

Demonstrace křivek v počítačové grafice

Projekt ITU, 2016Z

Číslo projektu: 9

Číslo a název týmu: 136. Tým xtisov00

Autor: Klára Nečasová (xnecas24)

Další členové týmu: Peter Tisovčík (xtisov00)

Termín řešení: 19. 9. – 14. 12. 2016

Abstrakt

Cílem projektu je vytvořit aplikaci, která bude vhodně objasňovat a demonstrovat problematiku Fergusonových křivek, B-spline křivek a také NURBS křivek.

Cílové požadavky na aplikaci a její rozhraní

Výsledná aplikace by měla vhodným způsobem ukázat odlišnosti mezi jednotlivými typy křivek a měla by také uživateli umožnit snadno měnit jejich parametry.

Uživatelské rozhraní by mělo být přehledné a uživatelsky přívětivé. Důraz bude kladen na snadné a intuitivní ovládání aplikace. Mezi hlavní požadavky na aplikaci patří srozumitelnost a názornost. Aplikace umožní nastavovat příslušné parametry jednotlivých typů křivek (např. měnit orientaci tečných vektorů křivky, pozici řídících bodů křivky, případně upravovat jejich váhy). Také by bylo vhodné, aby výsledná aplikace motivovala uživatele přemýšlet o vlivu parametrů na tvar křivky. K tomuto účelu by mohly sloužit jednoduché úlohy, které by spočívaly v tom, že by se uživatel snažil co nejvíce napodobit tvar vzorové křivky nastavováním jejich parametrů. Ke každé variantě křivky bude také zobrazen stručný popis jejich vlastností, jehož účelem je připomenout základní charakteristiky křivek.

Splnění cílů uživatelského rozhraní lze odvodit z toho, jak rychle se dokáže uživatel v aplikaci zorientovat a také, za jak dlouho porozumí problematice daných křivek. Spokojený uživatel by si měl aplikaci oblíbit a využívat ji, kdykoli si bude chtít problematiku daných křivek připomenout.

Přidanou hodnotou aplikace by mohla být možnost zobrazení informací k jednotlivým prvkům uživatelského rozhraní ihned po přesunu kurzoru myši na informační ikonu a možnost exportu vykreslené křivky jako obrázek.

Studium cílové skupiny a případy použití

Cílovou skupinu uživatelů tvoří převážně studenti, lektori či zájemci o počítačovou grafiku. Aplikace jim umožní ověřit si své poznatky o křivkách a porovnat různé vlastnosti daných křivek. Cílová skupina je specifická tím, že se v ní mohou nacházet lidé s různou úrovní znalostí dané problematiky.

Předpokládá se, že aplikace bude využívána zejména pro demonstraci křivek v průběhu přednášek předmětu IZG (Základy počítačové grafiky) a také v rámci cvičení, která budou zaměřena na problematiku křivek.

Jelikož musí být například výběr typu křivky realizován myší, je tento způsob zvolen k ovládání celé aplikace. Snahou je zvolit jednotný způsob ovládání, což pomůže uživateli si zautomatizovat její ovládání.

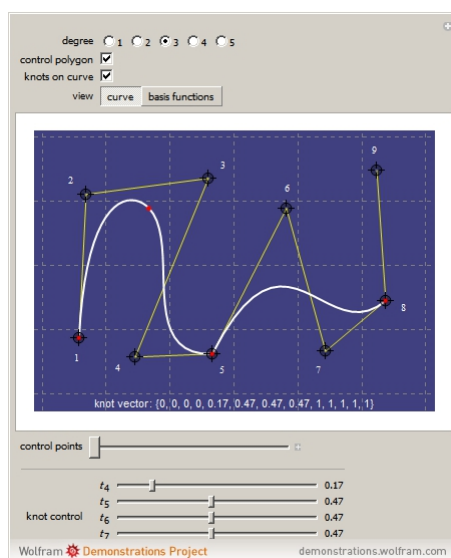
Kvůli požadavku spustitelnosti aplikace na různých operačních systémech je jistým omezením při návrhu uživatelského rozhraní nutnost použití multiplatformních knihoven pro tvorbu uživatelských rozhraní.

Existující řešení

Mezi vybrané aplikace patří následující.

Aplikace 1: Wolfram Demonstrations Project: B-spline Curve with Knots

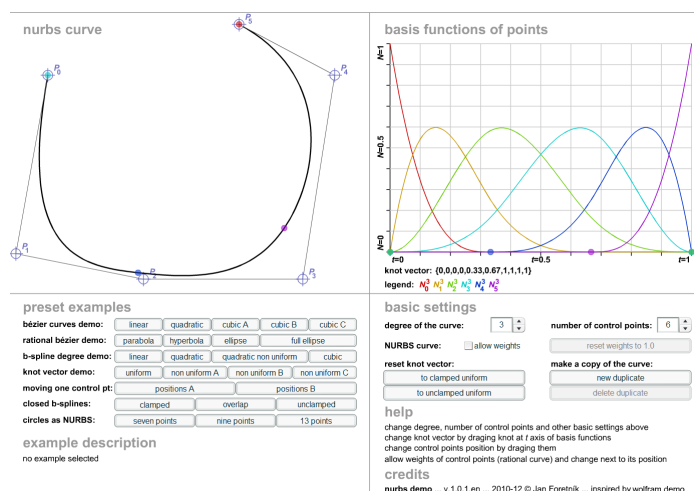
Aplikace umožňuje vykreslovat B-spline křivky¹. Její výhodou je přehledné uživatelské rozhraní, které přispívá k rychlému pochopení nastavování parametrů křivky. Velmi úzká specializace aplikace (umožňuje vykreslovat jen B-spline křivky) a nepříliš vhodně zvolené barevné odlišení prvků aplikace (černé řídicí body na tmavě modrém pozadí) se však řadí k nevýhodám. Další nedostatek vidím v jejím ovládání, konkrétně se jedná o poměrně neintuitivní přidávání řídicího bodu (nutné stisknout klávesu Alt a tlačítko myši) a také to, že se jedná o proprietární software.



Obrázek 1: B-spline Curve with Knots

Aplikace 2: Geometrie Foretník

Aplikace podporuje mnoho typů křivek – např. Beziérovky křivky, racionální Beziérovky křivky, B-spline křivky či NURBS křivky². Výhodou aplikace je její přehledné zpracování, které je realizováno rozdělením okna aplikace do čtyř částí, možnost výběru typu křivky a také zobrazení jejího krátkého popisu. Ovládání aplikace patří k jejím nevýhodám, konkrétně nepříliš intuitivní manipulace s řídicími body křivky (přidávání/odebírání řídicích bodů je nutné provádět v pravé dolní části aplikace, což nemusí být na první pohled zřejmé). Další nevýhodou je přílišná složitost aplikace, která může uživateli komplikovat orientaci.



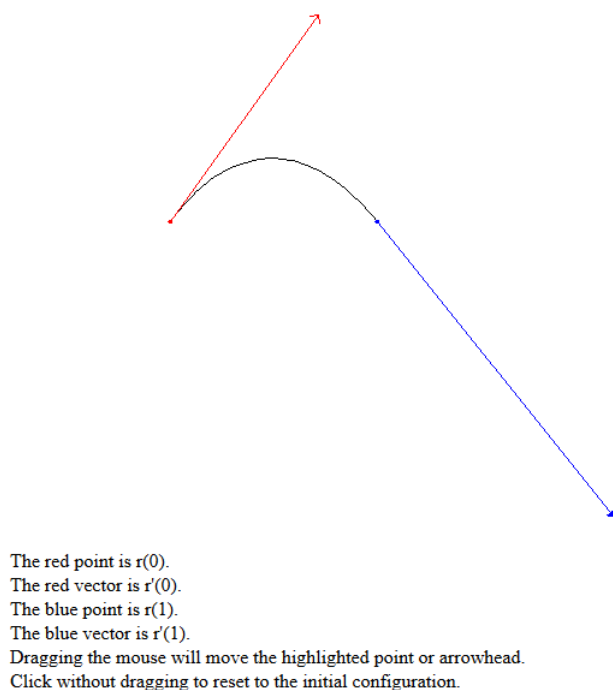
Obrázek 2: Geometrie Foretník

¹<http://demonstrations.wolfram.com/BSplineCurveWithKnots/>

²<http://geometrie.foretnik.net/files/NURBS-en.swf>

Aplikace 3: VIDIGEO: Ferguson Curve applet

Aplikace umí vykreslit pouze Fergusonovy křivky³. Pod vykreslenou křivkou je zobrazeno stručné a výstižné vysvětlení týkající se ovládání aplikace, což lze řadit k výhodám aplikace. Další výhodou je, že se koncový bod, případně šipka tečného vektoru zvýrazní, pokud na ně přesuneme kurzor myši. Zvýraznění by však obzvláště u šipky tečného vektoru mohlo být výraznější. K nevýhodám také patří způsob ovládání aplikace. Po kliknutí tlačítkem myši mimo zobrazovanou křivku se aplikace vrátí do původního stavu.



Obrázek 3: Ferguson Curve applet

Z analýzy existujících řešení lze vyvodit následující. Méně zkušený uživatel nemusí využít všechny možnosti aplikace a přílišná složitost aplikace jej může spíše odradit. Pro manipulaci s jedním typem objektu (např. s řídicím bodem křivky) je vhodné manipulovat jedním způsobem. Vhodná je například manipulace pomocí myši. Také je vhodné mít možnost měnit váhu řídicích bodů například kombinací přesunu myši na příslušný bod a rolováním kolečka myši. Je vhodné informovat uživatele o významu prvků aplikace a také uvést základní charakteristiky typu křivky. Přiměřené používání barev může významně ovlivnit přehlednost celé aplikace.

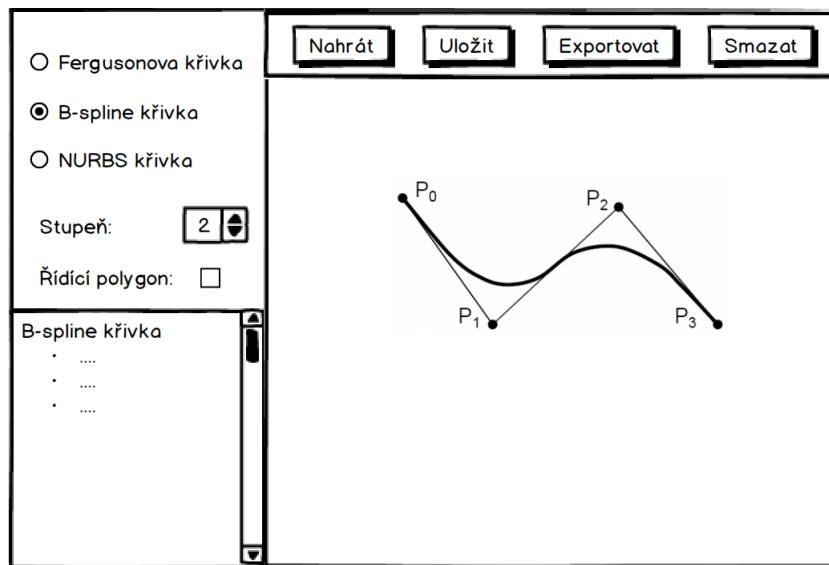
Návrh GUI

Aplikace by měla uživateli nabízet možnost vybrat si požadovaný typ křivky a nastavovat její parametry. Funkcionalita aplikace se bude lišit dle zvoleného typu křivky. U Fergusonových křivek bude možné měnit orientaci tečných vektorů a manipulovat s řídicími body. Co se týče B-spline křivek a NURBS křivek, bude taktéž možné manipulovat s řídicími body, navíc u NURBS křivek bude možné nastavovat jejich váhu.

V horní levé části okna by mohla být umístěna skupina prvků Radio Button, která by sloužila pro výběr typu křivky. Uvažovala jsem také o použití prvku Combo Box, ale uživatel zřejmě spíše ocení, pokud bude mít neustále zobrazeno, jaké další křivky aplikace podporuje. Po spuštění aplikace bude vykreslena zvolená křivka (implicitně křivka typu B-spline). Pro zobrazení řídicího polygonu křivky by mohl být využit prvek Check Box (implicitně by byl řídicí polygon zobrazen). Spodní levá část okna by sloužila k zobrazení charakteristických vlastností daného typu křivky. Zbylý prostor bude převážně využit pro vykreslení křivky, protože pro uživatele je klíčové, aby měl dostatek prostoru pro tvorbu křivek a mohl s nimi jednoduše manipulovat. Do pravé horní části okna by bylo vhodné umístit tlačítka pro načtení dat křivky ze souboru, jejich uložení do souboru, pro export vykreslené křivky a také tlačítko pro smazání kreslící plochy. Po kliknutí na tato tlačítka by si uživatel mohl zvolit soubor, ze kterého budou data načtena, v jakém formátu mají být data křivky uložena či do jakého formátu má být křivka exportována.

³<http://www2.mat.dtu.dk/info/mathematics/VIDIGEO/applets/book5.html>

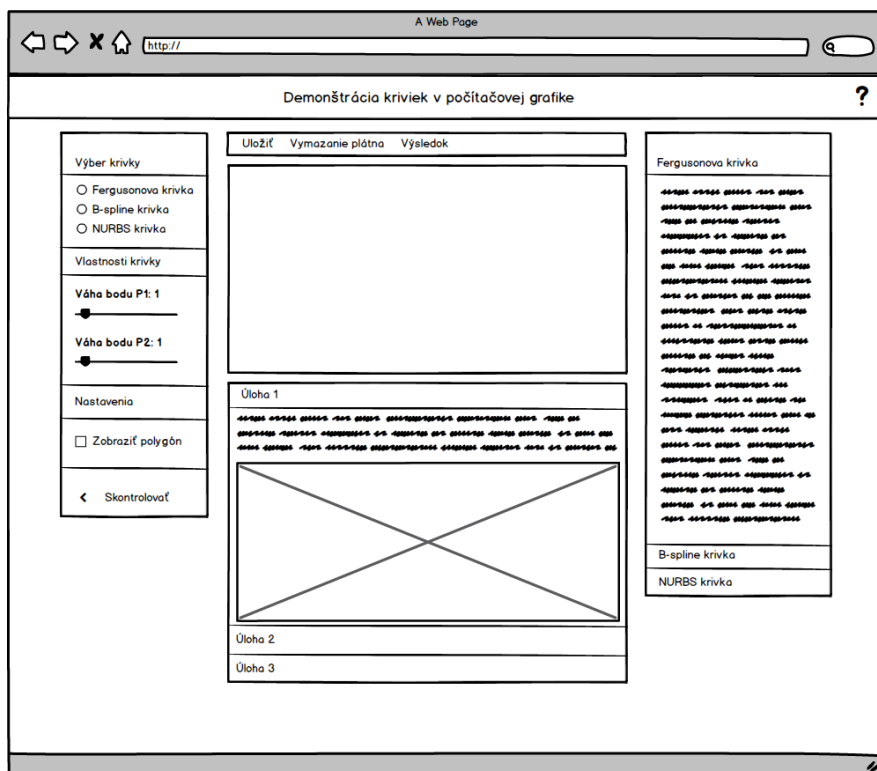
Řídící body křivek mohou být zobrazeny jako vyplněná kolečka, po přesunu myši na daný bod je vhodné průměr kolečka zvětšit. Podobný princip zvýraznění prvku by byl použit i v případě přesunutí myši na tečný vektor. Cílem je, aby byla manipulace s body či tečnými vektory křivky pro uživatele co nejvíce přívětivá. Manipulaci s řídicími body by bylo vhodné realizovat pomocí myši. Váhu daného bodu křivky by bylo možné měnit pomocí prvku Spin Box, který by se zobrazil po přesunu myši nad daný bod křivky. Samotnou změnu váhy by bylo možné realizovat buď pomocí myši (kolečko, tlačítko), nebo šipkami klávesnice. Počáteční a koncový bod by měl být vždy vložen automaticky. K vizualizaci tečných vektorů křivky mohou sloužit úsečky zakončené šipkou a manipulace s tečnými vektory může být realizována tažením myši.



Obrázek 4: Návrh GUI – 1. verze

Po konzultaci s kolegou došlo ke změně návrhu. Součástí lišty v horní části okna bude kromě názvu aplikace také ikona otazníku, po kliknutí myši na ni se zobrazí nové okno s popisem aplikace včetně způsobu jejího ovládání. Okno bude možné zavřít kliknutím myši na tlačítko *Zavřít*. Dále bude v levé části okna menu, kde bude možné si vybrat typ křivky, případně nastavit její vlastnosti, zobrazit řídicí polygon a také vyhodnotit shodu vytvořené křivky se vzorovou křivkou. Pro výběr typu křivky bude sloužit skupina prvků Radio Button. Uvažovali jsme také o použití prvku Combo Box, ale uživatel spíše ocení, pokud bude mít neustále zobrazeno, jaké další křivky aplikace podporuje. Po spuštění aplikace bude vykreslena zvolená křivka (implicitně NURBS křivka). K přidání nebo odebrání řídicího bodu bude sloužit tlačítko *Přidat bod* a *Odstranit bod*. Nastavení váhy řídicích bodů NURBS křivky bude realizováno pomocí prvku Slider. Aktuálně nastavená hodnota váhy se bude navíc zobrazovat nad tímto prvkem. Pro zrušení zobrazení řídicího polygonu B-spline a NURBS křivky bude využit prvek Check Box (implicitně se řídicí polygon bude zobrazovat). Vyhodnocení správnosti řešení úlohy bude použito tlačítko *Zkontrolovat!*. V pravé části okna bude umístěno také menu, jehož obsahem bude stručná charakteristika jednotlivých typů křivek. Po kliknutí myši na příslušnou položku se zobrazí charakteristika dané křivky (prvek Accordion) a její součástí bude také odkaz na webovou stránku s podrobnějším popisem křivky. Důvodem je omezený prostor okna aplikace. Navíc je vhodné uživateli poskytnout nejzákladnější informace o dané křivce a zbytečně jej nezatěžovat informacemi, které nepotřebuje.

Většinu prostoru bude zabírat kreslicí plátno pro vykreslení křivky, protože pro uživatele je klíčové, aby měl dostatek prostoru pro její tvorbu a mohl s ní jednoduše manipulovat. Součástí horní části kreslicího plátna bude lišta, kde bude tlačítko *Uložit* pro export křivky do obrázku a tlačítko *Reset plátna* pro nastavení křivky do výchozího stavu. Dále zde bude zobrazen výsledek vyhodnocení úlohy. Pokud bude řešení úlohy správné, bude text *Výsledek* a následný symbol fajfky zvýrazněn zelenou barvou, v opačném případě bude následovat za textem *Výsledek* symbol křížku a budou zvýrazněny červenou barvou. Pod kreslicím plátnem se budou nacházet vždy tři úlohy pro každý typ křivky. Zadání úlohy bude zobrazeno po jejím výběru. Úkolem bude modifikovat výchozí křivku na kreslicím plátně tak, aby co nejvíce odpovídala vzorové křivce v dané úloze. Vypracovaná úloha bude vyhodnocena po stisknutí tlačítka *Zkontroluj!*, výsledek se zobrazí, jak již bylo řečeno, na liště kreslicího plátna.



Obrázek 5: Návrh GUI – 2. verze

Návrh a implementace back-endu

Pokud by se jednalo o desktopovou aplikaci, bylo by nutné rozhodnout, na kterých operačních systémech bude podporována. V případě webové aplikace by nebylo nutné řešit přenositelnost mezi operačními systémy, ale webovými prohlížeči.

Pro implementaci by bylo vhodné zvolit Qt Framework, který umožňuje snadnou tvorbu uživatelských rozhraní a nabízí integrované vývojové prostředí Qt Creator. Další výhodou je možnost tvorby multiplatformních aplikací. Mezi nevýhody se řadí problém s distribucí aplikací, který je například způsoben nárůstem celkové velikosti aplikace.

Napojení grafického uživatelského rozhraní na back-end je přímé, proto není potřeba přidávat žádnou další funkcionalitu. Back-end aplikace zahrnuje funkce pro práci s křivkami. Aby bylo možné křivku vykreslit, bylo by pro tento účel vhodné použít specializované knihovny. Uvažovala jsem nad knihovnou GSL⁴ (GNU Scientific Library), která je napsána v jazyce C. Další užitečnou knihovnou může být NURBS++⁵, která je napsána v jazyce C++ a umožňuje vygenerovat NURBS křivky různého stupně či uložit jejich data do souborů různých formátů.

Implementace back-endu byla realizována v jazyce JavaScript. Pro vykreslení B-spline a NURBS křivek byla použita JavaScriptová knihovna `ibiblio`⁶. Vykreslení Fergusonovy křivky je realizováno pomocí prvku Canvas jazyka HTML, konkrétně se jedná o metodu `bezierCurveTo()`⁷. Pro zpřístupnění aplikace testovaným osobám byla využita subdoména na adrese `www.itu.mienkofax.eu`.

Návrh uživatelských testů

Testování bych prováděla na skupině lidí v rozmezí 20–60 let (studenti, pedagogové, lidé se zájmem o počítačovou grafiku). Vzhledem k tomu, že se jedná primárně o aplikaci, která bude využívána na vysoké škole, tak by měl být největší počet testovaných z řad studentů. Nicméně je vhodné do testování zahrnout i lidi pokročilejšího věku, kteří se řadí mezi zájemce o počítačovou grafiku. Jejich počítačová gramotnost nemusí být na tak vysoké úrovni, což by mohlo být využito

⁴<http://libnurbs.sourceforge.net/>

⁵<https://www.gnu.org/software/gsl/>

⁶<http://www.ibiblio.org/>

⁷<http://blogs.sitepointstatic.com/examples/tech/canvas-curves/bezier-curve.html>

k otestování dostatečné intuitivnosti a přehlednosti uživatelského rozhraní i celé aplikace. Testování bude probíhat iterativním způsobem ve dvou fázích. Uživatelé budou testováni v přirozeném prostředí (doma, v práci) a expert bude dohlížet na průběh testování aplikace.

V první fázi budou mít uživatelé chvíli čas seznámit se s aplikací (cca 5-10 minut) a poté si zkusí provést základní operace s křivkami – tzn. zvolit typ křivky, modifikovat řídicí body křivky, případně nastavovat jejich váhy či měnit orientaci tečných vektorů křivky.

V rámci druhé fáze se bude cílit na operace, které způsobovaly uživatelům největší potíže. Budou připraveny úkoly, které bude muset uživatel splnit. Otestována by měla být i pokročilá práce s aplikací, tzn. zobrazení řídicího polygonu nebo exportování křivky do obrázku. Časový limit se bude pohybovat v rozmezí 10-15 minut.

Spokojenost uživatele nejlépe zjistíme buď přímo z rozhovoru s uživatelem, nebo formou dotazníků, které by se měly skládat z otevřených i uzavřených otázek. Touto cestou uživatel může plně vyjádřit svoje názory, připomínky a postřehy.

Testování by také mohlo vypadat tak, že uživatelé budou rozděleni na dvě skupiny – na laiky a na lidi věnující se počítačové grafice. V průběhu plnění jednotlivých úloh uživatelem by nebyl přítomen expert, vyhodnocení by probíhalo formou nepřímého pozorování s nahráváním/natáčením. Výhodou je, že by uživatel nebyl vystaven stresu z přítomnosti experta, což by mohlo příznivě ovlivnit výsledky jeho testů. Na druhé straně expert v průběhu testování nemůže pokládat dodatečné otázky testovanému, jehož odpovědi by mohly být pro experta užitečné.

Klíčovým prvkem GUI je zejména skupina prvků Radio Button, pomocí které si uživatel zvolí požadovaný typ křivky. Dalším důležitým prvkem je prvek Slider pro nastavení váhy řídicích bodů křivky a prvek Check Box pro zobrazení řídicího polygonu. Mezi důležité prvky patří také tlačítko Reset plátna, které nastaví křivku do výchozího stavu.

Efektivita prvků se pozná podle doby, která uplyne od spuštění aplikace po dobu, než si uživatel všimne daného prvku a použije jej. Dalším způsobem může být analýza pohybu myši, díky které lze poznat, kde uživatel podvědomě předpokládá umístění klíčových prvků.

Během testování se budou provádět nejdříve nejméně složité úlohy, jejichž složitost se bude postupně zvyšovat. Uživatel se seznámí s aplikací a v průběhu práce se bude postupně zdokonalovat. Seřazení úloh od nejsložitější k nejméně složité není vhodné, protože by pravděpodobně uživateli trvalo mnohem déle, než by se s aplikací naučil efektivně pracovat.

Nástroje pro tvorbu GUI

Pro implementaci jsme využili webových technologií. Konkrétně se jedná o CSS framework `Bootstrap`⁸, který je jedním z nejvíce používaných frameworků a je snadno rozšiřitelný. Dále je využit jazyk HTML pro definici objektů stránky a jazyk CSS pro jejich grafické úpravy. Ovládání aplikace je implementováno za pomoci jazyka JavaScript.

Výhodou webových technologií je nezávislost na použitém zařízení a operačním systému, díky čemuž bude moct aplikaci využívat široká skupina uživatelů. Díky uvedené nezávislosti není tedy například nutné řešit instalaci či vytváření verzí pro různé operační systémy. Možnost správy systému z jednoho místa a zpřístupnění pouze části, ve které se aplikace zobrazuje, se řadí k jejich dalším výhodám. Mezi nevýhody patří znemožnění uživatelům aplikaci použít v případě, že je místo, kde je aplikace zobrazována, nedostupné. Za omezení může být považována potřeba přístupu k internetu při práci s aplikací.

Programování GUI

Pro implementaci grafického uživatelského rozhraní byly použity jazyky HTML, CSS a JavaScript. Pro vytvoření kostry šablony jsme použili `Bootstrap` frameworku, který využívá jazyky HTML a CSS. Poté jsme začali implementací vykreslení podporovaných křivek, tedy Fergusonovy, B-spline a NURBS křivky. Následně bylo třeba doplnit vyhodnocení vypracované úlohy, které je založeno na porovnání uživatelem definované křivky se vzorovou křivkou. Nakonec bylo naprogramováno nastavení váhy řídicích bodů křivky, zobrazení řídicího polygonu, reset kreslicího plátna, export křivky do obrázku a dynamické zobrazování panelů. Na závěr jsme doladili celkový vzhled aplikace, což zahrnovalo úpravu použitých barev a rozložení prvků.

⁸<http://getbootstrap.com/>

Pro dynamické zobrazování jednotlivých prvků jsou využity události poskytované jazykem JavaScript. Nejvíce jsme využívali událost `onclick`, která byla použita při výběru typu křivky, kdy se současně zobrazí popis charakteristických vlastností dané křivky. Stejná událost byla zvolena i pro zobrazení jednotlivých úloh k vybrané křivce.

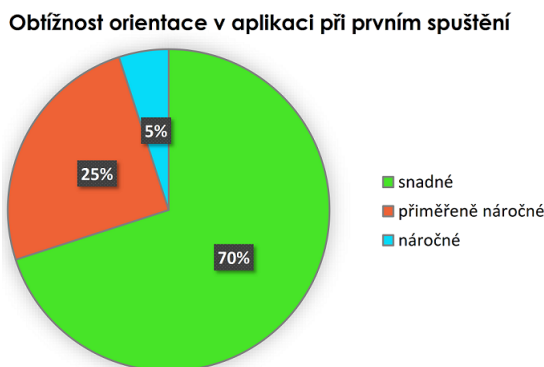
Element třídy `container` obsahuje jednotlivé elementy zarovnané a vycentrované na střed a je rozdělen na tři části. Levá část je definována třídou `box` a třídou `col-md-2`. K návratu kreslicího plátna do původního stavu se využívá funkce `reset2()`. Pravá část je definována třídou `box` a třídou `col-md-3`. Střední část je tvořena třídou `center-box`. Jednotlivé křivky se kreslí v elementu `canvas`, který má jedinečný identifikátor, na základě kterého je možné element `canvas` zobrazit nebo skrýt. Jedinečný identifikátor úlohy slouží taktéž pro zobrazení či skrytí jejího zadání.

Funkcionalitu aplikace zajišťují skripty v jazyce JavaScript – `bootstrap.cs`, `bootstrap.min.js`, `bspline.js`, `curves.js` a `nurbs.js`. Při kliknutí na daný typ křivky se vyvolá příslušná funkce. V případě Fergusonovy křivky se vyvolá funkce `fer()`, v případě B-spline funkce `bspline()`, pro křivku NURBS funkce `nurbs()`. Přidávání řídicích bodů zajišťuje funkce `addPointInto()`, jeho odstranění funkce `removePointFrom()`. Zobrazení řídicího polygonu je implementováno ve funkci `zobraz()`. Při kontrole vypracované úlohy je volána funkce `compare2()`. Export zobrazené křivky do obrázku zajišťuje funkce `ExportObrázka()`.

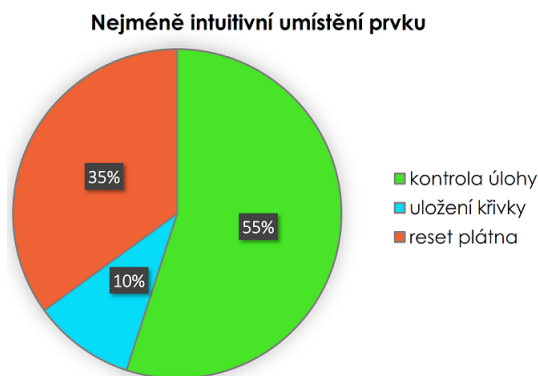
Uživatelské testy

Cílem testování bylo odhalit, zda aplikace splňuje svůj účel. Mým záměrem bylo tedy zjistit, které části aplikace jsou pro uživatele nejméně srozumitelné a také, co by uživatelé z pohledu grafického uživatelského rozhraní dále ocenili.

Testování bylo provedeno dle navrženého testovacího protokolu a bylo testováno celkem dvacet lidí, mezi kterými byli jak studenti, tak i starší lidé se zájmem o počítačovou grafiku.



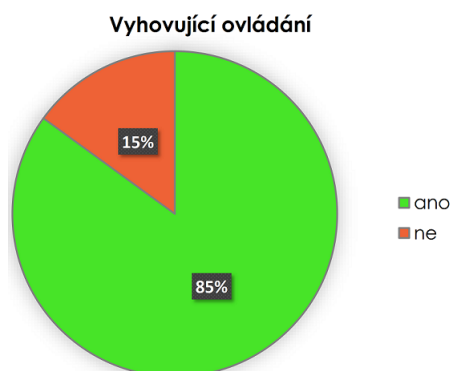
Obrázek 6: Graf k otázce 1



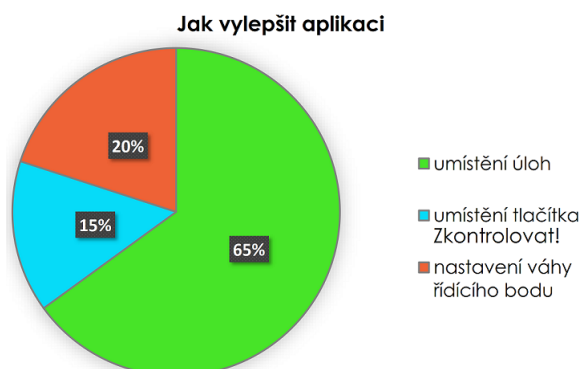
Obrázek 7: Graf k otázce 2



Obrázek 8: Graf k otázce 3



Obrázek 9: Graf k otázce 4



Obrázek 10: Graf k otázce 8

Výsledky a závěr

V první fázi se všichni testovaní s aplikací seznámili velmi rychle, přibližně za 5 minut. V rámci druhé fáze se testovaným podařilo splnit všechny úkoly v daném časovém limitu. Vyhodnocením dotazníků bylo zjištěno, že orientace v aplikaci při jejím prvním spuštění nedělá uživatelům větší problémy. Co se týče rozmístění prvků aplikace, uživatelé byli s řešením víceméně spokojeni. Za nejméně intuitivní bylo označeno umístění tlačítka *Zkontrolovat!*, které by uživatelé očekávali na liště kreslicího plátna. Dále by mnozí předpokládali, že tlačítko *Uložit* pro uložení křivky jako obrázek bude součástí levého menu. Testovaní jako důvod většinou uváděli, že při uložení obrázku křivky se v podstatě jedná o uložení jejího nastavení. Velká část uživatelů navrhla jako možné vylepšení přesunutí zadání úloh tak, aby se vůbec nemuselo pohybovat s obsahem okna. Nejobtížnějším úkolem bylo odhadnout, kdy je křivka natolik podobná vzoru, že je úloha vyhodnocena kladně. Řešením by bylo zvýšit toleranci odlišnosti definované křivky od vzorové křivky.

Z výsledků testování vyplývá, že bylo dosaženo snadného a intuitivního ovládání, což bylo jedním z cílových požadavků na aplikaci. Tento závěr je podložen rychlým seznámením se s aplikací, a to do 5 minut. Navíc uživatelé sami v dotaznících uváděli, že považují ovládání za intuitivní. Doba, za kterou si uživatelé po spuštění aplikace všimli potřebného prvku, se pohybovala v rozmezí 2 – 5 sekund, což svědčí o vhodně rozvrženém umístění prvků. Testovaným byla na závěr položena otázka, zda jsou spokojeni s náročností prezentované problematiky, nebo zda si myslí, že by mělo být o křivkách uvedeno více či méně informací. Podrobnost popisu křivek byla hodnocena velmi kladně. Mnozí ocenili stručnost a výstižnost popisu křivky bez zbytečného uvádění podrobností. Náročnější uživatele využili odkazu pro detailnější popis vlastností křivky.

Týmová spolupráce

Během vzájemné spolupráce nevznikly žádné závažné neshody. Velký přínos týmové spolupráce vidím ve větším množství nápadů na realizaci grafického uživatelského rozhraní, než by byl schopen člověk vymyslet sám. Nejdříve vytvořil svůj návrh každý z členů týmu, následně jsme návrhy prokonzultovali a vznikl návrh, který vycházel z námi vytvořených návrhů a se kterým jsme byli oba spokojeni. Další výhodou je možnost konzultovat své problémy při implementaci s dalším členem týmu. Mnohdy totiž člověka napadnou řešení až při vysvětlování problému svému kolegovi.

Omezující byla nutnost informovat druhého člena týmu o změnách, které byly prováděny v aplikaci. Nesměla chybět také následná diskuze, jejíž cílem bylo zdůvodnit provedené změny a zjistit, zda s nimi druhý člen týmu souhlasí. Dále byl proces vývoje aplikace zpomalen tím, že každý implementoval jen část aplikace, tudíž členové museli pochopit jimi neimplementované úseky kódu. Obtížnější bylo rovněž najít takové barevné kombinace prvků grafického uživatelského rozhraní, se kterými by byli spokojeni oba členové týmu.

Závěr

Cílem bylo vytvořit aplikaci, která napomůže studentům předmětu IZG, ale i lidem se zájmem o počítačovou grafiku, pochopit problematiku Fergusonových, B-spline a NURBS křivek. Snažili jsme se zejména motivovat uživatele k přemýšlení o vlastnostech křivek a o vlivu jejich parametrů na výsledný tvar křivky. Z tohoto důvodu jsme do webové aplikace zahrnuli také jednoduché úlohy s možností vyhodnocení správnosti řešení. Na základě dosažených výsledků testování lze říct, že aplikace splnila svůj účel a většině testovaných usnadnila pochopení problematiky zmíněných křivek. Jednou z nejvíce oceňovaných možností bylo plnění úkolů, díky čemuž se stalo studování vlastností křivek pro mnohé zábavou.

Přílohy

Testovací protokol aplikace pro demonstraci křivek v počítačové grafice

Úkoly

- spustit aplikaci
- vybrat Fergusonovu křivku
- nastavit libovolně orientaci tečných vektorů
- vybrat NURBS křivku
- modifikovat (posunout) jeden řídící bod křivky
- změnit váhu u jednoho bodu křivky
- zobrazit řídící polygon
- exportovat křivku jako obrázek
- resetovat kreslící plátno
- vypracovat a poté vyhodnotit úlohu 1 uvedenou u B-spline křivky

Otázky

1. Jaká je vaše úroveň vzdělání v oblasti křivek v počítačové grafice?
Zvolte pouze jednu možnost.
 - jsem laik, o křivky se zajímám ve svém volném čase
 - nejsem laik, mám odbornější znalosti o křivkách
2. Jak moc náročné pro Vás bylo se zorientovat v aplikaci po jejím prvním spuštění?
Zvolte pouze jednu možnost.
 - snadné
 - přiměřeně náročné
 - náročné
3. Vyhovovalo Vám ovládání aplikace?
Zvolte pouze jednu možnost.
 - ano
 - ne
4. Jaké umístění prvku aplikace Vám připadalo nejméně intuitivní a proč?
5. Podařilo se Vám splnit všechny úkoly?
Zvolte pouze jednu možnost.
 - ano
 - ne
6. Pokud ne, tak které se Vám nepodařilo splnit?
7. Který úkol považujete za nejobtížnější a proč?
8. Co byste v aplikaci změnil/a (rozmístění prvků aplikace, způsob jejího ovládání, ...) a proč?