

INTUITIVE DRAWING ON THE ANDROID PLATFORM

Martin Appl

Master Degree Programme (3), FIT BUT

E-mail: xapplm00@stud.fit.vutbr.cz

Supervised by: Adam Herout

E-mail: herout@fit.vutbr.cz

Abstract: This paper deals with the implementation of a bitmap drawing application on the Google Android platform. Intuitive streamlined and fast user interface is the primary focus. The solution also offers extensive undo functionality and interpolation which guaranties smooth curves even with fast strokes.

Keywords: Android, User Interface, Drawing, Sketching, Bitmap editor

1. ÚVOD

Tématem práce je vývoj kreslicí a skicovací aplikace pro mobilní zařízení na platformně Android. Důraz je kladen na intuitivní uživatelské rozhraní, které je navíc plně přizpůsobitelné a slibuje zrychlení práce umístěním často používaných nástrojů a nastavení na příhodná místa podle preferencí uživatele. Další usnadnění práce nabídnou optimalizace funkcí jako krok zpět. Pro dosažení vizuálně kvalitních výsledků bylo nutné zpracovat interpolaci křivkami u bodů popisujících tah prstem. Tato práce popisuje převážně základ nutný k fungování aplikace, vlastní grafické uživatelské rozhraní je stále ve vývoji.

Konkurenční aplikace nezávislých vývojářů se zaměřují více na kvantitu poskytovaných funkcí, než na komfort používání. Profesionální aplikace, kterou jsem testoval, Autodesk® SketchBook® Mobile, má velmi dobře zpracované uživatelské rozhraní. Chybí mu však přizpůsobitelnost a možnost rychlého přístupu k předem připraveným nastavením.

2. IMPLEMENTACE FUNKCÍ PROGRAMU

2.1. INTERPOLACE

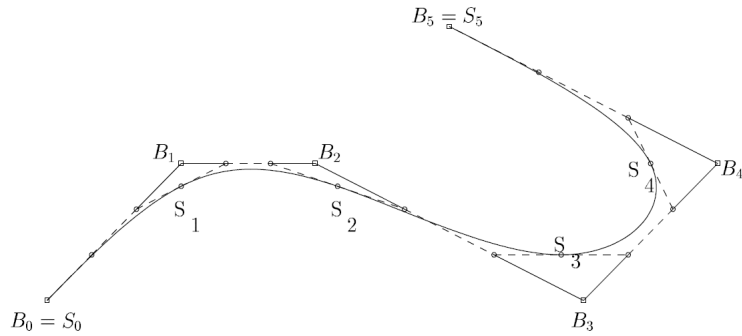
Knihovna platformy dodává data z dotykové obrazovky ve formě posloupnosti bodů. Pokud provedeme rychlý pohyb, je vzdálenost bodů od sebe velká. Použijeme-li naivní řešení a pouze pospojujeme body úsečkami, budou ve výsledné kresbě patrné ostré hrany. Tímto neduhem trpí vzhledem k netriviálnímu řešení mnoho aplikací nezávislých vývojářů.

Základní princip algoritmu štětce spočívá v otiskování bitmapy představující hrot v konstantních vzdálenostech od sebe podél dráhy tahu [1]. Ilustraci principu vidíme na obrázku 1. Nejsnazší na implementaci konstantních vzdáleností je reprezentace dráhy jako posloupnosti úseček navazujících v koncových bodech. Interpolátor tedy převádí reprezentaci křivkami na reprezentaci úsečkami a dopočítává nové body pro lepší rozlišení.

Nejprve je ale nutné převést posloupnost bodů na spline křivku. Konkrétně je použit uvolněný kubický uniformní B-spline. K výpočtu řídicích bodů se využívá podmínek spojitosti křivky. Výsledný spline je spojený v druhé derivaci [2]. Na obrázku 2 vidíme souvislost řídicích bodů spline křivky označených písmeny *B* s koncovými a řídicími body jednotlivých Bézierových křivek tvořících segmenty splinu. Koncové body jsou označeny písmeny *S* a řídicí body nejsou označeny. Dohromady body sousedících segmentů tvoří tzv. *A-frame*.



Obrázek 1: Vliv vzdálenosti otisků hrotu štětce na stopu



Obrázek 2: Konstrukce uvolněného uniformního kubického B-spline

2.2. ZVĚTŠENÍ A POSUNUTÍ

Uživatel může během kreslení zvětšovat, zmenšovat a posouvat zobrazenou výseč plátna. Slouží k tomu standardní gesto dvěma prsty pro zvětšování a tažení dvěma prsty s konstantní vzdáleností od sebe pro posunutí.

Implementace využívá podporu transformačních matic v grafické knihovně platformy Android. Informace z gest se přenáší do transformační matice, která je pak použita při vykreslování plátna. Inverzní matice je pak použita pro převod souřadnic z dotykové obrazovky na souřadnice plátna.

2.3. HISTORIE

Velkým hendikepem grafických aplikací především na mobilních zařízeních je omezený počet akcí, které lze vrátit zpět. Obecně je to způsobeno velkými paměťovými nároky nutnými pro uchování historických bitmapových dat. Řešení historie v této práci se pokouší obejít paměťové omezení vektorovým ukládáním informací o dráze nástroje spolu s informacemi nutnými pro renderování výsledku, jako je nastavení štětce nebo barva. Na akce uživatele lze pohlížet jako na příkazy [3]. Jejich opětovným provedením docílíme stejného výsledku.

Řešení naráží s přibývajícím množstvím stop které je nutné renderovat k obnovení obrazu na výkonové omezení, které ho dělá bez dalších optimalizací nepoužitelným. Mezi vektorové informace, které se renderují, je nutné vkládat ve stanoveném intervalu bitmapový obrázek, takzvaný „key frame“. Starší informace pak už není nutné renderovat. Opět se ale zvýší paměťová náročnost. Řešení je tedy určitým kompromisem mezi výpočetní a paměťovou náročností. Je snadno odladitelné nastavením intervalu mezi jednotlivými „key frames“.

2.4. ŠTĚTCE

Aplikace implementuje několik základních štětců. Nejzajímavější je nástroj tužka, který reaguje silou čáry na rychlost pohybu prstu po obrazovce. V budoucnu je v plánu štětec který bude plně uživatelsky konfigurovatelný včetně reakce na rychlost a směr pohybu po obrazovce.

Zajímavá je také možnost využití polohových senzorů telefonu k ovlivňování parametrů štětce.

3. UŽIVATELSKÉ ROZHRANÍ

Základem uživatelského rozhraní je menu s tlačítky. Menu je možné gesty posouvat do stran, což řeší pozdější rozšiřitelnost uživatelského rozhraní o nové funkce. Tlačítka je rovněž možné tažením přemísťovat. Chování „Drag & Drop“ je podobné jako ve standardní Launcher aplikaci Androidu.



Obrázek 3: Posuvné menu

Některá tlačítka v hlavním menu vyvolají podmenu, které vypadá a chová se stejně jako hlavní menu. Jiná vyvolají dialog s příslušnými nastaveními.

Tažením tlačítka na plochu aplikaci umístíme buď zástupce tlačítka nebo widget, umožňující rychlou volbu několika uživatelských nastavení parametrů spojených s dialogem, které tlačítko vyvolá.

Sémantika akce vyvolané přetažením tlačítka závisí na ovládacím prvku nad kterým tažené tlačítko pustíme.

4. ZÁVĚR

V aplikaci se podařilo implementovat základní funkce kreslicího programu v rozumné kvalitě. Interpolátor garantuje hladké křivky i při rychlých tazích a historie umožňuje návrat do libovolného bodu s téměř konstantní časovou složitostí.

Hlavní důraz je kladen na přizpůsobitelné uživatelské rozhraní, jenž je pravidelně konzultováno s grafikem, který má s podobnými programy dlouholeté zkušenosti. Zde popsané funkce tvoří základní sadu schopností podobných programů, avšak zejména historie a interpolace je zpracována kvalitněji než je jinak běžné. V dalších fázích se práce zaměří na optimalizaci uživatelského rozhraní a ovládání.

REFERENCE

- [1] POSCH, K. C. a W. D. FELLNER. The circle-brush algorithm. *ACM Transactions on Graphics*. 1988, roč. 8, č. 1, s. 1-24. ISSN 07300301. DOI: 10.1145/49155.49156. Dostupné z: <http://portal.acm.org/citation.cfm?doid=49155.49156>
- [2] BAKER, Kirby. *Cubic spline curves* [online]. Los Angeles, 2002 [cit. 6.1.2012]. Dostupné z WWW: http://www.math.ucla.edu/~baker/149.1.02w/handouts/dd_splines.pdf. UCLA University of California
- [3] ARCHER, James E., Richard CONWAY a Fred B. SCHNEIDER. User Recovery and Reversal in Interactive Systems. *ACM Transactions on Programming Languages and Systems*. 1984, roč. 6, č. 1, s. 1-19. ISSN 01640925. DOI: 10.1145/357233.357234. Dostupné z: <http://portal.acm.org/citation.cfm?doid=357233.357234>