

Dokumentace k zápočtovému programu

# SKart 1.0

Programování II (NPRGo31)

1. ročník Bc. studia MFF UK - Informatika



Autor: Jakub Kinšt

## **Obsah**

1	Úvod.....	3
2	Základní popis programu .....	3
3	Možnosti a funkce .....	3
4	Ovládání programu .....	4
5	Použité datové struktury a vysvětlení nejdůležitějších metod.....	4
5.1	Objekt MainWindow .....	4
5.2	Objekt Car .....	4
5.3	Objekt LapTimeStats.....	5
5.4	Objekt TimeCounter.....	5
6	Možná rozšíření hry .....	5

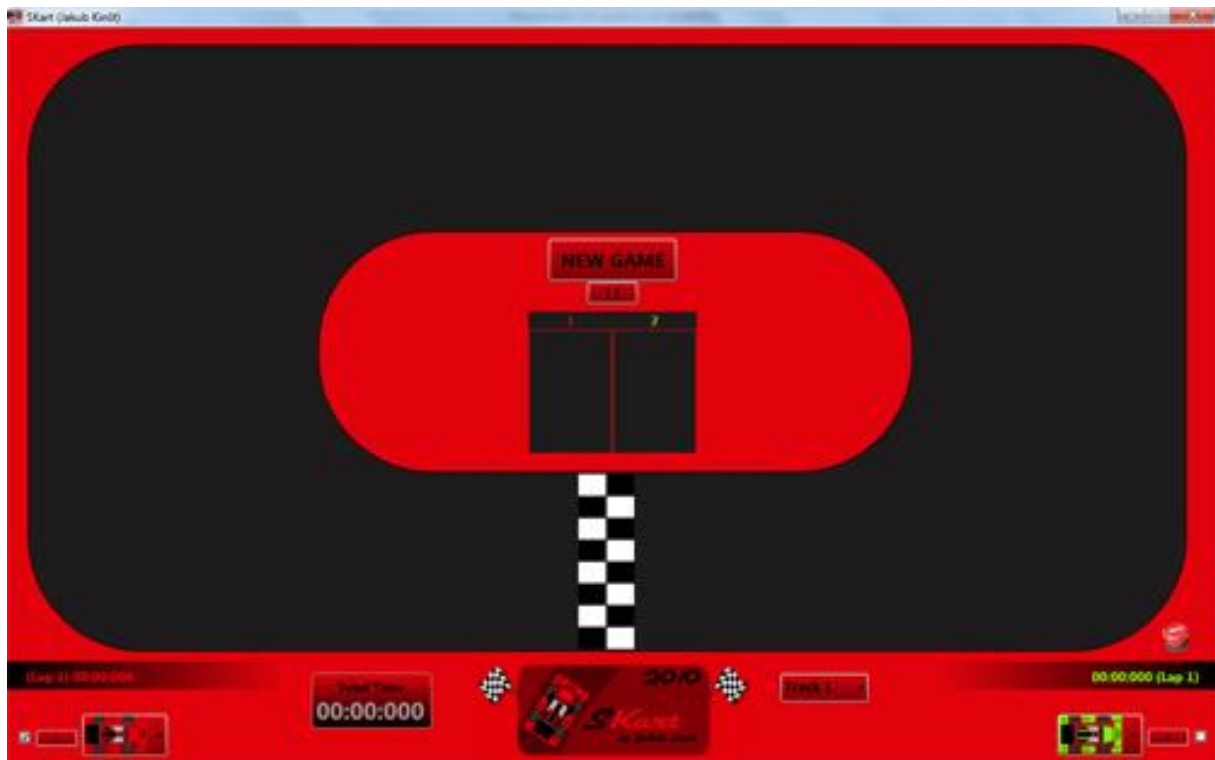
## 1 Úvod

Tuto dokumentaci doplňují zdrojové kódy jednotlivých objektů a oken a komentáře, které tyto zdrojové kódy obsahují. Podrobnou strukturu funkcí a procedur lze nejlépe pochopit právě v těchto zdrojových kódech.

## 2 Základní popis programu

Zápočtový program SKart je klasická závodní hra, ve které je cílem projet trať během co nejmenšího časového intervalu. Jedná se o závody motokár z pohledu ptačí perspektivy (tedy 2D) s počítadlem času, možností hry dvou hráčů na jednom počítači a s možností výběru trati.

Program SKart je vytvořen ve vývojovém prostředí Microsoft Visual Studio 2010. Je založen na technologii Windows Presentation Foundation (dále WPF), která chytře kombinuje možnost navrhovat design i pokročilé funkce v kódu XML a také klasické programování v jazyce



C#. Ke spouštění programu je potřeba mít nainstalované rozhraní Microsoft .NET Framework 4.

## 3 Možnosti a funkce

Hra SKart je určena pro jednoho nebo dva hráče na jednom počítači. Hráč projíždí se svojí motokárou určenou trať a snaží se dosáhnout co nejnižšího času. Program nabízí měření času projetí každého kola včetně funkce zvýraznění nejlepšího a nejhoršího v tabulce všech časů. Program i automaticky počítá ujetá kola. Aby uživatel nemohl podvádět a neustále jen projíždět startem, je také zajištěno, aby musel objet celé kolo za účelem zapsání času (neviditelný checkpoint). Hráč nemůže vyjet z určené trati. Pro lepší efekt jsou naimplementovány i funkce

jako rozsvícení brzdových světel na vozidle při brzdění nebo také efekt smyku společně se stopou po pneumatikách zanechanou brzděnými koly.

Každý hráč si může vybrat jeden z barevných profilů, dále své jméno, které se zobrazí i na motokáře.

## 4 Ovládání programu

Ovládání hry SKart je jednoduché. Hráč 1 ovládá svou motokáru klasicky šipkami na klávesnici a hráč 2 používá k pohybu kombinaci kláves W, S, A, D. V hlavním okně je několik ovládacích prvků. Jednak zde nalezneme hlavní tlačítko pro novou hru (NEW GAME), dále pak tlačítko pauzy (PAUSE), které na libovolně dlouhý časový interval hru včetně časovače pozastaví. Dále v okně nalezneme nastavení hráčů, které se sestává ze zaškrtačacího políčka pro zapnutí nebo vypnutí hráče, kolonka pro nastavení jména hráče a také výběr barvy.

V programu lze vyvolat okno nápovědy. Lze toho docílit buď stisknutím klávesy F1 nebo kliknutím na určenou ikonu v pravém dolním rohu okna.

## 5 Použité datové struktury a vysvětlení nejdůležitějších metod

### 5.1 Objekt *MainWindow*

*MainWindow* je kořenem celé aplikace. To tento objekt se vytvoří po spuštění hry. *MainWindow* obsahuje časovač typu *Timer*. Tento typ poskytuje metodu, která se vyvolá každých  $x$  milisekund, kde  $x$  je jedna z vlastností *Timeru*. Ten, pokud je hra spuštěna, opakovaně spouští metody k aktualizaci pozicí motokár a také aktualizací zobrazovaných časů. Dále tento objekt obsluhuje ovládací prvky v okně. Při spuštění nové hry se inicializují proměnné, vytvoří se dvě instance třídy *Car* (viz. dále), spustí se timer a o zbytek se již starají objekty *Car*. *MainWindow* ještě v každém provedení metody *Timer\_Tick* (zmiňovaná opakovaná metoda) kontroluje, zda žádná z motokár nekoliduje se zdmi, s jinými motokárami nebo se startem či s checkpointem. Podle toho pak na dílčí objekty zavolá obslužnou metodu. Kontrola kolizí je řešena integrovanou metodou *WPF*, která umí zjistit, zda se překrývají dva objekty typu *Path*. Je proto potřeba udržovat obrys všech objektů v typu *Path*. Což není vždy jednoduché. Je totiž potřeba neustále převádět objekty typu *Shape*. Metoda umí zjistit, zda se obrazce překrývají, nedotýkají nebo jestli se navzájem obsahují. Trať je reprezentována několika objekty typu *Shape*, které musí být jednotlivě podrobovány kontrole kolize.

XAML soubor obsahuje hlavní layout aplikace. Najdeme zde ovládací prvky, tabulku časů typu *LapTimeStats* a počítadlo času typu *TimeCounter*.

### 5.2 Objekt *Car*

Tento objekt reprezentuje samotnou motokáru. XAML soubor obsahuje kód určující podobu motokáry. Konstruktor má několik parametrů pro pohodlné vytváření nových motokár. Je možno navolit startovací pozici motokáry, jednu z dostupných barev, jméno hráče, které bude zobrazeno na karoserii a dokonce i řídicí klávesy. Samotná třída má několik konstant a proměnných určujících chování při pohybu nebo nárazu. O pohyb se stará metoda *UpdatePos*. Ta

je volána neustále během hry. Uvnitř se spočítá směr pohybu pomocí základních trigonometrických funkcí, motokára se podle toho natočí a následně se posune do vypočítaných souřadnic. Při držení plynu se zvětšuje proměnná *Speed*, která je pokaždé použita pro výpočet nových souřadnic. Motokára také musí umět brzdit ale i zpomalovat samovolně vlivem odporu vzduchu. Proměnná *Speed* tedy samovolně klesá. Je potřeba také zajistit realističnost zatáčení vozidla. Chceme, aby bylo možné motokárou zatáčet v závislosti na aktuální rychlosti. Každé vozidlo je lépe manipulovatelné při nižších rychlostech a naopak při vysoké rychlosti již nelze s vozidlem tolik manévrovat. Tudíž rotaci s určitými koeficienty dělíme rychlostí. Ještě se ale musíme postarat o to, aby se při nulové nebo úplně minimální rychlosti auto neotáčelo jako zběsilé. O to vše se tato metoda stará. Dále zde nalezneme i obsluhu nárazu, která se stará o vytvoření odrazu od zdi/motokáry. Ten je realizován tak, že při kolizi se motokáře upraví proměnná *Speed* do opačné hodnoty společně s uplatněním určité ztráty energie. Zde také narážíme na různé problémy jako například kmitání vozidla ve zdi (přepínání rychlosti opakovaně ze záporné hodnoty do kladné a zpět) a podobně. Rychlost motokáry je pochopitelně omezena konstantou *MaxSpeed*, která může být snadno upravena, což je možná příprava na přidání bonusů nebo například různých typů vozidel.

### 5.3 Objekt *LapTimeStats*

Tento objekt má za úkol monitorovat časy kol obou hráčů, ukládá je do tabulky a v té také zvýrazňuje nejlepší i nejhorší zajeté časy. Objekt má metodu, která obsluhuje přidání času. Během přidávání zkontroluje časy předešlé, a pokud je to potřeba, aktualizuje zvýraznění časů.

XAML soubor obsahuje vizuální podobu tabulky zajetých časů.

### 5.4 Objekt *TimeCounter*

*TimeCounter* je použit k měření časů jednotlivých kol pomocí rozdílů času startu a času projetí do dalšího kola. Čas pak nastaví v uživatelském rozhraní a zároveň ho pošle do tabulky typu *LapTimeStats*. Čas je změřen pomocí kontroly kolize motokáry s obdélníkem reprezentujícím cílovou čáru. Pokud nastane kolize, zavolá se metoda, která patřičnému hráči připíše jedno kolo a odečte čas vyjetí od současného času. Tím snadno spočte čas zajetého kola a zapíše ho do tabulky. Hráč nemůže podvádět tak, že by stále jen projížděl cílem. Kvůli tomu je na trati ještě jeden obdélník, tentokrát neviditelný, který musí nejdříve zaznamenat kolizi s motokárou a až teprve potom může být započítáno další kolo.

XAML soubor dává podobu měřiči časů, který se objevuje na hlavním okně.

## 6 Možná rozšíření hry

Závodní hru SKart lze velmi snadno rozšířit, tak aby byla ještě hratelnější a zábavnější. Rozšíření může spočívat ve vytvoření složitějších tratí, zabudování bonusů (například zrychlení, zpomalení...). Lze hru změnit tak, že by na obrazovce byla zobrazena jen část trati, která by byla mnohem větší a kamera by se hýbala po trati společně s motokárou. Je možné více zapracovat na řešení kolizí, může se přidat podpora hraní přes síť nebo internet. Lze zabudovat i různé nerovnosti na trati a graficky pak znázornit nadskočení vozidla. Ve zdrojovém kódu lze jednoduše upravovat fyzikální vlastnosti pohybu po trati. Je možné přidat efekty navíc (např.

kouř z výfuku, osvětlení motokáry, pohyby řidiče, natočení kol atd.). Další možností je vytvoření několika typů vozidel s kompletně odlišným modelem jízdy. Vozidla by se lišila jízdními vlastnostmi jako je maximální rychlost, zrychlení, poloměr otáčení atd. Hra je od začátku tvorby uzpůsobena k budoucímu rozšiřování.