VŠB – Technická univerzita Ostrava
Fakulta elektrotechniky a informatiky
Katedra informatiky

2D animace procesu měření

2D animation process measurements

2012 **Jakub Hendrych**

VŠB - Technická univerzita Ostrava
Fakulta elektrotechniky a informatiky
Katedra informatiky

Zadání bakalářské práce

Student: **Jakub Hendrych**

Studijní program: B2647 Informační a komunikační technologie

Studijní obor: 2612R025 Informatika a výpočetní technika

Téma: 2D animace procesu měření
2D Animation Process Measurements

Zásady pro vypracování:

Cílem bakalářské práce je návrh a realizace modulu na animaci procesu měření v rámci systému FOTOM 2009.

1. Seznamte se s problematikou počítačového zpracování fotografií za účelem následného numerického zpracování.
2. Seznamte se s počítačové animace obecně a počítačové animaci vyřešené v rámci systému FOTOM 2008.
3. Navrhněte a realizujte modul na 2D animaci zájmových objektů na snímků a snímků.
4. Implementujte navržený modul v prostředí systému FOTOM 2009.
5. Proveďte zhodnocení dosažených výsledků.
6. Vypracujte uživatelskou a programátorskou dokumentaci.

Seznam doporučené odborné literatury:

[1] GONZALEZ, C. Rafael; WOODS, E. Richard. Digital Image Processing, 3rd Edition. 2008. 954 str. ISBN 978-0131687288
[2] Russ, C. John. The Image Processing Handbook, 5th Edition. 2007. 817 str. ISBN 0-8493-7254-2
[3] Ličev, Lačezar: Analýza, modelování, rozpoznávání a vizualizace procesu měření objektů na snímcích, 128 str., Knihy vydané prostřednictvím www.vydejteknihu.cz, Computer Press, a.s., ISBN 978-80-2513-296-8, EAN 9788025132968
[4] Y. Daniel Liang, Introduction to Java Programming, Comprehensive, Prentice Hall; 8 edition, 2010, ISBN-13: 978-0132130806
[5] Sojka, E.: Digitální zpracování obrazu, skripta FEI VŠB - TUO, 1999, ISBN-13: 978-0132130806

Formální náležitosti a rozsah bakalářské práce stanoví pokyny pro vypracování zveřejněné na webových stránkách fakulty.

Vedoucí bakalářské práce: **doc. Ing. Lačezar Ličev, CSc.**

Datum zadání: 18.11.2011

Datum odevzdání: 04.05.2012

doc. Dr. Ing. Eduard Sojka	prof. RNDr. Václav Snášel, CSc.
vedoucí katedry	*děkan fakulty*

Prohlašuji, že jsem tuto bakalářskou práci vypracoval samostatně. Uvedl jsem všechny literární prameny a publikace, ze kterých jsem čerpal.

Dne 2. května 2012 Podpis: .....Jakub Rudych.....

Rád bych poděkoval vedoucímu mé bakalářské práce, panu doc. Ing. Lačezaru Ličevovi, Csc., který mi poskytl cenné rady a připomínky v průběhu zpracování bakalářské práce.

Dále bych chtěl poděkovat mé rodině a přátelům, kteří mě podporovali v průběhu celého studia.

Abstrakt

V mé bakalářské práci, s názvem 2D animace procesu měření, se zabývám vytvářením nového modulu, který bude připojen do systému FOTOM^{NG}.

Modul bude sloužit ke 2D animacím sledovaných objektů, jako je například krční tepna na ultrazvukových snímcích. Tento modul bude implementován na platformě NetBeans v programovacím jazyce Java.

Dále se v této práci krátce zmiňuji o fotogrammetrii, jakožto měřicí technice, o samotném systému FOTOM^{NG} a řeším problematiku spojenou s tvorbou nového modulu.

Na základě zadání mé bakalářské práce, bude mým úkolem vytvořit i uživatelskou příručku a programátorskou dokumentaci pro tento nový modul.

Klíčová slova

fotogrammetrie, 2D animace, FOTOM^{NG}, modul, platforma NetBeans, zájmové objekty, měřický snímek, Java

Abstract

In my bachelor thesis, titled 2D animation process measurement, I deal with creating a new module that will be attached into the FOTOM^{NG} system.

The modul will serve to 2D animations of monitored objects, such as the carotid artery in ultrasound images. This module will be implemented on the NetBeans Platform in the Java programming language.

Next, in this work I writing about photogrammetry as a measurement technique, the FOTOM^{NG} system itself and issues associated with creating a new module.

I will create a user´s manual and programming documentation for the new module.

Keywords

photogrammetry, 2D animation, FOTOM^{NG}, module, NetBeans platform, object of interest, measurment images, Java

Seznam použitých symbolů a zkratek

2D	Two Dimensions – Dvourozměrný
3D	Threen Dimensions - Trojrozměrný
API	Application Programming Interface
CD	Compact Disc
GUI	Graphic User Interface
IDE	Integrated Development Environment
JAI	Java Advanced Imaging
JAR	Java Archive
JRE	Java Runtime Environment
JDK	Java Development Kit
MVC	Model, View, Controller
NG	Next Generation
NTSC	National Television System(s) Committee
PAL	Phase Alternativ Line
PDF	Portable Document Format
RCA	Rich Client Application
RGB	Red, Green, Blue
SECAM	Séquentiel couleur à mémoire - postoupení barevné informace do paměti
VŠB-TU	Vysoká škola Báňská – Technická univerzita
VVUÚ	Vědeckovýzkumný uhelný ústav
XML	Extensible Markup Language

Obsah

Seznam obrázků

1. Úvod

Odjakživa se člověk snaží zdokonalovat a vyvíjet v různých odvětvích svého života. Jedním z příkladů tohoto tvrzení jsou stále zvětšující se nároky na systém FOTOM. Na začátku, v roce 1997, zde byla myšlenka na vytvoření PC systému, který by zpracovával a analyzoval jednotlivé snímky důlních jam na základě vědního oboru důlní fotogrammetrie. Tak vznikl systém FOTOM 2000 pod vedením katedry informatiky VŠB-TU Ostrava. Postupem času byly přidávány nové možnosti a funkce tohoto systému, což vedlo ke vzniku dalších verzí. FOTOM měl být schopen pracovat nejen s důlními, ale také medicínské snímky. Tak spatřil světlo světa systém FOTOM 2008. Tato aplikace byla implementována v jazyce C++, ale postupem času se stala zastaralou. FOTOM 2008 se skládal z několika modulů, kdy každý plnil potřebnou funkci. Tyto moduly mohly být spuštěny každý samostatně. V roce 2008 bylo rozhodnuto, že je systém třeba aktualizovat. Proto byly shromážděny všechny výhody a poznatky při tvorbě předchozích verzí a byl vytvořen FOTOM 2009. Systém kladl velký důraz na modularitu, což mělo vést k jednoduchému rozšiřování tohoto programu o nové moduly a funkce, které by se spojily v jeden celistvý systém z názvem FOTOM^{NG}.

Cílem mého bakalářského projektu, je vytvoření nového modulu do systému FOTOM^{NG}, který je poslední a nejnovější verzí. Tento modul je znám z předchozí verze jako modul FOTOM4. Díky novému modulu, bude mít uživatel možnost přehrávat 2D animace pomocí snímků s měřeným objektem. Celou bakalářskou práci jsem rozdělil do několika částí, které budou reprezentovat postup při tvorbě nového modulu.

V první části se budu zabývat pojmem fotogrammetrie, jakožto vědním oborem pro měření objektů pomocí měřických snímků. V této kapitole také krátce nastíním historii vzniku tohoto pojmu. Cílem bude pochopení samotného pojmu a jeho účast v tomto bakalářském projektu. Pak volně navážu na teorii animací. Jelikož cílem tohoto projektu bude vytvoření modulu, který se zabývá 2D animacemi, je nezbytně nutné, tuto teorii rozebrat. Popíšu první historické zmínky o tomto pojmu. Vysvětlím také princip animací samotných a proberu aplikování animací na teorii fotogrammetrie. Ujasněním těchto dvou pojmů, získám teoretické znalosti, které budou základním stavebním kamenem pro samotnou tvorbu modulu.

Další část věnuji analýze stávajících systémů FOTOM^{NG} a jeho předchůdce. V tomto ohledu bude důležitá analýza modulu, který byl vytvořen u předcházející verze FOTOM 2008. Shromáždím všechny klady a zápory, díky kterým mohu lépe realizovat nový modul pro 2D animace. Proberu i stávající vývoj systému FOTOM^{NG}, včetně dosavadních integrovaných modulů a jejich funkcí.

Závěrečná část je z pohledu implementovaného modulu velice důležitá. Cílem bude celková specifikace požadavků, které jsou kladeny na nový modul, včetně prvotního návrhu GUI a use-case diagramu. Dále se chci věnovat softwarovým požadavkům a teorii samotné platformy NetBeans, na které je postaven celý systém FOTOM^{NG}. Díky těmto poznatkům, získám potřebné znalosti, které využiji při implementaci modulu a následné integraci do systému. Jelikož se jedná o velmi rozsáhlou problematiku, soustředím se jen na ty nejdůležitější faktory.

Výsledkem se stane samotná implementace modulu pro 2D animace integrovaného do systému FOTOM^{NG}. Mým cílem bude i prezentovat tento modul z pohledu implementace a uživatelského rozhraní, včetně celkového zhodnocení a ověření funkčnosti.

2. Fotogrammetrie

Abychom porozuměli celému projektu, je třeba se zaměřit na základy a na samotný pojem fotogrammetrie. Podstatou systému FOTOM je zpracovávání série měřičských snímků, které jsou pořizovány pomocí měřící techniky – fotogrammetrie. Takto pořízené snímky jsou nedílnou součástí tvořeného modulu pro 2D animace. V následujících odstavcích čerpám z literatury [1] a [2].

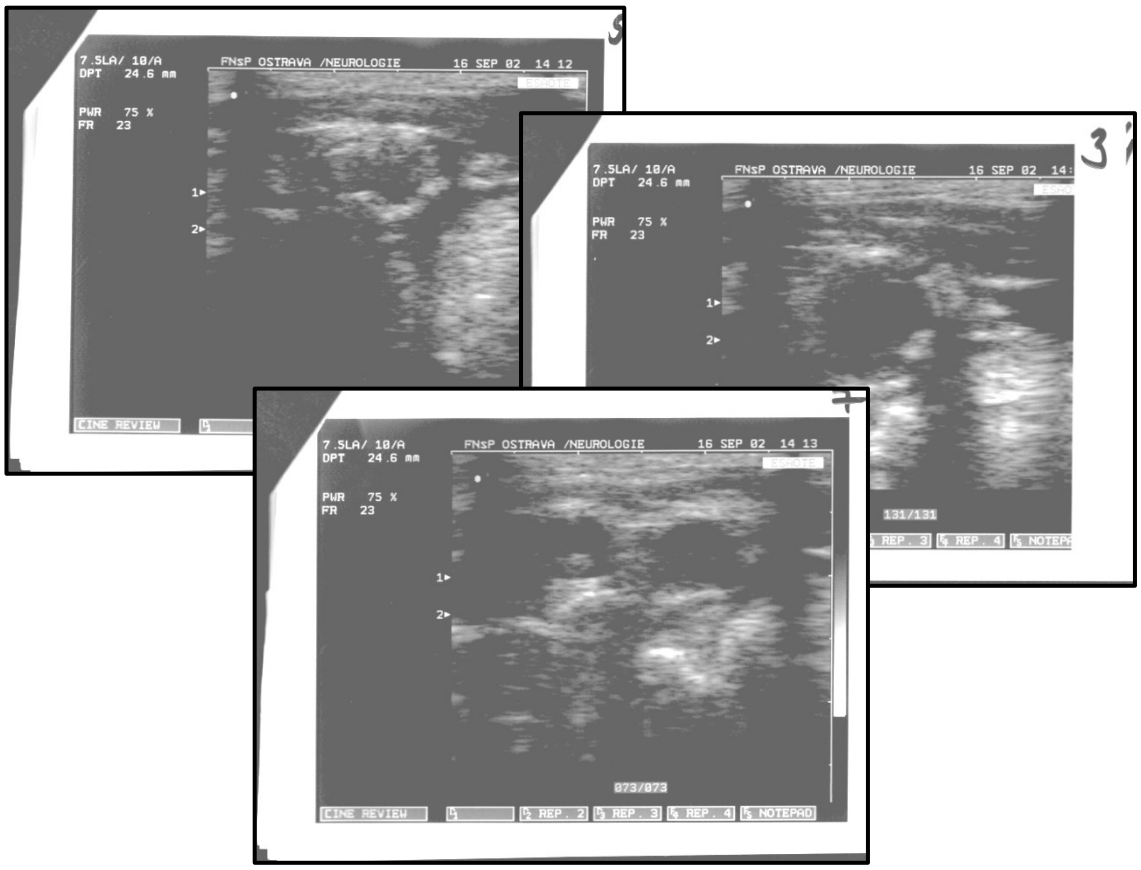
Slovo fotogrammetrie vzniklo složením tří řeckých slov: „photos“ = světlo, „gramma“ = nakresleno, zapsáno a „metron“ = měřit. Poprvé tento pojem použil Němec A. Meydenbauer při zaměřování stavebních objektů v roce 1867. Pro účely praktické fotogrammetrie byly fotografie poprvé využity ve Francii A. Laussedatem v roce 1861. V Čechách byla fotogrammetrii věnována pozornost již od roku 1862, kdy profesor pražské techniky doktor K. Kořistka provedl první fotogrammetrické měření. Při tomto měření určil polohu významných bodů na území tehdejší Prahy.

Fotogrammetrie nalezla své uplatnění v celé řadě odvětví vědy a techniky. Například v zeměměřictví při pořizování map a plánů, v lesnictví při plánování výsadby, pěstování a těžbě dřeva. Také v zemědělství, geologickém průzkumu, v archeologii, v architektuře, atd. Fotogrammetrie se také využívá v lékařství, při pořizování ultrazvukových snímků, ve vojenství při družicovém mapování nepřátelských objektů nebo i v meteorologii. S fotogrammetrií se můžeme setkat i v běžném životě. Například samotný lidský zrak, kdy rozlišujeme velikosti a tvary různých objektů, které momentálně sledujeme.

Ve fotogrammetrii, jako v měřicí technice, nezískáváme informace přímým měřením těchto objektů, avšak měřením jejich fotografických snímků. Základem je, že fotografický snímek je za určitých podmínek exaktním středovým průmětem fotografovaného předmětu. Tedy existují geometrické vztahy mezi předmětem a jeho fotografickým snímkem, které určíme numericky, graficky nebo mechanicky pomocí speciálních přístrojů. Takto pořízený snímek má velmi vysokou rozlišovací schopnost, která se při zvětšování rozměrů snímků stále zvyšuje. Velkou výhodou je krátká expoziční doba a možnost takto pořízené snímky uchovat a později celé měření opakovat.

Avšak tato věda nemusí sloužit jen k měření, ale i jako prostředek ke sledování průběhu a dokumentaci. Můžeme uvést i konkrétní případy, kdy byla využita fotogrammetrie. Například při měření VVUÚ Ostrava – Radvanice při diagnostice jam. Princip spočíval v postupném spouštění fotogrammetrické soupravy, která po expozici zaznamenávala profil důlní jámy. Za další příklad můžeme považovat i různé medicínské snímky tělních objektů. V těchto případech bude výsledkem série snímků, které jsou řazeny za sebou v určitém pořadí. U důlního díla se bude jednat o pořadí vyfotografovaných snímků při spouštění fotogrammetrické soupravy. U medicínských snímků, konkrétně u ultrazvukových, se bude jednat o postupné zmapování, například krční tepny, popřípadě jen jejího konkrétního úseku. Mimo jiné, takovou sérii medicínských snímků mám k dispozici při tvorbě mého modulu. Máme-li k dispozici takovou sérii, můžeme na sledovaných objektech pozorovat různé deformace. U krční tepny může lékař sledovat onemocnění koronární aterosklerózou, což je zužování krční tepny v důsledku usazování tukových tělísek. Takové onemocnění má za následek snížení průtoku krve do mozku a vede k velmi vážným zdravotním problémům.

Snímky série můžeme samozřejmě sledovat postupně. Avšak významným vylepšením je použití animací, kdy máme možnost tyto snímky promítat za sebou a vidět je v plynulé návaznosti.



Obrázek 1 – Úryvek ultrazvukové série krční tepny

3. Animace

S animacemi se setkáváme každý den a jsou velmi důležitou součástí našeho života. V následující kapitole se seznámíme s pojmem animace a přiblížíme si teorii, kterou tento pojem obnáší. Tato kapitola je i důležitá pro pochopení funkce tvořeného modulu pro FOTOM^{NG}, kde je právě využívaná 2D animace. V odstavcích čerpám z literatury [4].

Pojem Animace vzniknul z latinského názvu animātiō. Skládá se z „animo“ = oživit, nebo dát život a „ātiō“ = zákon o něčem. Tento pojem lze tedy přeložit jako „akt uvedení do života“.

Pokud mluvíme o zachycení pohybu, můžeme v naší historii najít mnoho zmínek, kde byly využity velmi jednoduché formy. Mezi takové příklady patří například paleolitické jeskynní malby, kde jsou znázorněna zvířata s více nohami v navrstvených pozicích. Tyto malby se jasně snaží zprostředkovat vnímání pohybu. Jako další příklad můžeme zařadit misku, která byla nalezena na území dnešního Iránu. Vědci její stáří datují na 3000 let př. n. l. Unikátnost této misky byla v tom, že po svém obvodu znázorňovala pět obrazců běžící kozy. Tyto obrazce můžeme nazývat snímky. Tento nález byl považován za příklad rané animace. Jelikož v těchto dobách neexistovaly přístroje k zobrazení maleb a obrazců, nemůžeme takovou sérii snímků považovat za animaci v pravém slova smyslu. Pokud budeme chtít jmenovat první vynálezy, které využívaly animací, bude to určitě čínský zoetrop. Ten byl vynalezen kolem roku 180 n. l. Zoetrop si můžeme představit jako rotující nádobu, která má na vnitřní straně dolní poloviny nakreslenou sérii obrázků, kde každý obrázek zaznamenává momentový pohyb objektu. V horní polovině jsou pak vyřezány otvory, kterými pozorovatel tyto objekty sleduje. Můžeme také uvést pozdější vynálezy, jako byly například phenakistoskop, nebo praxinoskop, který už využíval zrcadla a v pozdější verzi i obrazce promítal [5]. Princip založeny na zoetropu však zůstával stejný. V tehdejší společnosti byl také oblíbený klasický flipbook, který měl na každé stránce nakreslený momentový obrazec. Všechny tyto vynálezy vytvářely pohyb pomocí sekvenčních obrazců. Avšak skutečný rozvoj animace nastal s příchodem kinematografie.

Průkopníkem animací v kinematografii byl francouzský iluzionista a filmař Georges Méliès. Jako první vymyslel a použil (1896) techniku, kterou dnes nazýváme jako stop-motion animační techniku. Je třeba také zmínit amerického filmaře J. Stuarta Blacktona, který jako první spojil techniku stop-motion a ručně kreslené animace a dodnes je považován za prvního skutečného animátora.

V průběhu dějin se rozvíjely různé techniky animací. Mezi tyto techniky, můžeme zařadit například tradiční techniku animací. Pokud chceme vytvořit iluzi pohybu, potřebujeme sérii obrázků, na kterých se každý obrazec mírně liší od toho předcházejícího. Je nutno také zmínit i techniku stop-motion. Tato technika je založena na fyzické manipulaci skutečnými objekty. Tyto objekty jsou pokaždé vyfotografovány. Ve výsledku taková série fotografií vytvoří iluzi pohybu.

3.1 Princip animace

Z předchozí kapitoly jsme se dozvěděli, že animace jsou vlastně rychlá zobrazení sekvence obrázků (snímků), které tvoří iluzi pohybu. Výsledkem je tedy optický klam o pohybu. Takový klam vzniká v důsledku setrvačnosti lidského zraku.

Setrvačnost lidského zraku je fenomén, kdy stopa po obrazu zůstává ještě přibližně jednu čtvrtinu sekundy otisknutá na sítnici oka. Jinými slovy, pokud se zraku bude dostávat

kolem 24 snímku za jednu sekundu, bude se nám takový obraz, při této frekvenci, jevit jako spojitý. Avšak pokud uvidíme méně snímku za jednu sekundu, bude se nám takový obraz jevit jako blikající [6]. Počtu snímku za jednu sekundu obecně nazýváme pojmem framerate (*frame/s*).

Na tomto fenoménu je tedy založeno promítání filmů v kinech nebo sledování televize. U televizního přenosu se framerate rozlišuje podle aplikovaných norem (PAL, SECAM, NTSC, atd.). Nutné je také zmínit i počítačové monitory, které rovněž potřebují určitý framerate, aby se obraz jevil jako spojitý.

3.2 Počítačové animace

S příchodem informačních technologií přišel i velký rozvoj animací, které dnes volně nazýváme jako počítačové animace. Při počítačových animacích se setkáváme s celou řadou animačních technik, které však mají jedno společné a to, že jsou vytvářeny digitálně na počítači.

Počítačové animace můžeme rozdělit podle toho, jak jednotlivé algoritmy řeší pohyb objektu a to na nízko úrovňovou a vysoko úrovňovou animaci. U nízko úrovňové animace se řeší většinou pohyb hmotného bodu po křivce; jak padají kapky vody z mraku na zem, jaký tvar má oheň, atd. U vysoko úrovňové animace jsou dílčí úlohy chápány jako bloky, které se nemusejí řešit. A z těchto bloků se skládá komplikovanější pohyb. Příkladem mohou být pohyby člověka, který vykonává nějakou činnost. Výpočty dynamiky a jiných fyzikálních jevů necháváme nízko úrovňové animaci. V důsledku tohoto tvrzení můžeme říct, že se nejedná o dvě odlišné sféry, ale vysoko úrovňová animace je přímo založena na nízko úrovňové animaci [2].

3.2.1 2D animace

2D animace jsou v počítači tvořeny pomocí 2D bitmapové, nebo 2D vektorové grafiky. Bitmapová grafika je založena na matici bodů (pixelů) celé bitmapové struktury. Takovou strukturu si můžeme představit jako čtvercovou mřížku, kde každý bod je vyjádřen hodnotou základních barev RGB. Vektorová grafika využívá jednoduchých geometrických tvarů, například body, křivky, přímky a mnohoúhelníky. Výhodou oproti bitmapové grafice nejsou žádné ztráty kvality pří zmenšování nebo zvětšování obrázku [4].

U 2D animací většinou hovoříme o dvourozměrných modelech, jako jsou například obrázky. Pokud mluvíme o dvourozměrných objektech, pracujeme jen v osách x a y. Při práci ve 2D máme několik možností, jak s daným objektem pracovat. Mezi tyto možnosti patří například změna velikosti (scaling), rotace (rotation) a transformace (transformation) [7].

S 2D animacemi se dnes běžně setkáme při práci s počítačem. Mezi ty nejznámější patří Flash animace nebo všem známe GIF obrázky.

3.2.2 3D animace

Moderní 3D animace využívají 3D počítačové grafiky, která je příbuzná vektorové 2D grafice. Také pracuje se souřadnicemi bodů a informacemi o úsečkách, křivkách a plochách. Avšak tato data jsou na rozdíl od 2D grafiky uloženy v trojrozměrném souřadnicovém systému.

3D grafika je dnes bohatě používaná například v počítačových hrách, filmech, různých simulačních softwarech, výrobních i navrhovacích softwarech, atd. [4].

3.3 Animace ve fotogrammetrii

Hlavním úkolem animací ve fotogrammetrii je prezentovat naměřené údaje úplně jinak, než jsou tyto výsledky měření zobrazovaných grafů. Tím získáváme dva nezávislé úhly pohledu, ze kterých můžeme vycházet. V následujících podkapitolách jsou uvedeny druhy animací, se kterými se můžeme setkat ve fotogrammetrii. V těchto podkapitolách čerpám z literatury [2].

3.3.1 Animace snímků

U animace snímků zobrazujeme sérii měřičských snímků, které nemusí být transformovány. I když transformace snímku nebude provedena, stále si můžeme udělat představu o kvalitě jednotlivých snímků série. Můžeme tedy zjistit kvalitu jejich naskenování nebo kvalitu expozice. Výsledkem takové série bude ucelený přehled o kvalitě celé série snímků.

V dalším případě budeme počítat s transformacemi. To znamená, že každý měřický snímek je podroben rotaci, změně měřítka nebo posunu. Výsledkem budou snímky, které na sebe navzájem správně navazují a při aplikaci animace získáme vjem pohybu sledovaným objektem (například krční tepna).

3.3.2 Animace objektů

S animací objektů jsme se setkali například u systému FOTOM 2008 v modulu FOTOM4 a budou také součástí navrhovaného modulu pro systém FOTOM^{NG}.

Jedná se o znázornění různých zájmových objektů na určitém pozadí. Takovým pozadím většinou myslíme samotný měřický snímek. Výsledkem je pak animace, díky které můžeme sledovat geometrické vlastnosti zájmových objektů, které nám prozradí například různé deformace nebo jak bylo zmíněno v kapitole 2., onemocnění krční tepny.

Tyto animace snímků spolu se zájmovými objekty mají velký význam ve spojitosti s výpočtem transformací, které jsou prováděny kvůli správné vzájemné orientaci zobrazovaných objektů. Jedná se zde o transformace lokálních lícovacích bodů na pozici lokálních lícovacích bodů ve výchozím profilu. Výchozím neboli referenčním profilem, nazýváme většinou první snímek série. Tyto transformace použijeme na parametry zájmových objektů, které pak zobrazujeme na pozadí.

4. Systém FOTOM^{NG}

Vývoj systému FOTOM^{NG} začal v roce 2008 jako odpověď na zvyšující se požadavky předešlých verzí. FOTOM^{NG} kladl velký důraz na modularitu, aby bylo možno tuto aplikaci neustále rozšiřovat o nové funkce. Z těchto důvodů bylo rozhodnuto, že systém bude implementován na platformě NetBeans [3], o které pojednávám v kapitole 3.1. Výsledkem se stává systém, který odpovídá trendům moderních počítačových aplikací. FOTOM^{NG} vznikl pod vedením pana doc. Ing. Lačezara Ličeva, Csc. Nutno je také zmínit pány Ing. Lukáše Krahulce a Ing. Jana Krále, kteří pro tento systém vytvořili jádro a prototyp GUI.

Systém FOTOM^{NG} je stále ve vývoji. O jeho současném stavu můžeme říci, že je ve velmi pokročilém stádiu. Je hotova většina funkcí, které známe z předešlé verze ze systému FOTOM 2008. Posledním hotovým modulem je *Modelování* [1], které implementoval Tomáš Pytlík. Všechny stávající moduly a funkce popíšu v kapitole 4.2. Systému chybí modul pro znázornění 3D animace. Mým úkolem je tvorba modulu pro 2D animace. Tento modul byl znám v systému FOTOM 2008 jako modul FOTOM4.

4.1 Analýza předchůdce

V této kapitole se seznámíme s předchůdcem stávajícího systému FOTOM^{NG}. Systém se jmenoval FOTOM 2008. Avšak hlavní náplní této kapitoly bude analýza modulu, který se zabýval, stejně jako můj modul, 2D animacemi.

Celý systém byl implementován v jazyce C++. FOTOM 2008 se skládal z několika navzájem nezávislých modulů. Jak jsem již nastínil v úvodu, každý modul pracoval samostatně (mohl být spuštěn samostatně) a plnil různé funkce. Důležitým modulem byl pak FOTOM1, který plnil funkci rozcestníku. Z tohoto místa měl uživatel možnost se dostat volně do ostatních modulů a to pomocí tlačítek, které tyto moduly spouštěly. Další důležitou funkcí tohoto modulu byla definice zájmových objektů na snímcích. Mezi další moduly patřily například FOTOM2, který sloužil pro 2D modelování, FOTOM3 zobrazoval 3D znázornění měřeného objektu.

4.1.1 FOTOM4

Tím se dostáváme k modulu FOTOM4. Modul plnil všechny funkce, které byly nezbytně nutné pro 2D animace, jako byl například výběr souborů, které reprezentovaly sérii měřických snímků pro animace. Dále to byly klasické prvky pro manipulaci s animací, atd.

Při prvotním spuštění tohoto modulu, bylo po nás dožadováno vybrání *.ftm* souborů, které obsahovaly požadovaný snímek s definovanými zájmovými objekty. Na toto právě doplatila samostatnost všech modulů, jelikož neexistovalo společné API a jádro, nemohly si moduly mezi sebou přímo předávat požadovaná data. Proto byl uživatel nucen tyto soubory vyhledat, označit a pak se teprve spustilo okno pro 2D animace. Právě tato nevýhoda byla odstraněna v systému FOTOM^{NG}, který popisuji v kapitole 4.2.

Hlavní okno modulu FOTOM4 se skládalo z klasického vertikálního filmového pásu, který obsahoval jednotlivé snímky série. Po kliknutí se pak tento snímek zobrazil v plné velikosti napravo od filmového pásu. Uživatel mohl tento maximalizovaný snímek zvětšovat i zmenšovat podle potřeby.

Na hlavním panelu nástrojů jsme také nalezli funkce pro přehrávání, výběry objektů, souborů i tisk maximalizovaného náhledu. Avšak pokud jsme se chtěli dostat do rozšířenějších funkcí, museli jsme vyhledat požadovanou položku v menu, která spustila nové okno, kde se nacházely všechny ostatní potřebné funkce pro 2D animace. Mezi tyto funkce patřila například volba uživatelsky definované prodlevy mezi snímky, volba zpětné animace a jiné. Bohužel tyto funkce postrádaly uživatelskou nápovědu formou klasického *toolTipText*. Jinými slovy, pokud zhodnotíme celkové umístění nástrojů, mohly být všechny nástroje umístěny na *toolBar* v hlavním okně modulu FOTOM4.

Veškeré výše zmíněné nedostatky bude navrhovaný modul odstraňovat, aby práci s 2D animacemi uživateli maximálně usnadňoval. Je ovšem nutné podotknout, že všechny implementované funkce pracovaly správně.

Obrázek 2 – Okna modulu FOTOM4

4.2 FOTOM^{NG} – moduly a funkce

Při prvotním spuštění systému FOTOM^{NG} bude okno celé aplikace defaultně rozděleno do pěti částí. Tyto části můžeme kdykoliv spravovat přes položku *Window* v hlavním menu aplikace. Okna můžeme kdykoliv přesouvat na jiné pozice.

V levém horním rohu je umístěno okno *Prohlížeče* (Explorer). Zde má uživatel možnost vytvářet nové adresáře. Do těchto adresářů pak vkládá jednotlivé snímky, se kterými bude v aplikaci pracovat. V levém dolním rohu, pod oknem *Prohlížeče*, se nachází okno *Manažer objektů*. Uprostřed hlavního okna aplikace najdeme *Editor* pro prácí s jednotlivými snímky.

Každý snímek je otvírán jako nové okno *Editoru* formou záložek. *Editor* poskytuje uživateli jednoduché funkce, jako například tlačítka pro přiblížení nebo oddálení. V pravém horním okně je situována *Paleta nástrojů* (Pallete) s různými nástroji pro práci se snímkem. Pod *Paletou* jsou umístěny *Vlastnosti* (Properties), které slouží pro definování různých možností. Některé z těchto modulů a další funkce FOTOM^{NG} jsou podrobněji popsány v následujících podkapitolách, kde čerpám z literatury [2].

4.2.1 Manažer objektů

Tento modul vzniknul pro potřeby spravovat velké množství objektů na snímku. Pokud máme na snímku definováno několik desítek zájmových objektů, stává se pak práce méně přehlednou. Tento manažer zobrazuje seznam definovaných objektů a poskytuje další způsob jak vybrat objekty, se kterými chce uživatel pracovat. *Manažer objektů* samozřejmě poskytuje i více násobné označení pomocí stisku kláves Shift nebo Ctrl. Při takovém to více násobném označení, poskytuje tento modul znázornění spojnice mezi těmito objekty. Spojnice obsahují textové informace o vzdálenosti v obrazových bodech (v pixelech).

4.2.2 Paleta nástrojů

Toto okno nabízí panel nástrojů, ve kterém jsou zobrazeny dostupné nástroje pro měření, definici objektů a skicování. Při implementaci nového rozhraní, můžeme kdykoli přidat další modul, který nám poskytne nový nástroj. Panel nástrojů tento nový nástroj vyhledá a zařadí jej do zvolené kategorie, kde k němu bude mít uživatel přístup. Platforma NetBeans také poskytuje vytvoření nástroje, který bude dostupný z hlavního menu aplikace. Způsoby, jak používat různé nástroje, jsou popsány v příslušných uživatelských příručkách.

Panel nástrojů – Měření

- Lícovací body – modul lícovací body se implementuje do nástroje pro *Měření*. Poskytuje uživateli nástroj pro definici referenčních bodů na zpracovávaném snímku. Podle toho je pak možné přesně určit pozici snímku v prostoru. Tento nástroj je velmi důležitý při práci ze sérií snímků, které jsou na sobě závislé. Např. série snímků ultrazvuku krční tepny. Důležitou funkcí tohoto nástroje je i definice měřítka mezi skutečnými souřadnicemi (mm, cm, m) a obrazovými jednotkami (pixely). Lícovací body jsou definovány dvěma body, měrnou jednotkou a zadáním reálných souřadnic.

Panel nástrojů – Definice objektů

Nástroje implementované v tomto modulu slouží pro *Definici objektů* na snímku. Mezi tyto nástroje patří:

- Bod – slouží k jednoduchému definování bodu zájmu. Bod jako zájmový objekt patří mezi nejjednodušší objekt. Sledovaným parametrem je souřadnicová poloha bodu na snímku.
- Kružnice – slouží pro měření odchylek a vzdáleností na snímcích kruhového tvaru (důlní šachty, tepny, aj.). Kružnice se zadává definováním dvou parametrů a to souřadnicemi středu a poloměrem kružnice. U kružnice je také sledována plocha tohoto objektu, jak v obrazových jednotkách, tak i v měrných jednotkách (musí být zadány lícovací body).

- Průsečík - stejně jako nástroj bod, slouží k definování jediného bodu. Avšak tento nástroj definuje bod jiným způsobem. Jsou vytvořeny dvě přímky pomocí čtyř bodů. Průsečíkem těchto přímek je námi sledovaný bod.
- Mřížka – nástroj sloužící pro ultrazvukové snímky. Mřížka je tvořená soustřednými kružnicemi a paprsky, vycházejícími z definovaného bodu.
- Matice – nástroj vytvoří soustavu horizontálních a vertikálních přímek na snímku.
- Polygon – nejpoužívanější nástroj. Pomocí polygonu definujeme oblasti, které nejsou pravidelné, a nelze použít například kružnici. Oblast definujeme pomocí bodů. Polygon je uzavřená hranice spojením n bodů a n-1 hranami. Sledovanými parametry je těžiště, obsah polygonu v obrazových i měrných jednotkách (musí být zadány lícovací body) a počet hran.

Panel nástrojů – Skicování

Nástroje implementované v tomto modulu poskytují uživateli prostředek k úpravám nedostatků snímku. To většinou probíhá dokreslením nebo zvýrazněním chybějících hran. Jedná se například o tyto nástroje:

- Úsečka, Lomená – tyto nástroje nám umožňují kreslit do snímku různé úsečky a lomené čáry o vlastní barvě a tloušťce.
- Kvadratická, Kubika, Bézirova křivka - tento nástroj je užitečný pro dokreslování chybějících částí. Jelikož sledované objekty většinou nemají pravoúhlý tvar, použijeme tyto křivky. Jednotlivé křivky jsou definovány počtem bodů.
- Plechovka – nástroj, který nám vyplní oblast námi definovanou barvou.
- Hranice – nástroj, který zvýrazní hranice oblasti námi definovanou barvou.

4.2.3 Properties

Toto okno nám poskytuje různé možnosti při práci se snímky. Jedná se o obdobu *Vlastností*, kterou známe například z NetBeans IDE. Nalezneme zde jednotlivé sledované parametry zájmových objektů až po uživatelsky nastavitelné barvy (velice rozsáhlá paleta barev, možnost výběru mezi RGB a CMYK odstíny nebo přednastavenými barvami).

4.2.4 Filtry

Položka *Filtry* nám poskytuje různé metody pro úpravu obrazu (snímku). Mezi tyto metody patří například *Prahování*. Jedná se o velmi jednoduchou metodu, která je založena na rozlišení jasových bodů (pixelů) od pozadí. Pak je nutné najít jasový práh, který by oddělil objekty od pozadí. Výsledkem je pak obraz v binárním tvaru, kde hodnotu 1 mají ty body, které náleží objektu. Hodnota 0 pak patří pozadí. Avšak nalezneme zde i podobné metody. Všechny většinou slouží k efektivnímu nalezení hran pro definování zájmových objektů.

Jako další metody můžeme jmenovat *Rozmazávání*, pak metody, které pracují s barvami obrazu *Převod na stupně šedi*, *Inverze barev*, nebo metodu pro *Jas a kontrast*.

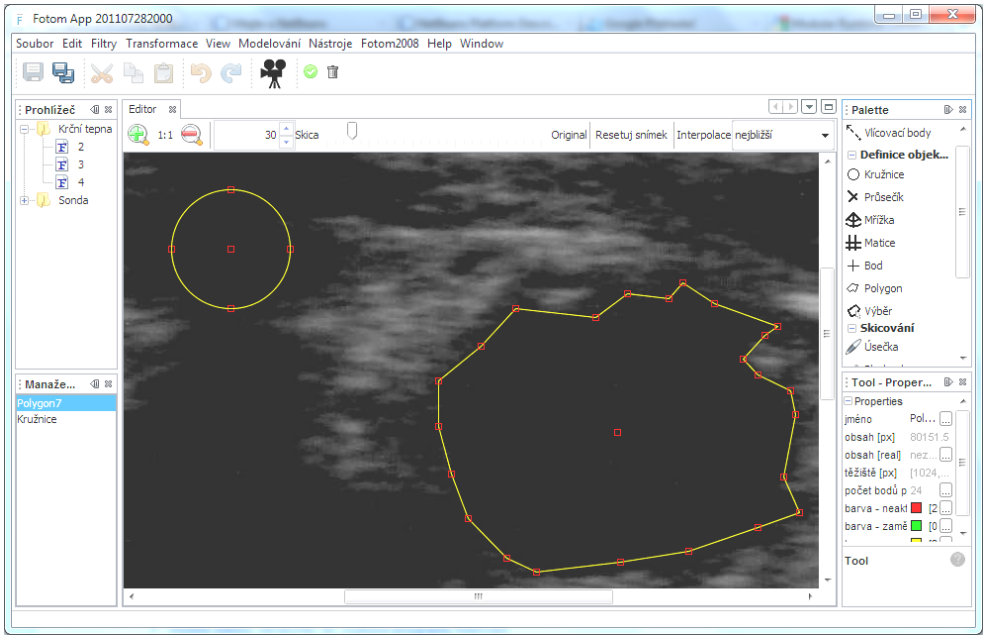
4.2.5 Transformace

Položka *Transformace* nám umožňuje fyzicky upravovat sledované snímky. Mezi nástroji nalezneme *Oříznutí obrazu*, jak název napovídá, bude se jednat o odstranění nepotřebných ploch na snímku. Dalším nástrojem je *Odstranění distorze čočky*, který slouží

svou funkcí k odstranění zkreslení obrazu, které je způsobeno optickými vadami čoček. Posledním nástrojem je *Rotace*, která umožňuje otočení o daný úhel.

4.2.6 Modelování

Nejnovější modulem, který byl v systému FOTOM 2008 znám jako FOTOM2 je *Modelování*. Modul pracuje ze sérií snímků. Výstupem jsou pak, podle zvoleného výpočtu, grafy, které nám znázorňují různé závislosti mezi sledovanými objekty. Položka nám poskytuje nástroje jako *Vzdálenosti mezi objekty*, *Odchylky parametrů a vzdáleností*, *Porovnávání dvou měření* nebo poskytuje i *2D model objektu*.



Obrázek 3 – Hlavní okno systému FOTOM^{NG}

5. Návrh modulu

V této kapitole zmiňuji obecnou vizi pro implementovaný modul. Budu také specifikovat požadavky, které jsou od tvořeného modulu vyžadovány. Z uvedených požadavků vytvořím use-case diagram, který bude tyto požadavky reprezentovat a poslouží mi jako dobré vodítko při implementaci modulu.

V dalších podkapitolách se zaměřím na softwarové prostředky. Cílem bude prvotní návrh uživatelského rozhraní a volba platformy NetBeans.

5.1 Vize

Navrhovaný modul bude poskytovat uživateli systému FOTOM^{NG} všechny funkce, které byly implementovány v systému FOTOM 2008 modul FOTOM4. Tyto funkce mohou být rozšířeny, ale hlavním úkolem bude předejít chybám a nedostatkům, které byly v systému FOTOM 2008 implementovány. Veškeré tyto funkce a požadavky na ně, jsou popsány v kapitole 5.2. Dalším důležitým faktorem je i uživatelský příjemné prostředí.

Modul pro 2D animace bude integrován do systému FOTOM^{NG}. Tudíž modul musí být schopen komunikovat s ostatními moduly, tím je myšleno předávání dat. V případě modulu pro 2D animace se bude jednat například o předávání série snímku, ze kterých bude vytvořena 2D animace.

Uživatelské rozhraní musí být implementováno tak, aby uživatel bez problému věděl, co přesně obnáší daná funkce 2D animací. Z toho vyplývá podpora češtiny v oblasti *toolTip* nápovědy. Vstupní uživatelské parametry musí být zabezpečeny vůči nesprávnému zadávání parametrů. Spouštění modulu musí být podmíněno správným počtem snímků, popřípadě jinými podmínkami, které jsou nutné pro správnou funkčnost tohoto modulu. Všechny funkce uživatelského rozhraní musí být implementovány tak, aby nijak neovlivnily průběh a funkcionalitu jiných.

Nutností je také vytvoření programátorské dokumentace všech implementovaných tříd a uživatelské příručky, která bude obsahovat podrobný popis jednotlivých funkcí modulu 2D animace.

5.2 Specifikace obecných požadavků

Úkolem této podkapitoly je seznámení se s obecnými požadavky na tvorbu tohoto modulu. Tyto požadavky jsou základním stavebním prvkem, který využiji při samotné implementaci.

Díky této specifikaci, mohu tyto požadavky i graficky znázornit pomocí use-case diagramu (diagram použití). Na základě tohoto diagramu, mohu také určit předpoklady a výsledky užití těchto funkcí (víc kapitola 5.3).

5.2.1 Funkcionalita animací

Všechny funkce obsažené v této podkapitole manipulují s animacemi. Všechny jsou umístěny na panelu nástrojů okna 2D animace (*toolBar*). Mezi tyto nástroje patří:

Výběr mezi animacemi snímků a animacemi objektů

Podrobný popis teorie těchto funkcí je uveden v kapitole 3.3.1 a 3.3.2. Volba mezi těmito funkcemi musí být jednostranná. Jinými slovy, v konkrétním okamžiku může být zvolena jen jedna z těchto možností. Po zvolení animace objektů se musí uživateli zpřístupnit okno pro volbu zájmových objektů (víc další odstavec).

Volba objektů, okno pro volbu objektů

V panelu nástrojů modulu bude také umístěno tlačítko, které zpřístupní okno pro volbu zájmových objektů.

V tomto okně se vypíší všechny zájmové objekty, které se vyskytují na snímcích celé série. Uživatel má možnost volby mezí jedním nebo více objekty a to pomocí klávesových zkratek Shift nebo Ctrl. V případě, že uživatel nechce zvolit žádný objekt, bude mu k dispozici tlačítko pro opuštění tohoto okna.

Přehrávání automatických animací

Uživatel bude mít k dispozici funkce pro manipulaci přehrávání animací. Bude se jednat o klasická tlačítka Play a Stop. Tlačítko Play musí být schopno kdykoliv spustit animaci a tlačítko Stop kdykoliv tuto animaci zastavit na aktuálním snímku. Přehrávání musí fungovat pro animaci snímků i animaci objektů. Po dosažení koncového snímku se musí animace automaticky vypnout.

Ruční animace

Jednou s možnosti animací bude také ruční animace. Tato volba bude uživateli sloužit pro rychlé procházení mezi snímky. Funkce bude podporovat klikání myší, ale především ovládání pomocí klávesnice. Na klávesnici budou vyhrazeny dvě klávesy pro prohlížení snímků nahoru a dolů. Informace o těchto klávesách musí být uživateli sděleny skrz *toolTip* nápovědu. Ruční animace musí fungovat pro animace snímků i animace objektů. Procházení pomocí ruční animace musí být omezeno prvním a posledním snímkem série.

Prodleva mezi snímky

V panelu nástrojů bude situována volba uživatelsky definované hodnoty prodlevy mezi jednotlivými snímky série v milisekundách. Tato volba musí mít defaultně nastavenou hodnotu od 500 – 1000 ms. Tato prodleva musí být aplikována pro oba druhy animací. Jak snímkovou animací, tak i animaci objektů.

Intervalová animace

Bude se jednat o novinku, která nebyla implementována v systému FOTOM 2008. Jedná se o funkci, která uživateli umožní provádět animaci snímků i objektů v určitém intervalu. Uživatel bude zadávat pouze číslo snímku prvního intervalu a číslo snímku posledního intervalu. Tato funkce musí být podmíněna určitými pravidly, aby byla za všech okolností správná. Hodnota prvního intervalu musí být menší než hodnota druhého intervalu, hodnoty musí být číselné, intervaly nesmí zasahovat mimo počet snímků.

Animace pozpátku

Při povolení této funkce se budou veškeré animace přehrávat pozpátku. Musí fungovat pro animace snímků a objektů i pro intervalové animace. V případě dosažení prvního snímku se musí animace automaticky vypnout.

Započítávání transformací

Teorie této funkce je uvedena v kapitole 3.3.1 a 3.3.2. Uživatel bude mít možnost volby, zda chce při animacích snímků a animacích objektů započítávat transformace.

Typ náhledu snímku

Po aktivaci animací objektů se zpřístupní nabídka pro volbu náhledu snímku. Uživatel si bude moci volit mezi třemi možnostmi. Bude se jednat o zobrazování aktuálních snímků při animaci. Další možností bude volba zobrazení referenčního snímku (víc kapitola 3.3.2). Poslední možností bude zobrazení pouze zájmových objektů. Tato možnost bude novinkou v navrhovaném modulu. Defaultní nastavení bude aktuální snímek.

Počet znázorňovaných objektů předešlých snímků

Po aktivaci animací objektů se zpřístupní tato nabídka. Volbou si uživatel bude moci měnit mezi znázorňováním zájmových objektů jen aktuálního snímku nebo aktuálního a předcházejícího snímku nebo aktuálního a dvou předcházejících snímků. Tuto volbu bude možno aplikovat i pro možnosti náhledu snímku v předešlém odstavci. Defaultně bude tato volba nastavena na zobrazování zájmového objektu aktuálního snímku.

5.2.2 Funkcionalita filmového pásu a náhledu snímku

Do této podkapitoly zařadíme všechny požadavky týkající se grafického znázornění samotného filmového pásu a hlavního maximalizovaného náhledu snímků.

Filmový pás

V okně pro 2D animace bude v levé části situován vertikální filmový pás, který bude obsahovat všechny zmenšené snímky celé měřické série, která byla požadována jako vstup do tohoto modulu. Každý snímek ve filmovém pásu bude opatřen nadpisem, který bude obsahovat pořadové číslo a globální název měřického snímku. Názvy těchto snímků nalezneme i v *Prohlížeči* (víc kapitola 4.2).

V reakci na uživatelem vyvolanou akci se daný snímek označí. Takový snímek bude indikovat jako právě zvolený a znázorní se do maximalizovaného náhledu snímku (víc další odstavec). Označení snímků bude probíhat ve dvou barvách, jedna pro klasické prohlížení a druhá, která bude indikovat spuštění animace a její následné procházení snímků.

Náhled snímku

Tato část bude situována napravo od filmového pásu. Bude sloužit pro maximalizovaný náhled měřického snímku. Na takovém to snímku se budou znázorňovat zájmové objekty, které

budou rozšířeny o funkcionalitu z předešlé kapitoly 5.2.1. V implementaci bude defaultně nastaveno zobrazení prvního snímku v maximalizovaném náhledu pro spuštění modulu 2D animace.

5.2.3 Ostatní funkcionalita

V této podkapitole je uvedena ostatní funkcionalita požadavků na modul 2D animace, která svým způsobem rozšiřuje funkcionalitu z kapitol 5.2.1 a 5.2.2.

Aktivace položky 2D animace, přidání série snímků

Pokud uživatel bude chtít spustit modul pro 2D animace, musí mít označeny minimálně dva snímky. Položka *2D animace* se bude nacházet v hlavním menu v položce *Modelování* na konci seznamu. Až po označení měřické série pomocí tlačítek Shift nebo Ctrl se zpřístupní položka *2D animace* a dojde k přidání měřické série a následnému spuštění modulu.

Možnost tisku

V hlavním okně modulu 2D animace bude mít uživatel možnost tisku. V předchozí verzi v modulu FOTOM4 je možnost tisku omezena pouze na tisk maximalizovaného náhledu snímku. V navrhovaném modulu bude tato možnost rozšířena i o tisk filmového pásu. V panelu nástrojů si uživatel bude moci vybrat právě mezi tiskem těchto dvou komponent. Bude zde samozřejmě situována ikona pro spuštění klasického dialogového okna pro tisk.

Manipulace obrazu

V panelu nástrojů bude uživateli k dispozici také funkce pro oddálení a přiblížení. Jedná se o klasický zoom-in a zoom-out. Tyto funkce budou doplněny o tlačítko, které vrátí přiblížený nebo oddálený snímek do původní polohy. Novinkou vůči modulu FOTOM4 pak bude oddalování a přibližování pomocí kolečka myši.

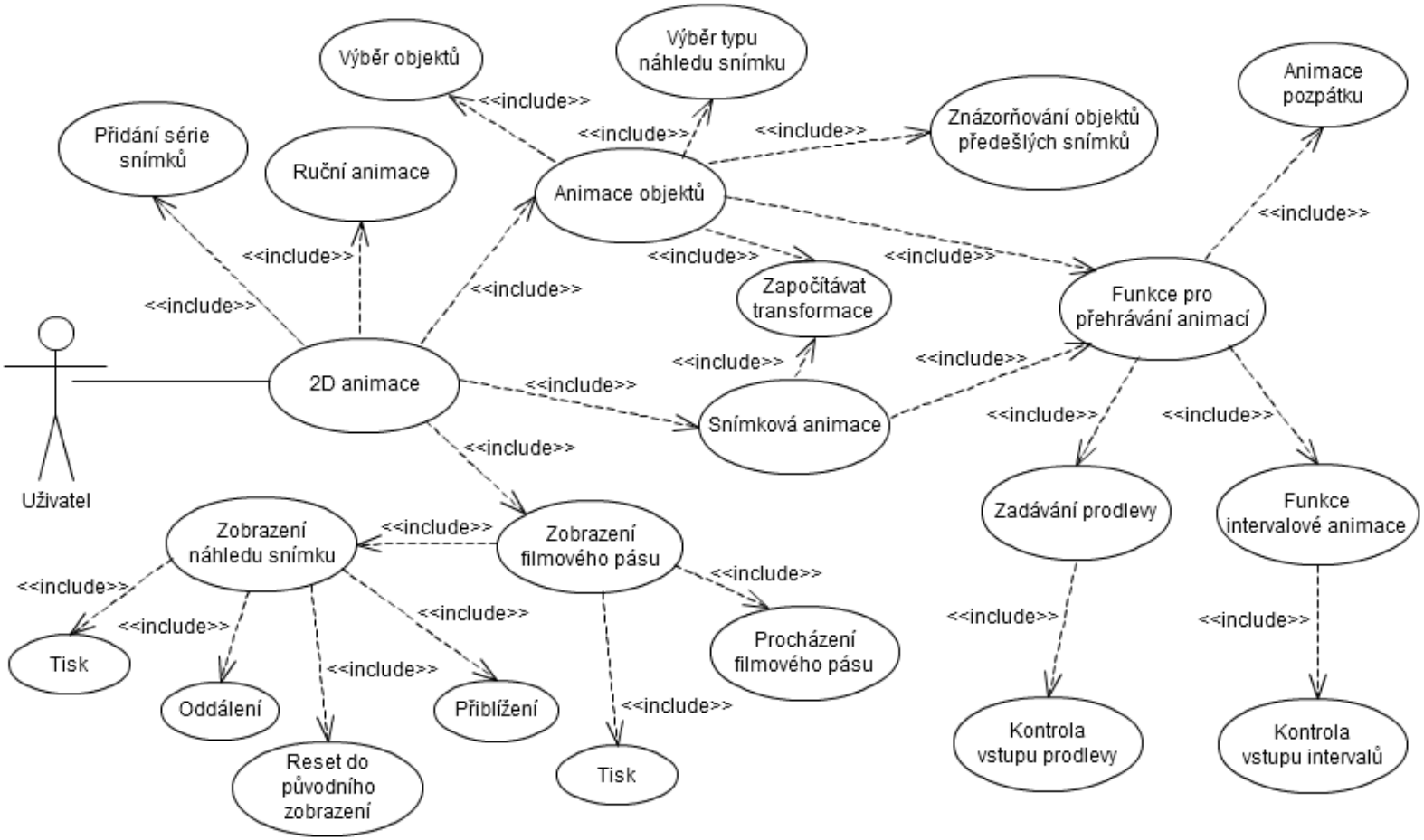
Dialogová okna, podpora české nápovědy

Podle vize bude v modulu implementována podpora všech funkcí v češtině pomocí klasické nápovědy *toolTip*. Nápověda by měla být u všech použitých komponent, aby zcela informovala uživatele o své dané funkci.

U všech komponent, které vyžadují vstup uživatele, budou implementována opatření proti špatnému vstupu. Konkrétně se bude jednat o zadávání prodlevy a zadávání intervalů u intervalové animace. Vstup těchto polí bude omezen jen na číslice.

5.3 Analýza požadavků

Na základě specifikovaných požadavků, které jsem uvedl v kapitole 5.2, mohu nyní sestavit use-case diagram. Tento diagram specifikuje práci účastníka, v našem případě, se bude jednat o uživatele, který bude pracovat s tvořeným modulem 2D animací. Diagram, který je uveden na obrázku 4., dále vykresluje jednotlivé asociace mezi funkcemi.



Obrázek 4 – Use-case diagram modulu 2D animace

Z tohoto diagramu vyplývá, že veškerá práce s ním je shrnuta v use-case bloku 2D animace. Odtud se dostáváme k ostatním blokům use-case diagramu. Mezi tyto případy užití patří například přidání série snímků, zobrazení filmového pásu, který dále zahrnuje práci s náhledem snímku, jeho manipulací a tiskem. Dále je také nutné zmínit bloky pro snímkovou a objektovou animaci, které dále zahrnují funkce pro přehrávání a další rozšiřující funkcionalitu, jako jsou například zpětné animace nebo intervalové animace.

5.4 Návrh GUI

V tomto prvotním návrhu uživatelského rozhraní, budeme vycházet z některých dekorací, se kterými jsme se setkali u modulu FOTOM4. Samozřejmě, jelikož se jedná o odlišné programovací jazyky, bude samotná implementace komponent vypadat úplně jinak.

V první fázi bude nutné přidat novou položku do menu. Podle specifikace požadavků, musí být tato položka s názvem *2D animace* přidána do sekce *Modelování* v hlavním menu systému. Před tuto položku se umístí separátor, kvůli oddělení od funkcí, které souvisejí s jiným modulem. Způsob přidávání bude podrobněji vyřešen v kapitolách 5.5.2 nebo 6.1.2. Po stisku této položky se spustí tvořený modul.

Celé GUI modulu bude situováno jako editační okno platformy NetBeans (víc kapitola 5.5). Z uvedených požadavků a ze zkušeností z modulu FOTOM4 a jiných modulů ze systému FOTOM^{NG}, jsem se rozhodl, že okno bude tvořeno třemi částmi. Jelikož jazyk Java podporuje velkou škálu nákresových manažerů, které definují rozložení komponent v GUI, ihned jsem se rozhodl pro *BorderLayout*. Tento nákres mi ve všech směrech podpoří navrhované GUI. V horní části (v nákresu známá jako PAGE_START) umístím horizontálně panel všech dostupných nástrojů (*toolBar*), které budou v tomto modulu využívány. Jelikož se jedná o větší počet funkcí, použiji pro systematické oddělení těchto nástrojů vertikální separátory, které by měly zvýšit přehlednost celého panelu nástrojů. V centrální části nákresu (CENTER), umístím samotný filmový pás s maximalizovaným náhledem jednotlivých snímků. K tomu bude ideální využit komponentu *jSplitPane*, která poskytuje rozdělení okna na dvě části, pomocí posuvného separátoru. Do levé části pak umístním vertikální filmový pás a do pravé části budou zobrazovány maximalizované náhledy právě označených snímků filmového pásu.

Pokud se bavím o návrhu GUI, je nutné zmínit i různá dialogová okna, která budou součástí tohoto modulu. Prvním druhem je dialogové okno, které nám poskytne funkci pro výběr objektů. Okno bude obsahovat *jList* pro znázornění všech zájmových objektů a tlačítka OK a Cancel. Dalším druhem budou dialogová okna pro varování uživatele. Budou využita například u špatného vstupu, který zadal uživatel. Budou popisovat konkrétní problém, aby bylo jasné, kde byla chyba zaznamenána.

5.5 Softwarové požadavky - platforma NetBeans

Jelikož je celý systém FOTOM^{NG} a jeho moduly implementován pod touto platformou, je nutné si v této kapitole přiblížit také danou problematiku, aby mohla započít samotná implementace modulu 2D animace. V následujících kapitolách čerpám z literatury [3], [8] a [9].

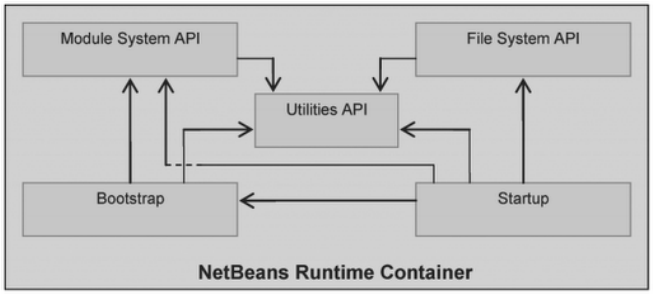
Platformu NetBeans řadíme mezi RCA aplikace. To jsou takové aplikace, které pracují na straně klienta, kde nejen data zobrazují, ale také je zpracovávají. Jedná se o modulární a rozšiřitelný aplikační framework pro Swing aplikace. Tato modularita přináší mnoho výhod jako například zjednodušení vytváření, přidávání a mazání nových prvků aplikace pro uživatele

nebo zjednodušení aktualizací již nainstalovaných prvků aplikace. Avšak největší výhodou této platformy je, že programátor nemusí vyvíjet žádné základní funkce, což ušetří mnoho hodin vývoje.

5.5.1 NetBeans Runtime Container

Slouží jako jádro platformy NetBeans a přebírá na sebe veškerou zodpovědnost za běh aplikace. Nahrává všechny dostupné NetBeans moduly a sestavuje je do jedné Swing aplikace. Dovoluje dynamické instalování a odinstalování modulů, které jsou v tu chvíli dostupné. Jinak řečeno NetBeans Runtime Container (dále jen NRC) je prostředí, které pochopí, jakou funkci mají jednotlivé moduly, pracuje s životními cykly modulu (spuštění, vypnutí, nahrávání a uvolňování modulu) a umožňuje komunikovat s ostatními moduly ve stejné aplikaci. Informace a závislosti o jednotlivých modulech jsou zadány v jejich manifestech. NRC se skládá z těchto pěti modulů, které jsou minimální sestavou, kterou aplikace požaduje pro svůj běh:

- Utilities – poskytuje základní komponenty, například pro komunikaci mezi moduly.
- Module System – odpovídá za správu modulů, jejich závislosti a nastavení.
- File system – zpřístupní virtuální datový systém s přístupem nezávislým na platformě. Zpočátku se použije k načtení zdrojů modulů do aplikace.
- Bootstrap – tento modul je spuštěn dříve než všechny ostatní. Umožňuje NRC pochopit, jaká je funkce daného modulu, načíst ho a poskládat do jedné aplikace.
- Startup – nastartuje aplikaci, inicializuje modulový systém a modul souborového systému.



Obrázek 5 – Závislosti modulů NetBeans Runtime Container

5.5.2 NetBeans Module System

Tento modul odpovídá za správu všech modulů aplikace. Jeho funkcí je také načítání modulů a jejich aktivace a deaktivace.

U tohoto modulu je nutno zmínit práci se soubory manifest. Tyto soubor popisují základní vlastnosti jednotlivých modulů a jejich závislosti na jiných modulech. V tomto souboru jsou aplikované i další atributy, jako například krátký a dlouhý popis modulu, lokalizace properties, lokalizace souboru *layer.xml*, název modulu, definice závislostí mezi moduly nebo verzi modulu.

Veškerá struktura modulu je definována v souboru *JAR*, který obsahuje právě zmiňovaný soubor *manifest.mf*, soubor *layer.xml*, různé *class* soubory a další zdroje jako jsou ikony, properties, atd.

Soubor layer.xml

Každý modul přidaný do aplikace má svůj vlastní XML Filesystem – soubor *layer.xml* (obsahuje informace, co vlastně modul umí). Formátem takového souboru je hierarchický souborový systém se složkami, soubory a atributy. Ve výsledku se sloučí tyto souborové systémy jednotlivých modulů a vytvoří tzv. MultiFileSystem. Jinými slovy, můžeme na něj nahlížet, jako na rozhraní mezi modulem a platformou NetBeans, které deklarativně popisuje integraci modulu do platformy. Takový souborový systém je schopen nám například ukázat, jak bude vypadat menu aplikace.

Strukturu samotného souboru *layer.xml* můžeme prohlížet pomocí prostředí NetBeans IDE. Tuto strukturu vidíme přímo ve stromovém *prohlížeči* projektu. Zvolíme si, zda chceme prohlížet položky, do kterých zasahuje náš soubor *layer.xml*, buď v rámci konkrétního modulu, nebo v rámci celého systému. Tyto položky jsou pak zvýrazněny tučným písmem.

5.5.3 Akce

Velice důležitým prvkem v aplikaci jsou akce. Tyto akce jsou integrovány například do nabídek, panelů nástrojů, apod. Akce pak reagují na uživatelské vstupy do aplikace.

Samotná platforma NetBeans poskytuje své vlastní třídy akcí, například *NodeAction*. Velkou výhodou těchto tříd je rozšíření funkčnosti těchto akcí o asynchronní úlohy.

Akce se v modulu registrují centrálně v souboru *layer.xml*, uvnitř složky *Actions*. Zde mohou být rozděleny do různých skupin, tedy odlišných složek v souboru *layer.xml*. Ostatní třídy pak reagují právě na tyto registrované položky.

Prostředí NetBeans IDE nám poskytuje i průvodce pro tvorbu nových tříd, které popisují akce. Velkou výhodou je, že výše zmíněný průvodce za nás vytvoří všechny potřebné záznamy do souboru *layer.xml*. Pochopitelně je i možná obyčejná ruční implementace, bez použití tohoto průvodce.

5.5.4 Windows System

Jedná se o framework poskytovaný platformou NetBeans. Pracuje a je zodpovědný za rozložení a zobrazení všech oken aplikace. Rozšířením je také možnost, která uživateli dovoluje přizpůsobit si vlastní rozložení oken v pracovním prostředí.

Už v kapitole 4.2 jsme se setkali s možným rozložením oken, jako je u systému FOTOM^{NG} nebo u vývojového prostředí NetBeans IDE. Základním rozdělením je tedy sekce editoru (*Editor*), sloužící k zobrazování několika souboru formou záložek, sekce pohledů (*View*), které jsou rozmístěny kolem sekce editoru a svým způsobem poskytují další editační funkce pro prácí s editorem. Jedná se například o okno *vlastností*, okna *prohlížeče* nebo okno *výstupu*.

Okno TopComponent

Windows System API poskytuje třídu *TopComponent*, která je potomkem třídy *JComponent*. Na této třídě jsou založena všechna okna, která se integrují do platformy NetBeans. Všichni potomci, kteří budou implementování, se můžou během své existence nacházet v různých stavech, jako jsou stavy otevřený, uzavřený, viditelný, neviditelný, aktivní a neaktivní.

5.5.5 Lookup

Jedná se o základní komponentu v platformě NetBeans. Poskytuje prostředky pro komunikaci mezi jednotlivými moduly. Lookup si můžeme představit jako strukturu *Map*, ve které je klíčem pro vyhledávání třída *.class* a hodnota je instance objektu.

5.5.6 Nodes API

Úkolem Nodes API je vizuálně reprezentovat data. Právě s tímto API bude spolupracovat *prohlížeč* (Explorer), který slouží jako kontejner uzlů.

Samozřejmě uzly nemusí sloužit jen k zobrazování. Můžeme je použít k poskytování akcí, jako jsou například poklepání na uzel nebo jeho označení.

6. Realizace modulu

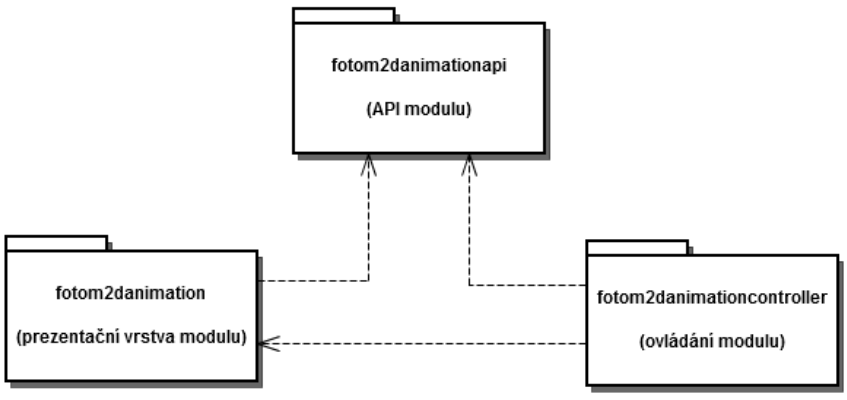
Díky teorii, specifikaci požadavků, návrhu uživatelského rozhraní a analýze z předchozích kapitol, jsem byl schopen naimplementovat tento modul pro 2D animace. Hlavním úkolem této kapitoly bude seznámení s implementovanými třídami, včetně třídního diagramu, struktura celého modulu, vývojové prostředí a všechny náležitosti, které jsou nutné pro samotné zhodnocení, jako například ověření funkčnosti a konečný vzhled modulu.

6.1 Struktura modulu

Celá funkcionalita modulu pro 2D animace byla rozdělena do tří balíčků. Výhodou tohoto rozdělení je zvýšení přehlednosti, případné jednodušší rozšíření funkcionality modulu. Avšak hlavním důvodem bylo následování trendů. Jinými slovy, struktura byla volena na základě jiných modulů. Důvodem je zachování nějaké společné konvence pro vývoj tohoto systému. Je nutné připomenout, že všechny třídy a jejich metody mají svou programátorskou dokumentaci, která je součástí přílohy této bakalářské práce. Toto jsou výše uvedené balíčky implementovaného modulu:

1. Balíček určený pro API modulu – fotom2danimationapi
2. Baliček určený pro prezentační vrstvu – fotom2danimation
3. Balíček určený pro ovládání – fotom2danimationcontroller

Na obrázku 6 jsou uvedeny jednotlivé závislosti mezi těmito balíčky. Na první pohled si každý uvědomí, že je tato struktura stejná, jako u návrhového vzoru MVC. Jak bylo napsáno v úvodu této kapitoly, snažil jsem se použít stejnou konvenci ostatních modulů. Jak později zjistíme, balíček pro ovládání obsahuje jen třídu, která realizuje akci pro spuštění modulu. V dnešní době jsou metody pro ovládání z 90% umístěny přímo u tříd prezentační vrstvy. Jedná se například o metody, které reagují na stisk tlačítka uživatelského rozhraní a přímo pracují s komponentami této třídy.



Obrázek 6 – Závislosti balíčků modulu 2D animace

6.1.1 Balíček pro ovládání modulu

Kompletní název balíčku je org.fotomapp.fotom2danimationcontroller. V těchto podkapitolách jsem si zvolil cestu, jak nejlepším způsobem popsat jednotlivé balíčky. Budeme následovat cyklus začínající od stisknutí menu položky *2D animace* uživatelem.

AddFtmForAnimation2dAction.java

Tato třída, poskytující akci (víc kapitola 5.5.3) pro stisk menu položky *2D animace*. Její předností je povolení nebo zakázání menu položky, do té doby, dokud nejsou splněny určité náležitosti. V tomto případě se jedná o splnění podmínky, zda do 2D animací vstupuje série měřických snímku v počtu dva a více. Tohoto se dosahuje pomocí metody *enable():*

```
...
if (activatedNodes == null || activatedNodes.length < 2)
   return false;
...
for(Node n : activatedNodes){
   Lookup nodeLookup = n.getLookup();
   if (nodeLookup.lookup(FtmObject.class) == null)
      return false;
}
return true;
...
```

Metoda je typu *boolean*. V první fázi se kontroluje, zdali jsou označeny nějaké nody (víc kapitola 5.5.6), v našem případě měřické snímky. Musí být přesně definováno, za jakých okolností bude položka ve vypnutém stavu. V další fázi se kontroluje Lookup (víc kapitola 5.5.5), zdali označené snímky jsou FTM objekty. Pokud je vše v pořádku, položka se zpřístupní.

Po stisku této položky je volaná metoda *performAction()*, která zkontroluje, zdali tato série splňuje podmínky pro 2D animace, aby se pak mohla spustit hlavní komponenta pro 2D animace. Tím ji také předává všechny FTM objekty formou nodů.

```
...
Animation2dTopComponent tc = new Animation2dTopComponent();
if (tc.setFtmObjectNodesAsSeries(activatedNodes)){
   tc.open();
   tc.requestActive();
}
...
```

6.1.2 Balíček pro prezentační vrstvu modulu

Kompletní název balíčku je org.fotomapp.fotom2danimation. V tomto balíčku jsou obsažena hlavní komponenta *TopComponent* spolu s dialogovým oknem pro výběr objektů. Dále obsahuje důležité třídy pro znázornění filmového pásu a maximalizovaného náhledu snímku a jiné soubory, jako například *layer.xml*, properties a ikony.

Animation2dTopComponent.java

Tato třída implementuje samotné uživatelské rozhraní modulu 2D animace. Konstruktor je volán po stisku položky v menu (víc kapitola 6.6.1 AddFtmForAnimation2dAction.java). Další dodatečné informace o komponentě *TopComponent* jsou uvedeny v kapitole 5.5.4.

Třída obsahuje velkou škálu implementovaných metod, které slouží pro manipulaci s animacemi a jejich rozšířenými funkcemi, které byly definovány v požadavcích (víc kapitola 5.2). Především velmi důležité metody jsou pro znázornění snímků filmového pásu a znázornění náhledu snímku. Jedná se o metody *displayFilm()* a *displayFtm()*.

V metodě *setFtmObjectNodeAsSeries()* jsme dostali jako parametr všechny potřebné snímky, které zkontrolujeme, zdali splňují podmínky pro sérii využitelnou ve 2D animacích. Pokud jsou tyto náležitosti splněny, postupně každý snímek předáme jako parametr metodě *displayFilm()*. Tato metoda pak vytvoří pro daný snímek panel s příslušným názvem snímku a na tento panel pak umístí *filmCanvas*. *FilmCanvas* reprezentuje miniaturu daného FTM snímku třídy *FilmDisplay*. Tento proces se vykoná pro celou sérii a sestaví nám pak kompletní filmový pás. Ještě je nutno dodat, že každý panel je vybaven vlastním posluchačem stisku myši, který vyvolá metodu *panelMouseClicked():*

```
...
if(evt.getClickCount() == 1){
    ...
    panelContainer[panelNumber].setBorder(BorderFactory.createMatteBord
    er(5, 5, 5, 5, Color.red));
    displayFtm(panelNumber);
    ...
}
...
```

Pokud se tedy jedná o jeden stisk myši, nastaví ohraničení miniatury snímku filmového pásu na červenou barvu. Následně je pak volaná funkce *displayFtm()*, která daný snímek zobrazí v maximalizovaném náhledu. Pochopitelně, že tato metoda udržuje vždy jen jeden ohraničený prvek. Toho docílí zrušením všech hranic a následné nastavení hranice jen požadovaného prvku. Zobrazení snímku v náhledu probíhá pomocí proměnné *canvas*, která je umístěna na panel náhledu. *Canvas* reprezentuje FTM snímky třídy *Display*.

Jak vyplývá z úryvku kódu, celá třída pracuje se skupinou polí, které usnadňují veškerou implementaci funkcí daného modulu. Důležitou je pak proměnná *panelNumber*, která slouží jako ukazatel na danou miniaturu filmového pásu. V tomto ohledu je nutné zmínit i řadu užitečných metod. Například *getCurrentFrame(),* která vrací číslo označeného snímku, *getPanelCount()* pro počet všech snímků, *getToolCount()* pro počet všech nástrojů snímků celé série, která se využívá u dialogového okna pro výběr objektů.

Pokud budeme hovořit o samotných animacích, jejich implementace je většinou založena na inicializaci a spuštění časovače pomocí příkazu *timer.Start()*. Konkrétně pro tyto funkce byl využit časovač swing.Timer. Poskytuje jednoduchou implementaci, včetně definování časových prodlev, které jsem použil i pro funkci uživatelsky definované prodlevy mezi snímky v automatické animaci. Například u obyčejné animace je nalezen snímek, který je ohraničen a ukazatel je posunut na další snímek v sérii. Tento snímek je pák znázorněn v maximalizovaném náhledu. Tento proces se stále opakuje, dokud není animace zastavena a to buď uživatelem, nebo bylo dosáhnuto konce série. Samotné ukončení je realizováno zastavením časovače příkazem *timer.Stop()*.

V této třídě nalezneme i mnoho metod, které reagují na stisk tlačítek, umístěných v panelu nástrojů modulu. Ty pak volají příslušné metody implementující požadované funkce.

ObjectChooserPanel.java

Tato komponenta je implementována pro funkci výběru objektů. Konstruktor komponenty se volá ze třídy *Animation2dTopComponent*, která vytvoří pro tuto komponentu dialogové okno opatřené tlačítky OK a Cancel.

Konstruktor přebírá jako parametry pole nástrojů všech snímků série. Nástroje uložené v tomto poli definují zájmové objekty, které jsou umístěny na jednotlivých snímcích. V dalších algoritmech je toto pole přefiltrováno a jsou z něj vybrány názvy zájmových objektů bez duplicit.

Uživatel pak prostým výběrem předá list těchto názvů zpět hlavní komponentě *Animation2dTopComponent*, která jej použije pro své metody.

FilmDisplay.java

Tato třída slouží pro znázornění miniatur filmového pásu. Je využívaná v hlavní komponentě *Animation2dTopComponent* pro přidání prvku *filmCanvas* na jednotlivé panely filmového pásu. Jedná se o konkrétní implementaci třídy *FilmDisplayer*.

Display.java

Tato třída slouží pro znázornění maximalizovaného náhledu měřického snímku. Také je využívaná v hlavní komponentě. Je konkrétní implementací třídy *Displayer*, která je například použita v systému FOTOM^{NG} pro editaci jednotlivých snímků. Avšak tato třída nepodporuje zakreslování nových zájmových objektů pomocí nástrojů popsaných v kapitole 4.2.2 a jejich případné reakce na klik myši.

layer.xml

V kapitole 5.5.2 jsem probral teoretickou stránku souboru *layer.xml*. Avšak zde si uvedeme konkrétní příklad tohoto souboru, který přímo řeší přidání položky do hlavního menu systému spolu s registrováním akce pro tuto položku.

```
...
<filesystem>
    <folder name="Actions">
        <folder name="2Dmodeling">
            <file name="org-fotomapp-fotom2danimationcontroller-
               AddFtmForAnimation2dAction.instance">
               ...
            </file>
        </folder>
    </folder>
    ...
</filesystem>
```

Ve složce *Actions* se provádí registrace akcí. Konkrétně pro tento případ jsou akce ještě rozděleny do dalších adresářů, jako je například složka *2dmodeling*. Jako parametr bloku *file* se uvádí plně kvalifikovaný název třídy, která zajišťuje provedení akce nad danou položkou. Postfix *.instance* znamená, že chceme vytvořit instanci tohoto objektu. Do bloku *file* se pak uvedou atributy.

```
...
<folder name="Menu">
   <folder name="Modeling">
      <file name="javax-swing-JSeparator.instance">
         ...
      </file>
      <file name="AddFtmForAnimation2dAction.shadow">
         ...
      </file>
   </folder>
</folder>
...
```

Pod registrovanou akcí nyní uvedeme samotné umístění položky v menu. Opět pomocí složek *file* systému si najdeme požadované umístění položky. Parametrem bloku *file* je pak reference (odkaz) na akci s postfixem *.shadow*. Do bloku *file* následně vpisují jednotlivé atributy. Jedná se například o atribut, který definuje cestu k instanci akce nebo o atribut, který určuje, na jaké pozici bude položka umístěna.

Pochopitelně jsem použil i separátor, který danou nabídku menu více zpřehlední. Do bloku separátoru se většinou umístí atribut, který vyjadřuje jeho pozici v nabídce.

6.1.3 Balíček pro API modulu

Kompletní název balíčku je org.fotomapp.fotom2danimationapi. V této podkapitole si popíšeme jednotlivé třídy tohoto balíčku. Tyto třídy nám realizují funkcionalitu celého modulu pro 2D animace. V tomto balíčku jsou umístěny i všechny výjimky tohoto modulu.

FilmDisplayer.java

Tato abstraktní třída slouží pro odvození třídy, která bude v hlavním okně modulu zobrazovat miniatury filmového pásu. Tato třída je potomkem třídy *DisplayJAI*. Třída neimplementuje žádné dodatečné metody, protože pro zobrazení ve filmovém pásu nejsou potřeba.

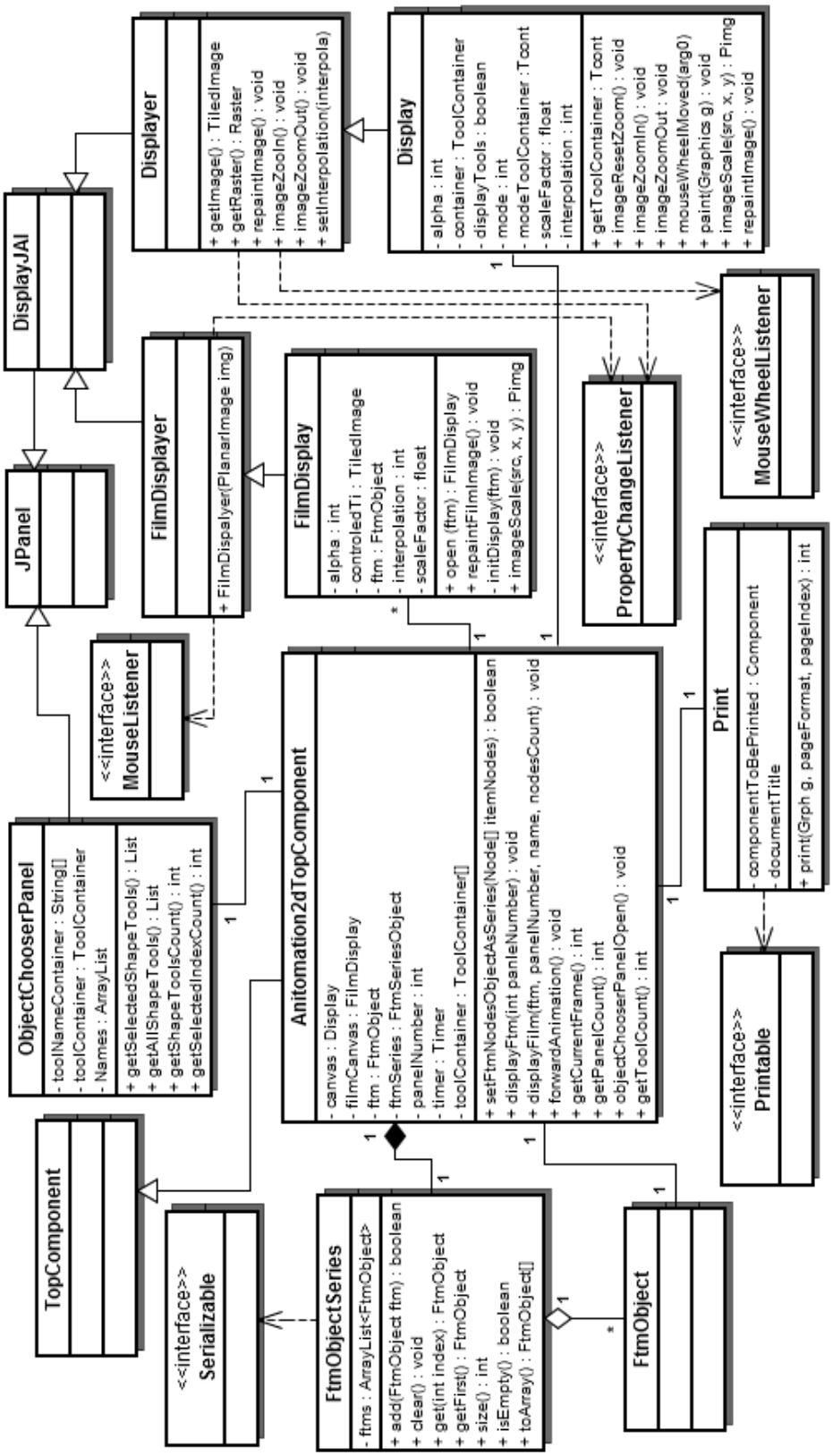
FtmSeriesObject.java

Tato třída reprezentuje sérii snímků, které vstupují do modulu 2D animace. Samotná série je zde reprezentovaná formou listu. Tato třída dále poskytuje řadu užitečných metod. Tyto metody vyplývají ze základní podpory, kterou známe u práce s *ArrayList*. Jedná se o metody jako *size(), clear(), getFirst(), toArray()* a jiné.

Důležitou je pak metoda, která kontroluje validitu snímků celé série. Tato metoda je volaná přímo z konstruktoru a kontroluje každý snímek. V případě, že nesplňuje podmínku, vyvolá se příslušná výjimka.

Print.java

Jak už název napovídá, jedná se o třídu, která obstarává tisk. Konstruktor třídy je volán přímo z hlavní komponenty *Animation2dTopComponent* přes tlačítko funkce tisku. Jak bylo uvedeno ve specifikaci, bude možnost tisku rozšířena o tisk filmového pásu i náhledu snímku.



Obrázek 7 – Třídní diagram modulu 2D animace

6.2 Vývojové prostředí – použitý software

Pro vývoj modulu 2D animace jsem použil vývojové prostředí NetBeans IDE nejnovější verze 7.0.1, včetně JDK verze 1.7.0.0. Je nutná i Java distribuce JRE.

Pro spuštění samotného systému FOTOM^{NG} bylo nutno doinstalovat ještě dva balíčky pro JAI. Jedná se o Java Advanced Imaging a Java Advanced Imaging CLASSPATH. Tyto balíčky poskytují třídy, které mají implementovanou pokročilou práci s obrázky.

6.3 Ověření funkčnosti

Modul pro 2D animace byl implementován v rozsahu, který mu udávaly specifikované požadavky z kapitoly 5.2.

Modul je vybaven několika dialogovými okny. Tato okna poskytnou uživateli v případě nějaké výjimky informace o dané chybě a jak jí předejít. Tímto postupem byly vybaveny části kódu, kde se pracuje s uživatelsky zadávanými daty.

Byla vytvořena testovací série snímků, na které bylo nadefinováno několik zájmových objektů. Nakonec byla ověřena celková funkčnost systému. Také znovu ověřeny jednotlivé funkce panelu nástrojů. Závěrem se testovala samotná integrace modulu do systému FOTOM^{NG}.

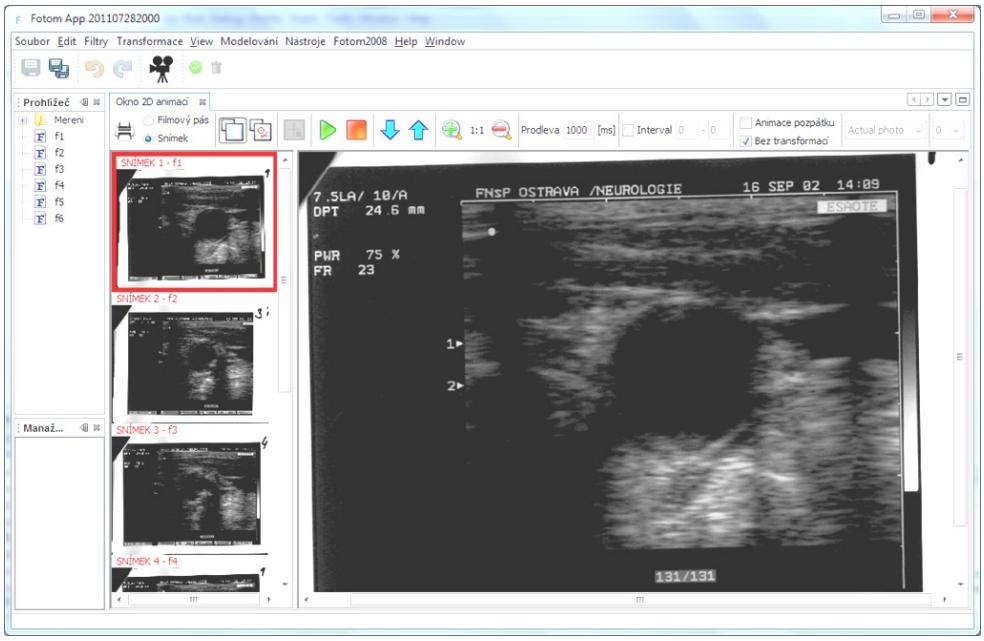
V následujícím soupisu jsou vypsány všechny funkce, které byly testovány:

- Tisk – připojením tiskárny jsem testoval tisk filmového pásu i náhledu snímku.
- Animace snímků – přehrávání od začátku série, prostředku série a konce série. Zastavení animace během přehrávání. Použití animace pozpátku.
- Animace objektů – přehrávání od začátku série, prostředku série a konce série. Zastavení animace během přehrávání. Použití animace pozpátku.
- Výběr objektů – testoval se výběr jednoho objektu, více objektů i žádného objektu, funkce tlačítek OK a Cancel.
- Ruční animace – ovládání pomocí kláves. Zamezení funkce v případě dosažení konce nebo začátku.
- Manipulace s obrazem – přiblížení, oddálení pomocí kolečka myši a restart snímku do původního zobrazení.
- Nastavení prodlevy – zadávání různých hodnot prodlevy. Zadávání řetězců a následné dialogové okno pro hlášení špatného vstupu.
- Intervalové animace – testování různých intervalů, jako například stejné, mimo rozsah, první bude větší než druhý interval. Zadávání řetězců a následné dialogové okno pro hlášení špatného vstupu.
- Typ náhledu snímku – testovány všechny tři možnosti u animací objektů.
- Počet objektů předcházejících snímků – testovány všechny tři možnosti u animace objektů.
- Zakázání některých funkcí při spuštění animace, aby uživatel nemohl jinými funkcemi zasahovat do probíhající animace.
- Testování podmínky, pro povolení položky *2D animace* v hlavním menu.
- Kontrola české nápovědy jednotlivých funkcí panelu nástrojů.

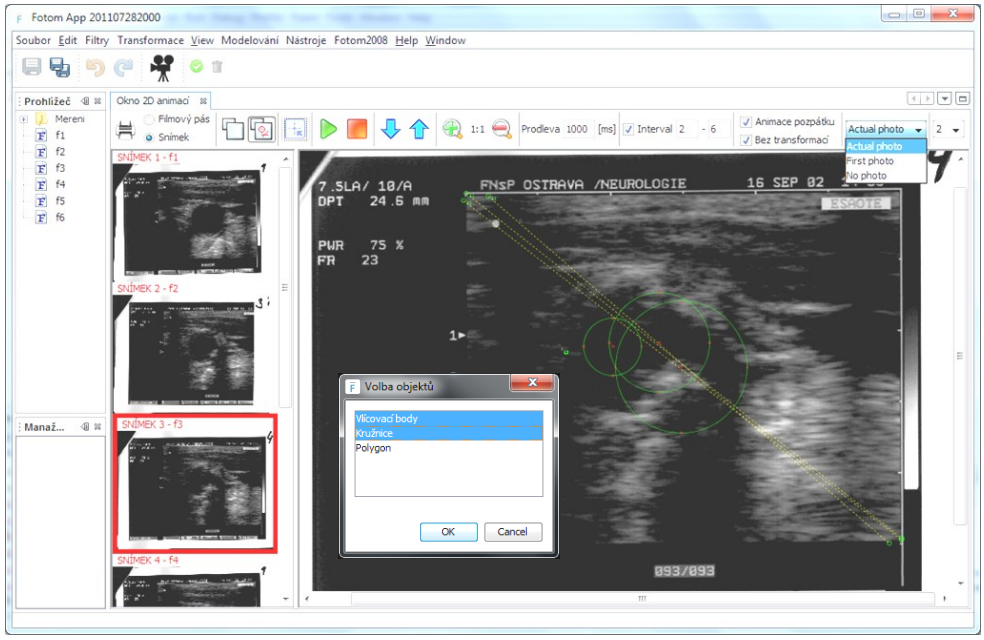
Po kontrole modulu, jsem dospěl k názoru, že je připraven k běžnému použití. Nicméně i přes velmi důsledné testování hrozí výskyt drobných chyb, které mohou být kdykoliv odstraněny.

6.4 Konečný vzhled modulu 2D animace

Jak bylo předpokládáno, konečný vzhled uživatelského rozhraní byl stejný jako návrh, který je uveden v kapitole 5.4. Jako ukázku si uvedeme několik obrázků z tohoto modulu.



Obrázek 8 – Modul pro 2D animace – Animace snímků



Obrázek 9 – Animace objektů se znázorněním dvou objektů předcházejících snímků

7. Závěr

Od zadání této bakalářské už uběhla dlouhá doba. V prvotních fázích seznámení se s touto prací, jsem se setkal s mnoha novými pojmy, jako například pojem fotogrammetrie. Už tehdy jsem si dal za svůj příští cíl nastudování této problematiky. I přesto, že mi byl tento pojem neznámý, usoudil jsem, že mnoho lidí pojem fotogrammetrie také nikdy neslyšelo. Bylo však pro mě velkým překvapením, že fotogrammetrie je běžnou součástí našeho života.

Mým dalším rozhodnutím bylo získat zkušenosti ze systému FOTOM 2008 a vyvíjeného systému FOTOM^{NG}. U systému FOTOM 2008 se stal mým středem zájmu čtvrtý modul systému, který se zabýval právě 2D animacemi. Jelikož byl modul včetně celého systému v plně funkčním a hotovém stavu, nejen že jsem získal zkušenosti z tohoto softwaru, ale také jsem měl k dispozici nějakou předlohu pro tvořený modul. Už tehdy jsem si uvědomoval některé nedostatky modulu FOTOM4, které jsem měl v plánu s tvořeným modulem odstranit. Velkým přínosem se pro mě staly také konzultace, kde jsem se dozvěděl mnoho užitečných informací, včetně podání požadavků, které se kladly na nový modul. Dalším úkolem bylo seznámení s vyvíjeným systémem FOTOM^{NG}. Systém je postaven na platformě NetBeans. Jelikož to bylo mé první setkání s platformou, musel jsem nastudovat danou problematiku, což se mi jevilo jako velmi důležité. K tomu mi velmi pomohla literatura. Díky těmto nabytým novým znalostem, jsem mohl pokročit k návrhu modulu 2D animace pro systém FOTOM^{NG}.

S návrhem nebyly žádné větší problémy. K dispozici jsem měl uživatelské rozhraní předchozí verze spolu se specifikací požadavků, mohlo se tudíž přikročit ke zpracování prvotního návrh GUI. Pochopitelně jsem aplikoval i změny, které měly mít za následek odstranění nedostatků z předchozí verze. Když byl hotov návrh modulu, mohl jsem přejít na samotnou implementaci nového modulu.

Závěrem jsem tedy implementoval modul 2D animace, který byl integrován do systému FOTOM^{NG}. Mým úkolem bylo také otestování tohoto modulu a odstranění případných funkčních nedostatků. Nyní jsou 2D animace připraveny k použití. V příloze této bakalářské práce je umístěna i programátorská dokumentace implementovaných tříd, včetně všech zdrojových kódů.

Vývoj samotného modulu pro 2D animace je velmi důležitý. Jako jediný s modulů systému FOTOM^{NG} poskytuje uživateli přehled o kvalitě měřických snímků. Žádný jiný z modulů tuto funkcionalitu poskytnout nemohl. Další velkou výhodou je získání pohledu na měřený objekt jinou a nezávislou formou, než je tomu například z grafu. Uživatel má také možnost využít uživatelské příručky, která je umístěna v příloze této bakalářské práce.

Během realizace tohoto bakalářského projektu, jsem se snažil využívat znalostí nabyté ze studia na VŠB-TUO. Velmi mi pomohly znalosti z oblasti tvorby uživatelských rozhraní, postupy při implementaci systémů nebo samotný jazyk Java. Nicméně velkým přínosem byla pro mě samotná bakalářská práce. Díky ní jsem se seznámil s platformou NetBeans, která je dnes běžně zdarma používanou platformou pro tvorbu systémů. Také jsem rozšířil své znalosti v oblasti programování větších projektů. Musím zmínit i nabyté znalosti z oblasti fotogrammetrie nebo animací.

Nápad samotného systému FOTOM je velice dobrý, už jen z pohledu, že se jedná o software, který poskytuje velkou škálu funkcí pro zpracování měřických snímků. S jistotou mohu tvrdit, že vývoj myšlenky FOTOM bude nadále pokračovat a zdokonalovat se. Do budoucna je třeba ještě naplánovat modul pro 3D animace. Další možností je rozšířit stávající modul 2D animace o nové funkce 3D animací. Jsem velice vděčný za to, že mi bylo umožněno se na tomto projektu podílet.

Literatura

[1] PYTLÍK, Tomáš. *2D počítačové zpracování fotografie na PC*. Ostrava, 2011. 46 s. Diplomová práce. Vysoká škola Báňská – Technická univerzita Ostrava.

[2] LIČEV, Lačezar. *Analýza, modelování, rozpoznávání a vizualizace procesu měření objektů na snímcích*. Vydání první. Brno: Computer Press, a.s., 2010. 125 s. ISBN 978-80-251-3296-8

[3] BÖCK, Heiko. *Platforma NetBeans: Podrobný průvodce programátora*. Vydání první. Brno: Computer Press, a.s., 2010. 320 s. ISBN 978-80-251-3116-9

[4] Animation. In *Wikipedia: the free encyclopedia* [online]. St. Petersburg (Florida): Wikipedia Foundation, Inc., 14 September 2002, last modified on 16 April 2012 [cit. 2012-04-18]. Dostupné z WWW: < http://en.wikipedia.org/wiki/Animation>

[5] Praxinoscope. In *Wikipedia: the free encyclopedia* [online]. St. Petersburg (Florida): Wikipedia Foundation, Inc., 14 September 2002, last modified on 22 January 2012 [cit. 2012-01-18]. Dostupné z WWW: <http://en.wikipedia.org/wiki/Praxinoscope >

[6] Persistence of vision. In *Wikipedia: the free encyclopedia* [online]. St. Petersburg (Florida): Wikipedia Foundation, Inc., 14 September 2002, last modified on 3 April 2012 [cit. 2012-04-18]. Dostupné z WWW: <http://en.wikipedia.org/wiki/Persistence_of_vision >

[7] 2D komputer graphics. In *Wikipedia: the free encyclopedia* [online]. St. Petersburg (Florida): Wikipedia Foundation, Inc., 14 September 2002, last modified on 15 March 2012 [cit. 2012-04-18]. Dostupné z WWW: <http://en.wikipedia.org/wiki/2D_computer_graphics >

[8] Vývojová platforma NetBeans. In *Wikipedia: the free encyclopedia* [online]. St. Petersburg (Florida): Wikipedia Foundation, Inc., 14 September 2002, last modified on 20 December 2010 [cit. 2012-03-15]. Dostupné z WWW: <http://cs.wikipedia.org/wiki/V%C3%BDvojov%C3%A1_platforma_NetBeans >

[9] *NetBeans* [online]. 2012 [cit. 2012-03-16]. NetBeans Platform. Dostupné z WWW: <http://netbeans.org/features/platform/features.html >

Přílohy

Přílohy k této bakalářské práci jsou uloženy na přiloženém CD. Příloha obsahuje:

1. Zdrojové kódy modulu 2D animace
2. Systém FOTOM^{NG} s integrovaným modulem 2D animace
3. Uživatelská příručka pro práci s modulem 2D animace ve formě PDF
4. Programátorská dokumentace modulu 2D animace ve formě Javadoc
5. Samotná bakalářská práce ve formě PDF a DOC