
TECHNICKÁ UNIVERZITA V LIBERCI
Fakulta mechatroniky a mezioborových inženýrských studií

Studijní program: B2612 – Elektrotechnika a informatika
Studijní obor: 1234R567 – Název studijního oboru

OFF-LINE přehrávání jízdy

OFF-LINE playback of drive

Bakalářská práce

Autor:	Martin Čičkán
Vedoucí práce:	Ing. Martin Vlasák
Konzultant:	RNDr. Klára Císařová

V Liberci 11. 5. 2007

Prohlášení

Byl(a) jsem seznámen(a) s tím, že na mou bakalářskou práci se plně vztahuje zákon č. 121/2000 o právu autorském, zejména § 60 (školní dílo).

Beru na vědomí, že TUL má právo na uzavření licenční smlouvy o užití mé bakalářské práce a prohlašuji, že **s o u h l a s í m** s případným užitím mé bakalářské práce (prodej, zapůjčení apod.).

Jsem si vědom(a) toho, že užít své bakalářské práce či poskytnout licenci k jejímu využití mohu jen se souhlasem TUL, která má právo ode mne požadovat přiměřený příspěvek na úhradu nákladů, vynaložených univerzitou na vytvoření díla (až do jejich skutečné výše).

Bakalářskou práci jsem vypracoval(a) samostatně s použitím uvedené literatury a na základě konzultací s vedoucím bakalářské práce a konzultantem.

Datum

Podpis

Anotace

Cílem bakalářské práce je vytvoření aplikace, která bude umět zobrazit záznam jízdy autíčka po autodráze. Práce se bude zabývat různými možnostmi zobrazení autodráhy a autíčka v počítači. Dále budou popsány způsoby převodu záznamu jízdy (hodnot napětí) na rychlosti autíčka při přehrávání. Bude vytvořena 3D aplikace, popsána její struktura a použité metody zobrazení záznamu jízdy. Aplikace bude schopna zobrazit záznam jízdy z několika pohledů s různými rychlostmi přehrávání.

Annotation

The aim of this bachelor thesis is the realization of the application that will be able to play the movement of the toy-car's course on the ride. The thesis will focus on different possibilities of display the ride and toy-car in the computer. Then the manners of conversion of toy-car movement (according to the amount of voltage) to the speed of the car during the play will be described. The 3D application will be created, described its structure and methods of play the recorded ride. The application will be able to play the record of the ride from few different views with different play speed.

Obsah

Úvod.....	7
1. Objekt autíčko a jeho možnosti pohybu po dráze.....	8
1.1 Objekt autíčko.....	8
1.2 Dráha.....	9
1.3 Pohyb autíčka po autodráze.....	10
2. Možné způsoby zobrazení záznamu jízdy.....	11
2.1 Videokamera.....	11
2.2 Fotoaparát.....	12
2.3 3D aplikace.....	13
2.4 Shrnutí jednotlivých řešení zobrazení jízdy.....	14
3. Zjištění rychlosti autíčka v závislosti na napětí.....	15
3.1 Digitální fotoaparát.....	15
3.2 Průměrná rychlost.....	17
4. Záznam jízdy autíčka.....	21
4.1 Úprava jízdy autíčka.....	21
4.2 Struktura datového souboru.....	21
5. Tvorba modelů autodráhy a autíček.....	23
5.1 Software pro 3D modelování.....	23
5.2 Vytvoření 3D modelu autodráhy.....	23
5.3 Vytvoření 3D modelů autíček.....	27
6. Získání trajektorií obou drah autodráhy.....	29
6.1 Načtení trajektorie.....	29
6.1.1 Přepis souřadnic trajektorie.....	29
6.1.2 Načtení souřadnic trajektorie.....	30
6.2 Úprava načtených trajektorií drah.....	31
7. Rozhraní 3D grafiky.....	32
7.1 Dostupná 3D rozhraní.....	32
7.2 Popis rozhraní OpenGL.....	32
7.3 Zvolené 3D rozhraní.....	34
8. Načtení 3D modelů do aplikace.....	35
8.1 Způsob uložení 3D modelů.....	35
8.2 Struktura souboru formátu 3DS.....	35
8.3 Unit f_3DS.....	35
9. 3D aplikace.....	36
9.1 Obecný popis.....	36
9.2 Struktura aplikace.....	36
9.2.1 Zobrazení 3D modelů.....	36
9.2.2 Reprezentace autíčka.....	39
9.2.3 Načtení a interpolace trajektorií.....	39
9.2.4 Jízda autíčka.....	41
9.2.5 Uživatelské rozhraní.....	43
10. Závěr.....	44
Použitá literatura.....	46
Přílohy.....	47
Ovládání a popis vytvořené 3D aplikace.....	47

Úvod

Práce je rozdělena na několik tematických částí, které se věnují problematice zobrazení záznamu jízdy autíčka po autodráze.

V kapitole s názvem *Objekt autíčko a jeho možnosti pohybu po dráze* jsou popsány základní rozměry a vlastnosti autíčka i autodráhy. Jsou zde také uvedena omezení pohybu autíčka vyplývající z konstrukce autodráhy.

Kapitola *Možné způsoby zobrazení záznamu jízdy* představuje různé způsoby zobrazení jízdy autíčka. Popsány jsou klady a zápory jednotlivých řešení.

V části *Zjištění rychlosti autíčka v závislosti na napětí* jsou popsány způsoby určení převodní funkce elektrického napětí na rychlost. Tato funkce umožní hodnotám elektrického napětí získaných z datového souboru se záznamem jízdy přiřadit odpovídající rychlosti autíčka.

Část *Záznam jízdy autíčka* se zabývá popisem struktury datového souboru se záznamem jízdy autíčka. Jsou zde také popsány nutné úpravy načítaného záznamu.

V kapitole s názvem *Tvorba modelů autodráhy a autíček* je představen software pro vytváření 3D modelů. Je zde podrobněji popsán postup vytváření 3D modelů autodráhy a autíček.

Část *Získání trajektorií obou drah autodráhy* řeší problematiku trajektorií, po kterých se autíčko na dráze pohybuje. Jsou zde představeny způsoby vytvoření, načítání a následné zpracování těchto trajektorií.

Kapitola *Rozhraní 3D grafiky* se zabývá výběrem zobrazovacího 3D rozhraní aplikace a přináší bližší popis 3D standardu OpenGL.

Kapitola *Načtení 3D modelů do aplikace* se věnuje problematice načtení souborů s 3D modely do vytvářené aplikace zobrazující jízdu autíčka. Jsou zde popsány nástroje umožňující načíst soubory obsahující 3D modely do datových struktur v aplikaci.

V části *3D aplikace* je popsána struktura aplikace zobrazující záznam jízdy autíčka. Popsány jsou datové struktury a jednotlivé objekty aplikace. Tato část je doplněna o vývojové diagramy, které popisují důležité funkce programu.

V části *Závěr* jsou shrnuty dosažené výsledky. Jsou zde také popsány další možnosti, které by mohly být použity při řešení některých problémů.

V příloze je popsáno uživatelské rozhraní aplikace a její ovládání. Na přiloženém CD se nachází vytvořená 3D aplikace včetně zdrojového kódu.

1. Objekt autíčko a jeho možnosti pohybu po dráze

1.1 Objekt autíčko

Jedná se o model autíčka ve tvaru formule o celkové šířce 7 cm, délce 13,5 cm a výšce 5 cm. Autíčko je vybaveno stejnosměrným elektromotorem napájeným maximálně dvanácti volty, který pohání zadní kola.



Obr. 1: Autíčko

Na spodní straně v přední části je autíčko vybaveno otočnou lyžinou, která má za úkol vézt pohybující se autíčko ve správné dráze. Lyžina je ze spodní strany vybavena kovovými sběrnými kartáčky, které jsou pomocí drátů připojeny přímo na elektromotor. Kartáčky zajišťují vodivé spojení s kontakty na dráze a tím napájení elektromotorku autíčka.

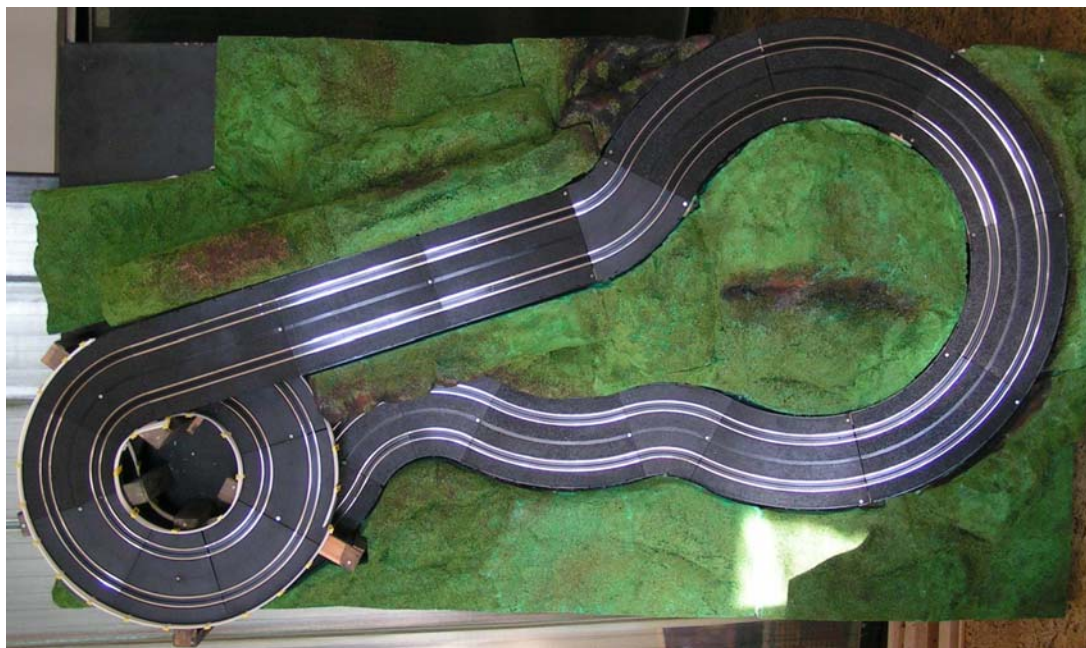


Obr. 2: Pohled na sběrné kartáčky a vodící lyžinu autíčka

1.2 Dráha

Celá dráha je postavena ze tří typů různých segmentů. Každý z nich obsahuje vždy dvě vodící drážky s dvojicí napájecích kontaktů pro vytvoření dvou nezávislých drah. Všechny segmenty lze k sobě různě připojovat a vytvořit tak téměř libovolný tvar dráhy. Autodráha je umístěna na dřevotřískové desce s rozměry 200×80 cm a tloušťce 2,5 cm. Většina dráhy je tvořena dlouhými a krátkými zatáčkami, pouze v jedné části se nachází rovný úsek složený ze tří segmentů. Všechny díly dráhy jsou umístěny na dřevěných podpěrách, ke kterým jsou přišroubovány. Toto řešení zajistilo, že jednotlivé části dráhy mohly být vedeny v různých výškách. Dráha je vedena v minimální výšce 5 cm a maximální 22 cm nad podkladovou dřevotřískovou deskou a překonává převýšení přibližně 17 cm.

Na ostatních místech podkladové desky, kde není vedena dráha, jsou umístěny různě tvarované části pěnového polystyrenu. Ty jsou z horní strany opracovány a nabarveny tak, aby vytvářely dojem krajinného reliéfu. Na většině míst vystupuje tento reliéf nad dráhu a zakrývá prázdná místa a podpěry pod ní. V případě vnější dráhy také částečně zabraňuje vylétnutí autíčka při vyšších rychlostech. Na nejvyšším místě vedle rovného úseku dráhy pak dosahuje polystyrénový reliéf své maximální výšky 30 centimetrů.



Obr. 3: Celkový pohled na autodráhu

1.3 Pohyb autíčka po autodráze

Jelikož každý segment dráhy obsahuje dvě vodící drážky s příslušnými napájecími kontakty, může se autíčko po autodráze pohybovat po dvou nezávislých drahách. Nejproblematictější částí celé autodráhy je patrový úsek složený z krátkých zatáček umístěný za rovinkou. Mimo prudkého zakřivení zde dráha poměrně rychle klesá. Kvůli velkému převýšení a použití mnoha segmentů s prudkými zatáčkami, není autíčko schopno jet úsekem směrem nahoru. Tím je omezena možnost jeho pohybu po dráze pouze na jeden směr (proti směru hodinových ručiček).

Další omezení pohybu autíčka po drahách je způsobeno spoji mezi jednotlivými částmi autodráhy. Hlavně spoje na úsecích, které překonávají převýšení, obsahují mezery a autíčko se může při jejich přejezdu s nedostatečnou rychlostí zastavit. Podrobněji bude problematika možných rychlostí autíčka probrána v kapitole 3.

2. Možné způsoby zobrazení záznamu jízdy

2.1 Videokamera

Jedna z možností jak zobrazit jízdu autíčka po autodráze by spočívala v natočení jeho jízdy na videokameru. Podle požadavků záznamu by bylo možné následně přehrávání zpomalovat nebo zrychlovat. Toto řešení by po vizuální stránce vypadalo velice realisticky, ale přináší s sebou určité problémy.

Aplikace musí být schopna zobrazit jízdu z několika pohledů, což by se sice dalo vyřešit natočením jízdy z různých pozic, ale přepínání kamer během jízdy by se pravděpodobně řešilo dosti obtížně. Pozice autíčka by totiž byla na různých kamerách v daný časový okamžik jiná. Pohled pro každou kameru by se musel natáčet zvlášť a jízda autíčka by určitě nebyla vždy totožná. Dalším problémem by bylo zoomování, jelikož při větším zvětšení by byl obraz hodně nekvalitní. Z tohoto důvodu by bylo nutné video natočit v nejvyšším dostupném rozlišení a zpracovávat jej v nekomprimované podobě. I přes to, že dnešní běžně prodávané videokamery obsahují snímací čipy se třemi miliony snímacích bodů (3 MPix), výstupní rozlišení videa je maximálně 720x576 bodů (přibližně 415 tisíc snímacích bodů). To odpovídá zobrazovací normě PAL, která je dnes u televizních obrazovek nejrozšířenější. Zoomováním by se dal obraz v takovémto rozlišení při zachování relativně dobré kvality zvětšit maximálně dvakrát, což je například při celkovém pohledu na dráhu shora nedostačující.

Vysoké rozlišení a nekomprimovaná podoba videa by navíc znamenaly velké nároky na hardware počítače. Jedna minuta nekomprimovaného záznamu v rozlišení 720x576 při 25 snímcích za sekundu potřebuje více než 200 MB diskového prostoru. Při použití pouze čtyř pohledů na dráhu, by se paměťové nároky pohybovaly kolem 1GB dat na jednu dráhu autíčka. Navíc následné zpracovávání takového videa by bylo nesmírně náročné na výpočetní výkon počítače. Další nevýhodou tohoto řešení je, že natočené video je sice možné bez problému zrychlovat, ale při zpomalování bude mít méně než 25 snímků za sekundu a pohyb autíčka nebude plynulý. Tento problém by se pravděpodobně dal omezit natočením základního videa pro každou kameru při minimální rychlosti autíčka a záznam by se

pak při vyšších rychlostech pouze zrychloval. Takto by se neplynulost pohybu projevovala pouze při zastavování nebo rozjíždění autíčka.

Zajisté by se daly udělat určité kompromisy v kvalitě obrazu při zoomování a bylo by možné použít video v komprimované podobě, avšak problémy s určením polohy autíčka na jednotlivých pohledech kamer by se řešily obtížně. Také neplynulost pohybu autíčka při zastavování a rozjíždění by se nejspíše odstranit nepodařilo.

2.2 Fotoaparát

Další možnost zobrazení jízdy autíčka se podobá první, popsané v kapitole 2.1 a spočívá v nafocení jednotlivých snímků s autíčkem digitálním fotoaparátem. Následně by tyto snímky bylo možné zobrazit za sebou s takovou rychlostí, aby to odpovídalo požadované rychlosti jízdy. Toto řešení odstraňuje problémy s kvalitou obrazu při zoomování a omezuje nedostatek snímků při zastavování a rozjíždění autíčka (větším počtem snímků například na jeden centimetr dráhy lze docílit větší plynulosti pohybu). Odpadá zde také problém s určením polohy autíčka při přepínání kamer, protože při pořizování fotografií jednotlivých pohledů lze autíčko umístit vždy do stejné pozice.

Bohužel toto řešení zvyšuje problém paměťových nároků. V případě jednoho snímku s rozlišením 1 MPix uloženém v souboru o velikosti přibližně 100 KB, by při 10 snímcích na 1 centimetr bylo potřeba 1 MB paměti na centimetr dráhy. Délka kratší dráhy je přitom přibližně 5 m. V tomto případě by na jednu kameru a jednu dráhu bylo potřeba 500MB datového prostoru, což jsou při 4 kamerách a 2 drahách téměř 4GB. Navíc pro jeden pohled kamery by bylo třeba vytvořit 5000 snímků, což by bylo velice obtížné. Další zmenšování rozlišení fotografií kvůli snížení paměťových nároků nepřipadá v úvahu, protože tím by se rozlišení snímku dostalo na úroveň záznamu z videokamery. Pravděpodobně by bylo nutné snížit počet pořizovaných snímků, ale tím by se opět objevil problém s plynulostí přehrávání při malých rychlostech autíčka.

Dalším problémem pak je, jak jednotlivé snímky dostatečně rychle zobrazovat (minimálně 25x za sekundu). Řešením by bylo využít některé z 3D rozhraní: OpenGL nebo DirectX. Jednotlivé snímky by se pak daly zobrazovat jako textury. Problémem ale je, že soubory s texturami nemohou být v komprimovaném

grafickém formátu. Proto je nutné snímky převádět, protože uložení všech snímků přímo v nekomprimovaném formátu nepřipadá kvůli paměťovým nárokům v úvahu. Bohužel díky převodům by byl opět výsledný program velice náročný na výkon počítače.

2.3 3D aplikace

Poslední uvažovaná možnost spočívá ve vytvoření kompletního modelu autodráhy a autíčka v některém 3D modelovacím programu. Modely lze poté pomocí 3D rozhraní OpenGL nebo DirectX zobrazovat a vytvořit tak animaci pohybujícího se autíčka.

Toto řešení je však oproti předchozím náročnější, protože právě kompletní tvorba modelu celé autodráhy a autíčka zabere mnoho času. Dále je také nutné zjistit trajektorie obou drah autodráhy a příslušné natočení autíčka na nich. Souřadnice jednotlivých bodů trajektorie je nutné volit dostatečně blízko sebe, aby byl výsledný pohyb autíčka dostatečně plynulý i při malých rychlostech.

Naopak výhody tohoto řešení jsou zřejmé: Odpadají zde problémy se zobrazováním autodráhy z různých pohledů. Pokud je totiž vytvořen kompletní 3D model, lze zobrazit libovolný pohled na autodráhu pouze správným nastavením kamery v příslušném 3D rozhraní. Tímto způsobem lze získat například kameru z místa pohybujícího se autíčka. Také zoomování nepředstavuje žádný problém, jelikož to se také provede pomocí příslušného nastavení kamery. Při dostatečném počtu bodů trajektorie také odpadá neplynulost pohybu autíčka v případě zastavování a zrychlování.

2.4 Shrnutí jednotlivých řešení zobrazení jízdy

Způsob řešení	Klady	Zápory
1) Videokamera	<ul style="list-style-type: none">- realistické zobrazení autodráhy	<ul style="list-style-type: none">- větší paměťové a hardwarové nároky- obtížné zoomování- problém s plynulostí při zastavování a rozjíždění autíčka- omezený počet pohledů na dráhu
2) Fotoaparát	<ul style="list-style-type: none">- výborná obrazová kvalita a realističnost- snadné zoomování	<ul style="list-style-type: none">- velké hardwarové a paměťové nároky v řádech GB- omezený počet pohledů na dráhu
3) 3D aplikace	<ul style="list-style-type: none">- libovolný pohled na autodráhu- libovolné zoomování- plynulost pohybu při zastavování a rozjíždění autíčka	<ul style="list-style-type: none">- časová náročnost vytváření modelu- nutnost zjistit trajektorie autíčka na jednotlivých drahách a jeho natočení

Tab. 1: Klady a zápory jednotlivých způsobů zobrazení jízdy autíčka

Po zhodnocení výhod a nevýhod jednotlivých způsobů zobrazení jízdy autíčka po autodráze, byla vybráno řešení spočívající ve vytvoření 3D aplikace. I přes větší náročnost při vytváření kompletního 3D modelu autodráhy a nutnosti zjišťovat trajektorie obou drah, poskytuje tato možnost nejlepší kontrolu nad pohybem autíčka a zobrazením celé autodráhy.

3. Zjištění rychlosti autíčka v závislosti na napětí

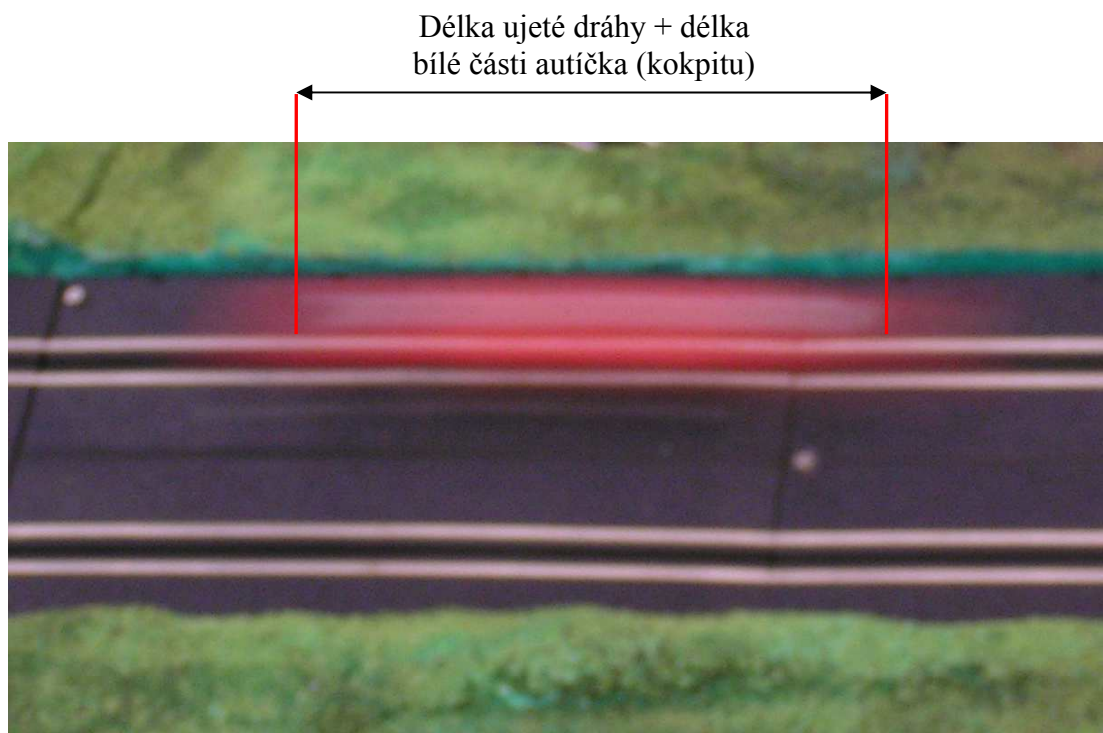
Aby bylo možné korektně zobrazit pohyb autíčka po autodráze, bylo nejprve nutné zjistit velikost rychlostí autíčka v závislosti na velikosti napájecích napětí a vytvořit tak převodní funkci elektrického napětí na rychlost. Tato převodní funkce byla zjišťována dvěma způsoby.

3.1 Digitální fotoaparát

První způsob zjištění rychlosti autíčka je pomocí digitálního fotoaparátu. Fotoaparát byl umístěn na stativu a byl přepnut do manuálního režimu. Cílem bylo zachycení dostatečné části trajektorie pohybujícího se autíčka. Toho bylo docíleno tak, že na fotoaparátu byla nastavena dostatečně dlouhá doba expozice a vyšší hodnota clony tak, aby se autíčko na fotografii zobrazilo rozmazané. Z rozdílu rozmazané dráhy a skutečné délky autíčka lze poté určit ujetou dráhu.

Jediné vhodné místo pro toto měření se vyskytuje v horní části dráhy, kde se nachází rovinka skládající se ze tří rovných segmentů o celkové délce 84 cm. Ze znalosti doby expozice a dráhy určené z fotografie byla poté počítána rychlost autíčka.

Již při pořizování snímků bylo zřejmé, že okraje autíčka jsou vůči dráze nevýrazné a na fotografiích nebude možné přesně určit počátek a konec dráhy autíčka, protože díky rozmazání budou v těchto místech splývat barvy autíčka a dráhy. Tento předpoklad se potvrdil a nepomohla ani volba autíčka s výraznější červenou barvou oproti barvě modré. Proto bylo nakonec využito výrazné bílé barvy, kterou je v autíčku znázorněn řidič. Navíc jeho původně žlutá helma byla obalena hliníkovou fólií, aby byla na fotografii ještě výraznější. Výsledky sice nebyly ideální, ale počátek a konec dráhy autíčka byl již na fotografiích znatelný (viz Obr. 4).



Obr. 4: Fotografie pohybujícího se autíčka (doba expozice: $\frac{1}{4}$ s)

Při zpracování pořízených fotografií se ale ukázalo, že měření pro jednotlivá napětí jsou dosti nepřesná. Hlavně při vyšších hodnotách napětí, vycházely velikosti rychlostí dosti neočekávaně. Místo aby se rychlosti zvyšovaly, docházelo k jejich snižování. Nepřesnosti byly pravděpodobně způsobeny úsekem dráhy, kde byly pořizovány fotografie. Jak je popsáno výše, jedná se o jediný rovný úsek na celé autodráze. Bohužel při vyšších hodnotách napětí jeho délka 84 centimetrů nestačí na to, aby se autíčko mohlo rozjet na danou rychlost a setrvat na tomto úseku dostatečně dlouhou dobu pro pořízení fotografie. Vyfotografovat autíčko na tomto úseku, bylo totiž velice obtížné. Při vyšších hodnotách napětí, se pohybovalo rychlostmi více než 1 m/s a na stisknutí spouště ve správný časový okamžik bylo k dispozici velice málo času. Proto bylo nutné, aby autíčko dosáhlo požadované rychlosti již před rovinkou. Avšak před vjezdem na tento rovný úsek, je umístěna zatáčka, kvůli které se autíčko při vyšších rychlostech hodně smýkalo. To následně způsobovalo snižování rychlosti na měřeném úseku.

Problémem tohoto řešení také je, že takto změřených rychlostí je autíčko schopno dosáhnout jenom na krátkém rovném úseku dráhy. V ostatních částech autodráhy budou pravděpodobně rychlosti z důvodu smýkání autíčka nižší.

Vzhledem k tomu, že větší část dráhy je tvořena těmito nerovnými úseky, bylo by lepší znát rychlosti na těchto úsecích. Bohužel změření rychlostí na nerovných úsecích by bylo pomocí fotoaparátu velice obtížné, ne-li zcela nemožné.

Kvůli výše popsaným problémům, proto nebyl nakonec tento způsob zjištění převodní funkce elektrického napětí na rychlost využit.

3.2 Průměrná rychlost

Druhý způsob zjišťování rychlosti autíčka spočívá v určení jeho průměrné rychlosti na celé dráze oproti určování rychlosti na jediném rovném úseku. Jak je popsáno v prvním způsobu řešení, největší rychlosti dosahuje autíčko na rovném úseku dráhy, protože zde není zpomalováno smýkáním a nárazy v zatáčkách. Tento úsek je však oproti zbylým částem poměrně krátký. Proto je výhodnější zjistit příslušné rychlosti autíčka pro ostatní části dráhy a rychlost na rovném úseku určitým způsobem oproti nim zvýšit (například vhodně zvoleným koeficientem, kterým se vynásobí rychlosti dosahované na tomto úseku). Jelikož ale byly zjišťovány průměrné rychlosti autíčka pro celé jedno ujeté kolo, budou muset být velikosti rychlostí korigovány i na ostatních částech dráhy. Korekce rychlostí na nerovných úsecích bude však zcela minimální, protože dráha se skládá z větší části právě z těchto úseků. Pokud je tedy zvýšena rychlost na krátkém rovném úseku, její snížení na ostatních úsecích je mnohem menší.

Všechna měření byla provedena pro obě dráhy autodráhy zvlášť. Protože vnitřní dráha (dráha 1) je oproti vnější (dráha 2) kratší a obě dráhy se liší pouze v délkách zakřivených částí, průměrné rychlosti na jedno kolo jsou různé.

Při určování rychlostí byl použit regulovatelný zdroj a stopky. Zjišťována byla doba, kterou autíčko potřebuje pro ujetí celého kola při určité hodnotě napájecího napětí. Aby byly omezeny nepřesnosti při odečítání času, autíčko ujelo tři celá kola a byla měřena doba k tomu potřebná. Z této hodnoty byl poté určen průměrný čas potřebný pro ujetí jednoho kola. Pro výpočet příslušných rychlostí, bylo dále potřeba znát délku obou drah, po kterých se autíčko může pohybovat. Aby se poloha autíčka v modelu přiblížila reálnému umístění, pro výpočet rychlosti byly použity délky drah získané z modelu autodráhy, které jsou oproti skutečným délkám drah menší. Tím je však zajištěno, že model autíčka při dané hodnotě napětí, ujede celé jedno kolo za stejnou dobu, která byla změřena na skutečné autodráze. Rychlost

zobrazovaná v aplikaci je však přepočtena tak, aby odpovídala skutečným délkám drah reálné autodráhy. Délky obou drah z modelu autodráhy jsou uvedeny v tabulce 2.

Dráha:	Délka drah (model) [cm]	Délka drah (skutečnost) [cm]
1 (vnitřní)	510,4	515
2 (vnější)	628,8	635

Tab. 2: Délky drah autodráhy

Pro každou změřenou hodnotu napájecího napětí byl proveden výpočet příslušné průměrné rychlosti na jedno kolo autíčka ze znalosti délky dráhy a průměrného času potřebného na odjetí jednoho kola podle vztahu:

$$v = \frac{d}{t_1}$$

d ... délka příslušné dráhy (dráha 1, dráha 2)

t_1 ... průměrná doba potřebná k ujetí jednoho kola

V tabulce 3 a tabulce 4 jsou zobrazeny hodnoty nastavených napájecích napětí společně se změřenými časy potřebnými pro ujetí třech kol. Dále jsou v tabulkách uvedeny průměrné doby potřebné pro ujetí jednoho kola a nakonec vypočtená průměrná rychlost v centimetrech za sekundu.

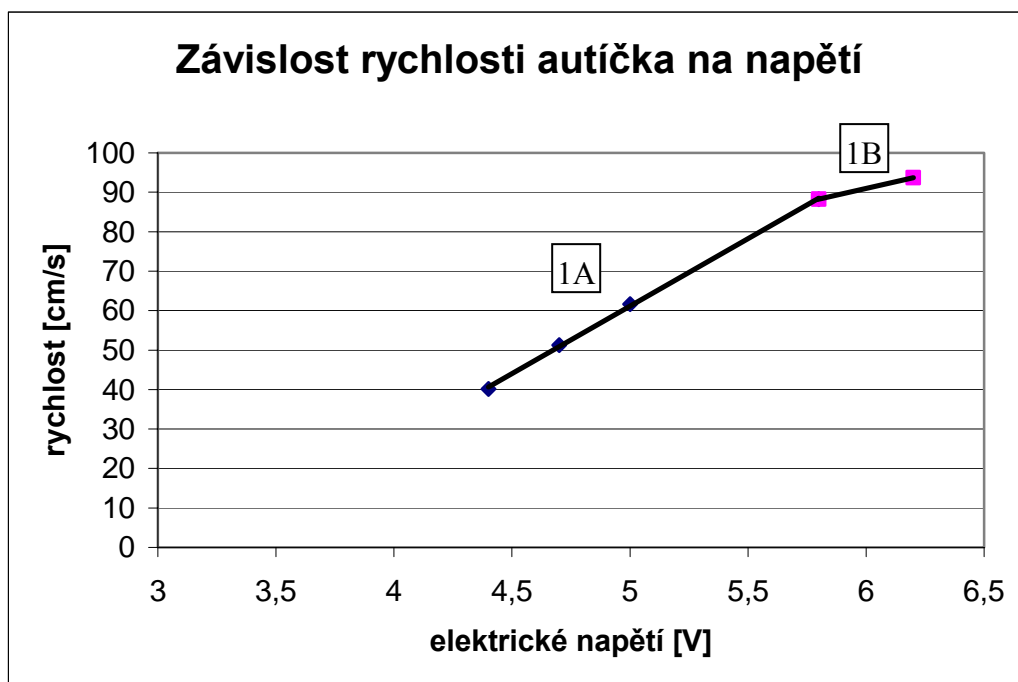
U [V]	t_3 /3 kola [s]	t_1 /1 kolo[s]	v [cm/s]
4,4	38,18	12,73	40,11
4,7	29,87	9,96	51,26
5	25,04	8,28	61,63
5,8	17,36	5,79	88,20
6,2	16,35	5,45	93,65

Tab. 3: Hodnoty napětí a rychlostí pro dráhu 1

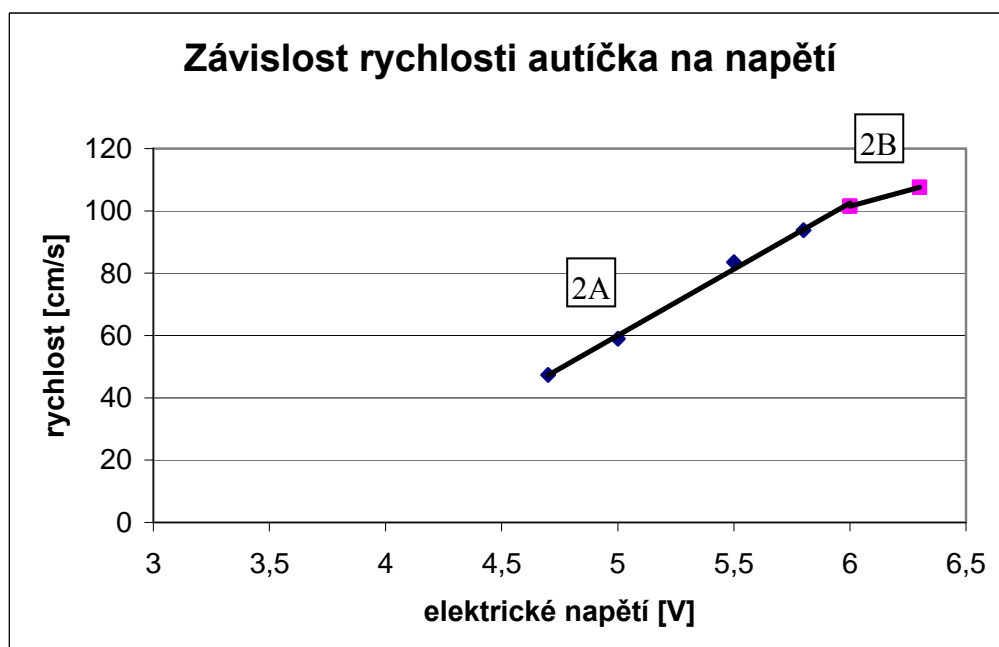
U [V]	t_3 /3 kola [s]	t_1 /1 kolo[s]	v [cm/s]
4,7	39,88	13,29	47,31
5,0	32,0	10,67	58,95
5,5	22,58	7,53	83,51
5,8	20,10	6,70	93,85
6,0	18,58	6,19	101,53
6,3	17,53	5,84	105,98

Tab. 4: Hodnoty napětí a rychlostí pro dráhu 2

Z vypočtených hodnot rychlostí, byl pro každou dráhu vytvořen graf závislosti rychlosti autíčka na jeho napájecím napětí (graf 1 a graf 2).



Graf 1: Základní převodní funkce el. napětí na rychlost pro vnitřní dráhu (dráha 1)



Graf 2: Základní převodní funkce el. napětí na rychlost pro vnější dráhu (dráha 2)

Z grafů je patrné, že rychlost autíčka je do určité hodnoty elektrického napětí závislá lineárně. Pro každou dráhu proto byly pomocí metody nejmenších čtverců určeny rovnice dvojice přímk (v grafech úseček), které aproximují průběh závislosti rychlosti na napětí. Z těchto rovnic bude pak v aplikaci sestaven základ převodní funkce elektrického napětí na rychlost autíčka. Rovnice přímk jsou popsány níže.

Dráha 1: **(1A):** $y = 34,16x - 109,64$ (pro el. napětí $U < 5,8 \text{ V}$)

(1B): $y = 13,62x + 9,20$ (pro el. napětí $U \geq 5,8 \text{ V}$)

Dráha 2: **(2A):** $y = 42,47x - 152,28$ (pro el. napětí $U < 6 \text{ V}$)

(2B): $y = 20,27x - 20,10$ (pro el. napětí $U \geq 6 \text{ V}$)

Dále bylo třeba změřit limitní napětí při kterých se autíčko rozjede. Na vodorovných a dolů se svažujících úsecích je autíčko schopno rozjet se při napětí 3,3V. Avšak na stoupajících úsecích je potřeba napětí 5 V, aby se autíčko dalo do pohybu. Komplikací je, že pokud se již autíčko pohybuje, stačí mu pro jízdu po takovéto části dráhy napětí nižší. Prakticky bylo zjištěno, že autíčko je schopno celou dráhu projet při napětí 4,4V. Při napětí nižším, není schopno vyjet stoupající část dráhy. Taktéž není schopno sjet klesající úsek za rovinkou, protože v těchto místech jsou klesání i poloměr zatáčky hodně prudké. Kvůli tomu, že jednotlivé části, ze kterých je autodráha postavena, do sebe úplně přesně nezapadají, dochází při přejezdů zlomů autíčkem ke ztrátě dotyku sběrných kartáčků s kontakty na dráze. V případě, že se autíčko pohybuje přes tato místa pomalu, dojde ke ztrátě napájení motorku a autíčko se zastaví.

Také bylo zjišťováno napětí, při kterém je autíčko schopno celou dráhu projet, aniž by z dráhy vylétlo. Toto napětí je pro obě dráhy různé. Podle předpokladu musí mít autíčko nižší rychlost pokud se pohybuje ve vnitřní dráze (dráha 1). Místa s největší pravděpodobností vyjetí jsou těsně za rovinkou a dále pak v celé navazující části dráhy, kde se vyskytuje prudké zakřivení a strmé klesání. V případě vnitřní dráhy (dráha 1) je maximální napětí 6,2 V. Při zvýšení napájecího napětí nad tuto úroveň autíčko z dráhy vylétne. Pokud autíčko jede po vnější dráze, lze použít vyšší napájecí napětí s hodnotou 6,3V.

Všechny výše uvedené skutečnosti komplikují převodní funkci napájecího napětí na rychlost autíčka. Při přehrávání záznamu jízdy se pro stanovení aktuální rychlosti autíčka bude muset brát v úvahu nejen velikost napětí v daném čase, ale také aktuální poloha na dráze, po které se autíčko pohybuje (dráha 1 nebo dráha 2).

4. Záznam jízdy autíčka

4.1 Úprava jízdy autíčka

Aby přehrávaná jízda odpovídala skutečné jízdě autíčka, nebude stačit brát v úvahu pouze převodní funkci popsanou v kapitole 3.2. Převodní funkce pro každou dráhu zohledňuje aktuální napájecí napětí autíčka a charakter dráhy. Jelikož se ale napětí může měnit skokově, je potřeba spočtené skokové změny rychlostí eliminovat. Pro tento účel budou upraveny hodnoty rychlostí tak, aby autíčko akcelerovalo a zpomalovalo plynule.

Další úpravou jízdy bude zajištěno, že autíčko ujede pouze takovou dráhu, která byla zaznamenána. Kvůli tomu budou muset být upraveny hodnoty všech zaznamenaných rychlostí. Pokud by nedošlo k takovéto úpravě, tak vzhledem k nepřesnostem v převodní funkci, eliminaci skokových změn rychlosti a nepředvídatelnému chování autíčka při průjezdech některými částmi dráhy, by jízda autíčka neodpovídala skutečnosti.

Pro snadné určení skutečně ujeté dráhy a následný přepočet rychlostí je nutné, aby v souboru se záznamem jízdy byla uložena data z průjezdu jednoho nebo více celých kol. Dále je nutné, aby při záznamu jízdy autíčko zůstalo v dráze, protože v případě vylétnutí autíčka nebo jeho zaseknutí, není možné pouze z velikostí napájecích napětí toto zjistit.

4.2 Struktura datového souboru

Záznam jízdy autíčka je uložen v datovém souboru, který má textový formát. Na každém řádku je vždy uveden jeden údaj. Na prvním řádku se musí vyskytovat textový řetězec *JIZDA*. V případě výskytu tohoto řetězce program zjistí, zda je načten soubor správného formátu se záznamem jízdy. Druhou hodnotou v souboru je perioda v sekundách, s jakou jsou snímány jednotlivé údaje o napětích. Třetí (celočíslná) hodnota reprezentuje číslo dráhy, po které autíčko jede (1 nebo 2). Čtvrtý řádek obsahuje hodnotu, která odpovídá maximálnímu dosažitelnému napětí příslušného zdroje autodráhy ve voltech. Na pátém řádku je uvedena celočíslná

hodnota, která udává pro kolik celých kol jsou v souboru uložena data se záznamem jízdy.

Následují data, popisující velikosti napětí v čase. Velikost napětí je reprezentována čísly od 0 do 255 . 0 odpovídá nulovému napětí a hodnota 255 maximálnímu dosažitelnému napětí příslušného zdroje autodráhy.

JIZDA	... kontrolní textový řetězec
0.04	... perioda s jakou jsou snímána napětí
1	... číslo dráhy, po které se autíčko pohybuje
10.0	... napájecí napětí autodráhy
1	... počet zaznamenaných okruhů v souboru
0	... první hodnota reprezentující velikost napětí
0	.
80	.
153	.
159	.
.	
.	
.	

Obr. 5: Příklad počátečních hodnot datového souboru se záznamem jízdy

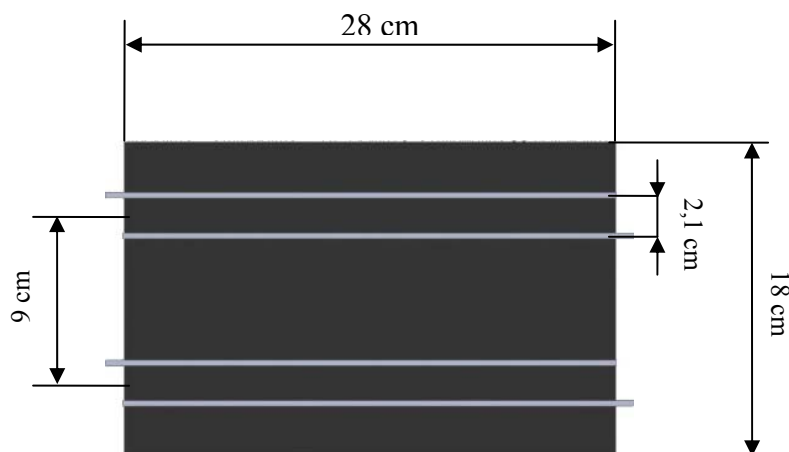
5. Tvorba modelů autodráhy a autíček

5.1 Software pro 3D modelování

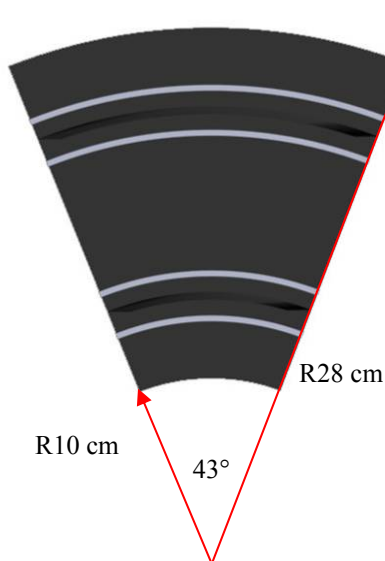
Pro 3D modelování byl použit software 3D Studio Max. Jedná se o jeden z nejrozšířenějších programů pro vytváření 3D modelů a animací. Modely autodráhy a autíček byly vytvořeny ve dvou verzích programu: 3D Studio Max 6.0 a 3D Studio Max 7.0. Jedná se sice o dnes již neaktuální verze tohoto programu, ale jsou k dispozici jako trial verze a každou z nich lze zdarma používat třicet dní.

5.2 Vytvoření 3D modelu autodráhy

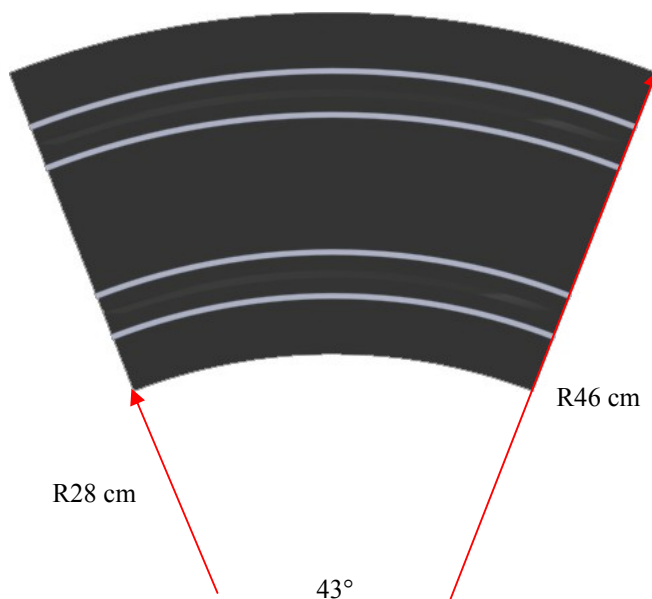
Pro vytvoření modelu odpovídající reálné podobě autodráhy, bylo nutné zjistit její přesné rozměry a pozice jednotlivých částí, ze kterých je postavena. Nejprve byla provedena měření vnějších půdorysných rozměrů autodráhy. Dále bylo třeba zjistit počet, rozměry a zakřivení jednotlivých typů segmentů, ze kterých je dráha postavena. Všechny segmenty mají tloušťku 0,5 cm a jejich tvary jsou zobrazeny na následujících obrázcích.



Obr. 6: „Rovný segment“

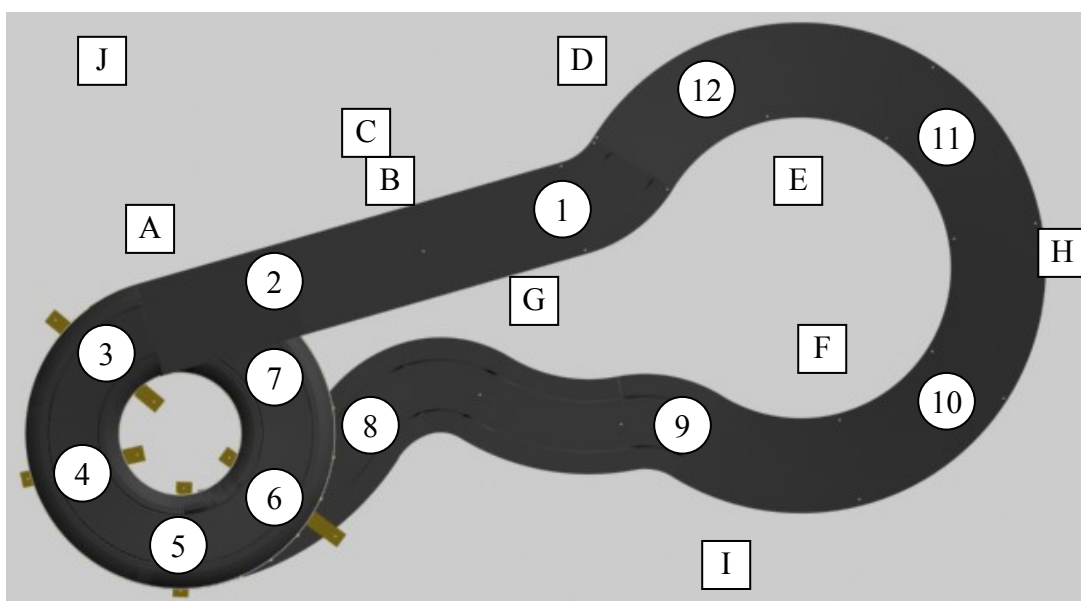


Obr. 7: „Krátká zatáčka“



Obr 8: „Dlouhá zatáčka“

Jelikož se na dráze vyskytuje mnoho výškových rozdílů, bylo také nutné změřit výšky, ve kterých se jednotlivé segmenty nacházejí. Také bylo nutné určit výšky polystyrénových částí, které vyplňují místa u dráhy a tvoří okolní reliéf. Všechny výšky jsou změřeny od podkladové desky, na které je autodráha postavena.



Obr. 9: Schematické označení různých výšek autodráhy

Dráha		Okolní reliéf	
Umístění	Výška [cm]	Umístění	Výška [cm]
1	19,5	A	34
2	22,5	B	24
3	22	C	19
4	19	D	21
5	17	E	20
6	15	F	15
7	13	G	23
8	6	H	20
9	5	I	10
10	13	J	10
11	18,5		
12	19		

Tab. 5: Hodnoty výšek jednotlivých částí autodráhy nad podkladovou deskou

Pomocí digitálního fotoaparátu byly pořízeny fotografie celé autodráhy shora a fotografie jednotlivých částí dráhy, aby podle nich bylo možné vytvořit některé vizuální detaily (umístění viditelných podpěr dráhy, tvar a barvu reliéfu).

Původním záměrem bylo vymodelování všech segmentů, ze kterých je dráha postavena, a jejich umístění v modelu tak, aby vytvořily celou dráhu jako v případě opravdové autodráhy. Protože však dráha na svých dvou zakřivených částech překonává velké výškové rozdíly, jsou v těchto místech jednotlivé pospojované díly různě zdeformovány a natočeny. První část, která je postavena z dvanácti segmentů krátkých zatáček a vytváří dvě patra dráhy byla v 3D Studiu Max nahrazena jedním 3D objektem ve tvaru pružiny (obrázek 10), jehož parametry: tvar, šířka, výška a poloměr odpovídají zjištěným parametrům dráhy. Druhá část dráhy, která překonává výškový rozdíl a tvoří ji původně šest segmentů dlouhých zatáček, je rovněž nahrazena pomocí objektu ve tvaru pružiny (obrázek 11).



Obr. 10: Patrový úsek autodráhy



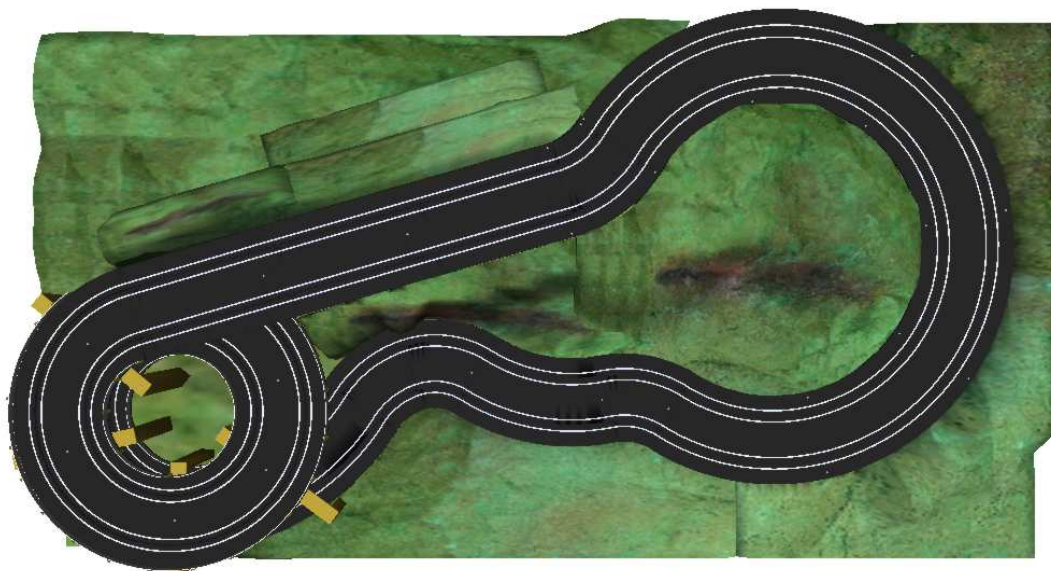
Obr. 11: Úsek dráhy s výškovým rozdílem

Ostatní části modelu dráhy jsou pak složeny ze samostatných segmentů jako v případě skutečné autodráhy. Celkem je tedy model trati vytvořen z osmi nezávislých 3D objektů. Každý z nich se dále skládá z dalších pěti objektů: základní černé části s vodící drážkou pro autíčko a čtyřech lesklých částí, které reprezentují napájecí kontakty pro každou ze dvou drah pro autíčka.

Reliéf kolem dráhy je tvořen celkem osmi 3D objekty. Aby model odpovídal reálné podobě autodráhy, je každý z objektů opatřen vlastní texturou. Textury jsou vytvořeny z digitálních fotografií autodráhy.

Dalšími objekty, které se v modelu vyskytují, jsou celkem čtyři viditelné podpěry, podírající dráhu v její dvoupatrové části. Kolem této části je z vnější a vnitřní strany vytvořeno bílé zábradlí včetně žlutých úchytů k dráze. Posledními 3D objekty, které dotvářejí celkový vzhled autodráhy, jsou modely šroubů, kterými jsou přichyceny jednotlivé segmenty dráhy k podpěrným částem.

Všechny části autodráhy jsou umístěny na podkladové dřevotřískové desce černé barvy s půdorysnými rozměry 200×80 cm a tloušťkou 2,5 cm. V případě 3D modelu autodráhy je tato deska místo černé barvy opatřena texturou s povrchem reliéfu. Tímto řešením bylo zamezeno splývání barev dráhy a viditelného podkladu při pohledu shora.

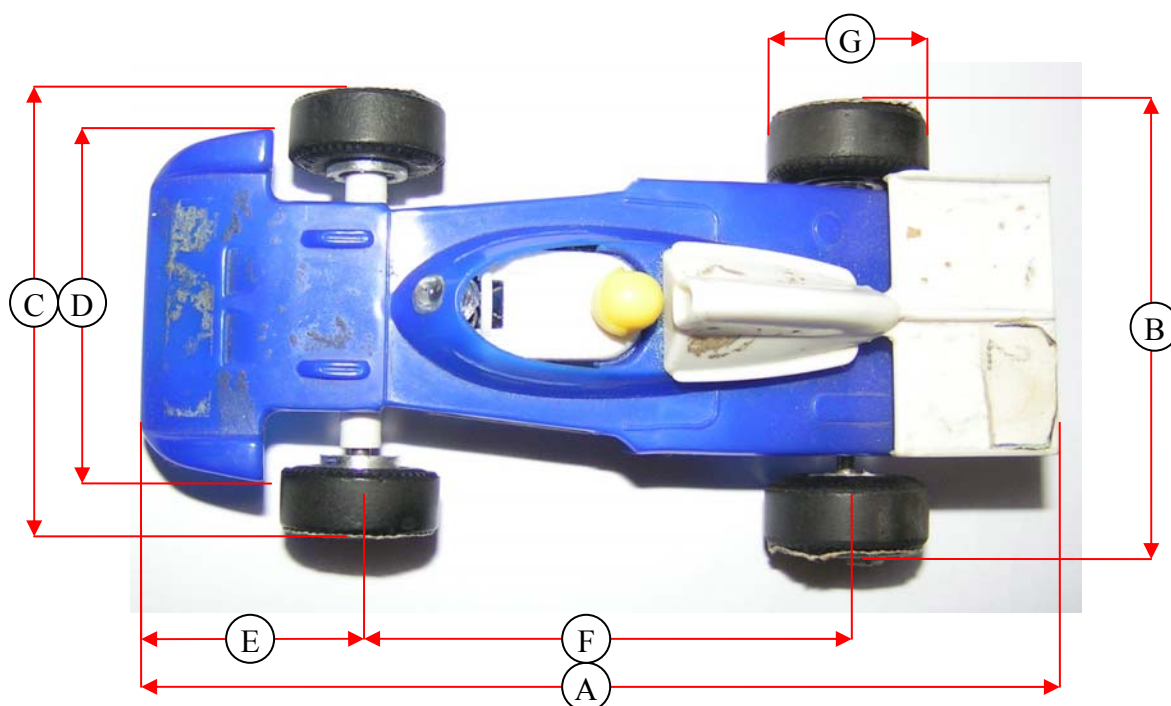


Obr. 12: Celkový pohled na 3D model autodráhy

5.3 Vytvoření 3D modelů autíček

Pro zobrazení jízdy byly vytvořeny 3D modely dvou autíček, která se mohou po dráze pohybovat. Tvarově a rozměrově se jedná o shodné modely, lišící se pouze svojí barvou a přítomností bílého přitlačného křídla. První autíčko je celé červené a neobsahuje žádné přitlačné křídlo. Druhé autíčko má modrou barvu a v zadní části za kokpitem je na něm umístěno bílé přitlačné křídlo. Pomocí digitálního fotoaparátu byla obě autíčka vyfotografována, aby je bylo možné podle fotografií a ze znalosti zjištěných rozměrů vymodelovat.

Podobně jako u vytváření modelu autodráhy, bylo pro správnou velikost 3D modelu nejprve nutné zjistit celkové rozměry skutečných autíček, aby jejich následné zobrazení na dráze odpovídalo skutečnosti. Fotografie jednoho z autíček včetně jeho rozměrů je zobrazena na obrázku 13. Vybrané rozměry jsou pak uvedeny v tabulce 6.

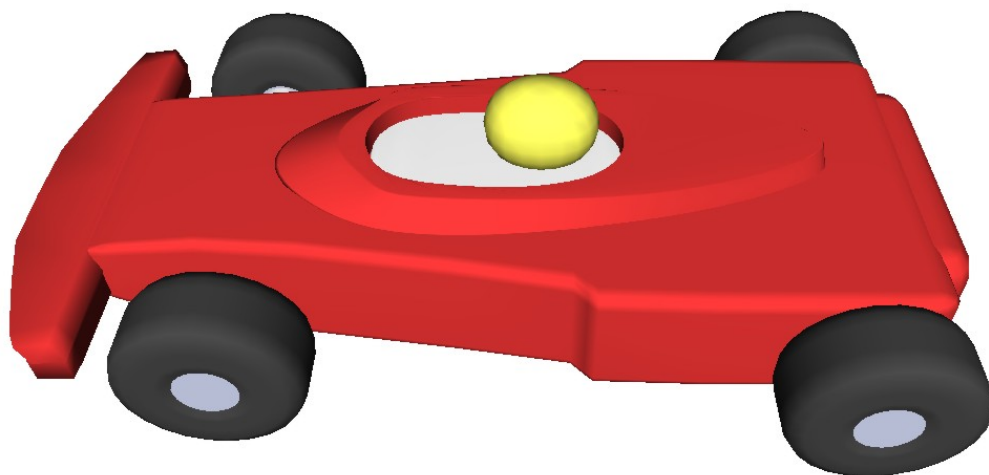


Obr. 13: Fotografie autíčka 2

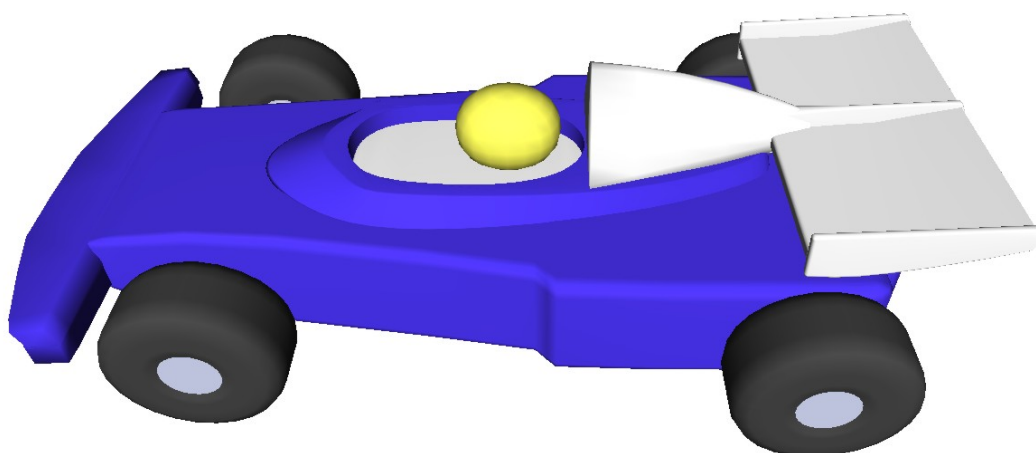
Pozice	Rozměr [cm]
A	13,5
B	7,0
C	6,5
D	5,0
E	3,5
F	7,0
G	2,3
Max. výška	5,0

Tab. 6: Rozměry vybraných částí autíčka

3D model červeného autíčka se skládá z osmi samostatných objektů: předního přitlačného křídla, samotného skeletu autíčka, kokpitu pro řidiče, čtyřech kol a hlavy řidiče. Model modrého autíčka se skládá ze stejných objektů, ale navíc obsahuje zadní bílé přitlačné křídlo. Oba 3D modely autíček jsou zobrazeny na obrázku 14 a obrázku 15.



Obr. 14: 3D model autíčka 1



Obr. 15: 3D model autíčka 2

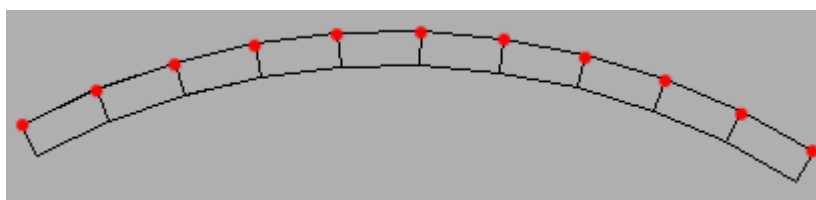
6. Získání trajektorií obou drah autodráhy

6.1 Načtení trajektorie

Aby bylo možné zobrazit jízdu autíčka po obou drahách umístěných na autodráze, bylo nutné získat příslušné trajektorie obou drah. Každá trajektorie se skládá z určitého počtu bodů. Souřadnice těchto bodů byly zjišťovány dvěma různými způsoby.

6.1.1 Přepis souřadnic trajektorie

První způsob spočívá v ručním přepisu souřadnic jednotlivých bodů trajektorie dráhy přímo z modelovacího programu 3D Studio Max. V 3D modelu autodráhy byly nejprve vytvořeny pomocné objekty, které byly umístěny tak, že vytvořily trajektorii dané dráhy. Každý objekt odpovídal jednomu z osmi segmentů, ze kterých je dráha vymodelována (viz kapitola 5). Každý objekt je dále tvořen více částmi (více body), aby bylo možné určit body dané trajektorie. Jednotlivé objekty jsou trojrozměrné útvary, proto je pro uzly trajektorie brán vždy každý čtvrtý bod. Na obrázku 16 je zobrazen objekt v pohledu shora s vyznačenými body trajektorie (na obrázku spodní část objektu splývá s horní).



Obr. 16: Příklad pomocného objektu s vyznačenými body trajektorie

Pro přepis bodů trajektorie byl vytvořen velice jednoduchý pomocný program, do kterého je možné zadat souřadnice jednotlivých bodů trajektorie. V modelovacím programu byly postupně označovány body trajektorie a jejich prostorové souřadnice (x, y, z) zadávány do pomocného programu. Výstupem programu je textový soubor, ve kterém jsou uvedeny souřadnice bodů trajektorie a jejich počet. Pro vnitřní dráhu autodráhy (dráha 1) bylo takto získáno 190 bodů.

Tímto způsobem lze sice vytvořit kompletní trajektorie obou drah, ale při jakékoli změně v 3D modelu autodráhy (posunutí, rotace) se musí celá trajektorie znova vytvořit. Dalším problémem tohoto řešení je časová náročnost přepisování souřadnic jednotlivých bodů.

6.1.2 Načtení souřadnic trajektorie

V tomto řešení je rovněž využito pomocných objektů, které vytvářejí trajektorii dráhy. Pomocné objekty pro každou dráhu zvlášť jsou v modelovacím programu 3D Studio Max uloženy do souboru formátu 3DS (viz kapitola 8). Pro zpracování 3DS formátu je použita unita pro vývojové prostředí Delphi *f_3DS*, která umožňuje načítat strukturu tohoto formátu souboru a uložit tak informace o všech 3D objektech do datových struktur (viz kapitola 8).

Každý pomocný objekt trajektorie má specifické jméno obsahující číslo, aby bylo možné objekty načíst ve správném pořadí tak, jak jdou na dráze za sebou. Aby na sebe navíc navazovaly počátky a konce trajektorií jednotlivých segmentů, je nutné body některých objektů načítat v obráceném pořadí. Toto je způsobeno tím, že se v 3D Studiu Max nedá zjistit, který bod je u 3D objektu počáteční. Jelikož pomocné objekty pro určení trajektorie vznikly úpravou objektů, které tvoří model autodráhy, nebylo je možné z důvodu natočení a zakřivení jednoduše otočit. Pořadí načítání bodů tak určuje první znak v názvu příslušného objektu. Pokud je první znak „T“, body objektu jsou načteny v normálním pořadí od prvního k poslednímu. V případě znaku „S“ jsou body naopak načteny od posledního k prvnímu.

Toto řešení odstraňuje nutnost znova vytvářet všechny body trajektorie při změnách v 3D modelu autodráhy. V případě změny stačí pouze upravit pomocný objekt trajektorie u příslušného změněného segmentu dráhy, popřípadě všechny objekty trajektorie posunout nebo otočit a uložit je do příslušného souboru formátu 3DS. Další výhodou je, že počet bodů trajektorie lze jednoduše zvýšit. Stačí totiž pouze vytvořit pomocné objekty skládající se z většího počtu segmentů.

Pro načtení trajektorií obou drah v 3D aplikaci zobrazující pohyb autíčka po autodráze bylo zvoleno toto řešení.

6.2 Úprava načtených trajektorií drah

Po načtení jednotlivých uzlových bodů trajektorie, bylo nutné dopočítat její další body, aby se 3D model autíčka mohl pohybovat po dráze plynule i malými rychlostmi. V různých částech dráhy je navíc mezi uzly trajektorie různá vzdálenost. Také přechody mezi jednotlivými úseky dráhy na sebe přesně nenavazují a jsou mezi nimi různě velké vzdálenosti. Při umísťování autíčka pouze na tyto body, by se tak pohybovalo v různých částech dráhy jinými rychlostmi.

Úseky mezi jednotlivými body jsou dalšími dopočtenými body lineárně rozděleny na více částí. Optimální vzdálenost mezi jednotlivými body je zvolena na 0,05 cm. Při 25 snímcích za sekundu se tak může autíčko plynule pohybovat rychlostí 1,25 cm/s. Reálně se však autíčko pohybuje menšími rychlostmi pouze krátký časový okamžik po rozjetí a před zastavením. Další zmenšování vzdáleností mezi jednotlivými body trajektorie tak nemá smysl, protože by se projevilo pouze většími paměťovými nároky.

Dráha:	Počet bodů trajektorie	Počet bodů trajektorie po interpolaci
1 (vnitřní)	686	10209
2 (vnější)	818	12577

Tab. 7: Počet uzlových a dopočtených bodů trajektorie obou drah

7. Rozhraní 3D grafiky

7.1 Dostupná 3D rozhraní

V současné době existují 2 rozšířená rozhraní, pomocí kterých lze v reálném čase vykreslovat 3D grafiku: OpenGL a Direct3D (součást DirectX). Tato rozhraní mají širokou hardwarovou i softwarovou podporu na většině dnes běžně prodávaných grafických akcelerátorech.

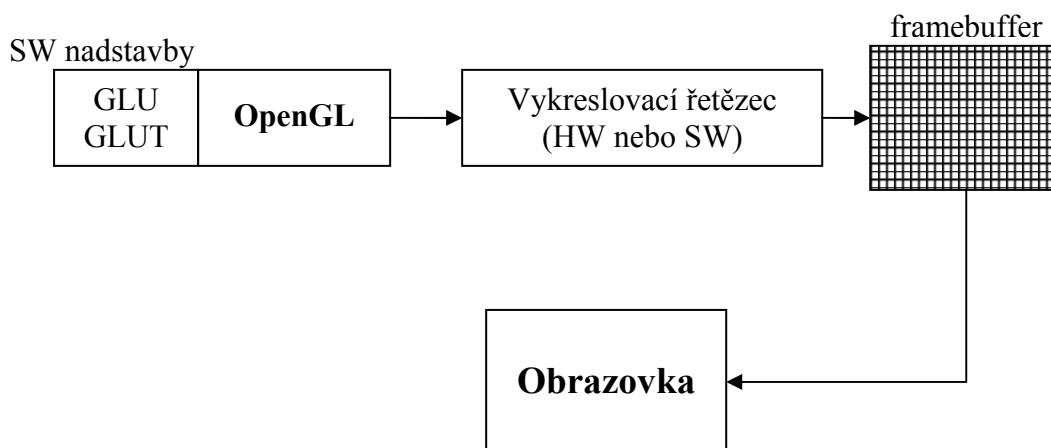
7.2 Popis rozhraní OpenGL

Standard OpenGL (Open Graphics Library) byl navržen firmou SGI (Silicon Graphics Inc.) v roce 1992. Jedná se o aplikační programové rozhraní k akcelerovaným grafickým kartám a celým grafickým subsystémům. Knihovna OpenGL byla navržena s důrazem na to, aby byla použitelná na různých typech grafických akceleratorů a aby ji bylo možno použít i v případě, že na určité platformě žádný grafický akcelerator není nainstalován (použita softwarová emulace). Knihovna OpenGL (narozdíl od Direct3D) je navržena tak, aby ji bylo možné použít na různých platformách. V současné době ji lze použít v unixových systémech (například Linuxu), OS/2 a v operačních systémech Microsoft Windows.

Knihovna OpenGL byla vytvořena tak, aby byla nezávislá na použitém operačním systému, grafických ovladačích a správcích oken (Window Managers). Kvůli tomu neobsahuje žádné funkce pro práci s okny (otevírání, zrušení, změnu velikosti), pro vytváření grafického uživatelského rozhraní ani pro zpracování událostí. Tyto funkce lze zajistit buď přímo voláním funkcí příslušného správce oken, nebo lze použít například nadstavbovou knihovnu GLUT (OpenGL Utility Toolkit). Pro dosažení co největší nezávislosti na použité platformě zavádí knihovna OpenGL vlastní základní datové typy (například GLbyte, GLint nebo GLdouble).

Programátorské rozhraní knihovny OpenGL je vytvořeno tak, aby knihovna byla použitelná v téměř libovolném programovacím jazyce. Primárně je k dispozici hlavičkový soubor pro jazyky C a C++. V tomto souboru jsou deklarovány nové datové typy používané knihovnou, také symbolické konstanty a sada cca 120 funkcí tvořících vlastní rozhraní. Existují však i podobné soubory s deklaracemi pro další programovací jazyky, například Fortran, Object Pascal či Javu.

Z programátorského hlediska se OpenGL chová jako stavový automat. To znamená, že během zadávání příkazů pro vykreslování lze průběžně měnit vlastnosti vykreslovaných primitiv (barva, průhlednost) nebo celé scény (volba způsobu vykreslování, transformace) a toto nastavení zůstane zachováno do té doby, než ho explicitně změníme. Výhoda tohoto přístupu spočívá především v tom, že funkce pro vykreslování mají menší počet parametrů a že jedním příkazem lze globálně změnit způsob vykreslení celé scény, například volbu drátového zobrazení modelu (wireframe model) nebo zobrazení pomocí vyplněných polygonů (filled model). Vykreslování scény se provádí procedurálně - voláním funkcí OpenGL se vykreslí výsledný rastrový obrázek. Výsledkem volání těchto funkcí je rastrový obrázek uložený v tzv. framebufferu, kde je každému pixelu přiřazena barva, hloubka, alfa složka popř. i další atributy. Z framebufferu lze získat pouze barevnou informaci a tu je možné následně zobrazit na obrazovce.



Obr. 17: Vykreslení 3D scény a její zobrazení na obrazovce

OpenGL *nezaručuje*, že při spuštění identického programu používajícího knihovnu OpenGL na různých platformách nebo různých grafických akcelerátorech dostaneme vždy přesně stejný výsledek. Pokud bychom oba výsledné rastrové obrázky porovnali pixel po pixelu, mohli bychom zjistit mírné rozdíly v barvách. Může to být způsobeno například odlišnou přesností reprezentace čísel na grafické kartě, odlišnými algoritmy pro interpolaci barvy, normály a texturových souřadnic nebo jinou bitovou hloubkou Z-bufferu. Celkové geometrické a barevné podání scény by však mělo být zachováno.

Pomocí funkcí poskytovaných knihovnou OpenGL lze vykreslovat obrazce a tělesa složená ze základních geometrických prvků, které nazýváme grafická primitiva. Mezi tato primitiva patří bod, úsečka, trojúhelník, čtyřúhelník, plošný konvexní polygon, bitmapa (jednobarevný rastrový obraz) a pixmap (barevný rastrový obraz). Existují i funkce, které podporují proudové vykreslování některých primitiv - lze například vykreslit polyčáru (line loop), pruh trojúhelníků (triangle strip), pruh čtyřúhelníků (quad strip) nebo trs trojúhelníků (triangle fan). Na vrcholy tvořící jednotlivá grafická primitiva lze aplikovat různé transformace (otočení, změna měřítka, posun, perspektivní projekce), pomocí kterých lze poměrně jednoduše vytvořit animace. Vykreslovaná primitiva mohou být osvětlena nebo pokryta texturou. Dále je možné celou vykreslovanou scénu "skrýt" v mlze.

Z hlediska datové reprezentace vykreslované scény poskytují funkce OpenGL pouze základní rozhraní pro přístup ke grafickým akceleratorům. Existují však rozšiřující knihovny, které funkcionalitu dále zvyšují. Jednou ze základních knihoven používaných společně s OpenGL je knihovna GLU (OpenGL Utilities Library), která umožňuje využívat *tesselátory* (rozložení nekonvexních polygonů na trojúhelníky), *evaluátory* (výpočet souřadnic bodů ležících na parametrických plochách) a vykreslovat *kvadriky* (koule, válce, kužely a disky). Další nadstavbovou knihovnou je knihovna GLAUX (OpenGL Auxiary Library), která umožňuje zobrazení složitějších grafických objektů (kvádr, koule, válec apod.).

7.3 Zvolené 3D rozhraní

Pro potřeby navrhované aplikace bylo zvoleno 3D rozhraní OpenGL. Vytvářená 3D aplikace bude mít za úkol pouze zobrazovat relativně jednoduchou 3D scénu bez použití dalších multimediálních funkcí (zvuk, síťová komunikace). Rozhraní OpenGL sice tyto funkce přímo v sobě neobsahuje (oproti DirectX), ale v případě zobrazení 3D grafiky přináší rozhraní DirectX (Direct3D) a OpenGL stejné možnosti. Výhodou OpenGL pak je, že se rozhraní chová jako stavový automat. Proto lze provést nastavení parametrů zobrazované scény pouze jednou a poté již stačí scénu zobrazovat pomocí několika základních funkcí. Další výhodou potom je, že OpenGL je nezávislé na použitém operačním systému (oproti DirectX od Microsoftu).

8. Načtení 3D modelů do aplikace

8.1 Způsob uložení 3D modelů

3D modely byly vytvořeny v programu 3D Studio Max (viz kapitola 5.1). Pro uložení 3D modelů byl zvolen soubor formátu 3DS (*3D Studio File Format*).

8.2 Struktura souboru formátu 3DS

3DS (*3D Studio File Format*) souborový formát byl navržený firmou Autodesk pro popis trojrozměrných scén vytvářených v programu 3D Studio a posléze i 3D Studio MAX. S tímto formátem pracují i mnohé další aplikace a to přesto, že v minulosti nebyl uvolněn ani definován jako standard. Dnes je proto možné nalézt dokumenty, které tento formát popisují. 3DS formát je binární podoby a lze v něm kromě popisu trojrozměrné scény ukládat i informace o animacích, texturách a osvětleních.

Soubor formátu 3DS je hierarchicky uspořádán a skládá se z bloků dat, které jsou označovány jako *chunks*. Každý *chunk* může obsahovat další bloky dat, které popisují vlastnosti jednotlivých položek. Na začátku každého bloku dat jsou vždy uvedeny jeho typ (*chunk_id*) a délka (*chunk_len*). *Chunk_id* udává jaká data se v daném bloku nacházejí. (například: pozice, barva, materiál objektu, informace o souboru s texturou apod.). *Chunk_len* pak udává délku těchto dat. Za samotnými daty může následovat opět další *chunk*.

Každý 3DS soubor musí vždy obsahovat datový blok s názvem *primary chunk*. Tento primární blok dat obsahuje základní informace o 3D scéně a skládá se z dalších *chunks*, které popisují kompletní 3D scénu uloženou v souboru.

8.3 Unit *f_3DS*

Unit *f_3DS*, která je k dispozici pro vývojové prostředí Delphi, umožňuje načíst vybraná data popisující 3D scénu ze souboru 3DS a uložit ji do datových struktur. Pomocí této unit lze načítat souřadnice bodů, které tvoří jednotlivé objekty 3D modelu, barvu materiálů a informace o texturách pokrývajících tyto objekty. Ostatní datové bloky, které se v souboru 3DS mohou vyskytovat (informace o animacích, osvětlení apod.), jsou při načítání přeskočeny.

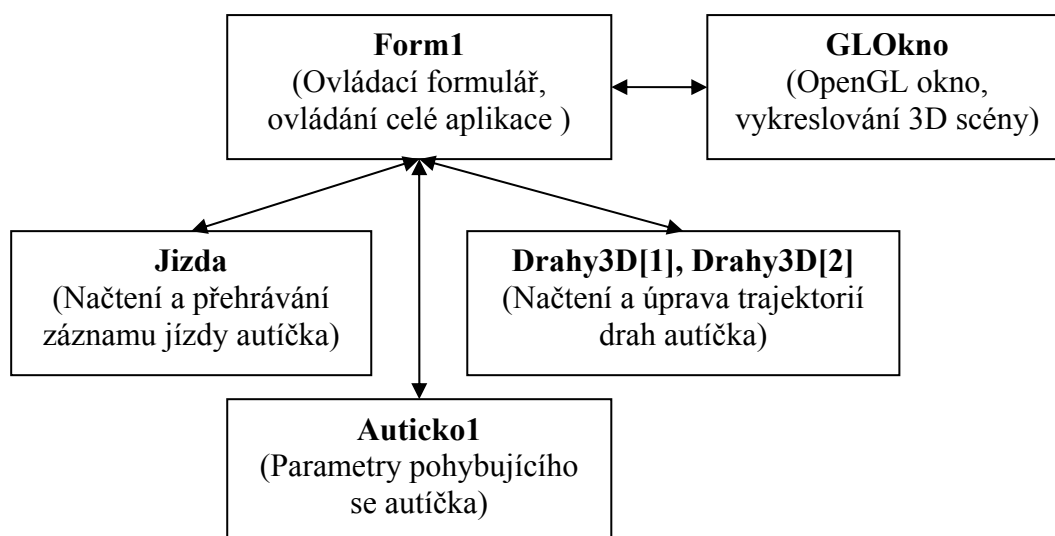
9. 3D aplikace

9.1 Obecný popis

Aplikace zobrazující jízdu autíčka je vytvořena pro platformu operačního systému Microsoft Windows. Jako vývojové prostředí bylo použito programu Borland Delphi 7.0. Pro vykreslování 3D scény je využito grafického rozhraní OpenGL, proto je pro správný běh aplikace vyžadována grafická karta s hardwarovou podporou OpenGL.

9.2 Struktura aplikace

V aplikaci je použito několik objektů. Každý objekt zajišťuje určitou část obsluhy funkcí programu (uživatelské rozhraní, vykreslování 3D scény, načtení a úpravu trajektorií autodráhy, přehrávání záznamu jízdy autíčka).



Obr. 18: Struktura objektů aplikace

9.2.1 Zobrazení 3D modelů

Zobrazení 3D modelů autodráhy a autíček má v programu na starost objekt *GLOkno*, který je instancí třídy *T3DOkno*. Třída *T3DOkno* také zajišťuje načtení a uložení 3D objektů ze souboru 3DS do datových struktur pomocí unit *f_3DS*, jež je popsána v kapitole 8.

Jelikož samotné rozhraní OpenGL neobsahuje žádné metody pro práci s okny ani pro zpracování událostí, zajišťuje vytvoření 3D okna, práci s ním a obsluhu událostí také objekt *GLOkno*. Vybrané metody třídy *T3DOkno* jsou popsány níže.

**procedure *CreateTexture*(var textureArray: array of UINT;
strFileName: LPSTR; textureID: integer);**

- Vytvoří texturu z bitmapového souboru a uloží ji do paměti. Procedura je volána při načítání 3D modelů z 3DS souboru.

function *Initialize*(window: PGL_Window; key: PKeys): boolean;

- Proveď inicializaci OpenGL, načte 3D modely, jejich materiály a textury.

procedure *Deinitialize*;

- Odstraní z paměti 3D modely, materiály a textury.

procedure *Klavesnice*;

- Obsluhuje reakce na stisk kláves.

procedure *Draw*;

- Vykreslí modely autodráhy a autíčka z pohledu vybrané kamery podle nastavených parametrů (jeden snímek 3D animace).

procedure *SouradniceAuta*(Draha3D: TDraha3D);

- Vypočte a nastaví aktuální souřadnice autíčka na dráze, provede nastavení pohyblivých pohledů kamer.

function *Okno*(hInstance: HINST; hPrevInstance: HINST;

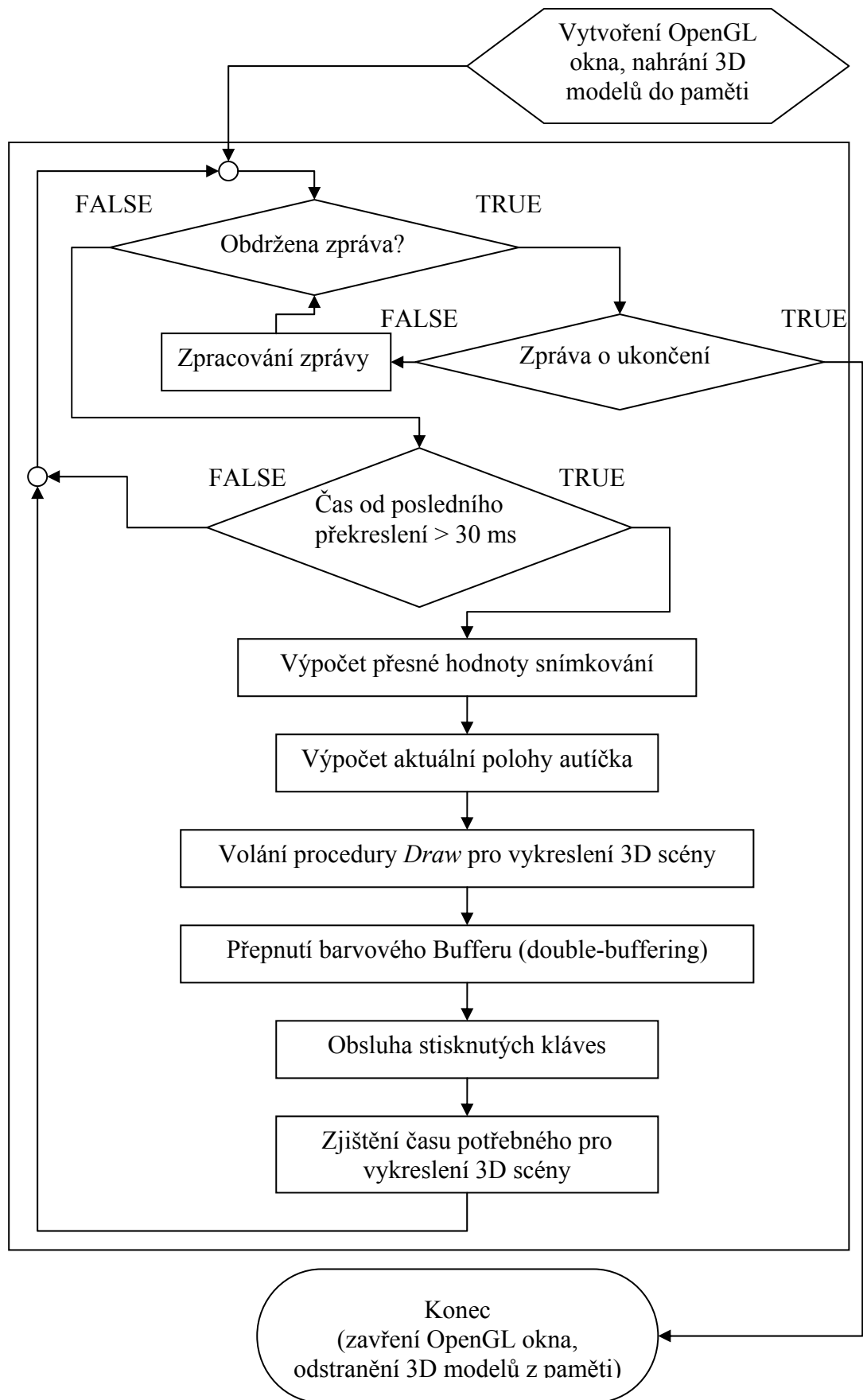
lpCmdLine: PChar; nCmdShow: integer): integer; stdcall;

- Zajišťuje vykreslování 3D modelů. Kvůli plynulosti pohybu autíčka, se jednotlivé snímky snaží vykreslovat s konstantním zpožděním přibližně 30 ms (33 fps). Zajišťuje aktualizaci polohy pohybujícího se autíčka.

function *WindowProc*(hWnd: HWND; uMsg: UINT; wParam: WPARAM;

lParam: LPARAM): LRESULT; stdcall;

- Funkce není součástí třídy *T3DOkno*. Avšak zajišťuje obsluhu zpráv pro vytvořené OpenGL okno (pohyb myši, stisky kláves). Funkce je volána operačním systémem v případě příchodu zprávy.



Obr. 19: Vykreslování 3D scény (funkce `Okno(...)`)

9.2.2 Reprezentace autíčka

Všechny důležité vlastnosti, popisující autíčko, jsou ukládány v objektu *Auticko1*, který je instancí třídy *TAuticko*. Objekt *Auticko1* neobsahuje kromě (constructoru) žádné metody a je využíván jako dočasná proměnná při pohybu autíčka po dráze. Jsou v něm vždy uloženy aktuální hodnoty polohy v prostoru (x,y,z), natočení autíčka kolem svislé a vodorovné osy, vektory rychlosti a předchozího směru. Hodnoty polohy a natočení využívá objekt *GLOkno* při umísťování a zobrazování 3D modelu autíčka na autodráhu.

9.2.3 Načtení a interpolace trajektorií

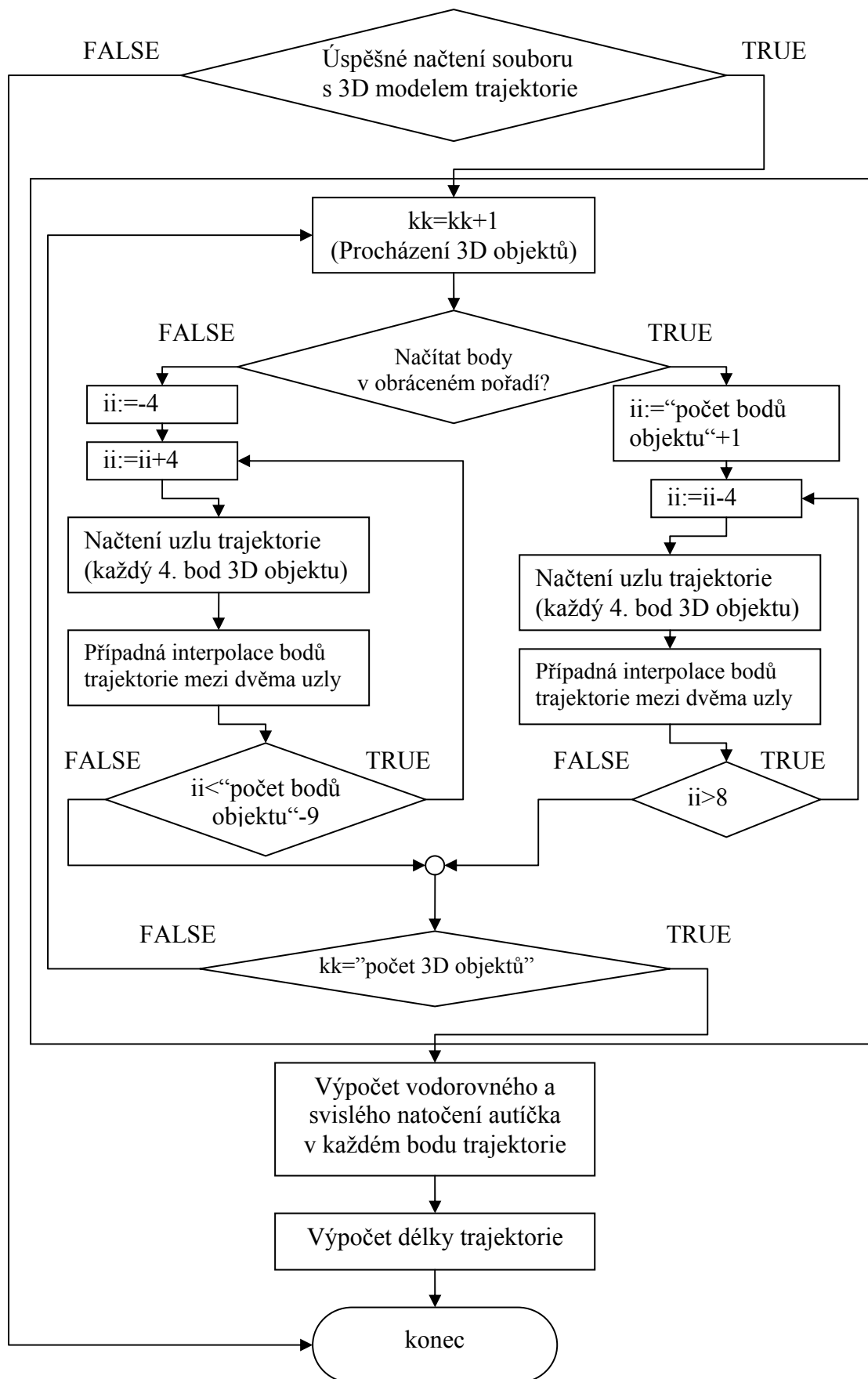
Načtení a interpolaci trajektorií jednotlivých drah, zajišťuje třída *TDraha3D*. Princip načtení a interpolace dráhy je popsán v kapitolách 6.1.2 a 6.2. Na autodráze se vyskytují dvě dráhy, proto je pro každou vytvořen samostatný objekt (*Drahy3D[1]*, *Drahy3D[2]*), který obsahuje všechny potřebné informace o jednotlivých trajektoriích (nejdůležitější informace: souřadnice všech bodů trajektorie, svislé a vodorovné natočení autíčka v každém bodě trajektorie). Jednotlivé metody třídy *TDraha3D* jsou popsány níže.

constructor *Create*;

- Vytvoří instanci třídy *TDraha3D* a provede počáteční nastavení hodnot proměnných vytvářeného objektu.

function *NactiDrahu3D(Cesta:string):boolean*;

- Funkce načte základní uzly trajektorie ze souboru formátu 3DS a provede interpolaci trajektorie se zvoleným krokem. Dále vypočte hodnoty svislého a vodorovného natočení autíčka v každém bodě trajektorie. Funkce vrátí hodnotu TRUE pokud došlo ke správnému načtení trajektorie.



Obr. 20: Načtení a interpolace bodů trajektorie (funkce *NactiDrahu3D(...)*)

9.2.4 Jízda autíčka

Samotné načtení a následné přehrání jízdy zajišťuje objekt *Jizda*, který je instancí třídy *TJizda*. Třída *TJizda* umožňuje načítání záznamu jízdy ze souboru, jehož formát je popsán v kapitole 4.2. Poté dojde k přepočtu načtených elektrických napětí na skutečné rychlosti podle základní převodní funkce napětí na rychlost (viz kapitola 3.2). Následně jsou tyto rychlosti upravovány tak, aby autíčko neměnilo svoji rychlost skokově. Aktuální rychlosti jsou také upraveny podle místa na dráze, po kterém se má autíčko pohybovat. Nakonec jsou jednotlivé rychlosti modifikovány tak, aby autíčko při přehrávání jízdy ujelo správný počet celých kol, jejichž jízda je zaznamenána v souboru. Metody třídy *TJizda* jsou popsány níže.

constructor *Create*;

- Vytvoří instanci třídy *TJizda* a provede počáteční nastavení hodnot proměnných vytvářeného objektu.

function *VratRychlost*(Napeti1:GLfloat; CisloDrahy:byte; Draha3D:TDraha3D):GLfloat;

- Funkce vrací hodnotu rychlosti v cm/s podle zadaného napětí a čísla dráhy.

function *UpravPrechodoveRychlosti*(Draha3D:TDraha3D):boolean;

- Funkce dopočítá plynulé změny rychlosti u původně skokových změn. Vrací hodnotu TRUE, pokud nedošlo při výpočtech k chybě.

function *UpravKolo*(Draha3D:TDraha3D):boolean;

- Funkce upraví rychlosti tak, aby celková jízda autíčka odpovídala zadanému počtu celých okruhů. Vrací TRUE, pokud nedošlo při výpočtech k chybě.

function *UpravRychlost*(CisloDrahy:byte; Draha3D:TDraha3D):boolean;

- Funkce upraví aktuální rychlosti podle pozice autíčka na dráze.

function *DopoctiJizdu2*(Draha3D:TDraha3D):boolean;

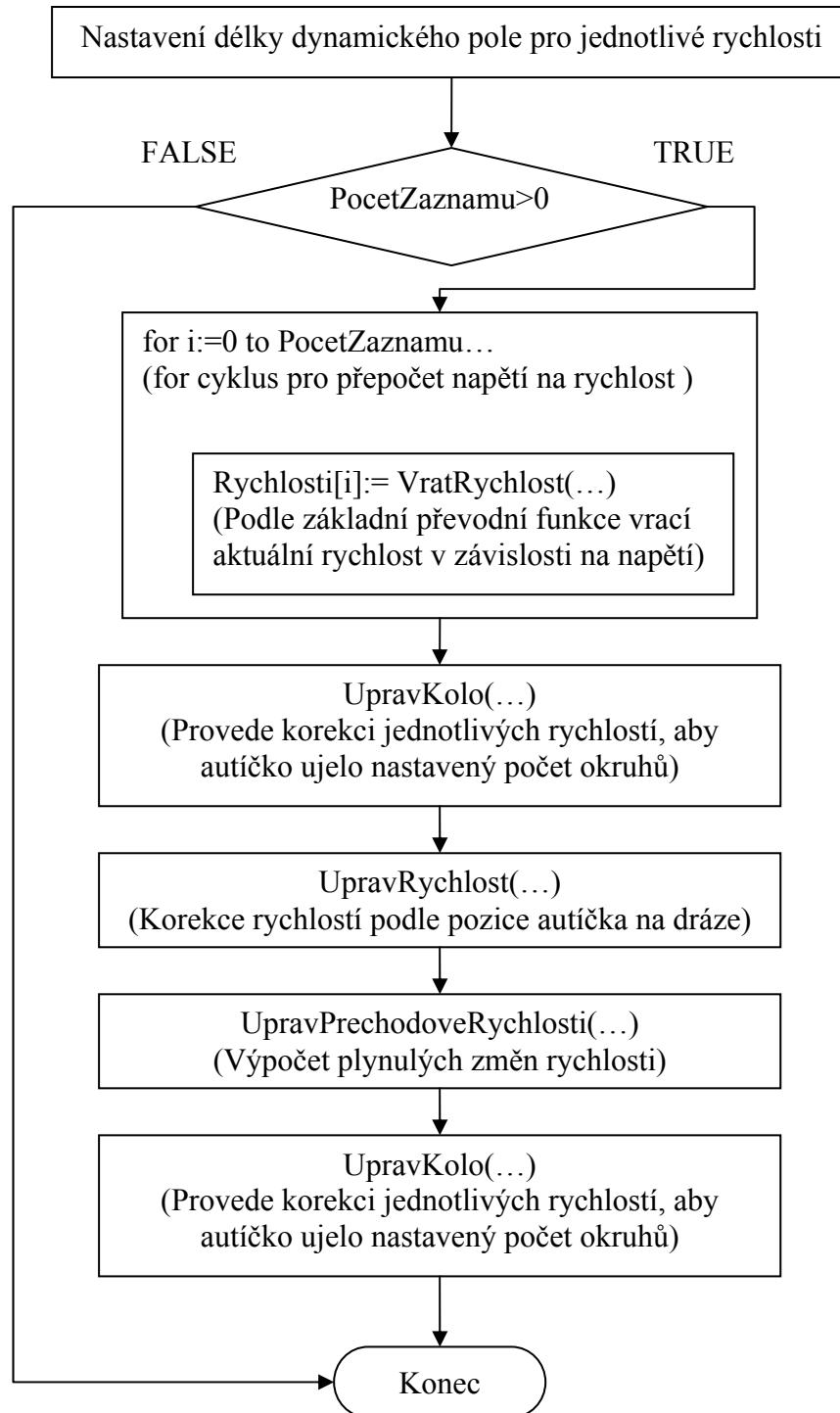
- Provede přepočet načtených napětí záznamu jízdy na rychlosti a jejich úpravu pomocí výše popsaných funkcí. V případě úspěšného přepočtu vrací funkce TRUE.

function *NactiJizdu*(Cesta:string):boolean;

- Funkce načte číslo dráhy, hodnotu napájecího napětí autodráhy, periodu zaznamenaných napětí a samotný průběh napětí uložených v souboru se záznamem jízdy. Pokud dojde k úspěšnému načtení, vrací funkce TRUE.

function *UjetaDraha*(Cas:glDouble):glDouble;

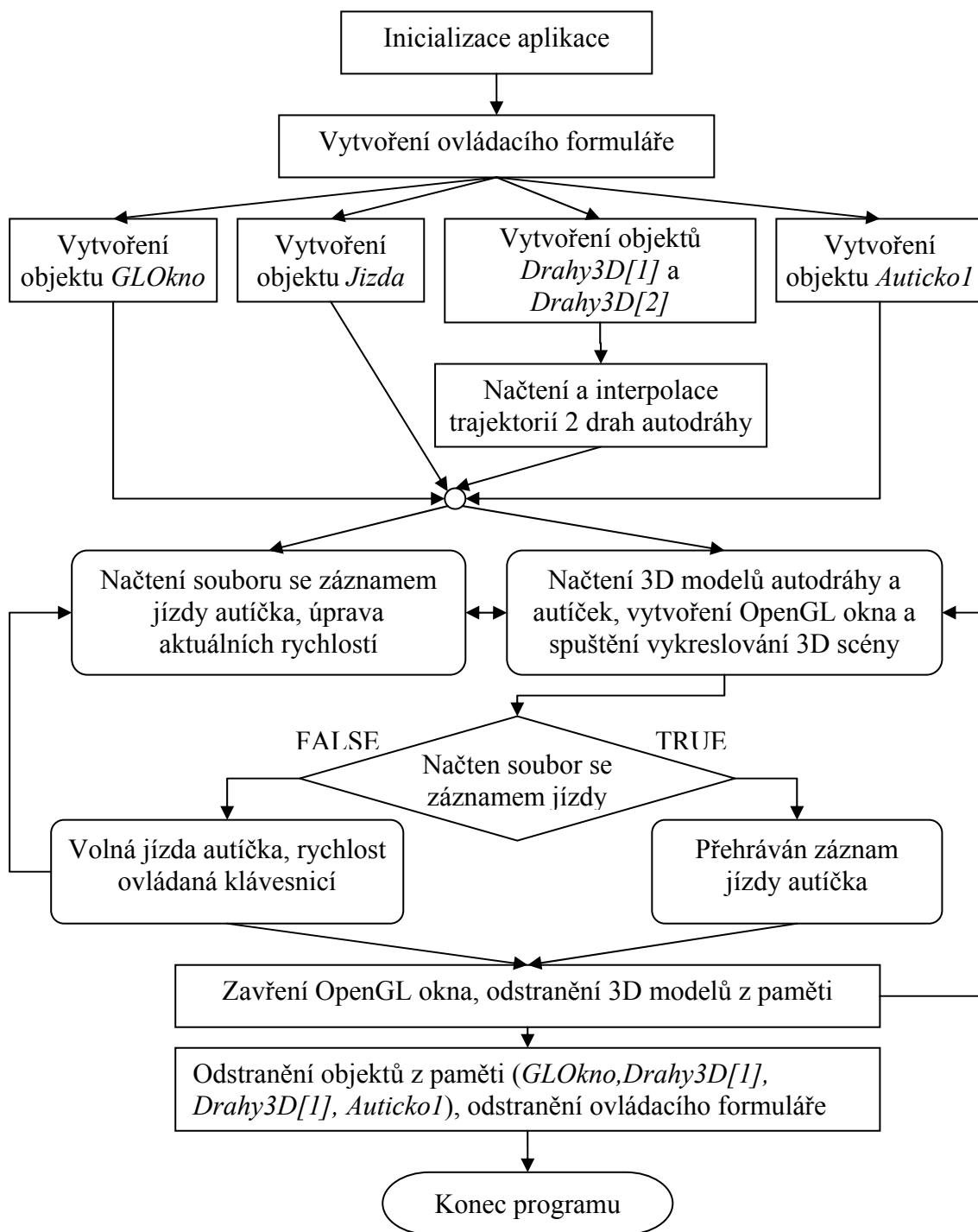
- Funkce vrací hodnotu ujeté dráhy v centimetrech podle zadané hodnoty času.



Obr. 21: Výpočet a úprava jednotlivých rychlostí podle hodnot napětí záznamu jízdy autíčka (funkce *DopoctiJizdu2(...)*)

9.2.5 Uživatelské rozhraní

Základní uživatelské a ovládací rozhraní (ovládací formulář) tvoří objekt *Form1*, který je instancí třídy *TForm1*. Při vytváření ovládacího formuláře dochází k vytvoření objektů *GLOkno*, *Drahy3D[1]*, *Drahy3D[2]*, *Auticko1* a *Jizda*.



Obr. 22: Běh aplikace

10. Závěr

Výsledkem bakalářské práce je 3D aplikace, která dokáže načítat soubory se záznamem jízdy autíčka po autodráze a tento záznam následně přehrávat. Za účelem zobrazení jízdy byly vytvořeny 3D modely autodráhy a dvou autíček. Tyto modely aplikace zobrazuje pomocí 3D rozhraní OpenGL.

Aplikace bez problémů zobrazuje pohyb autíčka po obou drahách autodráhy z pěti předdefinovaných pohledů s další možností jejich změny. Dále umožňuje zoomování obrazu a změnu rychlosti přehrávání záznamu jízdy.

Jedním z problémů při zobrazování 3D scény se ukázala být složitost 3D modelů autodráhy a autíček. Docházelo pak k pomalému překreslování scény (také závislé na výkonu použité grafické karty) s frekvencí nižší než 25 snímků za sekundu. Proto již nebylo možné do 3D modelu autodráhy zahrnout více detailů (textury segmentů drah, textury reliéfu s vyšším rozlišením apod.). Jedno z možných řešení by spočívalo v zjednodušování vykreslované 3D scény přímo za běhu programu. Vzdálenější objekty by se mohly skládat z menšího počtu polygonů nebo by nebyly zobrazeny vůbec. Protože je ale na většině pohledů zobrazována celá autodráha, bylo by toto řešení použitelné pouze při velkých zvětšeních nebo v případech pohledu z autíčka.

Dalším problémem práce bylo určení převodní funkce, která ze souboru s hodnotami zaznamenaných elektrických napětí určuje hodnoty aktuálních rychlostí autíčka. Jelikož je autodráha poměrně členitá, vyskytuje se na ni mnoho nerovností. Autíčko se pak při přejezdu těchto úseků chová nepředvídatelně (změna rychlosti, uvíznutí). Toto bylo vyřešeno tím, že přehrávaný záznam jízdy je záznamem jednoho nebo více celých úspěšně odjetých okruhů autíčka. Aplikace pak upravuje jízdu tak, aby při přehrávání autíčko ujelo zadaný počet okruhů za příslušnou dobu. Poloha autíčka se proto od skutečné jízdy liší pouze minimálně (nejvíce na problematických úsecích), protože větší lokální změny rychlostí jsou kompenzovány minimální úpravou ostatních rychlostí.

Srovnání přehrávaného záznamu jízdy se skutečnou jízdou bylo provedeno na obou drahách autodráhy. Pro srovnání bylo použito napájecí napětí autíčka o konstantní hodnotě (6 V) po celou dobu průjezdu jednoho kola. Nebylo sice možné určit přesné odchylky polohy autíčka zobrazovaného záznamu od skutečné

jízdy po autodráze, ale změny rychlostí autíčka odpovídaly charakteru drah (do zatáček se autíčko pohybovalo pomaleji než po rovném úseku, při průjezdu zatáček docházelo ke snižování rychlosti). Také při použití různých hodnot napětí v různých částech dráhy změna rychlostí odpovídala skutečné jízdě.

Přehrání záznamu jízdy tak zvládne aplikace s uspokojivými výsledky. Avšak platí zde určitá omezení pro zaznamenanou jízdu autíčka (záznam musí být vždy pro celé okruhy autodráhy a pro jízdu bez zaseknutí či vylétnutí autíčka z dráhy). Možnost řešení tohoto omezení by spočívala v doplnění autodráhy o senzory, které by určovaly přesnou polohu autíčka (například po každých dvaceti centimetrech dráhy). Záznam jízdy autíčka by poté obsahoval kromě hodnot napětí i informace o poloze autíčka. Pak by bylo možné upravovat rychlosti autíčka podle informací z těchto senzorů a bylo by možné zohlednit případné krátkodobé změny rychlosti autíčka, jeho zaseknutí v dráze nebo vylétnutí z dráhy ven. Avšak toto řešení by vyžadovalo kompletní návrh a sestavení měřicího rozhraní, což nebylo úkolem zadané bakalářské práce.

Použitá literatura

- [1] 3ds max pro začátečníky. [online]. Dostupné na www: <http://www.3dscena.cz>
- [2] Grafická knihovna OpenGL. [online]. Dostupné na www: <http://www.root.cz>
- [3] OpenGL. [online]. Dostupné na www: <http://www.opengl.org>
- [4] Tvorba her. [online]. Dostupné na www: <http://nehe.ceskehry.cz>

Přílohy

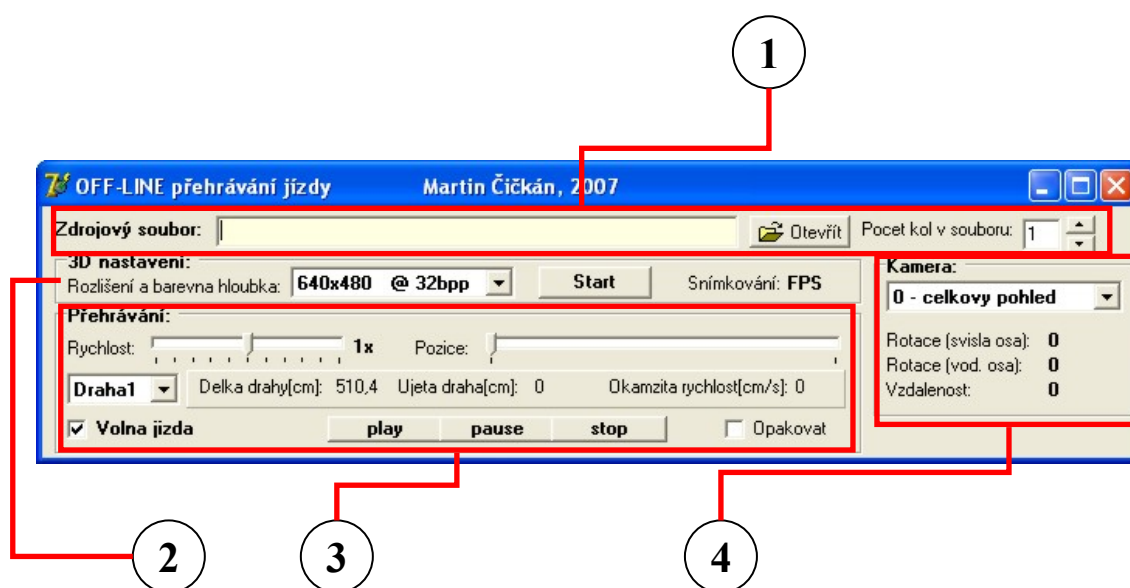
Ovládání a popis vytvořené 3D aplikace

Aplikace se skládá ze dvou oken: Ovládacího formuláře a okna pro zobrazení 3D scény.

Ovládací formulář

Na ovládacím formuláři lze nastavovat parametry zobrazované 3D scény, načítat soubory se záznamem jízdy autíčka a kontrolovat přehrávání tohoto záznamu.

Jednotlivé ovládací prvky jsou rozděleny do čtyřech skupin a budou popsány níže.



Obr. 1: Ovládací formulář

(1) Načtení souboru se záznamem jízdy autíčka

Otevřít – stisk tohoto tlačítka vyvolá dialog, ve kterém je možné vybrat soubor se záznamem jízdy.

Pokud je načten soubor správného formátu, je zobrazen jeho název v příslušném žlutém poli. Také je zobrazen počet kol, které jsou v souboru se záznamem jízdy uloženy. Po načtení je možné počet kol měnit pomocí

dvojice tlačítek a program vždy upraví záznam jízdy, aby odpovídal počtu nastavených kol.

(2) 3D nastavení

V příslušné nabídce, lze zvolit velikost 3D okna (v pixelech), ve kterém je zobrazena 3D scéna. Pokud bude později 3D okno přepnuto do celoobrazovkového režimu, bude použito zvolené rozlišení a barevná hloubka. Změnu rozlišení lze provést pouze v případě, že není zobrazeno 3D okno.

Start – při stisku tohoto tlačítka dojde k načtení 3D modelů autodráhy a autíček, vytvoření 3D okna a zahájení vykreslování 3D scény. Je také zobrazován dosahovaný počet snímků za sekundu (fps). Pokud již byl načten soubor se záznamem jízdy, začne přehrávání tohoto záznamu.

(3) Přehrávání záznamu jízdy

Pokud je načten soubor se záznamem jízdy a spuštěno vykreslování 3D scény, lze zde nastavovat parametry přehrávání.

První *posuvník* umožňuje změnu rychlosti přehrávaného záznamu. Rychlost přehrávání může nabývat jedenácti hodnot: $\frac{1}{6}$, $\frac{1}{5}$, $\frac{1}{4}$, $\frac{1}{3}$, $\frac{1}{2}$, 1, 2, 3, 4, 5, 6.

Druhým *posuvníkem* lze v případě pozastaveného přehrávání, posouvat záznam jízdy autíčka po celých vteřinách.

Další ovládací prvek umožňuje vybrat dráhu po které se autíčko pohybuje (vnitřní *dráha1*, vnější *dráha2*). Pokud je načten soubor se záznamem jízdy, je záznam upraven tak, aby odpovídal zvolené dráze.

Volba *Volna jizda* umožní ovládat rychlost pohybujícího se autíčka po autodráze pomocí klávesnice.

Play – tlačítko spustí přehrávání záznamu jízdy autíčka vždy od začátku. Pokud není načten soubor se záznamem jízdy nastaví polohu autíčka na počáteční pozici na dráze.

Pause – tlačítko pozastaví (nebo opět obnoví) přehrávání záznamu jízdy. V případě že není načten soubor se záznamem, nemá tlačítko žádnou funkci.

Stop – stisk tlačítka úplně zastaví přehrávání jízdy. V případě že není načten soubor se záznamem, nemá tlačítko žádnou funkci.

Volba *Opakovat* zajistí opakování přehrávaného záznamu jízdy. Po přehrání celé jízdy autíčka, začne přehrávání opět od začátku.

Na *panelu* jsou zobrazeny statistiky dráhy a záznamu jízdy: celková délka zvoleného okruhu, aktuální hodnota ujeté dráhy v případě přehrávaného záznamu a okamžitá rychlost autíčka.

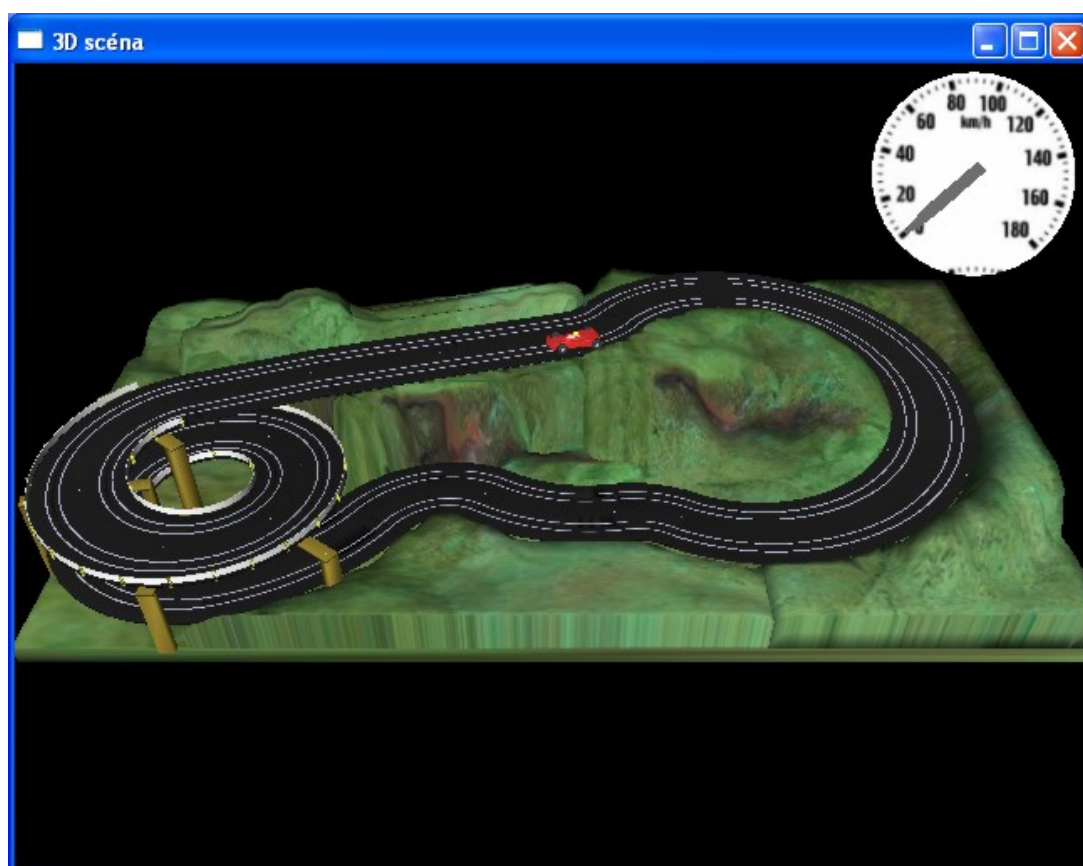
(4) Nastavení kamery

Poslední část ovládacího formuláře umožňuje nastavení pohledů na autodráhu. K dispozici je pět základních pohledů: celkový pohled na autodráhu shora, pohled z autíčka, pohled na pohybující se autíčko shora se sledováním jeho polohy, celkový pohled na autodráhu ze předu a boční pohled na dvoupatrovou část dráhy.

Parametry jednotlivých pohledů jsou zobrazeny na přítomném *panelu* (vzdálenost od středu, rotace kolem vodorovné a svislé osy). Jednotlivé pohledy lze dále upravovat pomocí klávesnice a myši (viz další text).

Okno pro zobrazení 3D scény

V okně jsou zobrazovány modely autodráhy a pohybujícího se autíčka. Dále je zde zobrazen „*rychloměr*“, který přibližně ukazuje aktuální okamžitou rychlost pohybujícího se autíčka.



Obr. 2: Okno se zobrazenou 3D scénou

Pomocí pohybů myši a stisků kláves, lze měnit některé parametry zobrazení modelu autodráhy:

Změna pohledu na dráhu – stiskem klávesy „C“ lze přepínat mezi pěti základními pohledy na autodráhu (viz předchozí část o ovládacím formuláři)

Posun 3D scény – stisknutím a držením levého tlačítka myši, se při pohybech myši pohybuje celá 3D scéna (toto neplatí při zobrazení pohledu z autíčka).

Rotace 3D scény – při stisknuté klávese „levý Ctrl“, stisknutém levém tlačítku myši a pohybech myši nahoru a dolů se celá scéna natáčí kolem vodorovné osy. V případě použití pravého tlačítka myši, se celá scéna otáčí kolem svislé osy. (V případě pohledu z autíčka, lze natáčet kameru pouze kolem vodorovné osy autíčka.)

Zoomování – při stisknuté klávese „levý Shift“ a pohybech myši nahoru a dolů, je celá zobrazovaná 3D scéna přibližována nebo oddalována.

Změna typu autíčka – stiskem klávesy „A“ je zobrazováno červené nebo modré autíčko.

Zobrazení „mlhy“ – stiskem klávesy „M“ je ve scéně zobrazena „černá mlha“. Pokud jsou objekty daleko od kamery, plynule se ztrácí v pozadí. Vhodné použít v případě perspektivních pohledů nebo pohledu z autíčka.

Celoobrazovkový režim – stiskem klávesy „F“ je přepínáno mezi zobrazením 3D scény v okně a na celé obrazovce.

Zavření 3D okna – stiskem klávesy „Esc“

Přehrání záznamu jízdy od začátku – stisknutí klávesy „R“ má stejný význam jako tlačítko „Play“ na ovládacím formuláři.