

Slovenská technická univerzita v Bratislave
FAKULTA INFORMATIKY A INFORMAČNÝCH
TECHNOLÓGIÍ

Tímový projekt
Rekonštrukcia 3D scény
Projektová dokumentácia - riadenie



Vedúci projektu:
Ing. Vanda Benešová, PhD.

Členovia tímu:
Bc. Lukáš Hudec (IS)
Bc. Róbert Birkus (IS)
Bc. Michal Löffler (IS)
Bc. Róbert Karásek (IS)
Bc. Martin Jurík (IS)
Bc. Michal Korbeľ (IS)
Bc. Katarína Janečková (IS)

Názov tímu: R3D (tím č. 5)
Web: <http://labss2.fiit.stuba.sk/TeamProject/2015/team05is-si/>
Kontakt: R3DteamTP@gmail.com
Akademický rok: 2015/2016
Dátum odovzdania: 14. december 2015

Obsah

1	Úvod	3
1.1	Prehľad dokumentu	3
2	Role členov tímu a podiel práce	4
2.1	Manažérske činnosti	4
2.2	Podiel práce	4
3	Aplikácie manažmentov	6
3.1	Manažment komunikácie a ľudských zdrojov	6
3.1.1	Komunikačné nástroje	6
3.2	Manažment rozvrhu a rozsahu projektu	7
3.3	Manažment plánovania projektu	7
3.4	Manažment kvality	8
3.5	Manažment rizík	9
3.6	Manažment integrácie a podpory vývoja	10
4	Sumarizácie šprintov	11
4.1	1. šprint	11
4.2	2. šprint	11
4.3	3. šprint	12
4.4	4. šprint	12
4.5	5. šprint	13
5	Používané metodiky	14
6	Globálna retrospektíva	15
7	Zoznam kompetencií tímu	17
8	Metodiky	18
8.1	Metodika komunikácie	18
8.2	Metodika písania zdrojového kódu	18
8.3	Metodika pre technickú dokumentáciu	28
8.4	Metodika evidencie úloh v TFS	28

8.5	Metodika pre evidenciu zmien zdrojového kódu	31
9	Export evidencie úloh	33

Kapitola 1

Úvod

V tomto dokumente sú opísané postupy, plány, a metódy riadenia tímového projektu stanovené naším tímom pre vývoj prototypu desktopovej aplikácie rekonštrukcie 3D scény z 3D dát. Charakter projektu je výskumne orientovaný, preto aj jeho hlavným cieľom je vytvoriť aplikáciu, ktorá bude spĺňať predpoklady testovacieho prototypu rôznych metód rekonštrukcie a segmentácie objektov. Projekt má predpoklad byť vývojovo a zdrojovo náročný, rovnako ako jeho prototyp náročný výpočtovo a pamäťovo. Preto je potrebné vytvoriť a dodržiavať predpísané pravidlá, aby sa možné komplikácie eliminovali skôr akoby mali skomplikovať ďalší vývoj. Vzhľadom na charakter projektu bolo potrebné upraviť aj metodiky riadenia, plánovanie vývoja a rozdelenie úloh na funkcionality, aby na ich základe vznikol požadovaný produkt.

Projekt 3DRecon je vyvíjaný pre výskumné centrum Joanneum v Grazi, aj preto nemá charakter klasického produktu pre koncového používateľa, ale skôr výskumne orientovaného prototypu, ktorý môže slúžiť ako základná knižnica pre ďalší vývoj v oblasti rekonštrukcie či segmentácie 3D dát.

1.1 Prehľad dokumentu

V časti 2 sú opísané manažérske činnosti pridelené jednotlivým členom tímu aj s opisom prislúchajúcej zodpovednosti (2.1) a uvedený podiel na jednotlivých častiach dokumentácie k riadeniu aj inžinierskemu dielu (2.2). V časti 3 sú opísané realizované manažmenty v rámci nášho tímového projektu. Sumarizácie doterajších šprintov sa nachádzajú v kapitole 4. V kapitole 5 je prehľad používaných metodík s krátkym opisom, celé znenie jednotlivých metodík sa nachádza v časti 8. Na záver sú v kapitole 9 vložené exporty úloh z nástroja TFS na jednotlivé šprinty.

Kapitola 2

Role členov tímu a podiel práce

2.1 Manažérske činnosti

meno	rola	zodpovednosť
Lukáš Hudec	team leader, manažér architektúry	návrh a implementácia architektúry, správa tímu a rozdelenie úloh, evidencia úloh v TFS, segmentácia
Róbert Birkus	manažér vývoja	vývoj hlavnej aplikačnej logiky, správa implementovaných metód, starostlivosť nad dodržiavaním konvencií
Martin Jurík	projektový integrátor	prepojenie modulov, kontrola funkčnosti, implementácia modulov
Michal Löffler	biznis manažér	analýza požiadaviek, návrh a rozdelenie úloh, komunikácia so zadávateľom, analýza rizík
Róbert Karásek	test manažér	tvorba testov, vykonanie testov, dohľad nad “definition of done”
Michal Korbeľ	manažér propagácie	správa webovej stránky
Katarína Janečková	manažér dokumentácie	udržiavanie dokumentácie a zápisník zo stretnutí

Tabuľka 2.1: Rozdelenie úloh v tíme

2.2 Podiel práce

KAPITOLA 2. ROLE ČLENOV TÍMU A PODIEL PRÁCE

časť	autor
Úvod	Katarína Janečková
Globálne ciele pre ZS	Martin Jurík
Celkový pohľad na systém	Lukáš Hudec
Rekonštrukčné triedy systému - úvod	Katarína Janečková
Metóda 1	Katarína Janečková
Metóda 2	Michal Korbel
Metóda 3	Róbert Karásek
Metóda 4	Róbert Birkus
Vizualizácia	Michal Löffler

Tabuľka 2.2: Podiel práce na dokumentácii k inžinierskemu dielu

časť	autor
Úvod	Lukáš Hudec
Role členov tímu a podiel práce	Katarína Janečková
Manažment komunikácie a ľudských zdrojov	Róbert Karásek
Manažment rozvrhu a rozsahu projektu	Lukáš Hudec
Manažment plánovania projektu	Róbert Karásek, Lukáš Hudec
Manažment kvality	Michal Löffler
Manažment rizík	Michal Löffler
Manažment integrácie a podpory vývoja	Martin Jurík
Sumarizácie šprintov	Michal Korbel
Používané metodiky	Katarína Janečková
Zoznam kompetencií tímu	Katarína Janečková
Metodika komunikácie	Róbert Karásek
Metodika pre evidenciu zmien zdrojového kódu	Róbert Karásek
Metodika vytvárania úloh v TFS	Lukáš Hudec
Metodika písania zdrojového kódu	Róbert Karásek
Metodika pre technickú dokumentáciu	Róbert Karásek
Export evidencie úloh	Katarína Janečková

Tabuľka 2.3: Podiel práce na dokumentácii k riadeniu

Kapitola 3

Aplikácie manažmentov

3.1 Manažment komunikácie a ľudských zdrojov

Oficiálne stretnutia tímu prebiehali každý týždeň vo štvrtok o 10:00. Ich témou bolo zhodnotenie vykonanej práce za predchádzajúci týždeň a naplánovanie ďalšej. Po stretnutí prebiehala diskusia o spoločnej implementácii, o dokumentácii, čo možno zlepšiť a ďalšie problémy, ktoré sa vyskytli počas týždňa. Prebiehali aj neformálne stretnutia, ktoré sa konali v škole alebo pomocou skype. Ich témou bolo hlavne riešenie menších problémov, ktoré vznikli mimo stretnutí a bolo ich potrebné vyriešiť čo najskôr.

3.1.1 Komunikačné nástroje

Pri výbere nástrojov sme vychádzali zo skúseností a preferencií jednotlivých členov tímu. Analyzovali sme najznámejšie nástroje a dohodli sme sa na nasledujúcich nástrojoch:

Slack

Je to aplikácia, ktorá slúži na komunikáciu a veľa používateľov si ju pochvaľuje aj vďaka zvýšenej produktivite. Dá sa používať aj ako samostatná aplikácia alebo vo webovom prehliadači. V aplikácii sme si vytvorili komunikačné kanály pre potrebné témy. Ako príklad jeden kanál je pre GUI ďalší pre datasety a podobne. Umožňuje posielanie súborov, zdieľanie kódov ale aj notifikácie, čo je užitočné hlavne pri naliehavých úlohách. V prvých fázach sa využíval Facebook, nakoľko tam bol registrovaný každý člen tímu.

Nástroj pre zdieľanie obsahu Google Drive

Google Drive používame pre zdieľanie a zálohovanie dokumentov, nie len k dokumentácii ale aj k implementácii. Nachádzajú sa tu aj datasety, logy k našej webovej stránke a pod. Prístup k tomuto priečinku má každý aj vedúci tímu.

3.2 Manažment rozvrhu a rozsahu projektu

Pre správne rozvrhnutie funkčných úloh pre členov tímu je dôkladná príprava a podrobná analýza problému, ktorým sa náš tím zaoberá. Bližšie informácie k tomuto problému poskytla diskusia so zadávateľom, ktorého zastupoval vedúci nášho tímu. Po tomto stretnutí a analýze požiadaviek sa vytvoril zoznam cieľov, ktoré si náš tím rozdelil medzi prioritné a voliteľné, podľa času kedy je ich žiadúce dosiahnuť. Následne bolo potrebné vytvoriť metodiky pre kooperatívne riešenie požiadaviek zadávateľa. Výsledkom tejto analýzy sa stali postupy práce a vývoja. Na základe dodržiavania metodík a postupovaní podľa nami zvolených princípov sme vytvorili sadu štruktúrou elementárnych úloh, na ktorých pracovali vybraní členovia tímu počas dvojtyždňových šprintov. Pre transparentnosť práce na projekte a vnútornú organizáciu jednotlivých úloh náš tím používal TFS – systém na správu projektu.

Práca a rozdelenie úloh medzi členov tímu prebiehalo počas scrum stretnutí na začiatku každého šprintu. Na začiatku sa skontrolovali úlohy v backlogu, podľa progresu na poslednom šprinte a ďalších požiadaviek od zadávateľa sa zhodnotil význam a potreba niektorých úloh. Následne sa podľa analýzy zo stretnutia vytvorili ďalšie „používateľské príbehy“ a priradili sa k nim implementačné úlohy. Tieto príbehy boli následne ohodnocované všetkými členmi tímu, čím sa určila ich odhadovaná obťažnosť. Podľa tejto hodnoty bola jednotlivým úlohám priradená časová rezerva na ich riešenie. Vzhľadom na štruktúru problému a zameranie členov, každý člen dostal priradenú úlohu podľa oblasti, ktorú si vybral.

Po stretnutí sú nové a priradené úlohy a príbehy evidované v nástroji TFS podľa vopred spísanej metodiky vytvárania „taskov“ v TFS. Evidovanie progresu na jednotlivých úlohách prebiehalo tiež prostredníctvom TFS.

3.3 Manažment plánovania projektu

O manažment plánovania projektu sa v našom tíme staral team líder, manažér dokumentácie pričom ich hlavou úlohou v tejto oblasti bolo udržiavanie konvencií a dodržiavanie metodík tvorby úloh.

Keďže pre implementáciu riešenia bolo zvolené vývojové prostredie Visual Studio 2013 (pôvodne 2015, z dôvodu problémom s knižnicami sme prešli na stabilnejšiu verziu) s potrebnými nástrojmi a knižnicami, bolo pochopiteľné že pre správu projektu a jeho verzií, bol zvolený systém TFS. TFS poskytuje pripravenú štruktúru pre tvorbu úloh z hľadiska agilného vývoja, a tak sa po viacerých diskusiách vytvorili konvencie pre tvorbu a udržiavanie úloh v TFS. Vzhľadom na charakter projektu a výskumného zamerania zadávateľa úlohy bolo potrebné pristúpiť k určitým úpravám z hľadiska názvoslovia „Používateľských príbehov“. Každéj úlohe na ktorej sa

začalo robiť bolo možné prideliť 3 stavy – „New“ nová, „Active“ aktívna, „Closed“ ukončená. Pre zistené chyby je to o stav „Resolved“ preriešený viac.

Konvencia tvorby, udržiavania a kategorizácie úloh bola dodržiavaná podľa metodiky vytvorenej pre tento účel. Takto vznikli vyššie štruktúry Epics a Features, ktoré predstavovali najväčšie funkčné prvky, ktoré požadoval zadávateľ vo výslednom produkte. Na týchto prvkoch sa postupne pracuje a vytvárajú sa nové podúlohy – „Používateľské príbehy“ (User Stories - UT), ktoré bližšie špecifikujú vlastnosti požadovaných Features. Do každého šprintu sa vybral set UT podľa počtu „story pointov“ a kapacity tímu. Rozdelenie bolo takmer vždy korektné až na ojedinelé prípady keď sa stalo, že sa museli preniesť nedokončené User stories do ďalšieho šprintu, prípadne že člen tímu skončil svoju prácu skôr a do konca šprintu nemal pridelený žiadnu ďalšiu úlohu.

Vedenie agile zasadnutí: Na začiatku každého stand-upu každý člen tímu prezentoval svoj progres za posledný týždeň a predpokladané smerovanie do ďalšieho týždňa. Táto časť bola vždy porovnávaná s agile boardom na TFS. Následne sa v otvorenej diskusii prebrali problémy s jednotlivými úlohami a prichádzalo sa ku konštruktívnym riešeniam. Podľa zistení a nových poznatkov sa podľa potreby vytvorili nové úlohy. Po konzultácií s vedúcim tímu sa navrhli nové „user stories“. Tým sa po ukončení analýzy priradila zložitost pomocou agile pokrových kariet. Na koniec sa vybrali UT pre jednotlivých členov tímu podľa ich kapacity. Podľa určenej zložitosti sa UT rozdelili na ďalšie implementačné úlohy pre priradeného člena. Bližšia špecifikácia úloh bola vždy obsiahnutá v opise úlohy, takže sa o jej postate dozvedel každý člen tímu aj keď na nej momentálne nepracoval.

Súvisiaca metodika: Metodika vytvárania úloh

3.4 Manažment kvality

Pod manažmentom kvality v našom tíme rozumieme všetky činnosti ktoré vedú k optimalizácii pracovných činností a tým ku zvýšeniu kvality výsledného produktu. Patrí sem: uplatňovanie dohodnutých pracovných a komunikačných metodík, rozdelenie kompetencií jednotlivým členom tímu ako aj používanie kvalitných vývojových nástrojov a knižníc. Súčasťou manažmentu kvality je aj pravidelné neformálne vyhodnocovanie plnenia stanovených úloh a výkonnosti jednotlivých členov tímu, na základe ktorého prispôbujeme úlohy členov za cieľom zefektívnenia práce. Asi najdôležitejšou činnosťou je však získavanie spätnej väzby od zadávateľa produktu a vyhodnocovanie jeho spokojnosti. Vzhľadom na špecifickú povahu nášho projektu kedy zadávateľom sú výskumníci z ústavu Joanneum Research v Rakúsku, pravidelné a priame stretnutia so zákazníkom nie sú možné. Napriek tomu pre lepšie vyhodnotenie priebežnej spokojnosti a získanie ďalších

požiadaviek sme sa dohodli na stretnutí so zástupcom z Joanneum Research na január roku 2016. Do tej doby sa musíme v ohľade požiadaviek vo veľkej miere spoliehať na vlastnú intuíciu a formálneho vlastníka produktu v osobe nášho pedagogického vedúceho p. Dr. Benešovej.

3.5 Manažment rizík

Jednou z hlavných činností ktoré sme vykonávali v počiatkoch projektu bola identifikácia rizík ktoré môžu ohroziť projekt a hľadanie vhodných protiopatrení. Každý z členov sa snažil identifikovať najmä riziká týkajúce sa jeho pridelených úloh a následne sme ich spoločne vyhodnotili a navrhli riešenia. Vybrané identifikované riziká:

Riziko: odchod člena tímu

Pravdepodobnosť: stredná

Preventívne opatrenie: rozdelenie kompetencií v tíme tak, aby boli do riešenia každého problému zapojení aspoň dvaja členovia tímu

Akcia na zmiernenie dopadu: pridelenie kompetencií odišlého člena jeho zástupcovi

Riziko: zadávateľ projektu nedodá testovacie dáta

Pravdepodobnosť: vysoká

Preventívne opatrenie: nájdenie dát na webe, vytvorenie syntetických dát

Akcia na zmiernenie dopadu: použitie záložných dát vytvorených v rámci prevencie

Riziko: nezvládneme dokončiť niektorý z dôležitých medzičlánkov, čo znemožní ďalšiu prácu na projekte

Pravdepodobnosť: nízka až stredná

Preventívne opatrenie: plánovanie úloh a dôkladná analýza možností uplatnenia existujúcich komponentov od tretích strán

Akcia na zmiernenie dopadu: použitie existujúcich komponentov nájdených v rámci prevencie

Riziko: nedostupnosť niektorých kľúčových funkcií pre v knižnici PCL

Pravdepodobnosť: nízka

Preventívne opatrenie: prieskum možností iných knižníc pre spracovanie oblakov bodov, napríklad CGAL, MeshLab

Riziko: nedostatočná kvalita implementácie metód kvôli ich veľkému počtu

Pravdepodobnosť: stredná až vysoká

Preventívne opatrenie: sústredenie sa na 2-3 metódy, pričom na každej

pracujú aspoň dvaja členovia tímu

3.6 Manažment integrácie a podpory vývoja

Pre zabezpečenie konzistentnosti počas riadenia tímového projektu s pohľadu jednotlivých častí a zaistenie požadovanej celistvosti modulov systému máme stanovený postup integrácie a podpory vývoja nasledovne:

Jednotlivý členovia tímu sú zodpovedný za ucelené časti tímového projektu ktoré spracúvajú. Majú na starosti ich kvalitu vyhotovenia a tým pádom aj výstup daného modulu a celkové spracovanie. Zodpovedný členovia komunikujú spoločne a aj s integrátorom projektu a podpory vývoja. Tým je dosiahnutá potrebná transparentnosť a lepšia spolupráca. Ďalej integrátor zozbiera od jednotlivých členov ich vytvorené moduly a vytvorí konečnú podobu procesu.

Pre prehľadnú spoluprácu tvorby softvéru, manažovanie práce, úloh a zdieľanie zdrojového kódu sme sa rozhodli použiť TFS - Visual Studio Team Foundation Server 2015, ktorý je jednoducho prepojitelný a synchronizovateľný s Microsoft Visual Studio. Vďaka tomu dokážeme dodržiavať dohodnuté postupy a konvencie, udržiavať kód prehľadný a zároveň všetkým dostupný. Poskytuje jednoduché aktualizovanie zmien v zdrojovom kóde od všetkých členov tímu.

Kvôli potrebe vytvoriť prehľadnú komunikáciu medzi všetkými členmi tímu sme si zvolili aplikáciu Slack, ktorá Umožňuje triediť si jednotlivé problémy do oddelených konverzácií aby sa medzi sebou nemiešali a tak tiež ponúka rôzne možnosti citovania, vytvárania útržkov zdrojového kódu či možnosť nainštalovania na smart zariadenia.

Kapitola 4

Sumarizácie šprintov

4.1 1. šprint

Projekt je postavený na rekonštrukcii 3D scény pomocou rozličných metód. V prvom šprinte sme sa preto sústredili na štúdium literatúry k daným metódam a štúdium pcl knižnice, keďže polovica tímu sa doposiaľ s ňou nestretla. Rovnako bolo potrebné vyriešiť dáta, ich načítavanie z rôznych formátov a najmä ich konverziu do jednej scény, keďže dáta boli tvorené len snímkami z rôznych pohľadov. V neposlednom rade sa riešila vizualizácia vstupných a výstupných dát, rovnako formou štúdia a malej implementácie v podobe pcl vizualizéra.

Po 1. šprinte sme odsúhlasili komunikáciu prostredníctvom komunikačného nástroja Slack. Doposiaľ sme používali na komunikáciu v tíme Facebook chat, avšak väčšina v tíme mala výhrady kvôli tomuto spôsobu. Manažment vytvárania a spravovania úloh v TFS zostal nezmenený, t.j. každý člen si vytvoril svoju vlastnú user-story na ďalší šprint a v nej definoval úlohy.

4.2 2. šprint

V tomto šprinte bola riešená prioritne architektúra celého projektu. Podarilo sa ju úspešne navrhnuť podľa MVC a implementovať. Okrem architektúry jeden člen tímu riešil ešte problém z 1. Šprintu ohľadne konverzie dát z formátu LAS. Úspešne sa mu podarilo prekonvertovať tento formát do PCD. Vytvorili sme grafické rozhranie, pomocou ktorého budeme interagovať s programom. Naďalej sme pokračovali v štúdiu literatúry ohľadne jednotlivých metód, z implementačnej časti sa nám podarilo spraviť výpočet normál a histogram normál. Napokon sme upravili dizajn webového sídla podľa pripomienok vedúcej tímu.

Po konzultáciách na predmete MIS a po jednotnom odsúhlasení sme od nasledovného šprintu, teda šprintu č. 3, zmenili manažment spravovania

úloh projektu v TFS. Doteraz každý člen tímu si pred šprintom vytváral storie na dva týždne a v rámci nej si zariadil úlohy. Všetky požadované úlohy boli rozložené na cca 16 hodín, aby ich stihol splniť. Tento manažment nebol správnym riešením, pretože ako tím, sme jednak ešte dobre nevedeli ohodnocovať jednotlivé úlohy (ohodnotili sme úlohu vysokým číslom a úloha bola ľahká) a na strane druhej tým, že sme mali presne vytvorený počet user-stories na každého člena (člen po dokončení svojej práce nemôže zobrať inú user-story a úlohy v nej zahrnuté a riešiť ich).

Nový manažment správy úloh v TFS poskytne lepší prehľad vykonanej práce a práce, ktorá nás ešte len čaká. Vytvorili sme jednotlivý „epics“, „features“, ktoré predstavujú veľké úlohy a budú otvorené počas celej fázy projektu a tie ďalej členíme do „user-stories“ a malých úloh, ktoré sa riešia počas daných šprintov. Rovnako sme zaviedli vytváranie úloh v rámci každej user-story, ktoré budeme vedieť presne ohodnotiť, či sú vykonané alebo nie, teda tzv. „definition done“ úlohy.

4.3 3. šprint

Tento šprint sme robili už „po novom“. Implementovali sme prototyp growing region metódy, ktorá je jednou zo segmentačných metód. Úspešne sme do vizualizéra pridali ovládanie scény pomocou kláves a rotovanie scény okolo kamery. Mnoho úloh je v tomto čase v rozpracovanom stave. V neposlednom prípade sme dokumentovali súčasný stav projektu. Retrospektíva po treťom šprinte bude vykonaná až na stretnutí tímu.

4.4 4. šprint

Šprint zahŕňal úpravy v architektúre, konkrétne v module grafického používateľského rozhrania, kde sme úspešne implementovali spúšťanie vizualizácie metód a ošetrili chybové stavy. Pre lepšiu interakciu s vizualizovanými metódami sme uvažovali použitie 3D myši SpaceNavigator, ku ktorej sme získali potrebné 3Dconnexion SDK. Podarilo sa nám implementovať ovládanie 3D myšou pomocou knižnice VTK, avšak do projektu táto funkcionálna zatiaľ nebola začlenená, nakoľko ten využíva PCL knižnicu. Napokon sme po spoločnej dohode rozhodli túto problematiku presunúť na letný semester, keďže sa momentálne zaoberáme vývojom jednotlivých metód, kde dáta nie sú natoľko obsiahle, žeby práca s nimi bola náročná na interakciu a prioritne sa musíme zamerať na doladenie architektúry a jej modulov. Pre overenie funkčnosti projektu sme riešili otázku testovania a vytvárania testov. Kvôli štúdiu vytvárania testov a pre mierne komplikácie zatiaľ neboli vytvorené žiadne testy. Okrem architektúry, modulu grafického používateľského rozhrania, problematiky vizualizácie a interakcie a testovania sme naďalej pokračovali v riešení (štúdiu, implementácii) rekonštrukčných metód. Pri

metóde growing region sme zisťovali rovinu z daných bodov v priestore a implementovali sme ju do projektu. Pri metóde č. 4 sme navrhli a implementovali označovanie bodov v oblaku dát a implementovali algoritmus pre nájdenie dominantných oblastí v oblaku dát. Pri metóde č. 1 sme vytvorili extrakciu rezov a implementovali mean shift. V metóde č. 3 sme neurobili žiadny pokrok, keďže osoba, ktorá má pridelenú túto metódu sa zaoberala štúdiom testov.

Migrácia vývojového prostredia Microsoft Visual Studio Community 2015 14.0.23107.0 (VS2015)-> Microsoft Visual Studio Professional 2013 12.0.21005.1(VS2013)

Nakoľko sa vyskytol problém pri kompilovaní knižníc OpenCV verzie 3.0.0 vo vývojovom prostredí VS2015 a navyše niektoré z knižníc OpenCV verzie 3.0.0 si vyžadovali podporu platformy CUDA a zároveň táto ešte nepodporuje VS2015, rozhodli sme sa jednohlasne pre migráciu z VS2015 do VS2013, kde knižnica OpenCV verzie 3.0.0 má plnú podporu.

4.5 5. šprint

V šprinte sme sa zaoberali rozšírením modulu grafického používateľského rozhrania o výber jednotlivých metód rekonštrukcie pomocou checkboxov. Používateľ pri interakcii s aplikáciou má navyše možnosť vidieť pôvodné dáta až potom rekonštruované a pod. Pri implementácii vizualizácie pôdorysu miestnosti z dát sme zaznamenali chybu v projekte, ktorá bude riešená a v prípade vyriešenia odstránená. Pri rekonštrukčnej metóde č. 4 sme pokročili s implementáciou submetódy pre nájdenie dominantných oblastí v histograme pomocou maxím. V rekonštrukčnej metóde č. 2 growing region sme hľadali priesečníky stien, pričom sme našli koeficienty priamky. Implementácia zatiaľ nebola otestovaná pomocou vytvorených automatických testov. Tieto ešte neboli zautomatizované kvôli migračným problémom s vývojovým prostredím.

Technická dokumentácia - Doxygen

Pre potrebu ľahkého dokumentovania kódu projektu sme sa rozhodli zaviesť používanie nástroja Doxygen. Dokumentácia sa generuje na základe komentárov pri jednotlivých úsekoch kódu. Doxygen prepája časti kódu a referencie na dokumentáciu k týmto častiam kódu, čím sa stáva projekt prehľadnejší z dlhodobého hľadiska pre všetkých záujemcov o prácu s projektom.

Kapitola 5

Používané metodiky

názov	opis	autor
Metodika komunikácie	Pravidlá pre používanie aplikácie Slack na komunikáciu v tímoch. Vytváranie kanálov, odpovedanie na správy a pod.	Róber Karásek
Metodika pre evidenciu zmien zdrojového kódu	Návod na správnu evidenciu zmien a vetvenie kódu podľa dohody v tíme.	Róbert Karásek
Metodika evidencie úloh v TFS	Pravidlá pre vytváranie nových úloh v nástroji TFS. Ktoré polia treba vyplniť a ako, ako správne nastaviť všetky parametre.	Lukáš Hudec
Metodika písania zdrojového kódu	Konvencie pre písanie zdrojového kódu v C++ v rámci projektu.	Róbert Karásek
Metodika pre technickú dokumentáciu	Pravidlá písania technickej dokumentácie pomocou Doxygen	Róbert Karásek

Tabuľka 5.1: Zoznam používaných metodík s ich opisom

Kapitola 6

Globálna retrospektíva

Ako tím sme sa posunuli počas riešenia projektu v zimnom semestri dopredu vo viacerých oblastiach. Ako prvý problém sme riešili komunikáciu v tíme. Na komunikovanie v rámci tímu sme prestali používať komunikačný nástroj Facebook a začali sme komunikovať prostredníctvom nástroja Slack, čím sme oddelili komunikačný pracovný kanál od iných komunikačných kanálov (zábava, priatelia, ...), ktoré výrazne ovplňovali úspešnosť dorozumievania v tíme pri riešení pracovných záležitostí.

Ďalším výrazným krokom vpred bola zmena manažmentu evidencie úloh v TFS (Team Foundation Server). Zmena nastala po 2. šprinte, dovtedy každý člen evidoval úlohy v rámci svojej user-story, ktorú si dopredu na stretnutí vytvoril. Časová náročnosť úloh bola rozložená na približne 16 hodín. Avšak po odpracovaní svojich úloh člen môže robiť na ďalších úlohách, v prípade, že má definované user-story a malé úlohy. Týmto spôsobom sa to nedalo. Aj ako začiatočníci sme nevedeli dobre ohodnocovať úlohy (ako začiatočníci všeob. v oblasti rekonštrukcie objektov) a mohlo sa stať, že sme ohodnotili úlohu vysokým číslom a úloha bola ľahko riešiteľná (nikto neoponoval). Nový manažment predstavoval definovanie množiny veľkých úloh v podobe “epics”, “features”, ktoré budú otvorené počas celej fázy riešenia projektu a nebudú sa meniť. Tieto ďalej členíme do “user- stories”, čo predstavujú väčšie úlohy riešené v rámci šprintu. Snažili sme sa zvoliť čo najväčšiu množinu týchto úloh, aby každý člen si potom mohol v šprintoch vyberať riešiť danú user-story a definovať k nej malé úlohy. Rovnako sme zaviedli vytváranie úloh v rámci každej user-story, ktoré budeme vedieť presne ohodnotiť, či sú vykonané alebo nie, teda tzv. „definition of done“ úlohy.

V ďalšej fáze projektu sme narazili na problém pri kompilovaní knižníc OpenCV verzie 3.0.0 vo vývojovom prostredí VS2015 a navyše niektoré z knižníc OpenCV verzie 3.0.0 si vyžadovali podporu platformy CUDA a zároveň táto ešte nepodporuje VS2015, preto sme sa rozhodli jednohlasne pre migráciu z VS2015 do VS2013, kde knižnica OpenCV verzie 3.0.0 má plnú podporu.

Pre potrebu ľahkého dokumentovania kódu projektu sme sa rozhodli zaviesť používanie nástroja Doxygen. Dokumentácia sa generuje na základe komentárov pri jednotlivých úsekoch kódu. Doxygen prepája časti kódu a referencie na dokumentáciu k týmto častiam kódu, čím sa stáva projekt prehľadnejší z dlhodobého hľadiska pre všetkých záujemcov o prácu s projektom.

Pre správnosť kódu v poslednom období plánujeme zaviesť testovanie. Testovanie kódu by sa malo uskutočňovať prostredníctvom Google testov. Testovanie pravdepodobne vykonáme až budúci semester nakoľko doteraz neboli naimplementované žiadne väčšie funkčné celky.

Kapitola 7

Zoznam kompetencií tímu

Každý v tíme zohráva dôležitú rolu, hlavné oblasti zodpovednosti sme si rozdelili nasledovne:

- manažér architektúry - Lukáš Hudec
- manažér vývoja - Róbert Birkus
- projektový integrátor - Martin Jurík
- biznis manažér - Michal Löffler
- test manažér - Róbert Karásek
- manažér propagácie - Michal Korbel
- manažér dokumentácie - Katarína Janečková

Viac o úlohách a zodpovednostiach členov tímu sa dočítate v kapitole 2.1.

Kapitola 8

Metodiky

8.1 Metodika komunikácie

Komunikácia

Nami zvolený systém pre komunikáciu medzi členmi tímu je Slack.

- Možnosť vytvoriť kanál ma každý člen tímu
- Do komunikačného kanála treba písať vždy k danej téme, ak neviem kde patrí konkrétna otázka, treba sa opýtať v general chate alebo konkrétnej osoby, na ktorú je smerovaná.
- Používať slack anotácie (napr. vloženie kódu do ‘code‘)
- Slack kontrolovať denne

Zdieľané dokumenty

Nami zvolený systém pre zdieľanie dokumentov je Google Drive.

- Štruktúru treba zachovávať ako je, prípadne požiadať o zmenu vedúceho tímu.
- Je dovolené meniť dokumenty ak sa nájdu v nich nezrovnalosti alebo presunúť súbor do správneho priečinka
- Nahrať súbory vždy načas

8.2 Metodika písania zdrojového kódu

Header Files

- **každý .cc súbor musí mať "pridružený" .h súbor**, okrem unit testov a malých .cc, ktoré obsahujú iba main()
- na konci .h
- **súbory, ktoré začleňujú text, ale nie sú hlavičkovými tak na konci .inc** (napr. ak sa používajú na viacerých miestach v kóde, alebo sú určené pre špecifickú platformu)
- každý by mal obsahovať "header guards"(nižšie), a mal by byť prepojený s ďalšími headermi
- ak je v headeri deklarovaná template alebo inline funkcia, tak definície týchto konštruktorov musia byť v každom .cc súbore, ktorý ich používa
- (As an exception, a function template that is explicitly instantiated for all relevant sets of template arguments, or that is a private member of a class, may be defined in the only .cc file that instantiates the template.)

The #define Guard

- každý header musí obsahovať #define guard
- formát <PROJECT>_<PATH>_<FILE>_H_
 - **napr. súbor v foo/src/bar/baz.h by mal vyzerat' nasledovne**

```
#ifndef FOO_BAR_BAZ_H_
#define FOO_BAR_BAZ_H_
...
#endif // FOO_BAR_BAZ_H_
```

Forward Declarations

- ako predísť zbytočným #includes
- [Pros / Cons](#)
- ak používam funkciu zadefinovanú v headeri, vždy #include konkrétny header
- pri použití class template, pre istotu #include jeho header
- ak používam bežnú triedu, spoliehať sa na poprednú deklaráciu je OK ale treba vždy skontrolovať
- **nenahrádzať datové členy s ukazovateľmi len preto aby sa zabránilo #include**

Inline Functions

- **iba funkcie, ktoré majú malý počet riadkov (napr. 10 a menej)**
- kompilátor zavolá funkcie inline, teda neprejde zvyčajným volacím mechanizmom
- [Pros / Cons](#)
- neodporúča sa aby inline funkcia obsahovala cykly alebo switch (može ak sa nikdy nebude vykonávať), nemali by byť ani rekurzívne

Function Parameter Ordering

- **najprv vstupy, potom výstupy** (aj keď pridáme nový vstup, tak na začiatok s ním)

Names and Order of Includes

- všetko od source zložky teda
 - google-awesome-project/src/base/logging.h → #include "base/logging.h"
- ak *dir/foo.cc* alebo *dir/foo_test.cc* má za úlohu implementovať alebo testovať *dir2/foo2.h*, **poradie includov** bude nasledovné
 - **dir2/foo2.h.**
 - **C system files.**
 - **C++ system files.**
 - **Other libraries' .h files.**
 - **Your project's .h files.**
 - každá sekcia zoradená podľa abecedy

- **Příklad**

- google-awesome-project/src/foo/internal/fooserver.cc
- #include "foo/server/fooserver.h"

```
#include <sys/types.h>
#include <unistd.h>
#include <hash_map>
#include <vector>
```

```
#include "base/basictypes.h"
#include "base/commandlineflags.h"
#include "foo/server/bar.h"
```

- **ak obsahuje podmienky, tak až nakoniec**
- #include "foo/public/fooserver.h"

```
#include "base/port.h" // For LANG_CXX11.
```

```
#ifdef LANG_CXX11
#include <initializer_list>
#endif // LANG_CXX11
```

Scoping

Namespaces

- nepomenované namespace je v .cc súboroch podporované
- ak chcem pomenovať tak v kombinácii s menom projektu a jeho path
- nepoužívať inline namespace
- (Namespaces subdivide the global scope into distinct, named scopes, and so are useful for preventing name collisions in the global scope.)
- [Pros / Cons](#)

```
- namespace {                      // This is in a .cc file.  
  // The content of a namespace is not indented.  
  //  
  // This function is guaranteed not to generate a colliding symbol  
  // with other symbols at link time, and is only visible to  
  // callers in this .cc file.  
  bool UpdateInternals(Frobber* f, int newval) {  
    ...  
  }  
  
} // namespace
```

```
- // In the .h file  
namespace mynamespace {  
  
  // All declarations are within the namespace scope.  
  // Notice the lack of indentation.  
  class MyClass {  
  public:  
    ...  
    void Foo();  
  };  
  
} // namespace mynamespace
```

- **// In the .cc file**
`namespace mynamespace {`

`// Definition of functions is within scope of the namespace.`
`void MyClass::Foo() {`
`...`
`}`

`} // namespace mynamespace`
- `#include "a.h"`

`DEFINE_bool(someflag, false, "dummy flag");`

`class C; // Forward declaration of class C in the global namespace.`
`namespace a { class A; } // Forward declaration of a::A.`

`namespace b {`

`...code for b... // Code goes against the left margin.`

`} // namespace b`
- **nič nedeklarovať v namespace std ani popredne deklarované classy z standard lib**

Nested Classes

- nerobiť vnorené triedy ak nie sú súčasťou interface
- používať nonmember funkcie s namespace alebo static member funkcie namiesto globálnych funkcií

Local Variables

- `int i;`
`i = f();`
- `int j = g();`
- `vector<int> v;`
`v.push_back(1);`
`v.push_back(2);`
- `vector<int> v = {1, 2};`
- [ďalšie ukážky](#)

Static and Global Variables

- **statické alebo globálne premenné typov tried sú zakázané**
- [ďalšie pravidlá](#)

Classes

- nikdy nevolať konštruktorom virtuálne funkcie
- C++ keyword **explicit** for **constructors callable with one argument**.
- používať delegovanie a dedičnosť konštruktorov, keď znižujú duplicitu kódu ([viac](#))
- poradie **public: pred protected: pred private: , metódy pred dátovými členmi (premenne)**
- Typedefs and Enums
- Constants (static const data members)
- Constructors
- Destructor
- Methods, including static methods
- Data Members (except static const data members)
- ak ma funkcia viac ako 40 riadkov = rozmýšľať či sa nedá rozdeliť

General Naming Rules

premenne pomenovávať zrozumiteľne, slová oddeľovať podčiarkovníkom

int price_count_reader; // No abbreviation.

int num_errors; // "num" is a widespread convention.

int num_dns_connections; // Most people know what "DNS"

mená súborov taktiež s _ alebo -

my_useful_class.cc

my-useful-class.cc

myusefulclass.cc

myusefulclass_test.cc

mená metód, tried.. CamelCase
static Pool<UrlTableProperties> pool;*

globálne s prefixom g_ napríklad

funkcie = CamelCase

enum CamelCase

makrá VELKYM_PISMOM

komentáre ako sa dohodneme ale asi klasicky
/ nad metódami */*
//v riadku

alebo všetko jedným štýlom

// TODO(kl@gmail.com): Use a "" here for concatenation operator.
// TODO(Zeke) change this to use relations.

môže sa použiť meno, kto by mal fixnúť ale môže to samozrejme aj niekto iný

komentáre písať zrozumiteľne bez spelling chýb

max dĺžka riadka 80 písmen
nepoužívať taby :(iba space a 2x => nastaviť aby po stlačení tab spravil 2x space

bool retVal = DoSomething(averyveryveryverylongargument1,
argument2, argument3);
argumenty funkcií pri volaní posúvať vždy na úroveň zátvorky

```
if (...) {  
  ...  
  ...  
  if (...) {  
    DoSomething(  
      argument1, argument2, // 4 space indent  
      argument3, argument4);  
  }  
}
```

PODMIENKY

```
if (condition) { // no spaces inside parentheses
    ... // 2 space indent.
} else if (...) { // The else goes on the same line as the closing brace.
    ...
} else {
    ...
}
```

```
if (x == kFoo) return new Foo();
```

jedno riadkový for

```
for (int i = 0; i < kSomeNumber; ++i)
    printf("I love you\n");
```

alebo

```
for (int i = 0; i < kSomeNumber; ++i) {
    printf("I take it back\n");
}
```

-viac podmienok

```
if (this_one_thing > this_other_thing &&
    a_third_thing == a_fourth_thing &&
    yet_another && last_one) {
    ...
}
```

POINTRE

```
// These are fine, space preceding.
char *c;
const string &str;
```

```
// These are fine, space following.
char* c; // but remember to do "char* c, *d, *e, ...;"!
const string& str;
```

```
RETURN
return result;           // No parentheses in the simple case.
// Parentheses OK to make a complex expression more readable.
return (some_long_condition &&
        another_condition);

// Good - directives at beginning of line
if (lopsided_score) {
#if DISASTER_PENDING      // Correct -- Starts at beginning of line
    DropEverything();
# if NOTIFY               // OK but not required -- Spaces after #
    NotifyClient();
# endif
#endif
    BackToNormal();
}

NAMESPACE
namespace {

void foo() { // Correct. No extra indentation within namespace.
    ...
}

} // namespace
```

každé namespace na novom riadku bez zač. posunu

OPERÁTORY

môže byť aj $x*y$ ale aj $x * y$, tak najlepšie sa bude dohodnúť (asi s medzerami)

MINIMALIZOVAT vertikálny space

HLAVNE BYŤ KONZISTENTNÝ

Tipy

[cpplint](#) nájde chyby vo formátovaní a pod.

nepoužívať exceptions RTTI

`static_cast<>()`. Do not use other cast formats like `int y = (int)x;` or `int y = int(x);`

Use prefix form (`++i`) of the increment and decrement operators with iterators and other template objects.

8.3 Metodika pre technickú dokumentáciu

Na vygenerovanie technickej dokumentácie sa používa Doxygen(verzia 1.8.10).

Inštalácia sa nachádza na <http://www.stack.nl/~dimitri/doxygen/download.html>.

- Každý autor zdrojového kódu je nútený komentovať a napísať hlavičku podľa <http://www.stack.nl/~dimitri/doxygen/manual/docblocks.html>
- Komentáre musia byť stručné a výstižné
- Treba komentovať len potrebné časti kódu
- Netreba spomínať znova zdrojový kód v komentári
- Nezakomentovávať zdrojový kód!!!
- Konfigurácia je vyobrazená na obrázkoch
- Už vygenerovanú dokumentáciu je potrebné ukladať na Drive, rozdeliť do priečinkov podľa formátov

8.4 Metodika evidencie úloh v TFS

Prístup k nástroju je možný cez portál tfs.fiit.stuba.sk, alebo s použitím pluginu vo Visual Studio 2015/2013 po presune do tohto vývojového prostredia. Postup vo vyplňaní jednotlivých polí je rovnaký a výstup musí spĺňať metodikou zavedenú normu. Pre polia, ktoré obsahujú všetky udalosti platia rovnaké pravidlá.

Momentálne existujú 3 najvyššie celky - Epic, Feature a User Story. Epic a Feature sa vytvoria na naplánujú s vlastníkom produktu a predstavujú hlavné celky požiadaviek. Ďalej sa už len pridávajú User Story do Feature podľa zamerania požiadavky a úlohy s ňou spojenej. Pri vytváraní úlohy sa riadíme najvyšším celkom - User story, jednotlivé úlohy pre členov tímu sú označené ako Task a Bug

User story:

- definovaný používateľský príbeh, abstrahovaná menšia časť požiadavky vlastníka produktu - implementačná oblasť problému
- má svojho zodpovedného vedúceho - ten je pridelený na stretnutí tímu, podľa osobnej žiadosti alebo určený podľa skúseností

- ak nie je určený, user story vytvorí team leader a úlohu nastaví podľa dohodnutých kritérií, zodpovedný vedúci bude následne určený hlasovaním v tíme

Task:

- jednotlivé úlohy používateľského príbehu
- ak sú jasné hneď na stretnutí, vytvorí ich zodpovedný vedúci a pridelí členovia tímu si ich rozdelia podľa vlastného záujmu tak, aby ich stihli za šprint vypracovať

Bug:

- označenie pre úlohu zaoberajúcu sa chybou v programe alebo dokumentácií
- jej vytvorenie má na zodpovednosť ten člen, ktorý chybu odhalil, ale riešenie môže po konzultácii so zodpovedným vedúcim user story ponechať na iného člena - zodpovedného za oblasť, v ktorej bola chyba odhalená

Vypĺňanie polí úloh

Celá dokumentácia a správa projektu je písaná výhradne v slovenčine s použitím diakritiky okrem explicitných výnimiek vyznačených v meto-dike(pokiaľ nie sú zistené žiadne problémy).

- Meno
 - začiatkové písmeno veľkým, ostatné podľa pravidiel slovenského jazyka
 - stručný a výstižný
- Tagy
 - využitie všeobecne známych a výstižných skratiek
 - bez diakritiky
 - môže byť nastavených viac, podľa potreby
 - v prípade zmeny opísať všetky zmeny, ktoré nastali
- Detail, Description
 - detailnejší popis úlohy, z ktorého bude každému jasné o čo v danej úlohe ide
 - spomenúť možné zmeny v úlohe
- Stav

- keď je úloha vytvorená, je označená ako “New” (prednastavené, nemožno zmeniť)
- Pridadenie (Assigned to)
 - vyplnené podľa typu úlohy a stavu, v ktorom sa nachádza
- Priorita
 - defaultne nastavená na hodnotu 2
 - upravená podľa dohody na stretnutí a momentálnych požiadaviek
- Iterácia
 - nastavuje sa podľa toho do ktorého šprintu daná úloha patri - táto hodnota musí byť vyplnená korektne, inak úloha nie je správne zobrazená v tabuľke
- Effort - vypĺňajú sa 3 hodnoty - “Original estimate”, “Remaining” a “Completed”
 - Original estimate - odhadovaný čas, ktorý riešenie zaberie; nastavená hodnota od ktorej sa počíta burndown chart
 - Remaining - original estimate - completed (odpracovaný čas) = zostávajúci čas; podľa tejto hodnoty sa počíta pokles burndown chart-u
 - Completed - riešiteľ si zadá koľko času na riešení tejto úlohy strávil

Stav úlohy závisí od toho či je to “User story” alebo “Task”.

Hlavné stavy:

- New - nastavená pokiaľ je priamo vytvorená
- Active - keď riešiteľ začne pracovať na úlohe zmení jej stav na aktívnu “User stories”, na ktorých sa pracuje majú tento stav od začiatku riešenia aj keď sa práve nerieši žiadna z ich taskov
- Resolved - “User story”, ktorej riešenie je hotové a prešlo aj testami (task item tento stav nemá)
- Closed - ukončená úloha, na ktorej sa predpokladá už viac nebude pracovať

Šprinty

Šprinty sa vytvárajú s dvojtýždňovým trvaním, najčastejšie od agile stretnutia po najbližšie stretnutie. V prípade technických alebo problémov v manažmente je možné tieto termíny posunúť - najviac však o 2 dni. Ak

šprint obsahuje úlohy k doplneniu dokumentácie, tieto úlohy sa evidujú v jednej špeciálnej User Story, vzhľadom na to, že dokumentáciu potrebuje aj vlastník produktu, jej štandardné ohodnotenie je 5 SP.

Šprinty vytvára a spravuje biznis manažér, ktorý zastupuje scrum mastera, prípadne tím líder, ktorý má prehľad o dianí v tíme a progrese práce na jednotlivých úlohách.

Vytvorenie úloh Úlohy sa plánujú na agile zasadnutí tímu, na ktorom sa plánuje nový šprint. Na začiatku každého stretnutia sa vykoná stand-up, na ktorom každý člen tímu opíše svoj progres za posledný týždeň, prípadné problémy. Ak sa vyskytnú problémy, ostatní členovia tímu produktívne prispedia s pomocou k riešeniu daného problému.

V prípade stretnutia s plánovaním šprintu, po stand-up scrum master vedie plánovanie nových úloh na nasledujúci šprint. Ak vznikli nové požiadavky vlastníka produktu, najprv sa určí ich priorita a podľa jej výšky sa vyberajú požiadavky s backlogu v kombinácii s novými požiadavkami. Vybrané požiadavky sa vložia do backlogu šprintu a ostatné sa uložia do backlogu a evidujú sa k Epicom a Featurám podľa tematiky. Ak majú vzniknúť nové úlohy, tím sa riadi vyššie spomenutým postupom vyplnenia úloh. Vybrané úlohy sa ohodnotia pokrom a prebehne analýza, ktorá má pomôcť zodpovednému riešiteľovi vytvoriť postup práce na úlohe - taktiež evidovať svoju prácu v TFS ako Task Items. Bližšie informácie k správe úloh podľa agilných metód sú v dokumente “Manažment plánovania projektu”.

Pokiaľ sú tieto úlohy/Task Items jasné počas stretnutia - zadáva ich do TFS team leader, zodpovední členovia tímu len vyplnia Description. V opačnom prípade si ich vytvárajú členovia postupne ako podľa vlastnej analýzy zistia, aké úlohy ich práca bude obnášať.

Zápisnice

Písanie zápisov zo stretnutia sa eviduje rovnako ako ostatné úlohy a ich zaradenie spadá do každého šprintu avšak bez priradenia k User Story. Zápisy majú svoju formu a predpokladaný obsah, čo je určené šablónou, ktorá sa nachádza v priečinku na Google Drive tímu.

- Názov úlohy je zadaný v tvare - “Zápisnica ‘RRRRMMDD’” (napr. Zápisnica 20151001)
- Description - obsahuje text zápisnice (stačí plain text)
- Attachements - formátovaná pdf verzia zápisnice

8.5 Metodika pre evidenciu zmien zdrojového kódu

Používa sa GIT integrovaný vo Visual Studiu.

- Commit musí obsahovať stručný opis, musí byť pri ňom autor.
- Commit je dovolený len pre skompilovateľnú verziu aby sa zbytočne neobmedzovali ostatní členovia tímu.
- Ohľadne verziovania je povolené použiť všetky možnosti ale zbytočne nekomplikovať verzie.
- Ak sú nezrovnalosti ohľadne commitov, treba kontaktovať integrátora

Kapitola 9

Export evidencie úloh

Tabuľka 9.1: Úlohy pred začatím šprintov

Title	Assigned to	Description	Tags
Zápisnica č. 1 (20150924)	Bc. Katarína Janeckova	Vytvorenie zápisnice z 1. stretnutia	doku
Štúdium literatúry [1]	Bc. Katarína Janeckova	Odporúčaná literatúra [1] k zadaniu témy projektu. Geometric Abstraction from Noisy Image-Based 3D Reconstructions	analýza
Štúdium možností OpenCV 3.0.0	Bc. Lukas Hudec	Zistenie možností knižnice OpenCV z hľadiska 3D Rekonštrukcie a 3D registrácie. Spísanie metodík pre vytváranie úloh v TFS. Metodiky budú dostupné na google drive v prílohu tímového projektu. Predpokladajú sa ešte zmeny, po odsúhlasení ostatnými členmi tímu	analýza
Metodiká tvorby úloh	Bc. Lukas Hudec	Analýza a navrhnutie rozhrania pre načítanie a správu datasetov - vstupov programu. Výsledok analýzy a návrh implementácie je priložený v prílohe.	analýza; dev
Analýza spracovania dát	Bc. Lukas Hudec	Vytvorenie zápisnice z 2. stretnutia	doku
Zápisnica č. 2 (20151001)	Bc. Martin Jurik	Analýzovanie spracovania dát, návrh hlavíciek tried	analýza; dev
Analýzovanie spracovania dát	Bc. Martin Jurik	Štúdium možností knižnice ITK	doku
Zápisnica č. 3 (20151008)	Bc. Martin Jurik		ITK
Štúdium knižnice ITK	Bc. Michal Korběl	Analýza možností knižnice PCL, prípadných pre použitie metódy Growing region.	
Implementácia webu (HTML + CSS + JS)	Bc. Michal Korběl	Nahratie všetkých súborov tvoriacich stránku do adresára /var/www/html na našom virtuálnom stroji team05-15-studenti.fkit.stuba.sk. Na nahrávanie je potrebné použiť protokol SFTP (port 22), pretože štandardný FTP port 21 nie je na stroji povolený.	
Štúdium metódy growing region	Bc. Michal Korběl		
Nahratie stránky na server	Bc. Michal Korběl		
Zápisnica č. 7 (20151116)	Bc. Michal Korběl		
Zápisnica č. 6 (20151029)	Bc. Michal Löffler	Vytvorenie zápisnice zo 6. stretnutia.	doku
Štúdium možností vizualizácie	Bc. Michal Löffler	Prieskum dostupných knižníc na vizualizáciu dát vo forme oblaku bodov a zhrnutie do dokumentu.	
Inštalácia virtuálneho servera	Bc. Michal Löffler	1. Pripojenie cez VNC pomocou pridelených prístupových údajov 2. Inštalácia konfigurácia OS Ubuntu Server 14.04.3 (LAMP, sshd) 3. Vytvorenie používateľských kont pre R. Karáska, M. Korbela	
Analýza spracovania dát	Bc. Robert Birkus	Analýza a návrh rozhrania pre spracovanie dát (Načítanie dát, konverzia dát a ukladanie dát)	analýza; dev
Štúdium knižnice Point Cloud Library	Bc. Robert Birkus	Rozbehnutie PCL knižnice vo VS 2015, vyskúšanie features knižnice a zistenie dátových typov	analýza; PCL
Zápisnica č. 4 (20151015)	Bc. Robert Birkus	Vytvorenie zápisnice zo 4. stretnutia.	doku
Zápisnica č. 5 (20151022)	Bc. Robert Karasek	Vytvorenie zápisnice z 5. stretnutia. Vytvorenie poznámok z dokumentu http://google-styleguide.googlecode.com/svn/trunk/cppguide.html	doku
C++ conventions	Bc. Robert Karasek		doku
Štúdium literatúry [2]	Bc. Robert Karasek		analýza

Tabuľka 9.2: Úlohy z prvého šprintu

Title	Assigned to	Iteration	Description	Tags
Analýza metód horizontálnych rezov	Bc. Katarína Janecková	Sprint 1	Analýzovať: - spôsob určenia a extrakcie rezu - projekcia bodov s určitou normálou na z-os	analýza
Analýza metód mean shift	Bc. Katarína Janecková	Sprint 1	Analýzovať metódu mean shift, spôsob jej využitia a implementáciu v knižniciach.	analýza
Načítanie súboru formátu PLY	Bc. Lukas Hudec	Sprint 1	PointCloud data je možné ukladať v rôznych typoch súborov a od toho sa odvíja aj formát akým sú uložené. Jedným možným spôsobom je načítavať dáta z formátu PLY (polygón)	
Načítanie súboru formátu OBJ	Bc. Lukas Hudec	Sprint 1	Ďalším možným formátom pre uloženie dát je formát OBJ (wavefront) knižnica pcd obsahuje funkcionality pre načítanie aj tohto formátu.	
Analýza dát a konfiguračných súborov	Bc. Lukas Hudec	Sprint 1	určenie pozície kamery v konfiguračnom súbore alebo hlavíčke jednotlivých načítavaných súborov.	analýza
Implementácia načítania súborov	Bc. Lukas Hudec	Sprint 1	Implementácia načítavacieho rozhrania s možnosťou načítať "multiple files" a automatické vyhľadanie konfiguračného súboru.	dev
Implementácia kombinácie snímok scény	Bc. Lukas Hudec	Sprint 1	Kombinácia snímok do jednej karteziánskej sústavy podľa určenia pozície kamery z konfiguračného súboru, alebo meta dát - hlavíčky - súborov obsahujúce jednotlivé snímky	dev
Testovanie a doimplementácia pre rôzne vstupy	Bc. Lukas Hudec	Sprint 1	Ako bolo spomenuté existujú rôzne typy dát a rôzne vstupy - všetky majú ale spoločnú podstatu - pozícia kamery a súradnice snímky - je potrebné preto docieľiť aby sa problém s načítaním týchto dát zvošedbecnil a výstup zjednotil.	dev, test
Načítanie dát formátu PCD	Bc. Martin Jurík	Sprint 1	Vyhľadanie informácií a načítanie dát formátu PCD, pokus o jednoduché zobrazenie a nastavenie zobrazených bodov do kamery	dev
Vyhľadanie informácií o formáte LAS	Bc. Martin Jurík	Sprint 1	Vyhľadanie informácií o formáte LAS, analýza štruktúry uložených dát, možnosti jeho načítania a konverzie do použiteľného formátu	dev
Načítanie dát vo formáte LAS	Bc. Martin Jurík	Sprint 1	Načítanie point cloud-ovo formate LAS a testovanie	dev
Štúdium PCL funkcií vhodných pre GR, výpočet normál bodov pomocou PCL	Bc. Michal Korbel	Sprint 1		
Vytvorenie hrubého prototypu modulu	Bc. Michal Korbel	Sprint 1	Štúdium triedy PCLVisualizer obsiahnutej v knižnici PCL	
Štúdium PCLVisualizer	Bc. Michal Löffler	Sprint 1	Štúdium možnosti vizualizácie vstupných - point cloud dát	
Vizualizácia vstupných dát	Bc. Michal Löffler	Sprint 1	Štúdium možnosti vizualizácie vstupných - point cloud/mesh dát	
Vizualizácia výstupných dát	Bc. Michal Löffler	Sprint 1	Štúdium možnosti vizualizácie výstupných - point cloud/mesh dát	
Štúdium formátu Autodesk DXF	Bc. Michal Löffler	Sprint 1	Štúdium v praxi rozšíreného parametrickeho formátu Autodesk DXF (Drawing exchange Format), ktorý sa používa na zakresovanie plánov budov, príp. strojárskych a elektrotechnických výkresov.	
Implementácia intuitívnejšieho spôsobu ovládania pohybu kamery	Bc. Michal Löffler	Sprint 1		
Analýza a hľadanie vhodného histogramu normál	Bc. Robert Birkus	Sprint 1	Pre implementáciu histogramu je potrebné najprv zanalýzovať možné spôsoby výpočtu normál a následného návrhu štruktúry histogramu normál.	analýza, design, histogram, normal
Výpočet normál point cloud-u pomocou	Bc. Robert Birkus	Sprint 1	Cieľom tejto úlohy je naimplementovať výpočet normál point cloudu pomocou PCL funkcií.	dev, normal, PCL
Vytvorenie histogramu normál	Bc. Robert Birkus	Sprint 1	Cieľom je implementácia histogramu normál podľa vypočítaných normál z point cloudu	dev, histogram, normal
Zistenie smeru stien z point cloud-u pomocou histogramu	Bc. Robert Birkus	Sprint 1	Cieľom tejto úlohy je pomocou dominantných oblastí v histograme normál zistiť smer stien v nasmerovanej miestnosti (point cloud).	analýza, design, dev, histogram, normal
Analýza Data Collection and	Bc. Robert Karasek	Sprint 1		
Analýza Inverse CSG	Bc. Robert Karasek	Sprint 1		
Analýza Reconstructing 2D CSG Models	Bc. Robert Karasek	Sprint 1		
Analýza Hough transformation	Bc. Robert Karasek	Sprint 1		
Implementácia Hough transformation	Bc. Robert Karasek	Sprint 1		
Zoznámenie sa s knižnicou a nájdenie vhodných funkcií pre Furukawu	Bc. Robert Karasek	Sprint 1		

KAPITOLA 9. EXPORT EVIDENCIE ÚLOH

Tabuľka 9.3: Úlohy z druhého šprintu

Title	Assigned to	Iteration	Description	Tags
Práca s normálami	Bc. Katarína Janecková	\Sprint 2	- prehľad metód v knižniciach - analýza a pochopenie - spolupráca s RB	analýza
Ortogonálna projekcia	Bc. Katarína Janecková	\Sprint 2	- analýza a pochopenie - prehľad implementácie v knižniciach - edit: projekcia bodov na os z	analýza
Mean shift	Bc. Katarína Janecková	\Sprint 2	- analýza a pochopenie metód - prehľad o implementácii v knižniciach	analýza
Implementácia využitia normál	Bc. Katarína Janecková	\Sprint 2	- implementácia metód pracujúcich s normálami	implementácia
Implementácia projekcie	Bc. Katarína Janecková	\Sprint 2	Premietnutie bodov na os z.	implementácia
Implementácia metódy mean shift	Bc. Katarína Janecková	\Sprint 2	Použití metód z PCL alebo OpenCV aplikujúce mean shift v našom projekte.	dev; implementácia
Implementácia architektúry	Bc. Lukas Hudec	\Sprint 2	Podľa architektonického návrhu projektu z posledného zasadnutia tímu je potrebné vytvoriť nový	dev; implementácia
Návrh architektúry	Bc. Lukas Hudec	\Sprint 2	ktoré budú potrebné pre oo prístup k riešeniu a implementácii aplikácie.	analýza; design; doku
Vytvorenie diagramov	Bc. Lukas Hudec	\Sprint 2	Pre čitateľnosť a vyjadrenie návrhu architektúry sú potrebné UML diagramy.	design; doku
Implementácia Base class	Bc. Lukas Hudec	\Sprint 2	Najvhodnejšie pre naše použítie budú pravdepodobne "class diagramy".	dev; implementácia
Vytvorenie testovacieho datasetu	Bc. Lukas Hudec	\Sprint 2	Hlavná trieda, resp. "abstraktná trieda", ktorá určí štruktúru tried obsahujúcich metódy a bude slúžiť ako "interface" medzi načítaním dát a vizualizáciou spracovaných dát.	dev; implementácia; test
Úprava triedy načítania a správy dát	Bc. Lukas Hudec	\Sprint 2	Existujúci a 100% funkčný a použiteľný dataset je príliš zložitý a rozsiahly, čo je pre potreby vývoja a testovania aplikácie zbytočne priveľa. Z tohto dôvodu je potrebné z tohto datasetu (apartmán) vybrať približenie a zdokonalenie objektovo orientovaného prístupu je potrebné zmeniť ráz metód a štruktúry triedy.	dev; implementácia
Doríšenie LAS	Bc. Martin Jurík	\Sprint 2	Táto trieda a jej instance budú obsahovať objekty metód 3D rekonštrukcie a pracovať nad dátami, ktoré	dev; implementácia
Konverzia LAS->PCD	Bc. Martin Jurík	\Sprint 2		
Debugovanie používaných libraries	Bc. Martin Jurík	\Sprint 2		
Vytvorenie prototypu Grafického rozhrania	Bc. Martin Jurík	\Sprint 2	Vytvorenie jednoduchého grafického rozhrania pre testovacie účely aplikácie.	
Analýza indexov	Bc. Michal Korbel	\Sprint 2	Načítanie súborov - multiselect	
Test example rozne PCD subory	Bc. Michal Korbel	\Sprint 2	Spustenie jednotlivých metód	
Test example rozne thresholds	Bc. Michal Korbel	\Sprint 2	Prepínanie možnosti výstupu	
Komplet uprava webu	Bc. Michal Korbel	\Sprint 2		
Analýza možnosti knižnice CGAL	Bc. Michal Löffler	\Sprint 2	Zistiť aké sú výhody knižnice CGAL oproti PCL,	
Analýza možnosti manipulácie s kamerou	Bc. Michal Löffler	\Sprint 2	najšf komponenty využiteľné pre náš projekt	
Uzamknutie osy kamery	Bc. Michal Löffler	\Sprint 2	Problém: vo východnom stave sa v PCL/Visualizeri kamera otáča okolo nejakého bodu pred kamerou (začiatok súradnicovej sústavy?), čo je trochu neprirodzené.	
	Bc. Michal Löffler	\Sprint 2	Problém: PCL/visualizer defaultne otáča kamerou v smere všetkých osí X,Y,Z	
	Bc. Michal Löffler	\Sprint 2	Riešenie: uzamknutie rotovania kamery okolo osí z	

KAPITOLA 9. EXPORT EVIDENCIE ÚLOH

Analýza princípov interakcie s	Bc. Michal Löffler	Sprint 2	Štúdium tried InteractorStyle, ktoré definujú spôsob interpretácie stlačenia kláves a hýbania myšou.	dev; histogram; normal
Implementácia histogramu normal	Bc. Robert Birkus	Sprint 2	Po analýze a návrhu je potrebné naimplementovať histogram normal.	analýza; normal
Analýza štruktúry pd::Normal	Bc. Robert Birkus	Sprint 2	Cieľom je zanalyzovať štruktúru pd::Normal, ako reprezentuje normálny, čo všetko iné obsahuje.	analýza; histogram; normal
Analýza možnosti PCL pre	Bc. Robert Birkus	Sprint 2	Pred samotnou implementáciou histogramu normal je potrebné analyzovať všetky možnosti knižnice	analýza; normal; PCL
Návrh výpočtu normal v PCL	Bc. Robert Birkus	Sprint 2	Je potrebné analyzovať možnosti knižnice PCL pre výpočet normal.	design; histogram; normal
Návrh histogramu normal	Bc. Robert Karasek	Sprint 2	Je potrebné navrhnuť štruktúru histogramu normal.	
Analýza rezov pre metódu	Bc. Robert Karasek	Sprint 2		
Porovnanie implementácií Hougha	Bc. Robert Karasek	Sprint 2		
Testovanie Hougha na dátasetoch	Bc. Robert Karasek	Sprint 2		

Tabuľka 9.4: Úlohy z tretieho šprintu

Title	Assigned to	Iteration	Description	Tags
Nájdienie bodov s normálou podobnou osi z	Bc. Katarína Janeckova	\Sprint 3	Po nájdieni normál pre všetky body vybrať len tie body, ktoré majú podobnú normálu s osou z.	implementácia
Dokončenie implementácie výpočtu normál	Bc. Katarína Janeckova	\Sprint 3		implementácia
Implementácia projekcie bodov na os z	Bc. Katarína Janeckova	\Sprint 3	Projekcia bodov, ktoré majú normálu podobnú s osou z, na os z.	implementácia
Vytvorenie kostry celkovej dokumentácie	Bc. Katarína Janeckova	\Sprint 3	Vytvorí základný dokument na dopĺňanie pre všetkých členov.	doku
Vytvorenie kapitoly Úvod (inžinierske dielo)	Bc. Katarína Janeckova	\Sprint 3	Napísať úvodnú kapitolu dokumentácie k inžinierskemu dielu.	doku
Vytvorenie podkapitoly Dátový model (inžinierske dielo)	Bc. Katarína Janeckova	\Sprint 3	Do dokumentácie k inžinierskemu dielu vytvorí dátový model aj s opisom.	doku
Zdokumentovanie metódy 1	Bc. Katarína Janeckova	\Sprint 3	Do dokumentácie k inžinierskemu dielu napísať kapitolu o metóde 1.	doku
Vytvorenie podkapitoly s odkazmi na priložené e-dokumenty (inžinierske dielo)	Bc. Katarína Janeckova	\Sprint 3	Do dokumentácie k inžinierskemu dielu napísať časť, ktorá bude obsahovať zoznam priložených e-dokumentov a ich opis.	doku
Vytvorenie kapitoly Úvod (riadenie)	Bc. Katarína Janeckova	\Sprint 3	Napísanie úvodnej kapitoly do dokumentácie k riadeniu.	doku
Vytvorenie kapitoly Role členov tímu a podiel práce (riadenie)	Bc. Katarína Janeckova	\Sprint 3	Do dokumentácie k riadeniu napísať kapitolu s vysvetlením jednotlivých zodpovedností členov tímu a tiež zapísať ich podiel práce na častiach	doku
Vytvorenie kapitoly Používané metodiky (riadenie)	Bc. Katarína Janeckova	\Sprint 3	Spísať zoznam používaných metodík s krátkym opisom ku každej do dokumentácie k riadeniu.	doku
Vytvorenie kapitoly Zoznam kompetencií tímu (riadenie)	Bc. Katarína Janeckova	\Sprint 3	Do dokumentácie k riadeniu opísať kompetencie tímu.	doku
Vytvorenie kapitoly Export evidencie úloh + urobiť exporty z TFS (riadenie)	Bc. Katarína Janeckova	\Sprint 3	Vytvorí v dokumentácii k riadeniu kapitolu exportov z TFS.	doku
Zápisnica č. 8 (20151112)	Bc. Katarína Janeckova	\Sprint 3	Vytvorenie zápisnice zo stretnutia č. 8	doku
Vytvorenie syntetických datasetov	Bc. Lukas Hudec	\Sprint 3	Vytvorenie testovacích datasetov pre účely vývoja.	
Vytvorenie podkapitoly Diagram tried (inžinierske dielo)	Bc. Lukas Hudec	\Sprint 3	Do dokumentácie k inžinierskemu dielu vytvorí diagram tried aj s opisom.	doku

Vytvorenie podkapitoly Architektúra (inžinierske dielo)	Bc. Lukas Hudec	\Sprint 3		
Implementácia oddelenia pointcloudu interiéru	Bc. Lukas Hudec	\Sprint 3	Implementácia metód na vybratie rozdielu dát stien a "zvyšku" pointcloudu	
Identifikácia zhukov bodov mimo stien	Bc. Lukas Hudec	\Sprint 3		
Rozdelenie vybraných oblastí bodov podľa pozície a veľkosti	Bc. Lukas Hudec	\Sprint 3		
Analýza segmentácie objektov z 3D pointCloudu	Bc. Lukas Hudec	\Sprint 3		
Vytvorenie kapitoly Globálne ciele pre ZS (inžinierske dielo)	Bc. Martin Jurik	\Sprint 3	Do dokumentácie k inžinierskemu dielu spísať ciele, ktoré sme si určili na ZS.	dok
Vyriešenie exception s boost library	Bc. Martin Jurik	\Sprint 3	Premyslieť ciele, ktoré je možné stiahnuť za ZS	
implementácia multiselect dialogu a otestovanie	Bc. Martin Jurik	\Sprint 3	Odstranenie exception odchytu programu pri return 0; Implementácia multiselect dialogu pre otvorenie viacerých suborov a navrat vektora stringov absolutných ciest pre metodu načítania point cloudov.	
Implementácia eventov a handlerov pre vyber a spustenie metódy	Bc. Martin Jurik	\Sprint 3	Implementovanie vyberu metódy v gui a naprogramovanie funkcií pre priestor implementácie jednotlivých metód.	
Implementácia spustenia vizualizácie v gui projekte	Bc. Martin Jurik	\Sprint 3		
Implementácia segmentácie podľa vlastností normál	Bc. Michal Korběl	\Sprint 3		
Segmentovanie jednej plochy z objektu podľa vlastností normál	Bc. Michal Korběl	\Sprint 3		
Segmentovanie celého objektu podľa vlastností normál	Bc. Michal Korběl	\Sprint 3		

Vytvorenie kapitoly Sumarizácie šprintov (riadenie)	Bc. Michal Korbelt	\Sprint 3	Do dokumentácie k riadeniu napísať kapitolu, v ktorej bude zhrnutie všetkých šprintov. Podľa úloh v TFS spraviť spätný opis/retrospektíva jednotlivých šprintov. Úspechy, neúspechy = "stop doing, start doing, keep doing" z TFS z časti Reporty vytiahnuť dáta o šprintoch (všetko možné čo sa dá ;))	doku
Zdokumentovanie metódy 2 (inžinierske dielo)	Bc. Michal Korbelt	\Sprint 3	Do dokumentácie k inžinierskemu dielu napísať kapitolu o metóde 4, ktorá bude obsahovať nasledovné podkapitoly: analýza, návrh, implementácia a testovanie. Problém: Knižnice PCL, resp. VTL uplatňujú vo svojich východiskách InteractorStyle- triedach správanie, pri ktorom sa kamera rotuje okolo bodu umiestneného v určitej vzdialenosti pred kamerou (focalPoint). Prirodzeným spôsobom je však otáčanie okolo samotného "ťažiska kamery" - podobne ako človek otáča svojou hlavou. Riešenie: Zadefinovať bod otáčania kamery do jej stredu resp. do zanedbateľne malej vzdialenosti pred ňu.	doku
Nastavenie stredu rotácie	Bc. Michal Löffler	\Sprint 3		

Vytvorenie kapitoly Aplikácie manažmentov (riadenie)	Bc. Michal Löffler	\Sprint 3	Kapitola do dokumentácie k riadeniu s opisom realizácie jednotlivých činností potrebných pre riadenie projektu, procesu a produktu. Manažment komunikácie a ľudských zdrojov (Róbert Karásek) - stretnutia tímu, komunikačné nástroje, nástroje na zdieľanie obsahu Manažment rozvrhu a rozsahu projektu (Lukáš Hudec) Manažment plánovania projektu (Lukáš Hudec) Manažment kvality (Mišo Löffler) - (zaistenie že projekt jeho výsledky uspokojuje potreby pre ktoré sa vytvoril/inicioval) Manažment rizík (Mišo Löffler) Manažment integrácie a podpory vývoja (Martin Jurík) - (zaistenie že rôzne elementy projektu sú správne koordinované)	dok
Posun dolava a doprava pomocou kláves A a D	Bc. Michal Löffler	\Sprint 3	Problém: Vo výchdzom nastavení PCL(VTK) je potrebné pre posun (pan) kamery držať stlačené (prostredné/pravé?) tlačidlo myši a súčasne hýbať myšou. Riešenie: Namapovať pohyby doprava a dolava na klávesy A a D.	
Posun dopredu/dozadu pomocou kláves W a S	Bc. Michal Löffler	\Sprint 3	Problém: Vo výchdzom nastavení PCL(VTK) je potrebné pre pohyb kamery dopredu a dozadu točiť kolečkóm myši alebo držať stlačené pravé tlačidlo myši a hýbať ňou dopredu/dozadu. Riešenie: Namapovať pohyby dopredu a dozadu na klávesy W a S.	

Zrušenie nutnosti mať stále stlačené tlačidlo myši pre rotovanie	Bc. Michal Löffler	\Sprint 3	Problém: Ak chceme natáčať pohľad kamery, musíme pritom držať stlačené ľavé tlačidlo myši, čo je trochu nepohodlné. Riešenie: Prvým kliknutím sa aktivuje natáčanie kamery pohybom myšou druhým kliknutím sa deaktivuje. Do dokumentácie k inžinierskemu dielu napísať kapitolu o vizualizácii, ktorá bude obsahovať nasledovné podkapitoly: analýza, návrh, implementácia a testovanie. Cieľom je zistiť možnosti vizualizácie 2D histogramu a zvoliť si ten najvhodnejší pre náš problém.	
Zdokumentovanie vizualizácie (inžinierske dielo)	Bc. Michal Löffler	\Sprint 3		doku
Analýza možnosti vizualizácie 2D histogramu	Bc. Robert Birkus	\Sprint 3		analýza
Implementácia vizualizácie histogramu	Bc. Robert Birkus	\Sprint 3	Cieľom je implementácia vizualizácie histogramu.	dev
Analýza možnosti zisťovania dominantných oblastí v 2D histograme	Bc. Robert Birkus	\Sprint 3	Cieľom je zanalyzovať rôzne algoritmy vhodné pre hľadanie dominantných oblastí v 2D histograme.	analýza
Návrh algoritmu pre nájdenie dominantných oblastí	Bc. Robert Birkus	\Sprint 3	Cieľom je navrhnúť algoritmus pre nájdenie dominantných oblastí v histogram. Predpokladá sa použitie mean-shift algoritmu.	design; histogram
Implementácia algoritmu pre nájdenie dominantných oblastí	Bc. Robert Birkus	\Sprint 3	Implementácia navrhnutého algoritmu pre nájdenie dominantných oblastí v histograme.	dev; histogram
Otestovanie správnosti algoritmu pre nájdenie dominantných oblastí	Bc. Robert Birkus	\Sprint 3	Cieľom je otestovať správnosť a vhodnosť algoritmu pre nájdenie dominantných oblastí v 2D histograme.	histogram; test
Nasadenie implementácie výpočtu a vizualizácie histogramu normál na hlavný projekt	Bc. Robert Birkus	\Sprint 3	Cieľom je nasadenie implementácie histogramu na náš hlavný projekt.	deploy; dev
Zdokumentovanie metódy 4	Bc. Robert Birkus	\Sprint 3	Do dokumentácie k inžinierskemu dielu napísať kapitolu o metóde 2,	doku
Zdokumentovanie metódy 3	Bc. Robert Karasek	\Sprint 3	Do dokumentácie k inžinierskemu dielu napísať kapitolu o metóde 3,	doku

Vytvorenie metodiky pre komunikáciu a pre committing/branching	Bc. Robert Karasek	\Sprint 3	Spísanie metodiky pre komunikáciu členov tímu v aplikácii Slack	doku
Príprava Testovacích scenárov	Bc. Robert Karasek	\Sprint 3		
Príprava konvencií pre generovanie technickej dokumentácie	Bc. Robert Karasek	\Sprint 3	Príprava základných konvencií a návodu ako vygenerovať technickú dokumentáciu	
Analyza histogramu Furukawa	Bc. Robert Karasek	\Sprint 3	Analyza histogramu z článku pre metódu Furukawa	
Implementácia histogramu pre Furukawa	Bc. Robert Karasek	\Sprint 3	Implementovanie histogramu, ktorý bude slúžiť na rozdelenie 3D priestoru na 2D rezy	
Testovanie histogramu Furukawa	Bc. Robert Karasek	\Sprint 3	Testovanie histogramu	
Analyza rozdelenia 3D pointcloudu na 2D rezy	Bc. Robert Karasek	\Sprint 3		
Implementácia rozdelenia 3D pointcloudu na 2D rezy	Bc. Robert Karasek	\Sprint 3	Implementácia bude zložitejšia nakoľko v papieroch nie je presný návod a web je nefunkčný, podobnú metódu na rovnakom princípe som nenašiel	
Testovanie rozdelenia 3D pointcloudu na 2D rezy	Bc. Robert Karasek	\Sprint 3		
Implementácia Gaussian smoothing	Bc. Robert Karasek	\Sprint 3		

Tabuľka 9.5: Úlohy zo štvrtého sprintu

Title	Assigned to	Iteration	Description	Tags
Analýza binary inside/outside segmentation	Bc. Katarína Janecková	\Sprint 4	-pochopenie princípu binárnej inside/outside segmentácie -prehľad o implementácii v knižniciach -urobiť prehľad o danej metóde pre zvyšok tímu	analýza
Implementácia extrakcie rezu	Bc. Katarína Janecková	\Sprint 4		implementácia
Implementácia metódy mean shift	Bc. Katarína Janecková	\Sprint 4		
Analýza outline simplification	Bc. Katarína Janecková	\Sprint 4		
Funkčná integrácia podporných modulov	Bc. Lukas Hudec	\Sprint 4	Prepojiť existujúce a vytvorené moduly s hlavným GUI	
Projekcia 3D primitív na 2D zobrazenie pôdorysu	Bc. Lukas Hudec	\Sprint 4	Pohľad zhora - zmena 3D na 2D zobrazenie a príprava na spracovanie pôdorysu	
Integrácia metód do gui	Bc. Lukas Hudec	\Sprint 4	Integrácia spúšťania metód cez gui.	
Reštrukturalizácia architektúry	Bc. Lukas Hudec	\Sprint 4	Z dôvodu vyššej efektivity a lepšieho rozvrhnutia dizajnu architektúry projektu je potrebné zmeniť implementáciu modulov, rozbiť zložité štruktúry podľa tematického zamerania a znížiť závislosť na includech.	implementácia
Zistenie roviny z bodov	Bc. Lukas Hudec	\Sprint 4	Pomocou štatistiky a metód analytickej geometrie získať funkčný predpis aproximovanej roviny, ktorá predstavuje na abstraktnej úrovni priestor v ktorom sa nachádzajú body rovnakého príznaku predstavujúce rovinu, ktorú hľadáme.	implementácia
Zápisnica č.9 (20151120)	Bc. Lukas Hudec	\Sprint 4	Spísanie zápisnice z 9. stretnutia a jej sprístupnenie pre celý tím.	
Implementácia spustenia vizualizácie v gui projekte	Bc. Martin Jurik	\Sprint 4	Riesenie vtk erroru a zatváranie okna vizualizácie z gui	
Analýza binary inside/outside segmentation	Bc. Martin Jurik	\Sprint 4		
Analýza outline simplification	Bc. Martin Jurik	\Sprint 4		
Zápisnica č.10 (20151126)	Bc. Martin Jurik	\Sprint 4		
Zistenie roviny z bodov	Bc. Michal Korbelt	\Sprint 4	- Ransac model plane - výpočet koeficientov roviny (ax + by + cz + d = 0)	
Oboznámenie sa s 3Dconnexion SDK	Bc. Michal Löffler	\Sprint 4	- preloženie roviny v smere plochy	
Ošetrovanie situácie keď je kurzor mimo okna vizualizácie	Bc. Michal Löffler	\Sprint 4		
Implementácia posunu nahor a nadol (elevation)	Bc. Michal Löffler	\Sprint 4		
Implementácia rotácie kamery	Bc. Michal Löffler	\Sprint 4		
Implementácia interakcie pomocou 3D myši (knižnica VTK)	Bc. Michal Löffler	\Sprint 4	1. prekomplikovanie knižnice VTK 2. nastavenie vtkRenderWindowInteractor 3. implementácia event observerov	
Analýza možnosti knižnice VTK pre ovládanie 3D myšou	Bc. Michal Löffler	\Sprint 4		
Navrhnuť spôsob označovania bodov	Bc. Robert Birkus	\Sprint 4		design

KAPITOLA 9. EXPORT EVIDENCIE ÚLOH

Implementácia označovania bodov	Bc. Robert Birkus	\Sprint 4	Implementovanie označovania všetkých bodov v point cloud pomocou dominantných normál. Podľa normály jednotlivých bodov sa priradí bod k najpodobnejšej dominantnej normále	dev
Orešovanie presnosti a správnosti označovania bodov	Bc. Robert Birkus	\Sprint 4	Je potrebné otestovať efektívnosť označovania bodov v pointcloud vizuálne z hľadiska segmentácie bodov reprezentujúcich stenu miestnosti.	test
Analyza možností zisťovania dominantných oblastí v 2D histograme	Bc. Robert Birkus	\Sprint 4	Cieľom je analyzovať rôzne algoritmy vhodné pre hľadanie dominantných oblastí v 2D histograme. Po analýze sme sa rozhodli, že najvhodnejší algoritmus pre daný problém bude mean-shift. A keďže knižnice, ktoré používame obsahujú implementáciu mean-shiftu len pre špecifické úlohy je potrebné si navrhnuť a naimplementovať vlastný mean-shift pre náš problém.	analýza
Návrh algoritmu pre nájdenie dominantných oblastí	Bc. Robert Birkus	\Sprint 4	Cieľom je navrhnuť algoritmus pre nájdenie dominantných oblastí v histograme. Predpokladá sa použitie mean-shift algoritmu.	design; histogram
Implementácia algoritmu pre nájdenie dominantných oblastí	Bc. Robert Birkus	\Sprint 4	Implementácia navrhnutého algoritmu pre nájdenie dominantných oblastí v histograme.	dev; histogram
Orešovanie správnosti algoritmu pre nájdenie dominantných oblastí	Bc. Robert Birkus	\Sprint 4	Cieľom je otestovať správnosť a vhodnosť algoritmu pre nájdenie dominantných oblastí v 2D histograme.	histogram; test
Nasadenie implementácie výpočtu a vizualizácie histogramu normál na hlavný projekt	Bc. Robert Birkus	\Sprint 4	Cieľom je nasadenie implementácie histogramu na náš hlavný projekt.	deploy; dev
Vytvorenie stručných pravidiel pre komentovanie	Bc. Robert Karasek	\Sprint 4	Jedná sa o správne anotácie v komentároch, podľa ktorých sa môžete držať.	
Vytvorenie prvých správne okomentovaných súborov	Bc. Robert Karasek	\Sprint 4		
Štúdium Unit Testov	Bc. Robert Karasek	\Sprint 4		

Tabuľka 9.6: Úlohy zo štvrtého šprintu

Title	Assigned to	Iteration	Description	Tags
Práca na dokumentácii	Bc. Katarína Janeckova	\Sprint 5	Spísanie nových vecí v rámci projektu do dokumentácie v častiach, v ktorých najviac prispievam.	doku
Finálne spracovanie dokumentácie k ZS	Bc. Katarína Janeckova	\Sprint 5	Doplnenie potrebných vecí, konečné úpravy, kompletizácia.	doku
Zjednodušenie prídania nového objektu steny	Bc. Lukas Hudec	\Sprint 5	Vzhľadom na implementačnú zložitosť vytvorenia nového objektu steny, je potrebná jeho reštrukturalizácia a zjednodušenie po maximálnu abstrakciu - overloading konštruktora.	
Hľadanie priesečníkov rovín - body ohraničenia steny	Bc. Lukas Hudec	\Sprint 5	Spísanie nových vecí v rámci projektu do dokumentácie v častiach, v ktorých najviac prispievam.	
Práca na dokumentácii	Bc. Lukas Hudec	\Sprint 5	Doplnenie a úprava metodiky evidencie úloh	
Primitívne 3D zobrazenie primitív	Bc. Lukas Hudec	\Sprint 5	Doplnenie a úpravy dokumentácie inžinierskeho diela v časti architektúra podľa posledných zmien	doku
Práca na dokumentácii	Bc. Martin Jurik	\Sprint 5	Dokumentácia Doxygen komentáre do kódu	
Implementovanie funkcionality check boxov	Bc. Martin Jurik	\Sprint 5	prvý pokus pre zobrazenie primitív bude ich jednoduché 3D premietnutie pci vizualizérom	
Rozšírenie GUI o featuru pre Birkého	Bc. Martin Jurik	\Sprint 5	Spísanie nových vecí v rámci projektu do dokumentácie v častiach, v ktorých najviac prispievam.	doku
Práca na dokumentácii	Bc. Michal Korbel	\Sprint 5	Implementovanie vyberu point cloudov na základe vybraných check boxov	
Nájdenie hraničných čiar objektu pomocou Houghovej	Bc. Michal Korbel	\Sprint 5	Spísanie nových vecí v rámci projektu do dokumentácie v častiach, v ktorých najviac prispievam.	doku
Segmentovanie jednej plochy z objektu podľa rezov rd	Bc. Michal Korbel	\Sprint 5		
Práca na dokumentácii	Bc. Michal Löffler	\Sprint 5	Spísanie nových vecí v rámci projektu do dokumentácie v častiach, v ktorých najviac prispievam.	doku
Zmenšenie veľkosti kroku pri pohybe do strán	Bc. Michal Löffler	\Sprint 5		
Zjernenie plynulosti pohybu do strán	Bc. Michal Löffler	\Sprint 5	Zatiaľ sa neporadilo naimplementovať dôvod: pri použití vtkRepeatingTimer je príliš dlhá odozva na stlačenie kláves. (vtkRepeatingTimer je automatický časovač, ktorý po stlačení klávesy pre pohyb začne v pravidelných krátkych intervaloch generovať pohyb o malý kusok dopredu/ do strany)	
Výpomoc pri hľadaní priesečníkov rovín	Bc. Michal Löffler	\Sprint 5		

KAPITOLA 9. EXPORT EVIDENCIE ÚLOH

Práca na dokumentácii	Bc. Robert Birkus	\Sprint 5	Spísanie nových vecí v rámci projektu do dokumentácie v častiach, v ktorých najviac prispievam.	doku
Návrh a implementácia metódy pre nájdenie dominant	Bc. Robert Birkus	\Sprint 5	Je potrebné navrhnuť a implementovať metódu pre nájdenie dominantných oblastí v histogram normál pomocou hľadania maxim.	design; dev
Kontrola správneho fungovania metódy	Bc. Robert Birkus	\Sprint 5	Je potrebné overiť, či naimplementovaná metóda vykonáva požadovanú funkcionality.	test
Nasadenie fungujúcich častí rekonštrukčnej metódy č.	Bc. Robert Birkus	\Sprint 5	Hotové časti metódy č.4 je potrebné nasadiť do hlavného projektu.	deploy
Nastavenie knižnice OpenCV 3.0.0 v debug móde v hlav	Bc. Robert Birkus	\Sprint 5	V hlavnom projekte sa knižnica OpenCV 3.0.0 bje s knižnicou boost a každá z knižnic si vyžaduje iné nastavenia. Je potrebné vyriešiť, aby v projekte fungovali obi dve knižnice naraz.	
Zápisnica č.11 (2015 203)	Bc. Robert Birkus	\Sprint 5	Táto úloha je vykonávaná v spolupráci s Martinom Juríkom	configuration
Vytvorenie Unit Testov	Bc. Robert Karasek	\Sprint 5	Spísanie zápisnice z agile stretnutia tímu	doku
Build Google Test pre projekt	Bc. Robert Karasek	\Sprint 5	Príprava buildu google testov pre náš projekt	
			Spísanie nových vecí v rámci projektu do dokumentácie v častiach, v ktorých najviac prispievam.	
			Metodika technickej dokumentácie	
			Metodika testovania	
Práca na dokumentácii	Bc. Robert Karasek	\Sprint 5	Príprava a vygenerovanie technickej dokumentácie	doku