.3	Game.cs	12
3.2.4	Form1.cs	17
3.2.5	ScoreTableUC.cs	18
3.2.6	Invader.cs	18
3.2.7	Barrier.cs	19
3.2.8	Constants.cs	19
3.2.9	ExtensionMethods.cs	19
3.2.10	Utils.cs	20
3.2.11	Další soubory	20
4	Závěr	21

1 O Space Invaders

Jedná se o arkádovou hru vytvořenou Tomohirem Nishikadem. Hra byla uvolněna již v roce 1978. Hráč pohybuje tankem s laserem a jeho cílem je eliminovat nepřátelskou invazi blížící se k tanku. Vzhledem ke stáří hry se jedná o jednu z prvních stříleček.



(a) původní hra



(b) moje verze

Obrázek 1: porovnání originální hry s mojí verzí

2 Využité technologie

Program, které se tato dokumentace týká, by nemohl vzniknout bez různých technologií, o kterých se rozepíšu v této kapitole.

2.1 C#

Celý program byl napsán v programovacím jazyku C#. Tento programovací jazyk jsem zvolil z důvodu jeho jednoduchosti, výkonnosti a možnosti objektově orientovaného programování.

2.2 WinForms

Pro jednoduchou práci s tlačítky jsem se rozhodl použít grafické rozhraní. Ačkoliv knihovna WPF je modernější, zvolil jsem knihovnu WinForms ze dvou důvodů:

1. kompatibilita s jinými operačními systémy než Windows,
2. jednoduchá tvorba uživatelského rozhraní.

2.3 Visual Studio 2017

U volby vývojového prostředí jsem neváhal. Konkurenční JetBrains Rider a MonoDevelop neobsahují designer pro tvorbu uživatelského prostředí aplikací využívajících knihovnu WinForms.

2.4 GDI+

Kvůli jednoduchosti se o vykreslování stará součást API OS Windows. Pro účely tohoto programu je dostačující.

2.5 Adobe Photoshop

Už jenom tento dokument obsahuje 24 obrázků. Několik dalších je součástí programu. Často bylo potřeba udělat přesný výběr objektu, odstranit pozadí a přidat nějaký prvek do obrázku. Jedná se sice o základní funkcionality, ale samotná možnost rozdělit si části obrázku do vrstev mi značně ulehčila práci.

3 Program

V této kapitole popíšu jednotlivé části programu ze dvou různých pohledů. Nejprve z pohledu uživatelského, pokračovat budu programátorskou částí, kde rozeberu vybrané bloky kódu.

3.1 Uživatelská část

V této části se, jak již název podkapitoly napovídá, zaměřím na části programu z takového pohledu, že i obyčejnému uživateli mohou přijít informace získané přečtením zajímavé a poučné, nacházejí se zde i informace, které se dají považovat za nutné ke hraní hry.

3.1.1 Ovládání

Pro ovládání tanku nám stačí 3 tlačítka viz následující tabulka.

pohyb tanku doleva	←
pohyb tanku doprava	→
střelba	mezerník, alt. x

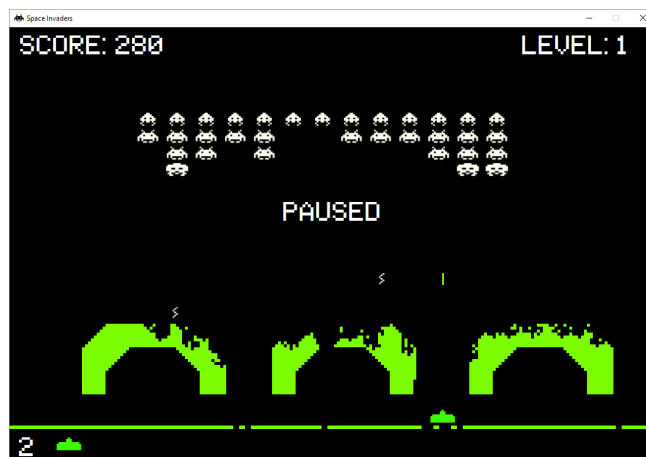
Tabulka 1: tlačítka ovládající tank

Cheaty Hra obsahuje celkem 5 podvodů, které nám pomůžou k poražení nepřátel. Opět jejich přehled v tabulce.

<i>popis cheatu</i>	<i>od slova</i>	<i>klávesová zkratka</i>
přidání jednoho života	health	h
regenerace bariéry	barrier	b
zakázání/povolení střelení nepřátel	missile	m
zničení náhodného nepřítele	delete	d
zvýšení skóre o 100	score	s

Tabulka 2: seznam cheatů

Pauza pokud okno hry ztratí *focus*, nebo hráč stiskne klávesu **p**, hra se pozastaví. Toto platí i při minimalizaci okna, což je pouze speciální případ.

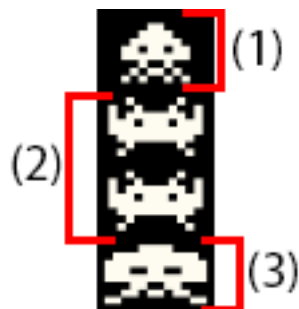


Obrázek 2: pozastavená hra

3.1.2 Modely postav

V následujícím textu budu rozumět pojmem *model* grafiku hry, která je vykreslená z předpřipraveného .png obrázku.

Nepřátelská invaze je složena z několika různých modelů.



Obrázek 3: sloupec nepřátelské invaze

(1) TopInvader – vzhledem k jeho menšímu hitboxu je za jeho eliminaci 30 bodů.

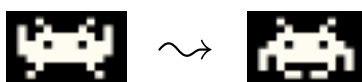
(2) MiddleInvader – těch je nejvíce, za eliminaci je 20 bodů.

(3) BottomInvader je první na ráně a hráč za něj získá 10 bodů.

Každý z těchto tří prochází při pochodu dvěma stavy pohybu.



Obrázek 4: stavy TopInvadera



Obrázek 5: stavy MiddleInvadera



Obrázek 6: stavy BottomInvadera

Tito střílí po našem tanku:



Obrázek 7: nepřátelské střelivo

Posledním speciálním členem nepřátelské invaze je tzv. mystery. Ten s náhodnou rychlostí 3 až 10 px za 15ms, v náhodném čase z rozmezí 10 až 20s prosvíští vrchní částí herní plochy. Podle jeho rychlosti jsou za jeho eliminaci přiděleny body.



Obrázek 8: mystery

Mystery jako jediný člen invaze neprochází mezi dvěma stavy a nestřílí. Pouze se plynule pohybuje.

Pokud se z tanku laserem trefíme do invadera, který umí střílet zahraje znělka [16]. Při zásahu mysteryho zahraje [17]. Mimo zvukového efektu se na pozici rozstřeleného nepřítele objeví buďto:



Obrázek 9: eliminace invadera

nebo



Obrázek 10: eliminace mysteryho

v případě eliminace mysteryho.

Nyní se zaměřím na modely týkající se hráčova tanku.



(a) tank



(b) tank zasažen

Obrázek 11: tank a jeho ruiny

Při zasažení tanku se v případě, že máme ještě rezervní život, spustí příslušná zvuková hláška [19] a po dobu 300ms se zobrazí jeho ruiny (b). Následně se objeví na náhodné pozici nový

tank, který bude po dobu 3s žlutý a imunní vůči nepřátelským střelám. Po uplynutí doby, kdy je tank neztříšitelný, se jeho barva změní opět na zelenou.



Obrázek 12: imunní tank

V opačném případě, kdy už život nemáme, se zobrazí konec hry.

Modely jsem vystříhal z různých variant her Space Invaders a upravil pomocí Adobe Photoshopu.

3.1.3 Konec hry

Hra může skončit ze dvou důvodů:

1. tank byl zasažen a nezbývá už žádný rezervní život,
2. invaderi se dostali na pozici tanku.



Obrázek 13: GAME OVER!

3.1.4 Pochod invaze

Invaze vždy začíná na levé straně herní plochy a pochoduje směrem k pravé straně do té doby, než dosáhne pravého okraje. Následně invaze sestoupí blíže k tanku, zvýší svoji rychlost pochodu a obrátí směr pochodu na levou stranu.

Invaze pochoduje periodicky a to po uplynutí času na časovači.

3.1.5 Úrovně

Hra začíná první úrovní. Při eliminaci všech střilecích nepřátel hra postoupí do další úrovně. Všechny úrovně mají stejné rozestavení nepřátel z invaze. Úrovně hry se liší pouze v následujícím:

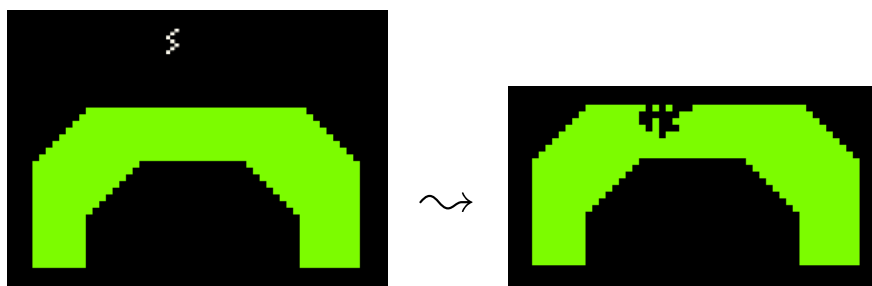
- invaze nepřátel z vyšší úrovně začíná na pozici blíže k tanku,
- invaze z vyšší úrovně sestupuje směrem dolů, blíže k tanku o větší vzdálenost,
- s každým přiblížením k tanku se zrychlí pochod celé invaze.
(čím vyšší úroveň, tím vyšší zrychlení)

3.1.6 Bariéry

Během vytváření nové úrovně hry se vygenerují 3 bariéry. Tyto bariéry jsou vytvořeny a vykreslovány po sloupcích ze čtverců o rozměru 5×5 px.

Nyní bych rád popsal, jak funguje plošné poškození bariéry.

Pokud dojde ke kolizi bariéry a nějaké střely, ať už nepřátelské, nebo naší pocházející z tanku, tak se nejprve zjistí kolikátý sloupec bariéry byl zasažen. V tomto sloupci se zničí všechny čtverečky, jejichž vzdálenost je menší než `SplashRadius`. Ve všech ostatních sloupcích se s pravděpodobností 70% zničí čtverečky, jejichž vzdálenost je menší než `SplashRadius` a s 30% pravděpodobností zůstanou zachovány. Tímto vznikne efekt plošného poškození.



Obrázek 14: před plošným poškozením a po

Pokud se nepřátelé dostanou na hranici bariéry, všechny bariéry jsou instantně zničeny, toto zničení je avizováno hláškou [21].

3.1.7 Střelba

Invadeři, jak již bylo několikrát zmíněno, střílejí po tanku. Nyní bych chtěl popsat jak funguje proces výběru invadera, který bude střílet.

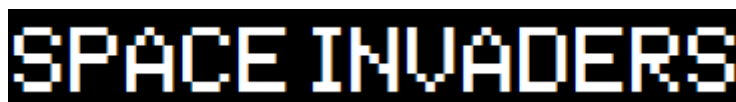
Invadeři střílejí po uplynutí času na příslušném časovači. Při vytvoření nové úrovně hry se interval časovače nastaví na 1s. Při eliminaci invadera se tento čas snižuje o 5ms. Pokud invadeři sestoupí blíže k tanku, tento interval se sníží o 30ms až do minima 420ms.

Vždy střílí 1 až 3 invadeři v závislosti na náhodném výběru. S 30% pravděpodobností se vybere invader, který má šanci zasáhnout tank, tj. invader, jehož souřadnice Y je nejbližší k souřadnici Y tanku. Zbýlá 70% pravděpodobnost pokrývá náhodný výběr invadera, který bude střílet.

Pokud zbývá už pouze poslední střílecí invader nastane speciální případ. Interval na časovači určující rychlost střelby se na nastaví na 350ms a interval na časovači určující rychlost pochodu na 100ms. Invader bude velmi rychle střílet a bude obtížné ho zasáhnout.

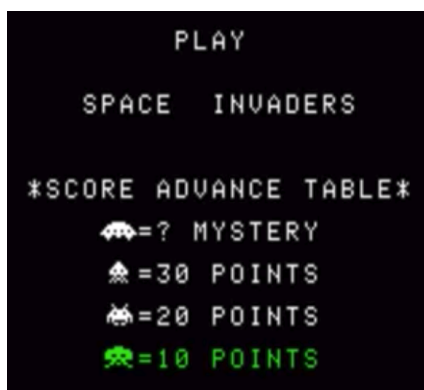
3.1.8 Autentičnost

Vzhledem k tomu, že Space Invaders je už 40 let stará hra, snažil jsem se mé verzi dát alespoň jakýsi nádech grafiky původní doby. Nejvýznamnější podíl na tom má písmo Space Invaders [1].

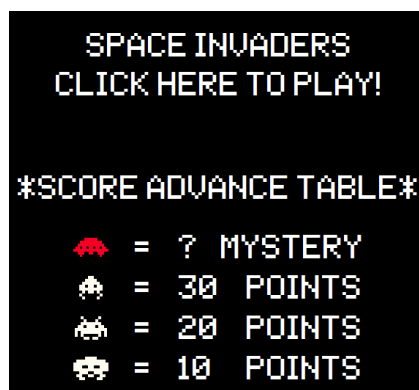


Obrázek 15: nápis Space Invaders v písmu Space Invaders

Dále jsem napodobil menu hry z původní verze z roku 1978.



(a) původní menu z roku 1978



(b) moje menu

Obrázek 16: porovnání původního menu z 1978 a mého menu

A reimplementoval jsem bariéry, které jsou plošně poškozovány při střelbě.



(a) původní bariéra



(b) moje bariéra

Obrázek 17: porovnání původní bariéry s mojí bariérou

3.1.9 Responzivita

Ačkoliv mi knihovna WinForms příliš nepomáhá, tak se hra přizpůsobí velikosti okna aplikace. Změnu velikosti okna není možné provést při spuštěné hře. Pro hraní hry v jiné velikosti okna než výchozí, musíme změnit velikost okna již v menu před započítím hry.



(a) hra s velkou výškou



(b) hra s velkou šířkou

Obrázek 18: porovnání hry s velkou délkou a šířkou

3.1.10 Zvukové efekty

Hra je doprovázena zvukovými efekty a hudbou.

Menu V hlavní nabídce hraje hudba [14] z roku 1980, která pochází z australského hudebního nočního vysílání Rage.

Další zvukové efekty:

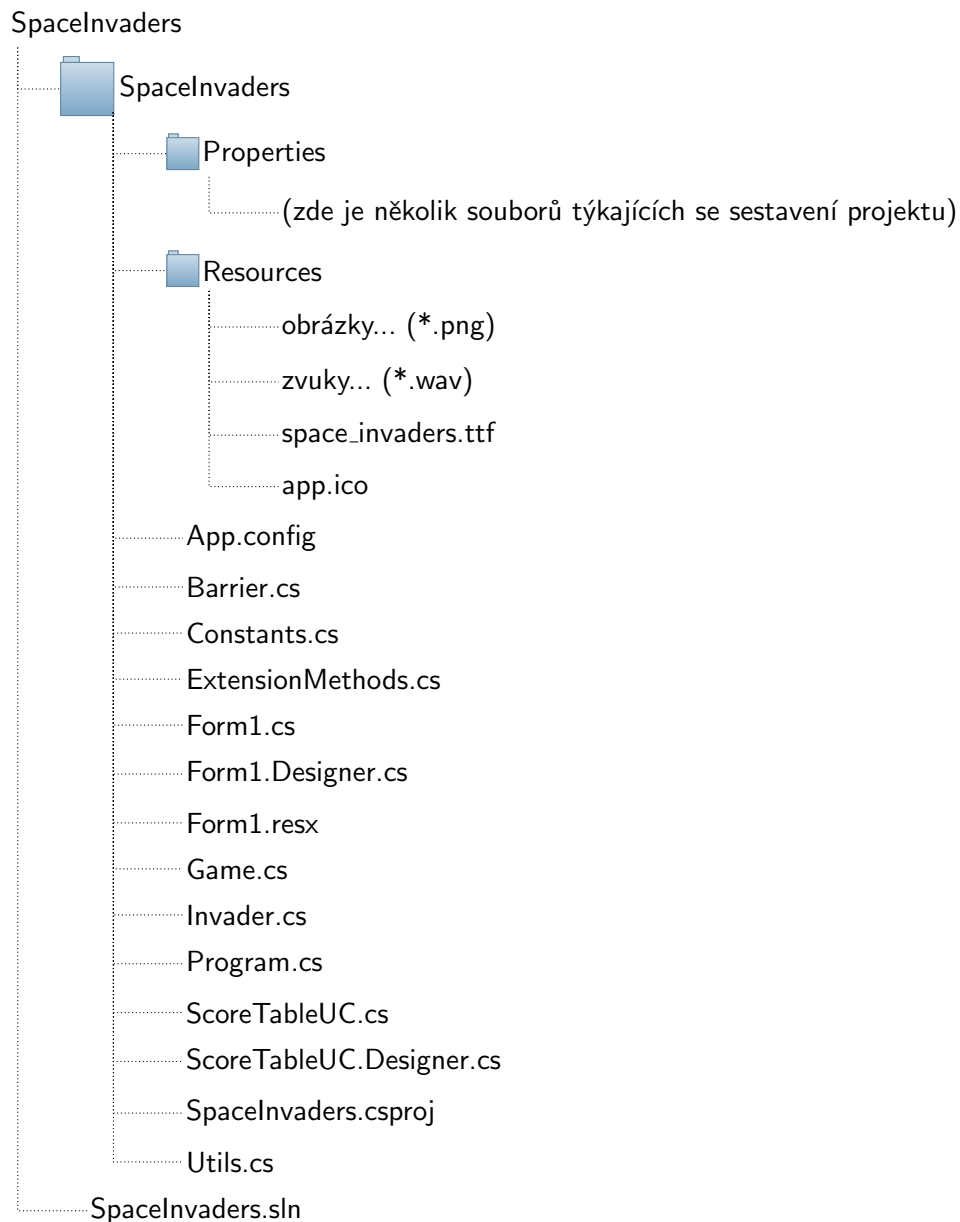
- výstřel z tanku [15],
- eliminace invadera [16],
- eliminace mysteryho [17],
- průlet mysteryho [18],
- zásah tanku [19],
- konec hry [20]

pochází z různých verzí hry Space Invaders. Zbylé zvukové efekty jsem přidal dle vlastního uvážení.

3.2 Programátorská část

Druhá část se podstatně liší od první, uživatelské části. Zde je popsána struktura projektu, jednotlivé soubory a bloky kódu, které implementují mechaniky hry. Čtenáři doporučuji, aby si nejdříve prošel část uživatelskou, neboť se tu budou často rozvíjet koncepty již částečně popsané v předchozí části.

3.2.1 Struktura projektu



3.2.2 Adresáře projektu

SpaceInvaders/Properties Zde se nachází několik souborů obsahujících verzi, název, jazyk,... výsledného spustitelného souboru hry.

SpaceInvaders/Resources Tato složka obsahuje soubory, které aplikace při svém běhu využívá. Při kompilaci projektu se tyto soubory přidají do spustitelného souboru hry.

Zbytek kapitoly 3 již bude obsahovat informace o souborech nacházejících se v adresáři **SpaceInvaders/**. Budu tedy předpokládat, že se v tomto adresáři pohybujeme a jeho název již budu vynechávat.

3.2.3 Game.cs

Součástí souboru je jádro celé hry, které obsahuje veškerou logiku herního módu. Tento soubor je jednoznačně nejdůležitějším a největším ze všech souborů projektu. Jeho popis je rozdělen do několika částí, z nichž v každé bude rozebráno několik důležitých a ne zcela jednoduše uchopitelných celků souboru.

V této části se vyskytuje několik seznamů, ať už proměnných nebo metod, jejichž pořadí je stejné jako jejich pořadí deklarace v programu.

Hlavní smyčka Každá hra potřebuje nějakou hlavní smyčku, ve které se skokově aktualizuje herní prostředí. V této hře to má na starosti časovač `drawingTimer` s nastaveným intervalem tiknutí na 15ms. Při tiknutí se zavolá metoda `drawingTimer.Elapsed`, kde se herní plocha překreslí a o odpovídající vzdálenost se posunou jednotlivé objekty. Dále se zde řeší veškeré kolize prvků, které mohou nastat.

Vlastnosti

- `Form1 Form { get; }` formulář aplikace nad kterým je spuštěna hra
- `int Life { get; private set; } = 5;` obsahuje zbývající počet životů
- `int Level { get... set... }` aktuální úroveň hry
- `int Score { get... set... }` současné skóre

Proměnné

- `List<PointF> tankMissiles` kolekce střel tanku, střela je uložena bodem s největší hodnotou souřadnice Y
- `List<PointF> explosions` obsahuje pozice padlých invaderů u kterých se ještě objevuje posmrtný efekt tj. obrázek 9
- `List<PointF> explosionsSafe` obdobné jako bod výše, jenom z této kolekce se posmrtný efekt vykresluje, aby nedošlo k synchronizačním problémům
- `List<PointF> invadersMissiles` součástí jsou střely pocházející od invaderů, uloženy jsou bodem s nejmenší hodnotou Y
- `List<PointF> invadersMissilesSafe` obdobné jako `explosionsSafe`
- `List<Invader> invaders` seznam všech střilecích invaderů
- `List<Barrier> barriers` kolekce bariér
- `List<RectangleF> bottomLine` jednotlivé obdélníčky tvořící spodní zelenou čáru
- `PointF? mysteryLocation` pozice mysteryho, null pokud žádný mystery není na herní ploše
- `PointF? mysteryExplosion` obdobné jako `explosions`, pouze pro mysteryho

- `PointF tankLocation` souřadnice tanku
- `Timer mysteryExplosionTimer` časovač, který tikne když by měl posmrtný efekt `mystery`ho zmizet
- `Timer drawingTimer` časovač hlavní smyčky
- `Timer mysteryShouldAppearTimer` tik nastane, když by se měl objevit nový `mystery`, interval tohoto časovače je nastaven jako `random.Next(10000, 20000)`
- `Timer invadersShouldMoveTimer` má na starost pochod invaze
- `Timer explosionsTimer` obdobné jako `mysteryExplosionTimer`, jenom pro střelící nepřátele z invaze
- `Timer invadersShouldShootTimer` při tiku vystřelí 1 až 3 invaderi v závislosti na náhodném generátoru
- `Timer tankExplosionTimer` tikne potom, co jsme 300ms viděli ruiny našeho předchozího tanku, umístí na náhodnou pozici tank nový a dá mu na 3s imunitu
- `Timer immunityTimer` tento časovač naopak tikne, když jsme si imunitu užívali 3s a sebere nám ji
- `Timer tankCanShootTimer` po výstřelu tank 1s nabíjí, časovač tikne, když tank už nabil a může opět střílet
- `bool canTankShoot` *flag* indikující zda tank může střílet, po výstřelu je nastaven na `false`, následně ho na `true` nastaví `tankCanShootTimer` po uplynutí doby nutné na nabití
- `bool keyLArrow, keyRArrow, keySpace` obsahují `true` pokud je daná klávesa stisknuta, v opačném případě `false`
- `bool invadersMovingRight` směr kterým pochoduje invaze
- `bool lockMovement` zámek na pohyb invaze, využívá se pokud invaze dojde na kraj herní plochy, před sestoupením invaze je totiž menší prodleva
- `bool lockingToken` udává pokud právě probíhá vykonávání kódu metody `drawingTimer_Elapsed`
- `bool shouldDrawTank` udává zda-li se má vykreslovat tank, nebo jeho ruiny
- `bool immunity` *flag*, v kterém je uloženo zda je imunita tanku aktivní
- `bool forcePause` zajišťuje funkci pauzy v případě pozastavení hry hráčem v okamžiku, kdy se právě zobrazovaly ruiny tanku
- `bool paused` byla hra pozastavena hráčem stisknutím klávesy `p`?
- `bool endGame` nastal konec hry?
- `bool enableExtraSpeed` indikuje aktivaci nastavení nejvyšší rychlosti pohybu invaderů, tato hodnota se nastavuje na `true`, pokud na hracím poli zbývá pouze jeden invader
- `bool enemyShootingDisabled` udává zapnutí/vypnutí cheatu, kdy nepřátelé přestanou střílet

- `bool barrierDestroyedPlayed` nastaven na `true`, pokud byla přehrána znělka [21], jinak je `false`
- `int mysterySpeed` rychlost prolétajícího `mysteryho`, jedná se o náhodně generovanou hodnotu x z intervalu 3 až 10 znamenající rychlost x px za 15ms
- `int movementAccelerationMs` při sestupu invaze se interval časovače `invadersShouldMoveTimer` sníží o hodnotu `movementAccelerationMs`, pochod se tedy zrychlí
- `int greenLineY` souřadnice Y spodní zelené čáry, její hodnota se spočítá na základě velikosti okna
- `double shootingPrecision` pravděpodobnost „inteligentního“ výběru střelce tj. výběr střelce, který má šanci zasáhnout tank
- `double invadersShouldMoveOriginalInterval` při sestupu invaze se na chvíli změní hodnota intervalu časovače `invadersShouldMove` pro docílení menší prodlevy při sestupování invaze, poté je třeba obnovit původní hodnotu intervalu časovače, ta se uloží do proměnné `invadersShouldMoveOriginalInterval`
- `float bottomInvaderY` souřadnice Y invadera s největší hodnotou této souřadnice, užitečné to je pro detekci, zda se už invadeři nedostali příliš blízko a nemělo by dojít k instantnímu zničení bariér
- `float barrierY` souřadnice Y bariéry, její hodnota je závislá na velikosti okna
- `float splashRadius` poloměr značící maximální vzdálenost od bodu střetu střely s bariérou a zbytkem bariéry, kde může dojít k jejímu poškození
- `StopWatch immunityStopwatch` pokud by hra byla pozastavena během imunity tanku, nikdy by nedošlo k tikům časovače `immunityTimer`, protože při pozastavení hry se všechny časovače zakazují, to znamená, že se nám znovu resetuje imunita na 3s. Jde o nežádoucí efekt a ten řeší tyto stopky, které zaznamenají již uplynulý čas imunity, interval `immunityTimeru` se o naměřený čas zkrátí
- `Random random` náhoda se vždy hodí, ať už se jedná o výběr střelce, rychlost `mysteryho` nebo místo, kde se po zasažení objeví nový tank

Konstrukce objektu Game Během vytváření instance třídy se v následujícím pořadí provede:

- inicializace časovače hlavní smyčky,
- registrace potřebných událostí `Form1`,
 - `Paint` vykresluje herní plochu
 - `Resize` pozastaví hru při minimalizaci okna
 - `KeyDown` nastaví stisk všech kláves na `true`
 - `KeyUp` nastaví stisk všech kláves na `false`
 - `Deactivate` pozastaví hru při ztrátě *focusu*
- vytvoření bariér,

- vytvoření spodní zelené čáry, tj. čára nacházející se těsně pod tankem



Obrázek 19: spodní zelená čára

- registrace událostí všech ostatních časovačů,
- spuštění 1. úrovně hry.

Uvolnění objektu Game Uvolnění objektu probíhá následovně:

1. zastaví se všechny časovače,
2. překreslí se okno hry,
3. zobrazí se konec hry,
4. uvolní se objekt **Game** zavoláním metody **Dispose**.

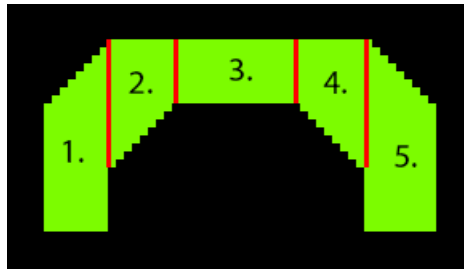
Veřejné metody Třída obsahuje pouze 2 veřejné metody a to sice:

- **CheatKey**, jenž je volána z přetížené metody **ProcessCmdKey**, metoda **CheatKey** zjistí zda byla stisknuta klávesa, která aktivuje cheat, a v případě, že ano, tak zavolá metody, které cheat aktivují,
- **End**, ta zastaví všechny časovače, zobrazí konec hry a uvolní objekt **Game**.

Soukromé metody Oproti veřejným metodám je soukromých o poznání více. Některé na vstupu dostávají více parametrů a nemusí být zcela jasné k čemu slouží. Všechny metody, kde hrozí zmatení ze vstupních parametrů, je mají popsány. Z popisu je vynechán návratový typ **void**, události formuláře a události časovačů které byly popsány v rámci popisů časovačů jako proměnných.

- **GenerateBarrier()** 3× zavolá metodu **CreateBarrier**, což odpovídá počtu bariér na herní ploše
- **PutTextIntoControl(int value, Label l)** bezpečně aktualizuje vlastnost WinForms komponent **Text** z jiného vlákna
- **PrepareForMovingDown()** zajišťuje menší prodlevu při sestupu invaze, dochází tu ke změně tiků časovače **invadersShouldMoveTimer** a k nastavení zámku pohybu **lockMovement**
- **MovingDown()** po zavolání metody **PrepareForMovingDown** se vykoná kód této metody, ta zajistí posun všech invaderů a úpravu intervalu časovačů
- **Invader RightInvader()** vrátí invadera z kolekce **invaders** s největší hodnotou souřadnice X
- **Invader LeftInvader()** oproti předchozímu bodu, tato metoda vrátí invadera s nejmenší hodnotou souřadnice X, tyto dvě metody společně se hodí k detekci zda-li se invaze už nedostala na kraj herní plochy
- **CreateBottomLine()** naplní proměnnou **bottomLine** obdélníčky tvořícími spodní zelenou čáru

- `CreateBarrier(PointF origin, float w, float h, int partN, ref Barrier b)`
naplní prázdnou bariéru předanou referencí obdélníčky
 1. parametr pozice levého horního rohu bariéry
 2. parametr šířka bariéry
 3. parametr výška bariéry
 4. parametr bariéra se vytváří po jednotlivých částech, tento parametr určuje část, která je právě vytvářena, standardně by tato metoda měla být volána s tímto parametrem nastaveným na 1



Obrázek 20: části bariéry

5. parametr bariéra, která se naplní obdélníčky
- `SmallShootingSpeedIncrease()` tato metoda je volána po zásahu invadera pro 5ms zrychlení tiků časovače `invadersShouldShootTimer`
 - `DisableAllTimers()` pomocí reflexe metoda zastaví všechny časovače
 - `bool CollisionWithBarrier(PointF missile, Barrier barrier, out int columnIndex, bool fromInvader)` návratová hodnota určuje, jestli došlo ke kolizi bariéry na vstupu a `missile`
 1. parametr střela
 2. parametr bariéra
 3. parametr pokud metoda vrátí `true`, tak do 3. parametru uloží index sloupce bariéry, kde došlo ke kolizi
 4. parametr indikátor původce střely
 - `SplashDamage(PointF impactPoint, Barrier barrier, int columnIndex, bool fromInvader)` aplikuje plošené poškození bariéry vymazáním příslušných čtverečků tvořících bariéru
 1. parametr bod bariéry, který byl zasažen střelou
 2. parametr objekt zasažené bariéry
 3. parametr index sloupce bariéry, v kterém se nachází `impactPoint`
 4. parametr pochází střela zasahující bariéru od invadera?
 - `ShowTankExplosion()` bez zobrazení zprávy o tom, že by hra byla pozastavena ji pozastaví, během té doby nastaví několik *flagů*, díky kterým se budou vykreslovat ruiny tanku, pauzu pak zruší `tankExplosionTimer`, který tato metoda aktivuje, časovač pak v okamžiku tiků nastaví tank imunním a zruší pozastavení
 - `NextLevel()` zvýšení úrovně hry, úprava intervalů časovačů pro zvýšení obtížnosti

- `List<Invader> NClosestInvaders(int n)` vrátí seznam N invaderů takových, aby platilo, že vzdálenost souřadnice X invadera a tanku je minimální
- `int InvaderComparator(Invader i1, Invader i2)` komparátor, díky kterému je možné seřadit kolekci invaderů podle vzdáleností souřadnice X invadera a tanku, to se hodí v případě, že chceme vybrat vhodné střelce
- `MakeInvaderShootMissile(Invader invader)` na vstupu dostane instanci třídy `Invader`, metoda přidá do kolekce `invadersMissiles` novou střelu s nastavenou pozicí tak, aby vypadala, jako že pochází od invadera na vstupu
- `GenerateInvadersForLevel()` naplní seznam `invaders` armádou nových invaderů
- `UpdateBottomInvader()` zaktualizuje proměnnou `bottomInvaderY`
- `ShowNormalExplosion(PointF invaderLocation)` na vstupu dostane pozici padlého invadera, metoda zajistí, aby se na této pozici zobrazila posmrtná exploze přidáním této pozice do kolekce `explosions`
- `ShowMysteryExplosion()` obdobné jako `ShowNormalExplosion`, pouze verze pro mysteryho
- `PauseWOMsg()` zastavení hry bez zobrazení nápisu „PAUSED“, hodí se pro vytvoření menší prodlevy při zásahu tanku a 300ms zobrazení jeho ruin
- `PauseGameWMsg()` standardní zastavení hry vyvolané uživatelem stisknutím klávesy `p`
- `ResumeGame()` spustí všechny časovače a zajistí pokračování hry

3.2.4 Form1.cs

Jediný formulář hry, spouští se hned po startu aplikace. Během startu programu se nejprve nastaví ikonka aplikace, poté se všem `Label`ům nastaví písmo `Space Invaders` viz Ukázka kódu 1. Jako poslední se spustí hudba [14], která hraje ve smyčce.

Po startu hry se zobrazí již nám známé (z části 3.1.8) menu aplikace s jedním tlačítkem pro spuštění hry. Kód tohoto souboru obsahuje logiku pro reakci aplikace při interakci s tlačítkem. Jmenovitě se jedná o metody:

- `playLabel_MouseClick` sloužící ke spuštění herního módu,
- `clickableLabel_MouseMove`, `clickableLabel_MouseLeave` měnící barvu textu a vzhled kurzoru při přejetí myši.

Další metody:

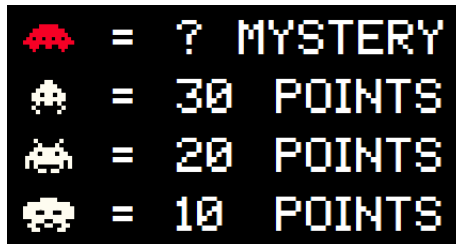
- `SetGameMode`,
- `EnableResizing`, `DisableResizing`,
- `SetScoreAndLevelLabelVisibility`,
- `SetMenuVisibility`,
- `ShowGameOver`

upravují uživatelské rozhraní při přechodu módů (menu, hra, konec hry).

Pokud se program nachází ve hře, potřebuje se dozvědět případné stisknuté klávesy. K tomu slouží přetížená metoda `ProcessCmdKey`, která volá instanční metodu `CheatKey`, jež náleží objektu `Game`.

3.2.5 ScoreTableUC.cs

Jedná se o seskupení 16 WinForms komponent na obrázku do logického celku.



Obrázek 21: uživatelská komponenta ScoreTableUC

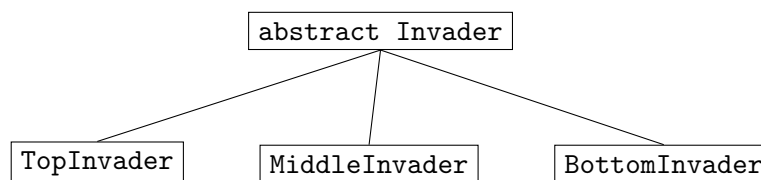
Jedinou věcí, kterou má tento celek na starost, je nastavení písma Space Invaders všem jeho 12 Labelům. To je naštěstí v C# primitivní:

```
foreach (Control c in Controls)
{
    if (c is Label)
        Utils.ChangeFontToSpaceInvaders(c);
}
```

Ukázka kódu 1: nastavení písma Space Invaders všem Labelům

3.2.6 Invader.cs

Soubor Invader.cs obsahuje hned 4 třídy. Jejich vztah je následující:



Obrázek 22: vztah tříd v souboru Invader.cs

Bázová třída obsahuje několik vlastností, z nichž většina je virtuálních. Nevirtuální jsou pouze 2 a jedná se o `Location` a `State`. První vlastnost je typu `PointF` a tedy určuje pozici invadera. Druhá je typu `bool` a její hodnota definuje stav invadera, více o stavech v části 3.1.2.

Mezi virtuální vlastnosti patří:

- `int ScoreGain`, což je skóre, které získáme při eliminaci invadera,
- `Image Look1`, `Image Look2`, každá z těchto vlastností obsahuje obrázek invadera v jednom stavu,
- `int Width`, `int Height` obsahující šířku a výšku obrázku invadera, což je v pořádku, protože všechny obrázky jednoho invadera mají stejné rozměry.

Co se týče metod, tak zde není potřeba využívat polymorfismus, protože invadeři pochodují společně a vždy se posunou o stejnou vzdálenost. Bázová třída obsahuje metody:

- `Image ActiveLook()`, ta vrací aktuální obrázek invadera podle jeho stavu,

- `void SwapState()` prohazující stav invadera,
- `void MoveLeft()`, `void MoveRight()`, obě dvě nejprve zavolají metodu `SwapState`, poté změni pozici invadera příslušným směrem o krok,
- `void MoveDown(int lvl)`, která vyžaduje jako vstupní parameter úroveň hry, podle parametru se nastaví velikost kroku, který invader udělá.

Žádné další nezmíněné vlastnosti ani metody nejsou obsaženy v žádném z potomků báze třídy. Samozřejmě všechny virtuální vlastnosti a metody jsou patřičně přetíženy.

3.2.7 Barrier.cs

Již bylo zmíněno v uživatelské části, že každá bariéra se skládá ze čtverečků o rozměru 5×5 vykreslovaných po sloupcích. Bude se tedy hodit čtverečky uskupit po sloupcích hlavně kvůli implementaci plošného poškození. Soubor vypadá následovně:

```
class Barrier
{
    public List<Column> Columns { get; set; } = new List<Column>();
    public float TopY { get; set; } = float.MaxValue;
}

class Column
{
    public List<RectangleF> Rectangles { get; set; } = new List<RectangleF>();
}
```

Ukázka kódu 2: soubor Barrier.cs

Poslední vlastnost s zatím nejasnou funkcí je `TopY`, do ní se ukládá souřadnice Y čtverečku s největší hodnotou Y. To se hodí v případě, kdy by nepřátelé měli instantně zničit bariéru.

Další informace o bariéře se nachází v části 3.1.6.

3.2.8 Constants.cs

V tomto souboru můžeme nalézt několik proměnných, jejichž hodnota se nikdy nemění. Jde např. o délku střely tanku, dobu po kterou je tank imunní, rozměr čtverečků tvořících bariéry a další...

3.2.9 ExtensionMethods.cs

Jak již je z názvu patrné, zde se nacházejí metody, které „zvenčí“ rozšiřují instanční metody třídy. Přesněji řečeno v tomto souboru se nachází pouze jedna metoda a tou je `DrawInvader`.

```
public static void DrawInvader(this Graphics g, Invader invader)
{
    g.DrawImage(invader.ActiveLook(), invader.Location);
}
```

Ukázka kódu 3: metoda DrawInvader

Na pozici invadera se vykreslí obrázek reprezentující invadera v závislosti na jeho aktuálním stavu.

3.2.10 Utils.cs

Pro případy, kdy využití statické třídy je přehlednější a dává smysl oproti vymyšlení *extension metod*, je tu soubor `Utils.cs`. V něm je možné nalézt klasické metody, jako např. vzdálenost dvou bodů nebo jestli se bod nachází uvnitř obdélníku.

O něco zajímavější je metoda `Play` [5], která na svém vstupu dostane `UnmanagedMemoryStream`. Tento proud dat bude reprezentovat zvuková data a ta se přehrají pomocí `SoundPlayeru`.

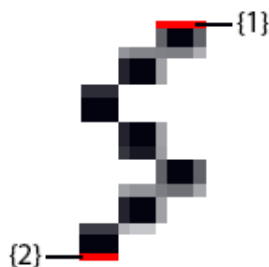
Dále se zde nachází dvojice metod umožňující nastavit komponentě WinForms písmo `Space Invaders`, jmenovitě: `InitSpaceInvadersFont` a `ChangeFontToSpaceInvaders` viz [4].

Poslední dvě metody `TopOfTankMissile` a `BottomOfInvaderMissile` mají symetrické chování. Obě na vstupu dostanou bod a vrátí bod. Nejprve se zaměřím na první metodu, ta bude na vstupu dostávat střely z tanku. Střela, znázorněná na následujícím obrázku, je uložena v paměti bodem (1). Pokud metoda na vstup dostane bod (1), vrátí bod (2).



Obrázek 23: zvětšená střela z tanku

Symetricky pro druhou metodu, střela od invadera je uložena v paměti bodem {1}. Pokud dostane na vstup bod {1}, vrátí bod {2}.



Obrázek 24: zvětšená střela od invadera (invertované barvy)

3.2.11 Další soubory

App.config XML soubor obsahující verzi prostředí .NET Framework nutnou ke spuštění hry.

SpaceInvaders.csproj Opět XML soubor, jeho součástí jsou informace o souborech přítomných v projektu Visual Studio.

4 Závěr

Povedlo se mi implementovat hru Space Invaders v takovém rozsahu, v jakém jsem si ji představoval. GDI+ je plně dostačující pro vytvoření grafiky takovéto hry. Nejedná se ale o žádného „rychlíka“. Použitím DirectX nebo OpenGL bych mohl snížit zatížení procesoru přesunutím výpočtů na grafickou kartu. Na druhou stranu práce s GDI+ je velmi jednoduchá.

Dále by bylo zajímavé rozšířit hru o:

- různá rozestavení invaze nepřátel v závislosti na úrovni hry,
- „padání“ bonusů z padlých nepřátel (inspirace z Arkanoidu).

Seznam obrázků

1	porovnání originální hry s mojí verzí	3
2	pozastavená hra	5
3	sloupec nepřátelské invaze	5
4	stavy TopInvadera	5
5	stavy MiddleInvadera	5
6	stavy BottomInvadera	6
7	nepřátelské střelivo	6
8	mystery	6
9	eliminace invadera	6
10	eliminace mysteryho	6
11	tank a jeho ruiny	6
12	imunní tank	7
13	GAME OVER!	7
14	před plošným poškozením a po	8
15	nápis Space Invaders v písmu Space Invaders	8
16	porovnání původního menu z 1978 a mého menu	9
17	porovnání původní bariéry s mojí bariérou	9
18	porovnání hry s velkou délkou a šířkou	9
19	spodní zelená čára	15
20	části bariéry	16
21	uživatelská komponenta ScoreTableUC	18
22	vztah tříd v souboru Invader.cs	18
23	zvětšená střela z tanku	20
24	zvětšená střela od invadera (invertované barvy)	20

Seznam tabulek

1	tlačítka ovládající tank	4
2	seznam cheatů	4

Seznam ukázek kódu

1	nastavení písma Space Invaders všem <code>Label</code> ům	18
2	soubor <code>Barrier.cs</code>	19
3	metoda <code>DrawInvader</code>	19

Zdroje

Grafické modely Veškeré grafické modely byly vystříhány z her Space Invaders a upraveny v Adobe Photoshop.

Autentičnost hry Písmo Space Invaders, který celá hra používá obzvlášť zlepšil herní zážitek.

[1] space_invaders.ttf

<https://fonts2u.com/download/space-invaders-regular.font>

Informace o hře Space Invaders Nějaké obecné informace o hře se mi hodily zejména pro napsání kapitoly 1 a usnadnily mi výběr zápočtového programu.

[2] anglická wikipedie

https://en.wikipedia.org/wiki/Space_Invaders

[3] různé pohledy a ukázky ze hry na youtube

<https://www.youtube.com/watch?v=D1jZaIPeD5w>

<https://www.youtube.com/watch?v=UZ1EX19xgR8>

<https://www.youtube.com/watch?v=MU4psw3ccUI>

C# inženýrské věci Během vývoje hry jsem si zaznamenával kolikrát, a co jsem sháněl jak udělat (kde jinde než na stackoverflow ☺).

[4] jak použít externí font ve WinForms

<https://stackoverflow.com/questions/1297264/using-custom-fonts-on-a-label-on-winforms>

[5] jak spustit zvukovou nahrávku

<https://stackoverflow.com/questions/3502311/how-to-play-a-sound-in-c-net>

[6] jak spojit všechny List<T> do jednoho pomocí LINQ

<https://stackoverflow.com/questions/1191054/how-to-merge-a-list-of-lists-with-same-type-of-items-to-a-single-list-of-items>

[7] jak používat resources

<https://stackoverflow.com/questions/90697/how-to-create-and-use-resources-in-net>

[8] jak zjistit, že okno ztratilo *focus*

<https://stackoverflow.com/questions/570021/forms-lost-focus-in-c-sharp/570045>

[9] jak updatovat GUI z jiného threadu

<https://stackoverflow.com/questions/10775367/cross-thread-operation-not-valid-control-textbox1-accessed-from-a-thread-oth>

[10] jak vytvořit cyklus přes všechny proměnné objektu

<https://stackoverflow.com/questions/9893028/c-sharp-foreach-property-in-object-is-there-a-simple-way-of-doing-this>

<https://stackoverflow.com/questions/7649324/c-sharp-reflection-get-field-values-from-a-simple-class>

[11] jak zjistit, zda-li se dva obdélníky překrývají

<https://stackoverflow.com/questions/14662796/check-if-two-areas-are-in-contact>

[12] jak spočítat počet řádek kódu celého VS projektu

<https://stackoverflow.com/questions/1244729/how-do-you-count-the-lines-of-code-in-a-visual-studio-solution>

[13] proč se nevolá událost KeyDown

<https://stackoverflow.com/questions/3172731/forms-not-responding-to-key-down-events>

Zvukové soubory

[14] hudba v menu

<https://www.youtube.com/watch?v=8teuyCYeDxQ>

[15] výstřel z tanku

<http://www.classicgaming.cc/classics/space-invaders/files/sounds/shoot.zip>

[16] eliminace invadera

<http://www.classicgaming.cc/classics/space-invaders/files/sounds/invaderkilled.zip>

[17] eliminace mysteryho

http://www.classicgaming.cc/classics/space-invaders/files/sounds/ufo_highpitch.zip

[18] průlet mysteryho

http://www.classicgaming.cc/classics/space-invaders/files/sounds/ufo_lowpitch.zip

[19] zásah tanku

<http://www.classicgaming.cc/classics/space-invaders/files/sounds/explosion.zip>

[20] konec hry

https://www.youtube.com/watch?v=AD0-L9L_NrM

[21] zničení bariéry

<https://uloz.to/!5878yRtYUY0C/barrier-destroyed-wav>