

Vstupní rozhraní mice2mouse

Projektová dokumentace do předmětu ITU 2011/12

Radek Fér xferra00@stud.fit.vutbr.cz
Miroslav Paulík xpauli00@stud.fit.vutbr.cz

22. května 2014, FIT VUT Brno

Obsah

1	Úvod	1
1.1	Princip činnosti	1
2	Návrh	2
2.1	Slučovač	2
2.2	Fyzické uspořádání	2
2.3	Existující zařízení	3
3	Implementace	3
3.1	Slučovač	3
3.2	Použité knihovny	4
3.3	Demonstrační aplikace	4
4	Experimenty a testování	5
4.1	Strategie experimentování a testování	5
4.1.1	Výběr účastníků experimentů	5
4.1.2	Experiment č. 1 - Požadavky na ovládání a orientaci v 3D prostředí . . .	5
4.1.3	Pilotní test	6
4.1.4	Experiment č. 2 - Efektivnost práce s rozhraním m2m	6
5	Závěr	6
A	Návrh zadání	7

1 Úvod

Chceme-li pokytnout uživateli nějaké počítačové 3D aplikace maximální komfort při manipulaci s objekty ve 3D prostředí, je nutné mu umožnit provádět operace ve všech 3 osách **nezávisle** – v současné době (vyjma použití speciálních vstupních zařízení, viz část 2.3) to není možné.

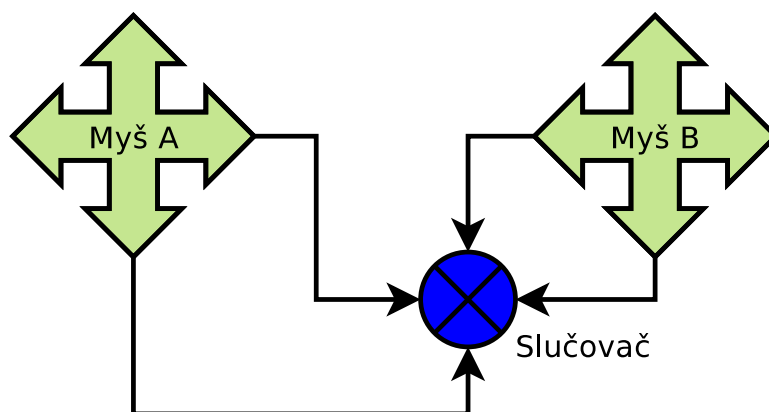
Tento projekt si klade za cíl odzkoušet možnosti využití speciálního vstupního zařízení (zkonstruovaného podle myšlenky jednoho z autorů), které **využívá 2 dostupná 2D polohovací zařízení (myš, touchpad, ...)** k vytvoření jednoho 3D polohovacího zařízení. A to tak, že pohyby oněch dvou 2D polohovacích zařízení jsou slučovány do pohybů nad 3D.

Dále si projekt klade za cíl navrhnout a otestovat různá uživatelská rozhraní využívající tohoto zařízení tak, aby se práce s virtuálním 3D prostorem stala více efektivnější, intuitivnější a příjemnější.

Možnosti využití tohoto zařízení jsou všude tam, kde je třeba efektivně pracovat ve virtuálním 3D prostoru (modelování, hry, ovládání skutečných modelů, ...)

1.1 Princip činnosti

Princip činnosti navrhovaného zařízení je zobrazen na obrázku 1.



Obrázek 1: Princip činnosti zařízení

Nejdůležitější část zařízení tvoří **slučovač**, kde dochází k mapování čtyřech vstupních os (osy x a y , u obou myší) na osy 3 (odpovídající osám 3D prostoru, ve kterém chceme pracovat). Odtud název „mice2mouse“, příp. zkratka „m2m“. Obecné informace jsou uvedeny v sekci 2.1, konkrétní implementace je pak popsána v sekci 3.1.

2 Návrh

Zařízení je navrženo tak, aby jeho služeb mohl využívat úplně každý. Hardwarové vybavení (tedy de facto pořizovací cena zařízení – software je šířen v otevřené podobě pod GPL licencí) obnáší pouze 2 klasické myši.

2.1 Slučovač

Výstupem ze 2 myší jsou 4 nezávislé osy, které chceme nějak sloučit do 3 a existuje mnoho kombinací, jak to udělat. Zařízení by mělo umožňovat definici vlastního mapování, dle chuti každého.

Pro jednoznačné určení mapování budeme nadále používat tzv. *mapstring*, který určuje, která osa na kterou bude mapována. Myši rozlišujeme pomocí písmen **A** a **B**.

Příklad mapstringu: *xyxz* – osa *x* myši A se mapuje na osu *x*, osa *y* myši A se mapuje na osu *y*, osa *x* myši B se mapuje na osu *x* a osa *z* myši B se mapuje na osu *z*. V kapitole o testování (??) se dozvíte, že mapování *xzxy* vyhovovalo většině lidí, na druhém místě pak skončilo mapování *xyxz*.

2.2 Fyzické uspořádání

Při návrhu zařízení jsme zohlednili praktickou stránku, jak zacházet se 2 myšmi:

- **Vše v 1 fyzické rovině**
 - + obě ruce stejně zatížené, žádná nemusí držet váhu myši a sebe u vertikálně položené podložky
 - ne tak intuitivní, jako následující uspořádání
- **Podložky obou myší fyzicky umístěny ortogonálně** – vlastnosti viz předchozí uspořádání, ale naopak
- **Varianta pro 1 ruku** – na klasické myši je zleva (myš pro praváky) umístěn drobný touchpad ovládaný palcem ovládající zbývající osu

Nabízí se ještě možnost ovládat zbývající osu kolečkem myši, nicméně tuto možnost jsme dále nezkoumali. Kolečka většiny myší mají navíc nedostačující rozlišení (což by nesplňovalo podmínku, že chceme ovládat všechny 3 osy nezávisle a rovnocenně).

2.3 Existující zařízení

Obecný termín pro takováto zařízení je **3D myš**.

Na trhu se již dlouho vyskytují výrobky s podobnou funkcionalitou. Jde např. od výrobky fy *3Dconnexion*, kupř. **SpaceNavigator** – zařízení určené pro pohyb a manipulaci s objekty ve 3D. Toto zařízení umožňuje ovládat 6 směrů pomocí jedné ruky (tažením, rotací, stlačením, vytážením apod.).

Další možnosti:

- výrobky fy **Axsotic** – něco podobného, využívá standardu HID
- **Wii Remote Nunchuk** – obsahuje 3D akcelerometr a tlačítka
- jistě by se našly i další

3 Implementace

Součástí projektu je implementace ovladače zařízení a dále několika demonstračních aplikací. Kód je psán jazykem C (slučovač) a C++ (demo aplikace). Slučovač je nyní možné přeložit pouze v OS GNU/Linux, díky silné a nutné vazbě na jeho vstupní subsystém. Demonstrační aplikace (a tedy jakýkoli klientský software, který by chtěl využívat naše rozraní) už přenositelné jsou (je zde pouze závislost na knihovnách SDL a GLUT).

3.1 Slučovač

Jak již bylo zmíněno, implementace slučovače je silně vázána na vstupní subsystém Linuxového jádra. V této části jsou tyto vazby popsány spolu s tím, jak přesně slučovač funguje.

Vstupní sybsystém v Linuxu

Každé vstupní zařízení rozpoznané Linuxem má v adresáři `/dev/input/` vlastní speciální soubor, který je možný použít z userspace k přístupu k tomtuto zařízení, typicky pro čtení událostí. Všechny události mají společný formát a jejich popis je možné nalézt v `/usr/include/linux/input.h`.

Nás zajímají události o pohybu myši, jejichž typ je `EV_REL` a dále se dělí na zprávy `REL_X` a `REL_Y`. Každá takováto zpráva nese navíc časové razítko a velikost změny polohy v dané ose.

Upozornění: Xka (Xorg server) si většinou myš, která je nastavena jako *CorePointer* zaberou pro sebe a ta je poté pro další čtení „němá“.

Slučovač tedy nedělá nic jiného, než že čte ze dvou souborů (např. `/dev/input/event10` a `/dev/input/event11`), tyto data interpretuje jako události a hodnoty výchylek os mapuje podle *mapstringu* na výchylky os výstupních.

Virtuální zařízení a modul `uinput`

Zde popisujeme, jak nabídnout výstup slučovače k dispozici aplikacím.

Původní prototyp našeho slučovače je obsažen v souborech `mice2mouse.{cpp|h}`. Aplikace si pomocí tohoto modulu mohla zaregistrovat callback funkce volané při nějaké události. Nicméně musela také zajistit pravidelné volání funkce `m2m_workHorse()`, která zjišťovala, zdali je nějaká událost k dispozici a příp. volala registrované funkce. Kompletní aplikace využívající tento modul je v souboru `main.cpp`.

Tento přístup není ideální, pro jsme hledali jiné řešení a to našli v modulu linuxového jádra s názvem **`uinput`**. Tento modul umožňuje vytvořit virtuální vstupní zařízení a z userspace ho „krmit“ událostmi. Implementace slučovače využívající služeb tohoto modulu je v souboru `m2m_device.c`.

Výsledný spustitelný program má 2 povinné parametry – názvy speciálních souborů zařízení pro myši **A** a **B** a jeden volitelný parametr (**-m**) udávající mapování, tedy *mapstring*. Výchozí mapování je *xyxz*.

Poznámka: pro úspěšné spuštění je nutné mít v jádře zavedený modul `uinput`. To se dá zjistit pomocí příkazu `lsmod | grep uinput` a případně napravit pomocí `modprobe uinput`.

Po spuštění se vytvoří virtuální zařízení (kde jinde než v `/dev/input/`) s názvem „m2m“. Toto zařízení, ač se tváří jako klasický joystick (události typu `EV_ABS` pro 3 osy a nějaká tlačítka) ale ve skutečnosti v události zasílá relativní odchylku. Toto je kvůli zjednodušení implementace. Díky tomu, že je zařízení knihovnami jako GLUT či SDL rozpoznáno jako joystick, nemusíme se o nic starat a pouze využívat standardních funkcí těchto knihoven (např. v SDL funkce `SDL_JoyAxisEvent()`).

Pouze je nutné se při výběru joysticku dotazovat na joystick se jménem `m2m`. Toto by šlo v budoucnu jistě vyřešit čistěji.

3.2 Použité knihovny

Jak již bylo uvedeno, klientské aplikace mohou využít knihovny GLUT či SDL, příp. jiné, které mají nějakou podporu pro joysticky.

Pokud chceme mít aplikaci minimální, může sama způsobem uvedeným výše zpracovávat události ze souboru náležejícímu zařízení `m2m` v `/dev/input/`.

3.3 Demonstrační aplikace

Součástí výstupu projektu je implementace demonstračních aplikací využívající naše rozhraní. Vytvořili jsme jednoduché **intro**, dále aplikaci pro **kreslení ve 3D**, základ pro **3D piškvorky** a nakonec aplikaci **3D moorhuhn**, která byla použita při testování rozhraní. Pro postupné spuštění všech aplikací slouží příkaz `make run`.

4 Experimenty a testování

Cílem experimentální části je získání užitečných informací o testovaném subjektu a z nich vycházející další vývoj. Tato kapitola bude tedy popisovat strategii testování a experimentů na rozhraní m2m, výběr vhodné skupiny účastníků těchto pokusů a popisem jednotlivých experimentů a analýzy jejich výsledků.

4.1 Strategie experimentování a testování

Úplně jako první bylo potřeba zjistit, jakým způsobem by měla být aplikace ovládána, které tlačítka obou polohovacích zařízení se budou používat ke kterým činnostem a jak správně nastavit osy pohybu každého ze vstupních zařízení. Jelikož má rozhraní pracovat v prostředí 3D, musí navíc existovat prostředek pro prostorovou orientaci. Cílem prvního experimentu se tedy stalo zjištění reakcí uživatelů na různé možnosti ovládání prostorového kurzoru a jeho přizpůsobení při operaci otáčení 3D scény. Získání výše zmíněných informací umožnilo vytvořit několik demonstračních aplikací, jež se později staly instruktážními nástroji pro poslední experiment. Než k němu ale došlo, byl proveden pilotní test celé sady instruktážních aplikací kvůli zjištění možných nedostatků ještě před samotným testováním. Cílem celého procesu experimentování a testování však bylo hlavně získání údajů o míře efektivnosti práce s rozhraním m2m. A právě to bylo náplní závěrečného experimentu.

4.1.1 Výběr účastníků experimentů

Největší vliv na výběr skupiny testujících uživatelů byla jejich schopnost pracovat s počítačem, uživatel mající problém obsluhovat jedno polohovací zařízení bude jistě mít problém i se zařízením druhým. Proto jsme požádali o účast v experimentování skupinu 25-ti vysokoškolských studentů, kteří s požadavkem běžného ovládání počítače nemají problém.

4.1.2 Experiment č. 1 - Požadavky na ovládání a orientaci v 3D prostředí

Jak již bylo uvedeno výše, cílem prvního experimentu bylo nalezení optimálního ovládání a zjištění dalších možných nastavení, které by si eventuálně uživatel mohl explicitně zvolit. Jako forma získání informací byl zvolen dotazník. Ten byl rozdělen na 2 části. První byla zaměřena na informace o dotazované osobě a jejich zkušenostech s prací v 3D, zatímco druhá na jejich očekávání ohledně práce s 3D objekty.

Přehled důležitých dotazů a vyhodnocení jejich odpovědí:

Preferujete stejnou funkcionalitu tlačítek na obou polohovacích zařízeních?

Odpovědi: ANO (25x), NE (0x).

Které tlačítko preferujete pro natáčení scény?

Odpovědi: LEVÉ (0x), PROSTŘEDNÍ (21x), PRAVÉ (4x).

Preferujete obrácený pohyb myši (posun myši vlevo znamená posun vpravo na obrazovce) před pohybem shodným?

Možné odpovědi: ANO (7x), NE (18x).

Preferujete obrácený význam levého a pravého tlačítka u druhého polohovacího zařízení?
Možné odpovědi: ANO (4x), NE (21x).

Z odpovědí, které byly většinově přikloněny k jedné z možností, vyplynuly požadavky pro ovládání polohovacích zařízení. Velmi překvapivě dopadl dotaz na obrácení významu levého a pravého tlačítka vzhledem opačnému pořadí prstů na druhé lidské ruce. Bylo možné očekávat, že dotazující budou upřednostňovat stejný význam tlačítka ovládaného ukazováčkem stejně jako u druhé ruky. Výsledky průzkumu však ukázaly, že tomu tak být nemusí.

4.1.3 Pilotní test

Pro ověření správné funkčnosti instruktážních aplikací byl proveden pilotní test. Toho se zúčastnili pouze 2 dobrovolníci, kteří však nepatřili do skupiny účastníků prvního a závěrečného experimentu. Tito dobrovolníci však velmi důsledně připomínkovali každou nejasnost a hlásili každou nalezenou chybu. Na základě jejich poznámek k funkcionalitě byl implementován např. tzv. „spinning mód“ umožňující automatickou rotaci kamery, která umožní snazší orientaci v 3D prostoru aniž by bylo nutné explicitně natáčet scénu. Dále bylo upraveno zvýraznění zaměření objektu v měřeném testu nebo vyladěn způsob, jakým uživatel měnil nastavení polohovacích zařízení.

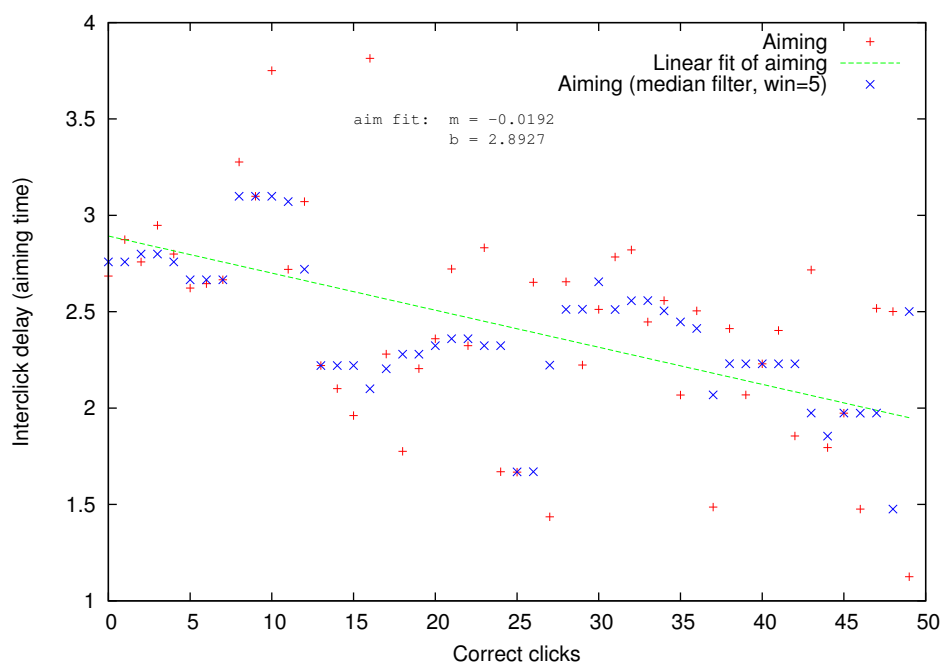
4.1.4 Experiment č. 2 - Efektivnost práce s rozhraním m2m

Cílem závěrečného pokusu bylo demonstrovat rychlost práce s rozhraním m2m a to měřením času potřebného ke splnění úkolu. Před samotným měřením byli všichni účastníci experimentu seznámeni s ovládáním ve třech demonstračních aplikacích, které sloužily k vyzkoušení práce v 3D prostředí a nastavení ovládání. Po tomto seznámení s prostředím byl každému zúčastněnému spuštěn test v podobě aplikace 3D moorhuhn, kde byl měřen čas „zastřelení“ 50-ti krychlí náhodně umístovaných v 3D prostoru.

V tomto experimentu byly měřeny časy mezi jednotlivými zásahy a ty pak byly dále zpracovány. Všechny hodnoty byly interpolovány do přímky, jež znázorňovala relativní zrychlení při zaměřování a zásahu cíle. Obrázek 2 ukazuje, že rychlost nalezení a provedení akce v 3D prostoru se nakonec přiblíží běžné práci v prostředí 2D. To je zajímavé, jelikož po hlubším seznámení se se zařízením bude tato rychlost jistě výrazně větší.

5 Závěr

V rámci projektu m2m bylo vytvořeno rozhraní umožňující intuitivní pohyb v 3D prostoru a ovládání objektů v něm umístěných. Toto rozhraní vzniklo spojením vstupů 2 polohovacích zařízení do jediného virtuálního kurzoru. Pro demonstraci využití m2m byly implementovány 4 aplikace, které dohromady tvoří instruktážní sadu pro přizpůsobení ovládání konkrétnímu uživateli, kterému tak bude umožněno vyzkoušet si alternativní způsob práce v 3D prostředí. S těmito aplikacemi následně proběhl experiment zjišťující informace o práci s tímto rozhraním. Ten poskytl cenné informace pro možné budoucí použití a poskytl nové poznatky k případnému dalšímu vývoji.



Obrázek 2: Graf zlepšení práce v závislosti na čase.

A Návrh zadání

Název: Pokus o vytvoření jednoduchého 3D polohovacího zařízení.

- 2 myši (nebo jiné polohovací zařízení), každá ovládá pozici na jedné rovině (xy , xz či yz).
- Hlavní fór by byl v tom, že by i druhá myš (a ne jen kurzor v rovině kolmé k desce stolu) fyzicky jezdila po rovině kolmé k desce stolu.
- Vhodnou kombinací vstupů z těchto 2 myší lze určit bod v 3D a případně s ním manipulovat (pozice na ose společné pro obě použité roviny by se určila jako průměr souřadnic).
- Možné módy (možno měnit např. kombinací stisků tlačítek na myši):
 - **základní** – posunování 3D kurzoru v prostoru, tlačítkem se provede výběr nejbližšího objektu
 - **rotační** – rotace vybraného objektu pomocí např. pomocí „circular-scrolling“
 - **morfní** – změna nějakého atributu objektu (velikost, barva, tvar, ...)
- Nebo celé úplně jinak, cílem by bylo zkoumat schopnosti interakce takového HW zařízení.
- K demonstraci by sloužila např. hra 3D piškvorky (pro jednoduchost by bylo implementováno pouze označování políček a ne logika hry – cílem projektu by nebylo vytvořit hru 3D piškvorky, ale „ověření zařízení pro práci ve 3D“).
- *openGL*
- Možné využití (jen co mě napadlo): modelování 3D scény, hry, ovládání jeřábu, ...