

# PRG 2 - Hra Frogger

---

**Milan Abrahám**

---

## Zadání

---

Cílem bylo vytvořit vlastní verzi hry z konzole ZX Spectrum, kde se současně pohybuje více objektů.

## Úvod

---

Hrou, kterou jsem se rozhodl napodobit, je Frogger z roku 1992 [1]. Inspirací mi byla ještě jiná stejnojmenná verze hry z roku 1981 [2] a to hlavně díky lepšímu grafickému zpracování. Hra je vytvořená v jazyce C# pomocí grafické knihovny Windows Forms.

Hráč ovládá žábu, které musí pomoci dostat se na druhou stranu herní plochy (na lekníny). V cestě jí ovšem překáží silnice, na které jezdí vozidla, nebo řeka po nichž se plaví klády. Žába ovšem neumí plavat, proto je nutné nespadnout do vody a přeskákat po kládách. Hra obsahuje několik úrovní s rozdílnou obtížností, včetně náhodně generované. Hra má jednoduché ovládání, žába se pohybuje pomocí šipek (nahoru, doleva, doprava, dolů). Hráč má 5 životů/žab a 3 z nich se musí dostat na lekníny, aby vyhrál.

## Herní objekty

---

Rozhodl jsem se pro hru nakreslit vlastní obrázky formou pixel artu. Všechny mají velikost 64x64 pixelů, stejně jako jeden dílek herního prostředí. K nakreslení všech prvků jsem použil webovou stránku Pixilart [3].

Pozadí je pak skládáno z jednotlivých čtverců, stejně tak jako kláda, která má více částí.

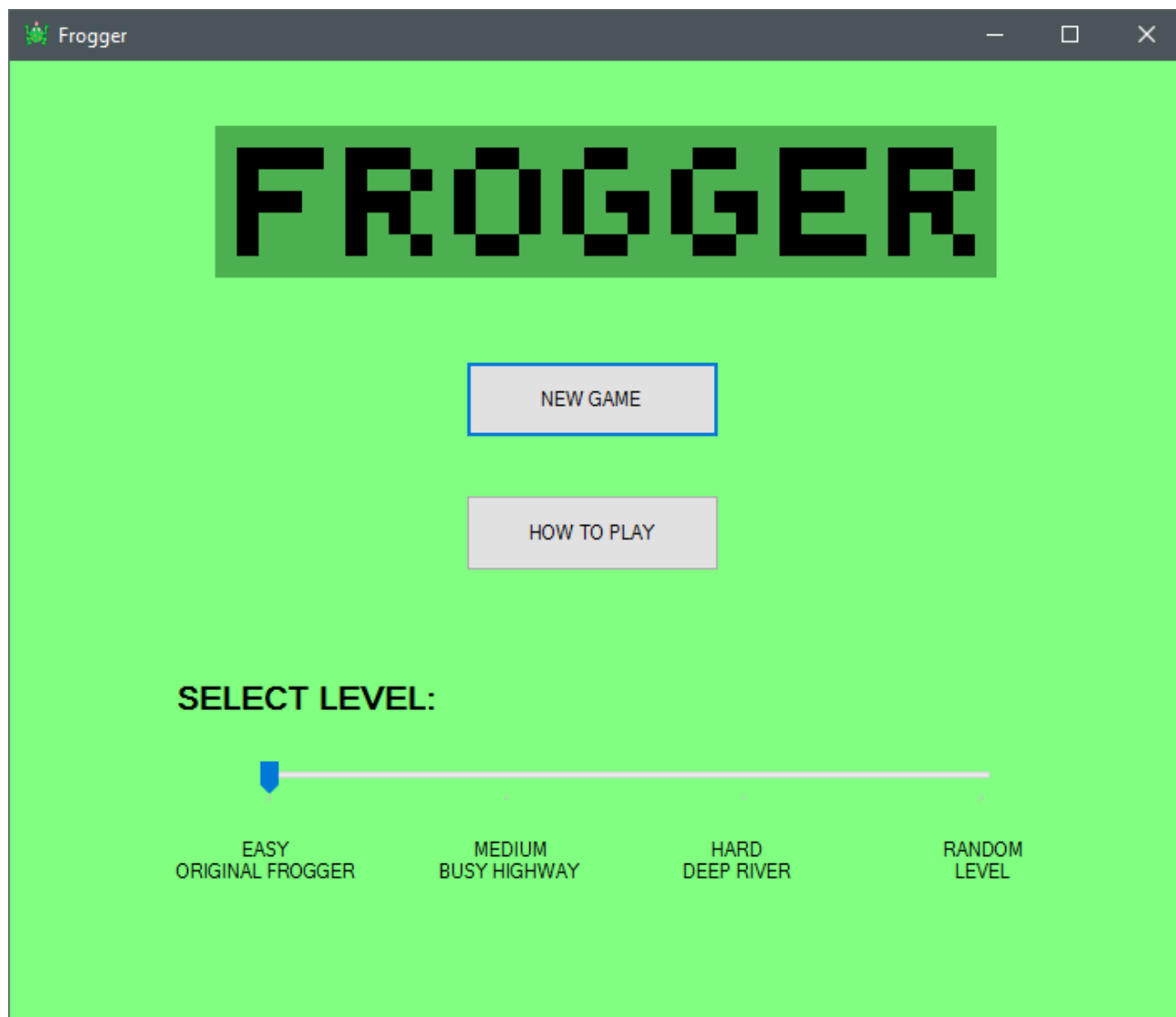
Obrázek	Název	Popis
	žába	Hlavní objekt ovládaný hráčem.
	leknín	Cílem hry je zaplnit všechny lekníny žábami.
	auto	Pohybuje se po silnici, žába se s ním nesmí střetnout jinak zemře. Má více možných barev.
	řeka	Žába nesmí do řeky spadnout, neumí plavat a utopí se.
	kláda	Kláda plave po řece, žába na ní může skočit, ale pohybuje se spolu s ní.
	tráva	Tráva je pro žábu bezpečná.
	pláž	Pláž je pro žábu bezpečná.
	silnice	Silnice je pro žábu bezpečná, ale musí dávat pozor na příjíždějící auta.

Pro žábu jsem vytvořil animaci, jednou za určitý čas vyplázne jazyk, to je vyřešeno měnícími se obrázky žáby.

## Hlavní menu

---

Hlavní menu obsahuje tlačítko pro začátek nové hry, tlačítko s nápovědou a výběr úrovně. Nápověda obsahuje ovládání a popis všech herních objektů a úrovní. Tlačítko pro začátek nové hry spustí hru v úrovni, která je vybrána posuvníkem.



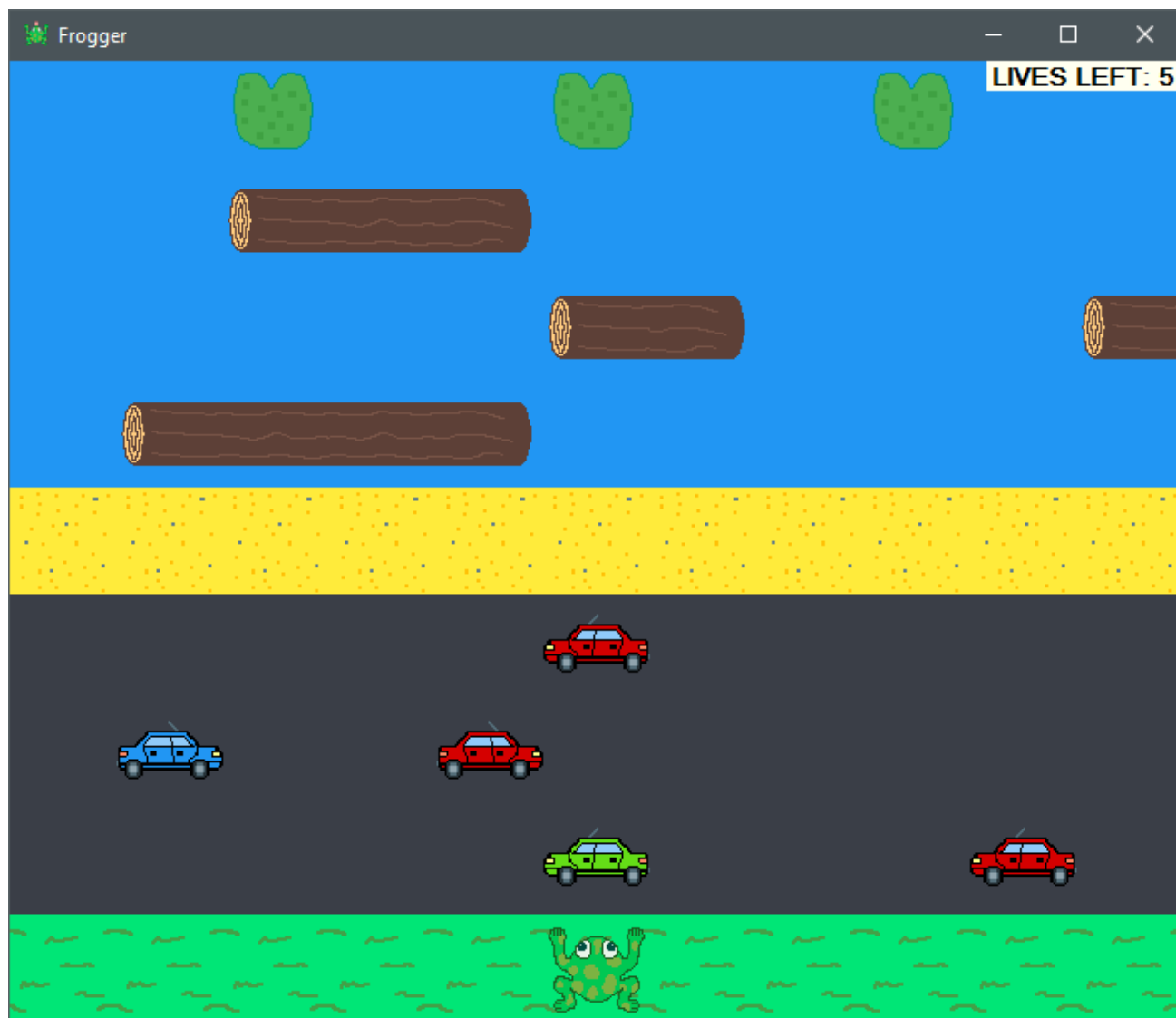
## Herní prostředí

---

Prostředí je tvořené z dílku, každý má velikost 64x64 bodů, právě kvůli velikosti obrázků. Nejdříve je nakresleno pozadí, podle přednastavených pravidel. Na pozadí jsou potom přikreslovány herní objekty. Prostředí obsahuje 9 dílků na šířku a 11 na výšku. Výšku jsem zvolil podle původní hry, šířku pak adekvátní výšce.

Prostředí se překresluje každých 130 ms. Objekty se pohybují každé čtvrté překreslení (cca půl vteřiny), kromě žáby, která se může pohnout každé překreslení. Uživatel ovšem není schopen pohybovat žábou tak rychle. Žába se tak může pohybovat z důvodu snadnějšího uhýbání jiným objektům.

Objekty včetně pozadí nejsou vykreslovány přímo na obrazovku. Nakreslí se nejdříve na skrytou bitmapu, která je potom zobrazena na obrazovku. Důvodem tohoto řešení je, že vykreslování přímo na obrazovku způsobovalo nepříjemné blikání jednotlivých objektů.

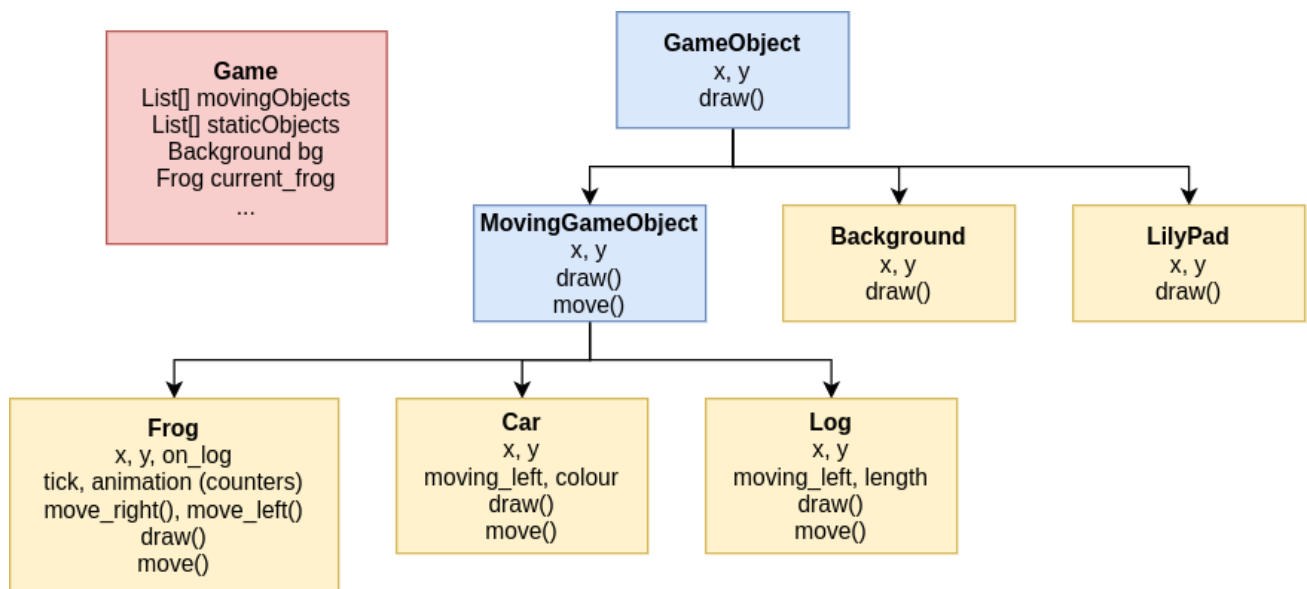


## Objektový návrh

Objektový návrh je znázorněn na následujícím diagramu. Všechny objekty dědí od abstraktní třídy `GameObject`. Mají metodu `draw`, která umožňuje jejich nakreslení na herní plochu. Objekty, kterými je třeba pohybovat, jsou potomky třídy

*MovingGameObject* a mají navíc metodu *move*. Celé pozadí je jedním objektem, vždy se nakreslí první a vykreslí celou plochu podle předem daných pravidel, která jsou uložena v objektu *Game*.

*Game* je hlavní objekt obsahující všechna data o hře, uchovává všechny objekty ve dvou polích, jedno s pohyblivými objekty a druhé se statickými. Dále obsahuje pravidla pro strukturu herního prostředí a vytváření nových objektů, udržuje stav hry, skóre atd.



## Herní úrovně

Hra obsahuje 3 rozdílně obtížné úrovně v různých prostředích a možnost nechat si prostředí náhodně vygenerovat.

Úroveň si uživatel vybere v hlavním menu, v konstruktoru hry (object *Game*) se poté přiřadí pravidla pro pozadí a vytváření objektů. Tato pravidla jsou pole řetězců s požadovanými vlastnostmi. Každý index reprezentuje, jednu y-ovou souřadnici.

V případě náhodné úrovně jsou jednotlivé řetězce vygenerovány náhodně, četnost vytváření objektů a délka klád je generována z předem vybraných intervalů, aby byla úroveň hratelná.

Ukázka pravidel pro jednoduchou úroveň:

```
background_tiles = new string[] {
    "river_end",
    "river",
    "river",
    "river",
    "beach",
    "road",
    "road",
    "road",
    "grass"
};
//handles spawning
//type of object | where to spawn it | (log length) | frequency (in game ticks)
spawning = new string[]
{
    "",
    "log left 3 23",
    "log right 2 19",
    "log left 4 27",
    "",
    "car right 24",
    "car left 12",
    "car right 16",
    ""
};
```

Úroveň	Obtížnost	Popis
Original Frogger	jednoduchá	Kopie prostředí z originální hry Frogger.
Busy Highway	střední	Prostředí tvořeno pouze silnicí, auta se objevují často.
Deep River	těžká	Prostředí tvořené pouze řekou, přeskákat ji na kládách není jednoduché
Náhodná		Prostředí je vygenerováno naprosto náhodně. Obtížnost je vždy různá.

## Průběh hry

Po začátku hry se nejdříve vytvoří objekt Game, pozadí, přidají se lekníny a žába a další potřebné objekty a proměnné. Jak již bylo zmíněno prostředí se překresluje každých 130 ms. Objekty se pohybují každé čtvrté překreslení. Hra je ovládána objektem Timer (časovač). Každých 130 milisekund provede jedno překreslení (**tick**).

Nejdříve se přepíšu zbývající životy a připravím seznam objektů k odstranění. Nelze je odebrat v průběhu překreslení.

Poté se podle pravidel vytváření nových objektů vytvoří nová auta a klády na určených souřadnicích. Oba objekty mají v pravidle frekvenci. Frekvence znamená, že objekt se vytvoří každý  $n$ -tý tick. Po vytvoření všech objektů se nakreslí pozadí. Všechny objekty se kreslí na nezbrazenou bitmapu, která se na obrazovku vykreslí až nakonec.

Následně je vykreslen každý pohyblivý objekt. Každý čtvrtý tick se objekty pohnou. Objekty, které se dostaly mimo obrazovku, jsou umístěna do pole pro vymazání. Následuje vykreslení všech statických objektů. Těmi jsou lekníny a žáby, které se již dostaly na lekníny.

Dále se zkontroluje zda nějaké auto není na stejném políčku jako žába, zda žába není na vodě bez klády, zda kláda nepohla žábu mimo obrazovku nebo jestli už žába nedoskákala na leknín. Ve všech případech se změní stav hry.

Stav hry může být následující: hra nezačala, hra běží, žába se dostala na leknín, žába zemřela, prohra a výhra.

Nakonec se odstraní všechny objekty k odstranění, bitmapa se vykreslí na obrazovku a vyřeší se, zda se změnil stav hry.

Pokud žába zemřela a zbývají ještě životy vytvoří se nová, v opačném případě hra skončí prohrou. Pokud se žábá dostala na leknín, vytvoří se nová, pokud se tam dostaly 3 hra skončí vítězstvím.

## Rozdíly oproti původní hře

---

Některé části jsem se rozhodl pozměnit od původní hry. Například se mi nelíbila grafika původní hry, proto jsem si nakreslil vlastní v jiném stylu.

Původní hra obsahuje pouze jedno prostředí, ve kterém se ale mění obtížnost. A hráči je počítáno skóre, čím rychleji vyhraje, tím vyšší je. Skóre a zvyšující se obtížnost jsem se rozhodl nahradit více úrovněmi s rozdílnou obtížností. Jelikož jsem měl vytvořený systém umožňující různou generaci prostředí na základě předurčených pravidel bylo snadné přidat náhodně generované úrovně.

## Závěr

---

Povedlo se mi vytvořit funkční hru, se kterou jsem spokojen. Hratelnost jsem nechal otestovat několik lidí. Získal jsem zpětnou vazbu, která mi pomohla hru vylepšit a opravit většinu chyb. Některé chyby jsou skoro nepostřehnutelné a vyžadovaly by přepsání velké části programu, proto jsem se rozhodl je neopravit. Získal jsem nové zkušenosti práce s grafikou a uživatelským prostředím. Navíc jsem se přesvědčil, že vývoj hry nemusí být tak náročný, jak jsem si původně myslel. Příště bych ovšem zvolil jiné prostředí. Na Visual Studio se mi nelíbí nedostupnost na systému Linux a grafická knihovna Windows Forms mi přijde docela omezující.

Hra je volně dostupná ve webové službě Github [4] jako projekt pro Visual Studio.

## Zdroje

---

- [1] [Frogger \(Deanysoft\) Walkthrough, ZX Spectrum - YouTube](#)
- [2] [Arcade Game: Frogger \(1981 Konami\) - YouTube](#)
- [3] [Pixilart - Free Online Art Community and Pixel Art Tool](#)
- [4] [GitHub - milan252525/Frogger: Remake of 1992 Frogger game](#)