

Ovládání leteckého simulátoru

Projekt ITU, 2016Z

Číslo projektu: 40

Číslo a název týmu: 24. Tým xdanhe04

Autor: František Daňhel (xdanhe04),

Další členové týmu: Adam Červenka (xcerve16)

Termín řešení: 19. 9. - 19. 12. 2016

Abstrakt

Cílem projektu je navrhnout a otestovat UI jednoduchého leteckého simulátoru. Rozhraní simulátoru by mělo být jednoduché a pro uživatele přívětivé. Nejedná se o návrh vyspělé hry a už vůbec ne o simulátor, který by se snažil co nejvíce podobat realitě.

Důležité je nabídnout uživateli rozhraní, které pro něj bude jednoduché a intuitivně ovladatelné. Bude schopen s ním pracovat bez jakéhokoliv delšího vysvětlování, či prozkoumávání. Uživateli musí být od první chvíle jasné, co se jak ovládá a jak dosáhne požadovaného chování.

Návrh aplikace bude zaměřen především na jednoduchost a použitelnost uživatelského rozhraní, nikoliv aplikace jako celku. Testováním na vzorku potencionálních uživatelů bude zjištěna a vyhodnocena použitelnost rozhraní. Následně je třeba vyhodnotit, zda chování uživatele odpovídá předpokladům o jednoduchosti a intuitivnosti UI. Pokud výsledky nebudou uspokojivé, je třeba se vrátit do fáze návrhu rozhraní a na základě zpětné vazby vylepšit, či kompletně změnit UI.

Cílové požadavky na aplikaci a její rozhraní

Hlavním cílem aplikace je demonstrovat možné ovládání leteckého simulátoru. Vzhledem ke komplexnosti leteckých simulátorů se jedná spíše o jednoduché ovládání, které bude přívětivé pro běžného nenáročného uživatele. Takové UI musí být především přehledné a uživateli musí být na první pohled jasné, co k čemu slouží a co se čím a jakým způsobem ovládá.

To, že UI své cíle naplňuje, se pozná podle testování, jak uživatel prací s aplikací zvládá. Pokud nebude mít uživatel problém se v rozhraní orientovat a bude téměř ihned vědět, co k čemu slouží, tak UI naplnilo svůj cíl.

Studium cílové skupiny a případy použití

Cílovou skupinou jsou především laičtí uživatelé, kteří ocení jednoduchost a orientaci v rozhraní, před věrným zobrazením a ovládáním přibližujícím se realitě. Pokročilejší

uživatelé a hráči pravděpodobně nebudou nadšeni jednoduchým rozhraním, které neodpovídá řízení skutečného letounu a proto je nelze považovat za cílovou skupinu.

Uživatel bude moci aplikaci ovládat pomocí běžných prostředků, jako jsou klávesnice a myš. V úvahu přichází ještě použití herního joysticku, což by ovládání uživateli nejen usnadnilo, ale zároveň by ho přiblížilo více skutečnému letadlu. Ovládání by bylo možné i prostřednictvím myši, což ovšem dle mého názoru komplikuje a znepráhledňuje celé UI a hlavně i samotná interakce se simulátorem by se stala těžkopádnou.

Existující řešení

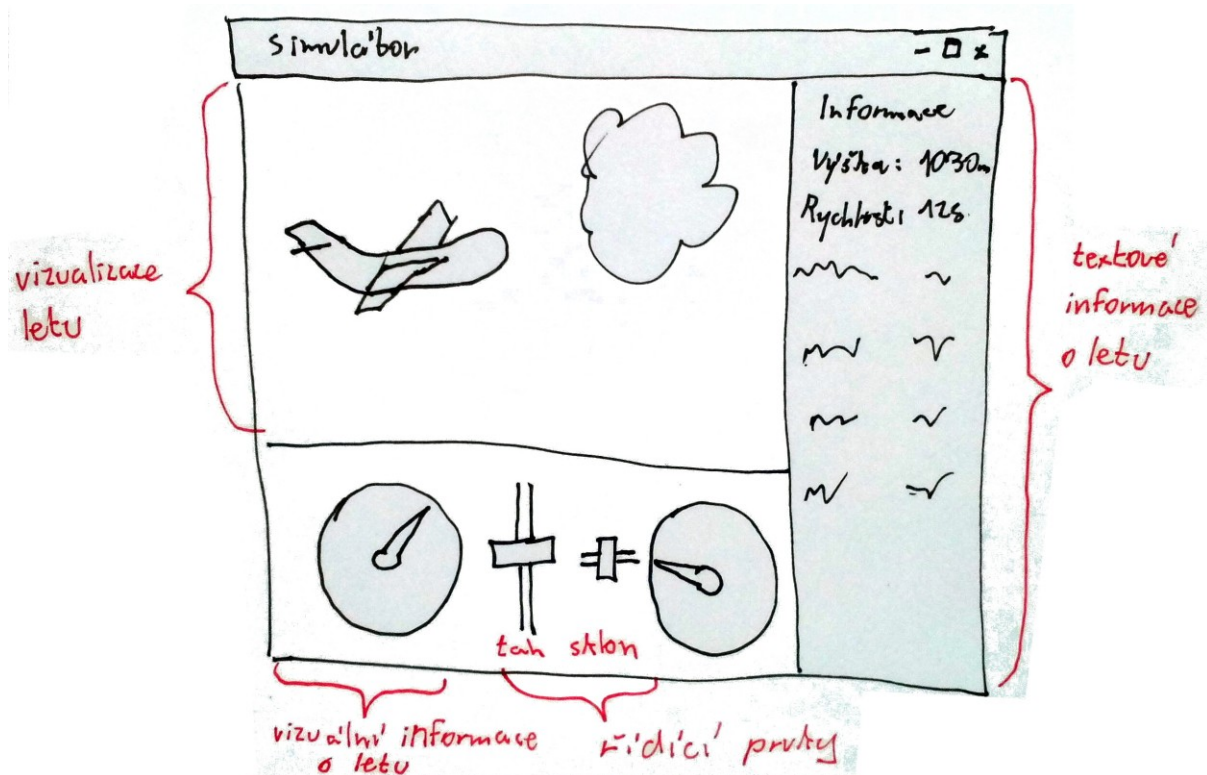
V dnešní době existuje spousta leteckých simulátorů na velmi vysoké úrovni. Může se jednat o hry, které nabízejí propracovanou 3D grafiku, fyzikální model i ovládání v podobě joysticku. Konkrétně lze jmenovat Microsoft Flight Simulator, jehož první verze byla uvedena již v roce 1982^[1]. Existuje ale i jednoduchý simulátor, který obsahuje reálné mapové podklady. Tento simulátor je obsažen jako zpestření aplikace Google Earth, která primárně slouží k procházení satelitních a leteckých záběrů povrchu Země.

Letecké simulátory slouží v praxi pro výcvik pilotů a trénování různých krizových situací, tak stejně jako i pro testování nového leteckého softwaru. Taková řešení jsou velmi nákladná a zahrnují mnoho speciálního vybavení, kdy ovladače, palubní přístroje a další prvky vypadají totožně jako v reálném letadle. Fyzikální model těchto simulátorů je velice propracovaný a složitý, aby mohl věrně napodobit reálné podmínky. Často se jedná o fyzickou část kabiny, která je hydraulicky všemožně naklápěna, aby se dosáhlo, co nejvěrohodnějšího chování letounu. Nejde jen o simulaci vizuální, ale i zvukovou a pohybovou. Nevýhodou je ovšem potřeba speciálního vybavení a s tím spojené ohromné náklady v porovnání s herními simulátory, které jsou většinou provozovány na osobním počítači, popř. herní konzoli.

Návrh GUI

Aplikace musí nabízet uživateli možnost ovládat letadlo a zároveň sledovat průběh jeho letu. Ovládání letadla bude prostřednictvím nastavení páky tahu a nastavení sklonu. Páka sklonu je samovratná, proto je třeba ji v žádané poloze držet dostatečně dlouho. Páky jsou v konečném návrhu reprezentovány komponentami `JSlider`, díky možnosti ovládání jak prostřednictvím myši, tak prostřednictvím klávesnice.

Aplikace se bude chovat jako jednoduchá simulace – později může být rozšířena na zajímavou jednoduchou hru.



Návrh a implementace back-endu

Jedná se o desktopovou aplikaci a proto veškerý backend se děje na klientské straně a nevyužívá se možnosti komunikace se serverem, popř. jinými službami. Backend aplikace představuje balíček `airsimulator.model` a vněmž obsažená třída `Airplane`.

Tato třída představuje velmi zjednodušený fyzikální model letadla. Samotné GUI (třídy v balíčku `airsimulator.GUI`) si vytvoří instanci třídy `Airplane` a pravidelně na ní volají (zhruba každých 200 ms) metodu `simulationStep()`, která aktualizuje model v čase. GUI získává z modelu informace (výška letu, rychlost, atd.), které vypisuje v informačních panelech. Zároveň ovlivňuje nastavení řídicích pák letadla.

Samotnou implementaci backendu jsem se zabýval já. Adam se pracoval převážně na GUI.

```

1. public class Airplane
2. {
3.     public Controls getControls();
4.     /* Vrátí aktuální naklonění letounu (od -30° do 30°) */
5.     public int getGradient();
6.
7.     /* Aktuální vertikální rychlost (v m/s) */
8.     public int getVerticalSpeed();
9.
10.    /* Aktuální horizontální rychlost (v m/s) */
11.    public int getHorizontalSpeed();
12.
13.    /* Výška letu letadla */
14.    public int getAltitude();

```

```

15.
16.     /* Je letadlo zničeno pádem? */
17.     public boolean isDestroyed();
18.
19.     /* Provede krok simulace (jakoby 1/5 vteřiny herního času */
20.     public void simulationStep();
21. }
22.
23. public class Controls()
24. {
25.     public void getGradient(); /* poloha páky sklonu (-1,0,1) */
26.     public void getPower(); /* poloha páky výkonu (-1, 0..100) */
27.     public void setGradient();
28.     public void setPower();
29. }

```

Kód 1 Soupis veřejných metod třídy Aiplane a přidružené třídy Controls

Návrh uživatelských testů

Pro danou aplikaci se mi jako nejvhodnější způsob testování jeví testování s uživateli za přítomnosti vývojářů. Uživatelem simulátoru může být jakýkoliv uživatel, znalý základní práce s PC.

Testování bude probíhat s několika uživateli, které budou převážně z našeho okolí. V mém případě se bude jednat o spolubydlící na bytě a rodinné příslušníky. Výhodou je, že nikdo z mých spolubydlících není zaměřen ani nijak odborně spjat s informatikou. Jedná se převážně studující nebo pracující v různých oborech a odvětvích. Lze jmenovat např. student stavební fakulty, studentka cestovního ruchu, studentka pedagogiky, pracující v kontrole kvality apod.

Test proběhne vyzkoušením aplikace uživatele za přítomnosti jednoho z vývojářů. Vývojář sleduje reakce a chování uživatele – především jeho práci s GUI. Odpovídá na dotazy a slouží jako malá nápověda v případě, že by měl uživatel s GUI potíže.

Uživatelé nebudou mít zadány konkrétní úlohy, vzhledem k povaze simulátoru bude ponecháno čistě na nich, jak si s letadlem „pohrají“. Jako vnímaný úspěch bude posuzováno dostání letounu do vzduchu - stejně tak případné přistání.

Nástroje pro tvorbu GUI

Pro řešení naší aplikace existuje spousta vhodných technologií. Napadlo mě, že bychom mohli udělat simulátor jako webovou aplikaci, ale protože tvorba takového GUI by byla mnohem komplikovanější, vydali jsme se osvědčenou cestou ostatních simulátorů, což je klasická desktopová aplikace.

Nespornou výhodou je absence serveru, a podpora různých GUI komponent, kdy nemusíme „znovu vymýšlet kolo“, ale můžeme se zaměřit na samotnou tvorbu GUI, přičemž se snažíme využít grafické komponenty poskytované přímo technologií.

Projekt jsme se rozhodli realizovat v prostředí a programovacím jazyku Java. Přináší to spoustu výhod. Já i Adam Javu umíme a tak jsme ušetřili čas, jinak strávený učením a poznáváním jazyka. Programování v Javě je jednoduché – nemusíme se zabývat správou paměti, složitými jazykovými konstrukcemi a přímo můžeme rovnou tvořit GUI. Tuto podmínku splňuje např. i WPF, ale Java ho předčila svojí nezávislostí na platformě. Uživatel může spustit simulátor, ať již používá Windows, Mac či Linux. A vždy přitom bude mít dostupné (téměř) totožné GUI.

Nevýhodou námi zvolené technologie je její architektura – virtuální stroj. Vzhledem k tomu, že se jedná o jednoduchý simulátor, výpočty jsou velmi jednoduché a není to komplikací. Pokud by ovšem se jednalo o sofistikovanější výpočty, které by byly náročné, je na místě zvolit nativně kompilovaný jazyk např. C++, který by svoji výkonností vynikal lépe.

Programování GUI

GUI aplikace je implementováno pomocí komponent platformy Java: `JFrame`, `JWindow`, `JButton`, `JLabel`, `JSlider`, `LayoutBorder`... Implementace GUI je především výsledkem práce Adama, já jsem dělal poté drobné vylepšení a napojení na backendovou část.

Uživatelské testy

Při provádění testů s uživateli jsme sledovali jejich chování, reakce a názory, které byly následující:

- doba potřebná pro vzlet (čas od spuštění hry po uvedení letadla do pohybu a stoupání do výšky)
- problémy při vzletu a letu (problémy s ovládáním simulátoru, nejasnosti ohledně letu a letových údajů)
- přistání (přistání, potíže s přistáním, náročnost)
- celkový názor

Výsledky a závěr

Všem uživatelům se podařilo bezproblémově otevřít hru. Líbilo se jim a oceňovali samotnou vizualizaci letu. Jednoduché a intuitivní pro ně bylo ovládání tahu letadla, kdy ihned pochopily, že tím přímo ovládají výkon letadla a pomocí toho jej uvedou do pohybu. O něco složitější byl vzlet s letadlem do výšky, kde obvykle nestíhali sledovat, co ovladačem ovládají a jaký má letadlo sklon.

Vzhledem k tomu, že uživatelé nestihnou sledovat a porozumět údajům v informační tabulce, tak bylo nezbytné vylepšit samotnou vizualizaci letu, do které jsme přidali naklápění letadla, aby bylo vidět aktuální sklon.

Ze začátku hra neobsahovala simulaci havárie, tzn. s letadlem šlo přistát za jakýchkoliv okolností. Toto chování bylo velmi nerealistické, a proto jsem přidal funkci, která hlídá maximální klesání při přistávání. Pokud je tato rychlost klesání překročena, letadlo havaruje.

Následující se ovšem ukázalo mírně nevhodné, protože uživatelé většinou nebyli schopni s letadlem přistát. Proto jsme se rozhodli místo budíku náklonu nahradit budíkem vertikální rychlosti.

V první chvíli tento budík zobrazoval absolutní hodnotu vertikální rychlosti, tudíž nezohledňoval stoupání nebo klesání, později byl vybaven o světýlko, které signalizuje výškový směr pohybu letadla.

Protože ovládání v letadle se liší od reality a každému není známo, přidali jsme do hry nápovědu, kde je stručně napsáno, jak hra funguje a jak ji lze ovládat.

K mému překvapení, si jeden z uživatelů tuto nápovědu otevřel, našel si jak hru ovládat a poté neměl žádné problémy. A to aniž by se mě při testování na něco dotazoval.

Týmová spolupráce

Spolupráce v týmu urychlila celkový vývoj GUI. Pro mě bylo také důležité, že Adam má zkušenosti s tvorbou GUI na platformě Java a v případě jakýchkoliv dotazů, jsem ocenil, že nemusím řešení dlouze hledat v tutoriálech, či API dokumentaci, ale stačilo se pouze zeptat kolegy.

Pro podporu týmové spolupráce na projektu jsme využívali systém Git. Díky němu, jsme mohli současně programovat a mít přehled o práci druhého. Kromě sdílení zdrojového kódu aplikace jsme jej využili také sdílení návrhů rozhraní a jiných souborů.

Závěr

Realizovali jsme jednoduchý letecký simulátor s jednoduchým GUI pro ovládání a vizualizaci samotné simulace letu. Po otestování jsme přišli na několik nedostatků, které jsme se snažili napravit buď změnou GUI - přidáním vizualizovaných prvků a přidáním nápovědy pro uživatele.

Našemu simulátoru bylo často vytýkáno, že byl jednoduchý a představovali by si jej mnohem propracovanější a zábavnější. Z mého pohledu měli v tomto všichni pravdu, na druhou stranu je třeba si uvědomit, že cílem nebylo vytvořit propracovanou hru, ale kvalitní uživatelské rozhraní. Doufám, že se podařilo.

Reference

- [1] https://cs.wikipedia.org/wiki/Microsoft_Flight_Simulator
- [2] <http://gml.vse.cz/data/oppa-webdesign/ui.html>
- [3] <https://phoenix.inf.upol.cz/esf/ucebni/gui-dostal.pdf>

Přílohy

Ukázka mého zápisu při testování s uživatelem:

Jméno: Jan Brčka

student FAST VUT

- doba potřebná pro vzlet: cca 2 minuty
(kocha se grafikou)
- problémy při vzletu a letu:
 - ~~se~~ naklápění stolu na druhou stranu
- přistání:
 - havaroval (až po vylepšení,
poprvé hra havárii nesimulovala)
- názor:
 - dobře, ale chtělo by to více funkcí, přetáčení atd.

Výsledný vzhled aplikace

