

Slovenská technická univerzita v Bratislave
Fakulta informatiky a informačných technológií

Tím 16

Interakcia a kolaborácia vo virtuálnej realite

Študijný program: Softvérové inžinierstvo, Informačné systémy

Študijný odbor: 9.2.5 Softvérové inžinierstvo, 9.2.6 Informačné systémy

Miesto vypracovania: Ústav aplikovanej informatiky, FIIT STU

Vedúci tímu: Ing. Peter Drahoš, PhD.

Členovia tímu: Bc. Lenka Kutlíková, Bc. Erik Bujna, Bc. Mário Csaplár,
Bc. Michal Dobai, Bc. Lukáš Doubravský, Bc. Martin Petráš, Bc. Ondrej Vlček

november 2015

Obsah

1	Úvod	1
2	Kompetencie členov tímu	2
2.1	Bc. Erik Bujna	2
2.2	Bc. Mário Csaplár	2
2.3	Bc. Michal Dobai	2
2.4	Bc. Lukáš Doubravský	2
2.5	Bc. Lenka Kutlíková	3
2.6	Bc. Martin Petráš	3
2.7	Bc. Ondrej Vlček	3
3	Roly členov tímu	4
4	Aplikácie manažmentov	5
4.1	Manažment komunikácie	5
4.2	Manažment ľudských zdrojov	6
5	Sumarizácia šprintov	7
6	Používané metodiky	8
6.1	Podiel práce členov tímu na častiach dokumentácie	9
	Literatúra	11
A	Metodiky	12
A.1	Metodika tvorby dokumentácie	12
A.2	Metodika písania kódu	14
A.3	Metodika verziovania	16
A.4	Metodika tvorby zápisníc	18
A.5	Metodika používania stroja v laboratóriu	20
A.6	Metodika prenosu projektu	22
A.7	Metodika prideľovania storypointov	28
A.8	Metodika vytvárania úloh v nástroji na evidenciu úloh	29

A.9 Metodika komunikácie v tíme	31
B Zapisnice	36
C Exporty z Jiry	47

Zoznam obrázkov

1	Stavový diagram odstavca.	14
2	Proces tvorby zápisnice.	21
3	Inštalácia TDM GCC.	23
4	Konfigurácia prostredia CLion.	24
5	Konfigurácia v Cmake.	25
6	Generovanie v Cmake.	26
7	Spustenie kompilácie cez príkazový riadok.	27
8	Inštalácia projektu cez príkazový riadok.	27
9	Kompilácia pomocou Clion.	28

Zoznam tabuliek

1	Zoznam krátkodobých úloh členov tímu.	4
2	Podiel práce na dokumentácii k inžinierskemu dielu.	10
3	Podiel práce na dokumentácii k riadeniu.	10
4	Zoznam kanálov v Slacku.	34
5	Zoznam kanálov v Slacku.	35

1 Úvod

Hlavnou úlohou predmetu Tímový projekt je naučiť sa rozdiel medzi prácou v tíme a prácou v skupine. Zatiaľ čo skupina je len niekoľko ľudí, ktorí pracujú na spoločnom ciele, v tíme by malo dôjsť k čo najlepšiemu využitiu potenciálu každého člena. Keďže takýto stav nie je jednoduché dosiahnuť, vytvára sa tento dokument, ktorý má úlohu zdokumentovať, akým spôsobom sa každý člen pokúsil prispieť k celkovej práci.

Pod vedením Ing. Petra Drahoša, PhD. pracuje tím pod názvom Vexta agilnou metodikou Scrum s dĺžkou šprintov 2 týždne. Každý člen tímu má v rámci riadenia a manažmentu pridelených niekoľko úloh, ktoré sú v tomto dokumente opísané. Ďalej tu čitateľ nájde zoznam používaných metodík, sumarizáciu jednotlivých šprintov a zoznam zápisníc zo stretnutí.

2 Kompetencie členov tímu

2.1 Bc. Erik Bujna

Študent odboru Softvérové systémy, kde sa prihlásil po absolvovaní bakalárskeho štúdia v odbore Informatika. Vo svojej bakalárskej práci skúmal oblasť spracovania prirodzeného jazyka. Má skúsenosti s programovacími jazykmi Java, C a databázovým systémom MySQL.

2.2 Bc. Mário Csaplár

Absolvent bakalárskeho študijného programu Počítačové a komunikačné systémy a siete má viac než 6 ročné skúsenosti s programovacím jazykom C najmä vďaka reverznému inžinierstvu a analýze dátových štruktúr. Vytvoril niekoľko rozsiahlych webových projektov v technológiách xHTML, CSS, JavaScript (jQuery), MySQL, PHP. Medzi ďalšie používané jazyky patria C#, Java a VHDL. Jeho prednosťou je znalosť jazykov, venuje sa prekladu anglických textov rôznych žánrov do slovenčiny a korektúre textu po jazykovej aj formálnej stránke.

2.3 Bc. Michal Dobai

Vytvára projekty najmä v programovacom jazyku C# na platforme .NET, pričom s uvedenými technológiami má viac ako 5 ročné skúsenosti. Vďaka tvorbe knižničného systému nadobudol skúsenosti s databázovými technológiami, najmä so systémom MS SQL. Taktiež má prax vo vývoji aplikácií s používateľským prostredím založeným na Modern UI pre operačný systém Winows 8.1, respektíve Windows 10. K jeho prednostiam patrí tiež zrozumiteľné a presvedčivé odprezentovanie výsledkov práce tímu.

2.4 Bc. Lukáš Doubravský

Zaujíma sa hlavne o hardvér na mieru a tvorbu prislúchajúceho softvéru. Má skúsenosti od jazyka symbolických inštrukcií až po vysoko-úrovňové jazyky, napr. C#. Pracoval s rôznymi architektúrami a zariadeniami, akými sú PIC16F628A,

8086, 386EX, PXA255, Arduino Nano. Je oboznámený s tvorbou webových stránok s použitím HTML, CSS, PHP a MySQL. Zaujíma sa aj o tvorbu grafiky a videí.

2.5 Bc. Lenka Kutlíková

Študentka informačných systémov, kde sa prihlásila po absolvovaní bakalárskeho štúdia v odbore informatika na Fakulte informatiky a informačných technológií. Má skúsenosti s programovacími jazykmi Java a C++, s databázami (PostgreSQL), grafikou (OpenGL) a webovými technológiami (HTML, CSS).

2.6 Bc. Martin Petráš

Študuje na Fakulte informatiky a informačných technológií v odbore Informačné systémy. Bakalárske štúdium absolvoval na tej istej fakulte v odbore Informatika. Stretol sa s programovaním v jazykoch C++, Java, JavaScript, HTML, SQL, CSS. Má skúsenosti s technológiami ako OpenGL, GLSL, Qt.

2.7 Bc. Ondrej Vlček

Absolvent bakalárskeho štúdia na Fakulte informatiky a informačných technológií v študijnom programe Informatika. Prvák na inžinierskom stupni na rovnakej fakulte v študijnom programe Softvérové inžinierstvo. Skúsenosti v jazykoch C, Java, Ruby, HTML a PHP. Skúsenosti s tvorbou internetových obchodov a webových stránok.

3 Roly členov tímu

Táto kapitola obsahuje krátkodobé a dlhodobé úlohy jednotlivých členov tímu. Okrem manažérskych úloh zahŕňa aj podiel autorov jednotlivých častí dokumentácie k inžinierskemu dielu a ku riadeniu.

Meno	Krátkodobá úloha	Dlhodobá úloha
Lenka Kutlíková	Manažérka komunikácie a ľudských zdrojov	Plánovanie v Jire, príprava programu na stretnutia, tvorba zápisníc, dohliadanie na dodržiavanie termínov, dohliadanie na komunikáciu.
Martin Petráš	Manažér rozsahu projektu, manažér kvality	Vedúci vývoja.
Mário Csaplár	Manažér rizík a incidentov, manažér dokumentácie	Korektúra písomností, programovanie.
Erik Bujna	Manažér verzií	Verziovanie kódu, zdieľanie dokumentov.
Ondrej Vlček	Manažér úlohového nástroja	Utužovanie tímu.
Lukáš Doubravský	Manažér údržby	Technická podpora pre web server a počítač v laboratóriu, správa webovej stránky.
Michal Dobai	Manažér testovania	Správa webovej stránky.

Tabuľka 1: Zoznam krátkodobých úloh členov tímu.

4 Aplikácie manažmentov

4.1 Manažment komunikácie

Komunikácia v tíme je jedným zo základných predpokladov fungovania tímu. Každý člen tímu musí byť oboznámený so všetkými okolnosťami, ktoré nastali, aby nedošlo ku nedorozumeniam.

Každý krok je potrebné vzájomne odkomunikovať a odsúhlasiť všetkými členmi tímu, aby nevznikali konflikty. Ak vznikne situácia, kedy sa niektorý z členov tímu zasekne na svojej úlohe, je dôležité, aby požiadal o pomoc iného člena tímu, aby projekt nestál celý.

Pre komunikáciu v tíme boli zvolené formálne aj neformálne nástroje na komunikáciu. Členovia tímu sú s nimi oboznámení a vedia ich efektívne využívať.

Skupinová konverzácia – Facebook. Na začiatku semestra bola vytvorená spoločná konverzácia. Avšak pre neprehľadnosť a veľmi zložité vyhľadávanie informácií bola táto konverzácia po druhom šprinte zrušená. Súčasne bol od začiatku vytvorený Slack, ku ktorému však neboli jasne určené pravidlá, ktoré informácie budú tam a ktoré v skupinovej konverzácii na Facebooku, preto vznikali nedorozumenia na základe straty niektorých informácií. Zrušením tejto skupiny sa zefektívnila komunikácia v tíme.

Slack. V prvom šprinte bol zriadený nástroj na komunikáciu v tíme - Slack. Vytvorili sa potrebné kanály, ktoré majú jasne stanovený účel. Každý večer si každý člen tímu pozrie nové informácie na Slacku, čím sa zabezpečí efektívna a rýchla komunikácia. Určené pravidlá sú opísané v prislúchajúcej metodike.

E-mail. Bol vytvorený tímový e-mailový účet prostredníctvom služby Google Groups (tim_16@googlegroups.com). Používa sa prevažne na komunikáciu s vedúcim tímu.

Kalendár. Vytvoril sa zdieľaný kalendár prostredníctvom Google Calendar, kde všetci zaznačujú dôležité termíny (odovzdania, kontrolné body) a neprítomnosť alebo iné povinnosti všetkých členov tímu, aby bolo možné odhadnúť vyťaženie jednotlivých členov a na základe toho dobre naplánovať úlohy.

Dropbox. Vytvoril sa zdieľaný priečinok na Dropboxe, kde sa zdieľajú materiály, ktoré by mohli byť užitočné pre všetkých členov tímu.

JIRA. Na manažment úloh bola zvolená Jira. Zaznamenávajú sa do nej úlohy, ktoré vyplývajú z projektu. Každý člen je povinný navštevovať Jiru, vytvárať a spracúvať vlastné úlohy. Pridávanie úloh do Jiry má pravidlá opísané v prislúchajúcej metodike.

Tímové stretnutia. Komunikácia prebieha aj na tímových stretnutiach formou diskusie k danej téme. Vždy je pripravený program na stretnutia a postupne sa diskutujú jednotlivé body. Týchto stretnutí sa zúčastňuje každý člen tímu.

4.2 Manažment ľudských zdrojov

Pre správne fungovanie tímu je dôležité, aby každý člen tímu mal pridelenú úlohu, ktorá odzrkadľuje jeho schopnosti. V tom prípade tím optimálne využíva pracovnú silu. Úlohou manažéra ľudských zdrojov je rozdeliť úlohy v tíme tak, aby každý mal zvládnutelné množstvo práce.

Na základe schopností členov tímu vždy na stretnutí pred začiatkom ďalšieho šprintu manažér ľudských zdrojov rozdelí úlohy jednotlivým členom po vzájomnej dohode.

5 Sumarizácia šprintov

Sumarizácia 1. šprintu

V prvom šprinte bolo hlavnou úlohou rozdeliť si manažérske úlohy v tíme a sformalizovať procesy, ktoré pod túto úlohu patria. Každý člen tímu sa stal manažérom nejakej oblasti, ktorú od tej doby riadil. V tomto šprinte došlo k zoznámeniu tímu, vytvoreniu spoločného plagátu, webovej stránky so základnými informáciami. Každý člen tímu sa začal oboznamovať s technológiami použitými v projekte. Taktiež boli zriadené komunikačné nástroje a začal sa používať nástroj na manažment úloh v tíme, Jira.

Sumarizácia 2. šprintu

Hlavnou úlohou druhého šprintu bola analýza technológií a spustenie projektu z predchádzajúcich rokov. Druhá spomínaná úloha zabrala značné množstvo času kvôli problémom s kompiláciou. Súčasne všetci členovia tímu nabrali nové skúsenosti s prácou v Jire.

Sumarizácia 3. šprintu

V tomto šprinte boli spísané metodiky, ktoré boli vytvorili počas troch šprintov. Bola napísaná analýza technológií, ktoré budú v projekte používané. Spustila sa práca s Oculusom a Kinectom, avšak zatiaľ bez integrácie projektu z minuloročných tímových projektov. Podarilo sa vytvoriť kostrový model prostredníctvom Kinectu.

6 Používané metodiky

Metodika tvorby dokumentácie

Cieľom metodiky [A.1](#) je formalizovať všetky potrebné kroky pre jednotné písanie dokumentácie. Na základe tejto metodiky môžu dokumentáciu písať súbežne všetci členovia tímu s minimálnymi odlišnosťami vo formalizme. Zároveň je pripravený spôsob na jednoduchú integráciu spoločne tvorenej dokumentácie do finálnej prostredníctvom systému revízií.

Metodika písania kódu

Metodika [A.2](#) definuje pravidlá pre písanie kódu v jazyku C++. Jej hlavným cieľom je zvýšiť čitateľnosť a konzistenciu napísaného kódu. Ďalším cieľom je zabezpečiť, aby výsledný kód bol efektívny a bezpečný. Metodika sa snaží zabrániť používaniu neefektívnych a nebezpečných jazykových konštrukcií.

Metodika verziovania

Obsahom metodiky verziovania dostupnej v [A.3](#) je definovanie takého postupu práce s verziami zdrojového kódu pri vyvíjaní projektu, aby bolo minimalizovaný počet konfliktov. Pre projekt bol využitý systém riadenia verzií Git s repozitárom uloženým na stránke GitHub.

Metodika tvorby zápisníc

Obsahom metodiky tvorby zápisníc v [A.4](#) je definovanie postupu, ako správne vytvoriť zápis zo stretnutia. Metodika slúži primárne zapisovateľovi. Metodika definuje proces tvorby zápisnice, jej nutné časti a jej formálnu stránku.

Metodika používania stroja v laboratóriu

Metodika [A.5](#) prináša základné informácie o spoločnom používaní stroja, ktorý sa nachádza v Laboratóriu počítačového videnia a počítačovej grafiky. Uvádza pravidlá pre používanie priamo v miestnosti, ale aj prostredníctvom vzdialeného prístupu.

Metodika prenosu projektu

Vzhľadom na to, že projekt *3DSoftviz* je rozsiahly a obsahuje veľa závislostí, bol vytvorený nový systém kompilovania. Metodika [A.6](#) prináša podrobný opis všetkých krokov potrebných na spustenie projektu pod operačným systémom Windows 7 a vyšším.

Metodika pridelovania storypointov

Stručná metodika pridelovania storypointov v [A.7](#) prináša spôsob, akým sa ohodnocujú jednotlivé úlohy pre nasledujúci šprint. Opisuje význam jednotlivých kartičiek a postup pri ohodnotení úlohy vysokým číslom.

Metodika vytvárania úloh v nástroji na evidenciu úloh

Metodika [A.8](#) formalizuje celý proces evidencie úloh. Cieľom je štandardizovať pomenúvanie úloh, životný cyklus úlohy a jej prioritu.

Metodika komunikácie

Obsahom metodiky komunikácie v [A.9](#) sú pravidlá pre komunikáciu tímu prostredníctvom všetkých kanálov, ktoré používame. Opisuje pravidlá pre mailovú komunikáciu, Slack, používanie kalendára, zdieľanie prostredníctvom Dropboxu a komunikáciu prostredníctvom Jiry. Je určená pre všetkých členov tímu.

6.1 Podiel práce členov tímu na častiach dokumentácie

Tabuľka [2](#) prináša zoznam členov tímu a ich podiel na vzniku dokumentácie k inžinierskemu dielu. V tabuľke [3](#) sa nachádza rovnaké rozčlenenie pre dokumentáciu ku riadeniu.

Časť	Autor
Úvod	Mário Csaplár
Ciele pre zimný semester	Mário Csaplár
Oculus Rift	Martin Petráš
Microsoft Kinect	Erik Bujna
Analýza projektu	Mário Csaplár
Korektúra a integrácia	Mário Csaplár

Tabuľka 2: Podiel práce na dokumentácii k inžinierskemu dielu.

Časť	Autor
Úvod	Mário Csaplár
Kompetencie, roly	všetci
Aplikácie manažmentov	všetci
Metodika tvorby dokumentácie	Mário Csaplár
Metodika písania kódu	Martin Petráš
Metodika verziovania	Erik Bujna
Metodika tvorby zápisníc	Lenka Kutlíková
Metodika používania stroja v laboratóriu	Michal Dobai
Metodika prenosu projektu	Lukáš Doubravský
Metodika prideľovania storypointov	Ondrej Vlček
Metodika vytvárania úloh	Ondrej Vlček
Metodika komunikácie	Lenka Kutlíková
Zápisnice	Lenka Kutlíková
Korektúra a integrácia	Mário Csaplár

Tabuľka 3: Podiel práce na dokumentácii k riadeniu.

Literatúra

A Metodiky

A.1 Metodika tvorby dokumentácie

Dokumentácie k inžinerskemu dielu a ku riadeniu sa vytvárajú v dvoch etapách. V prvej etape všetci členovia tímu dokumentujú oblasti, za ktoré sú zodpovední, respektíve tie, ktoré im boli menovite zadelené manažérom dokumentácie na kanále #dokumentacia v tímovom Slacku.

Spoločne vytváraná dokumentácia sa nachádza v službe Dokumenty Google¹. Kostra dokumentácie bola vytvorená manažérom dokumentácie, zásadné štrukturálne zmeny je potrebné konzultovať na kanále #dokumentacia. Prístup ku dokumentu je udelený výhradne členom tímu.

Každý odstavec a podnadpis, ktorý sa aktuálne vytvára a ešte nie je dokončený, je potrebné písať modrou farbou #365f91. Tým dáva autor obsahu najavo, že na ňom momentálne pracuje a bude ho ešte meniť. Po dokončení obsahu zafarbí autor celý dokončený úsek zelenou farbou #4f6228, čím ho vydá na revíziu.

V druhej etape manažér dokumentácie integruje zelené úseky v dokumente Google do dokumentácie vytvárajanej v prostredí LaTeX, ktorá sa považuje za finálnu. Integrácia spočíva v revízii zelených úsekov, prekopírovaní úseku do výsledného dokumentu a zafarbení zelených úsekov čiernou farbou #000000.

Po integrácii je text v prostrediach LaTeX aj Google zhodný, preto vždy majú všetci členovia tímu k dispozícii aktuálnu verziu dokumentácie spolu s prehľadom o tom, ktoré časti už boli integrované a ktoré nie. Ak sa autor už integrovaného úseku rozhodne doplniť ďalší odstavec, postupuje podľa pokynov ku písaniu nového odstavca, teda ho označí modrou farbou #365f91.

Ak uváži, že je potrebné upraviť obsah už integrovaného úseku, každú vykonanú zmenu v odstavci označí červenou farbou #c00000, aby bola jednoznačne identifikovateľná manažérom dokumentácie. Zmeny, ktoré nebudú označené červenou farbou, nebude možné integrovať.

Pri písaní dokumentácie v službe Dokumenty Google nehrá rolu použitý rez písma ani jeho veľkosť, pretože text bude integrovaný do prostredia LaTeX v jednotnej forme danej šablónou. Je však potrebné dodržiavať všetky jazykové

¹<https://docs.google.com/document/d/1Ik-vJtBt7L0F2-8I5wZPm7feJhMfDEMx5khMvqMFIEA>

náležitosti opísané v nasledujúcich odstavcoch.

Dokumentáciu k inžinierskemu dielu je potrebné písať po anglicky. Nepripúšťa sa prvá osoba jednotného ani množného čísla, ani autorský plurál. Pri analýze technologických prostriedkov sa používa prítomný čas (Príklad: *Kinect SDK provides three distinct output modes.*). Pri opise vykonaných činností sa používa minulý čas v trpnom rode (Príklad: *It was determined the provided functionality does not suffice the project's needs.*).

Vzhľadom na to, že dokumentácia k inžinierskemu dielu je formálny dokument, je potrebné dodržiavať formálny štýl písania. Ten zahŕňa najmä používanie formálnych tvarov modálnych slovies (Príklad: *can not* namiesto *can't, do not* namiesto *don't* a podobne).

Dokumentáciu ku riadeniu je potrebné písať v slovenskom jazyku. Je povolené hovoriť o kompetenciách jednotlivých členov tímu, ale opäť vo forme tretej osoby (Príklad: *[Meno člena] mal na starosti manažment revízií.*). Používajú sa horné a dolné stredoeurópske úvodzovky (klávesové skratky *Alt+0132*, *Alt+0148*).

Pri uvedení prekladu termínu je potrebné uviesť do zátvorky skratku jazyka a kurzívou samotný termín. Príkladmi sú *spracovanie* (*angl. processing*) alebo *subset* (*Slovak: podmnožina*). Zvýrazňovanie textu je povolené iba kurzívou, nie tučným písmom, zväčšením písma ani farbením.

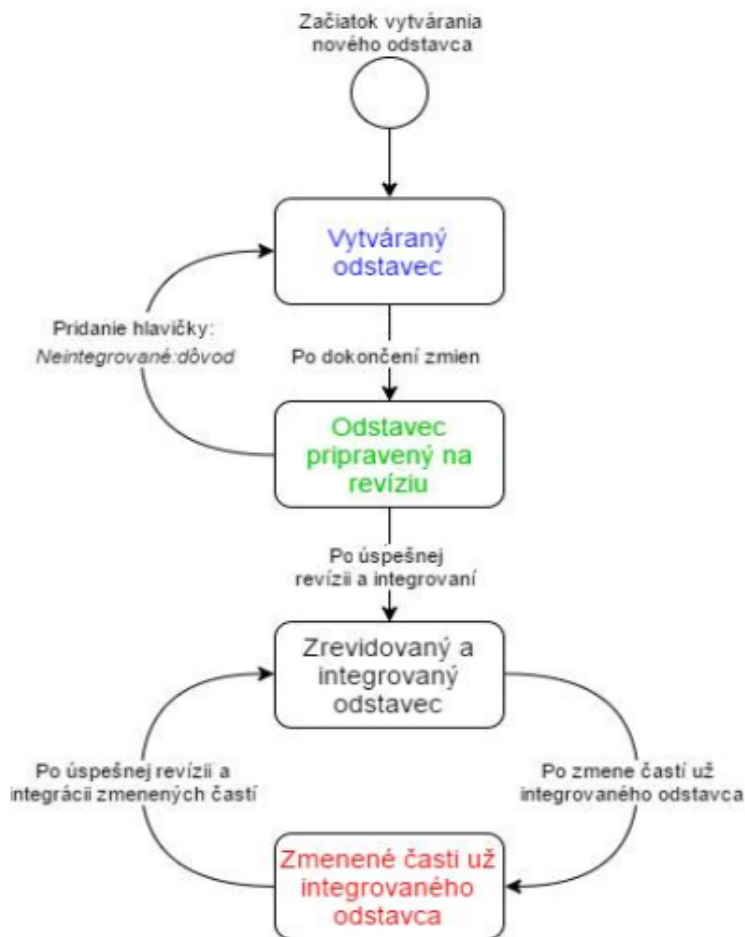
Každý obrázok musí obsahovať internetový odkaz, z ktorého sa dá obrázok prevziať. Ak bol vytvorený autorom, môže použiť ľubovoľný zdieľací server alebo kanál #dokumentacia na tímovom Slacku. Všetky obrázky a tabuľky musia obsahovať popis. Číslo obrázku alebo tabuľky je potrebné explicitne uvádzať iba vtedy, ak sa naňho v texte autor odkazuje.

Revízia úseku zahŕňa kontrolu vyššie opísaných aspektov spolu s kontrolou pravopisu. Pri menej závažných nedostatkoch vykoná nápravu priamo manažér dokumentácie a opravený úsek integruje.

Pri hrubom nedodržaní týchto princípov sa v dokumente Google označí chybný úsek modrou farbou #365f91 a na začiatok odstavca sa uvedie *Neintegrované: dôvod*, čím sa úsek vráti autorovi na prepracovanie. Je v kompetencii autora daného úseku, aby si skontroloval, či jeho úsek bol integrovaný alebo nie.

Na prvej strane dokumentu Google sa nachádzajú kolónky *Dátum a čas poslednej integrácie:* a *Dátum a čas najbližšej integrácie:*, ktoré musí manažér

dokumentácie priebežne aktualizovať, aby mali všetci členovia tímu prehľad o integračnom procese. Stavový diagram 1 zobrazuje životný cyklus odstavca.



Obr. 1: Stavový diagram odstavca.

A.2 Metodika písania kódu

Táto metodika slúži ako referencia pre programátora, aby bol výsledný kód dobre čitateľný a konzistentný. Všetok kód vrátane komentárov je nutné písať po anglicky.

Premenné. Názvy premenných sú písané ako CamelCase, pričom začínajú malým písmenom. Ak je potrebné uviesť informáciu o type premennej, musí sa

nachádzať v poslednom slove. Názov premennej musí napovedať, na čo slúži, zároveň nesmie byť dlhší ako 20 znakov. Názvy premenných dĺžky 1 znak sú povolené iba ako lokálne premenné vo funkciách a metódach.

Ak je potrebné špecifikovať účel premennej, je povolený prefix maximálnej dĺžky 5 znakov, ktorý je oddelený znakom „_“, za ktorým nasleduje zvyšok názvu začínajúci malým písmenom. Žiadnu informáciu nie je povolené uvádzať v názve viacnásobne.

Funkcie a metódy. Názov funkcie je v tvare CamelCase, začína sa malým písmenom. Názov musí jasne informovať o účele funkcie/metódy, ale zároveň nesmie presiahnuť 30 znakov. Prioritu má vždy informácia pred dĺžkou názvu.

Názvy parametrov funkcií musia vystihovať ich účel (*int nodeNumber* namiesto *int a*). Veľký dôraz je potrebné klásť na typ parametrov a spôsob predávania hodnôt pri volaní, aby sa predišlo zbytočnému kopírovaniu. Ak je to možné, voliteľné parametre musia byť implicitné parametre funkcií namiesto preťažovania funkcií (*void function(int i = defaultValue)* namiesto dvojice *void function()* a *void function(int i)*). Lokálne premenné sa deklarujú vždy na začiatku bloku.

Triedy. Triedy sú definované vždy v hlavičkovom súbore (.h), pričom každá trieda musí byť v samostatnom súbore. Implementácia metód sa musí nachádzať v samostatnom zdrojovom súbore (.cpp). V prípade tried alebo metód do 10 riadkov je možné spraviť výnimku, kedy je trieda alebo metóda iba hlavičkovom súbore. Názov triedy je v tvare CamelCase, začína sa veľkým písmenom.

Hotová trieda má implementované všetky potrebné konštruktory, virtuálny deštruktor a preťažené všetky potrebné operátory. Musí byť zabezpečené korektné rozhranie a bezpečný prístup k členom. Metódy na získanie a nastavenie hodnoty sú definované iba pre relevantné atribúty triedy. Členské premenné sú inicializované pomocou inicializačného zoznamu namiesto inicializácie v tele konštruktora (napríklad *Class(int arg1, int arg2) : a(arg1), b(arg2) { }*). Autor triedy je povinný vytvoriť dokumentáciu pomocou komentárov v hlavičkovom súbore.

Štruktúra kódu. Používa sa odsadenie virtuálnym tabulátorom, ktorý tvoria 4 medzery. Odsadenie sa zväčšuje každým novým vnoreným blokom. Znak „{“ je na samostatnom riadku. V prípade funkcie alebo jazykovej konštrukcie môže byť na konci toho istého riadku. Pri volaní, deklarácii a definícii funkcií (aj konštruktory) medzi názvom funkcie a znakom „(“ nie je medzera. V prípade jazykových

konštrukcií (podmienky, cykly) je medzi názvom konštrukcie a znakom „(“ znak medzera, čo platí aj pre funkciu *for_each* hlavičkového súboru *algorithm.h*.

A.3 Metodika verziovania

Obsahom metodiky verziovania je definovanie takého postupu práce s verziami zdrojového kódu pri vyvíjaní projektu, aby bol minimalizovaný počet konfliktov. Pre projekt bol využitý systém riadenia verzií Git s repozitárom uloženým na stránke GitHub. Ako základ pre metodiku verziovania bol použitý Git Flow navrhnutý Vincentom Driessenom.[?]

Získanie projektu

Pre nakopírovanie projektu zo stránky GitHub.com treba použiť príkaz *git clone –recursive (link na projekt)*, kde (link na projekt) je adresa projektu, v tomto prípade <https://github.com/vexta/3dsoftviz.git>. Vykonaním príkazu sa na lokálne úložisko skopíruje vetva master, ktorá je určená ako hlavná vetva. Na inú vetvu sa dá dostať príkazom *git checkout vetva*.

Konvencie vetiev

V projekte sa rozlišujú dva typy vetiev:

- Trvalé, ktoré sú aktívne počas celého vývoja projektu.
 - *Master*. Hlavná vetva s fungujúcim, odladeným kódom, pripraveným na produkciu.
 - *Develop*. Hlavná vetva, ktorej posledná verzia obsahuje funkcionality, ktorá sa pripravuje na odovzdanie do produkcie.
- Dočasné.
 - *Feature*. Hlavná vetva s fungujúcim, odladeným kódom, pripraveným na produkciu.
 - *Release*. Hlavná vetva, ktorej posledná verzia obsahuje funkcionality, ktorá sa pripravuje na odovzdanie do produkcie.

- *Hotfix*. Vedľajšia vetva slúžiaca na rýchlu opravu chýb, vytvára sa z vetvy *Master* vetvy, spája sa do *Develop* a *Master*.

Vytvorenie novej dočasnej vetvy

Postup vytvorenia novej dočasnej vetvy je nasledovný:

1. Nastaviť sa na verziu hlavnej vetvy, z ktorej sa bude dočasná vetva vytvárať príkazom *git checkout (hlavná vetva)*, napríklad *git checkout develop*.
2. Získať poslednú verziu danej hlavnej vetvy zo vzdialeného repozitára príkazom *git pull (vzdialený repozitár) (hlavná vetva)*, napríklad *git pull origin develop*.
3. Samotné vytvorenie novej vetvy je vykonané príkazom *git checkout -b (názov novej vetvy)*. Názov novej vetvy sa vytvára podľa šablóny (typ)-(názov), kde typ je jeden z typov dočasných vetiev, teda feature, release alebo hotfix, a (názov) musí vystihovať, čo je vo vetve implementované či opravené. Názov musí byť krátky a výstižný, zložený z najmenej jedného a najviac troch slov spojených cez podčiarkovník, a je písaný v angličtine.

Zapísanie zmien vo vetve

Po úprave kódu sa lokálne zmeny zapisujú nasledovným spôsobom:

1. Pre zistenie, ktoré súbory boli modifikované, slúži príkaz *git status*. Príkaz vypísal zoznam súborov, ktoré boli modifikované. Pre pridanie vybraného súboru do zápisu sa použije príkaz *git add (názov súboru)*. Príkaz sa použije na všetky súbory, ktoré sa majú lokálne uložiť do zápisu.
2. Nasledovným príkazom sa vykoná samotný lokálny zápis súborov: *git commit -m (správa o zmenách)*, kde (správa o zmenách) je krátka správa o vykonaných zmenách, teda úspešne opravených chybách, pridanej funkcionality či aktualizovaných častiach kódu.

Zapísanie zmien do vzdialeného repozitára

Akonáhle bola dokončená práca na funkcionalite v dočasnej vetve a bol vykonaný zápis upravených súborov, treba upravené súbory uložiť na vzdialený repozitár. To sa vykoná príkazom *git push* (*vzdialený repozitár*) (*vetva*), napríklad *git push origin feature-stereoscopic_vision*.

Spojenie vetiev

Po ukončení prác na dočasnej vetve je potrebné ju spojiť s hlavnou nasledovným spôsobom:

1. Presunúť sa na hlavnú vetvu, s ktorou sa má dočasná vetva spojiť, a aktualizovať ju dvojicou príkazov
 - (a) *git checkout* (hlavná vetva)
 - (b) *git pull* (*vzdialený repozitár*) (hlavná vetva)
2. Vykonať spojenie príkazom *git merge --no-ff* (*vedľajšia vetva*). Značka *--no-ff* zabraňuje strateniu informácií o vetve (*no fast forward*), inak by boli zmeny zapísané tak, akoby boli vykonané priamo v hlavnej vetve. Týmto spôsobom je uľahčené vrátenie zmien, pretože by inak neboli v skupine zápisov dočasnej vetvy.

A.4 Metodika tvorby zápisníc

Touto metodikou sa riadi zapisovateľ zápisníc v danom týždni. Zapisovateľ si robí počas stretnutia poznámky a po stretnutí to sformuje do zápisnice, ktorá spĺňa aj formálne požiadavky spísané v tejto metodike. Po schválení zápisnice ostatnými členmi tímu je zápisnica zverejnená na internetovej stránke projektu.

Skratky a pojmy

- *MS Word* – Microsoft Word

Obsah zápisnice

V každej zápisnici sa nevyhnutne nachádzajú tieto položky:

- Poradové číslo stretnutia
- Číslo tímu
- Dátum a čas stretnutia
- Miesto stretnutia
- Zapisovateľa stretnutia
- Prítomných členov tímu
- Tému stretnutia
- Opis stretnutia v bodoch
- Rozdelenie úloh

Opis stretnutia obsahuje informácie o aktuálnom stave úloh, zhodnotenie splnenia úloh z predchádzajúceho týždňa a rozhodnutia, ktoré boli na danom stretnutí uskutočnené, prípadne odporúčania od vedúceho tímu.

Rozdelenie úloh je zapísané v jednoduchnej tabuľke, ktorá ma tri stĺpce:

- Poradové číslo úlohy
- Popis úlohy
- Meno člena, ktorý úlohu vypracuje

Formátovanie

V dokumente je použitý štýl s názvom *Word 2010*. Je použité písmo *Century Gothic (Headings)*. Veľkosti, zvýraznenia a zarovnania sú nasledovné:

- *Názov dokumentu*. 26pt, zarovnané na stred.
- *Nadpis*. 20pt, zarovnané na stred.

- *Text*. 12pt, zarovnané naľavo.
- *Text (názov kategórie)*. 12pt, zarovnané naľavo, tučné.

Opis stretnutia je opísaný v bodoch. Používa sa číslovanie prvej úrovne s arabskými číslicami. Riadkovanie v celom dokumente je 1,15. Na rozdelenie úloh sa používa jednoduchá tabuľka. Písmo v tabuľke je zarovnané naľavo.

Proces tvorby zápisnice

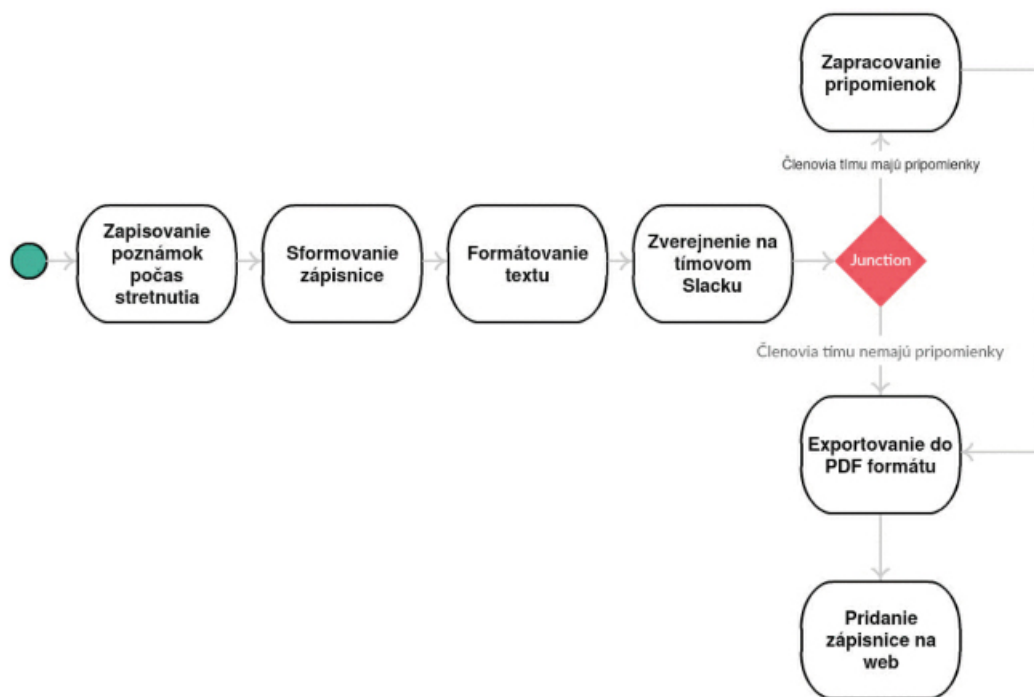
1. Zapisovateľ si píše poznámky.
2. Zapisovateľ sformuje zápisnicu.
3. Nastaví formátovanie podľa príslušných pravidiel.
4. Zverejní zápisnicu na tímovom Slacku.
5. Členovia tímu overia zápisnicu.
 - V prípade námietok, zapisovateľ zapracuje pripomienky do zápisnice.
6. Overená zápisnica sa vyexportuje do pdf formátu.
7. Zápisnica sa zverejní na webovej stránke.

Tento postup je znázornený na diagrame [2](#).

A.5 Metodika používania stroja v laboratóriu

Touto metodikou sa riadia všetci členovia tímu počas prístupu na počítač dostupný v Laboratóriu počítačového videnia a počítačovej grafiky (miestnosť 1.27). Každý člen má na tomto počítači vytvorené konto, pomocou ktorého sa naň prihlasuje buď priamo v laboratóriu, alebo pomocou vzdialeného prístupu. Na počítači je tiež vytvorené konto 1.27 s administrátorskými oprávneniami, ku ktorému majú prístup členovia Lukáš Doubravský a Michal Dobai.

Práca na PC pri priamom prístupe z laboratória:



Obr. 2: Proces tvorby zápisnice.

- V prvom kroku je nutné overiť, či je laboratórium v požadovanom čase voľné. Laboratórium je obsadené v časoch oficiálnych stretnutí tímov č. 4 a č. 5. Časy týchto stretnutí sú uvedené v rozvrhu pri dverách do laboratória. Laboratórium môže byť obsadené členmi iných tímov aj mimo oficiálnych stretnutí. V takom prípade je potrebná ústna dohoda medzi tímami.
- Dohody o časoch prístupu členov tímu Vexta do laboratória je nutné komunikovať výlučne cez kanál #labak.
- V prípade, že je laboratórium v požadovanom čase voľné, člen tímu si zapožičia kľúče na vrátnici.

Pri práci na PC v laboratóriu majú členovia tímu zakázané:

- Odpájať od počítača sieťový kábel.
- Vypínať a reštartovať počítač.

- Pripájať sa na počítač pomocou funkcie vzdialeného prístupu programu TeamViewer.

Práca na PC pri vzdialenom prístupe:

- Členovia tímu vytvoria vzdialené pripojenie k počítaču podľa návodu dostupnom v tímovom priečinku Dropbox, ako aj v prílohe ?? tejto dokumentácie.

Uvedené pravidlá je potrebné dodržať, pretože k počítaču je vytvorený vzdialený prístup. Ostatní členovia tímu teda môžu byť k počítaču v rovnakom čase pripojení vzdialenene. Pripájanie pomocou funkcie vzdialeného prístupu cez TeamViewer predstavuje bezpečnostné riziko. Osoby v laboratóriu môžu vidieť a zasahovať do práce vzdialene prihláseného člena tímu.

Každý člen tímu má k dispozícii svoj vlastný účet a osobný priečinok. Ten slúži ako priestor na umiestnenie zdrojových kódov vytváraného produktu. Pre všetkých členov tímu je prístupný aj priečinok C:

Work, ktorý slúži na zdieľanie súborov medzi členmi tímu.

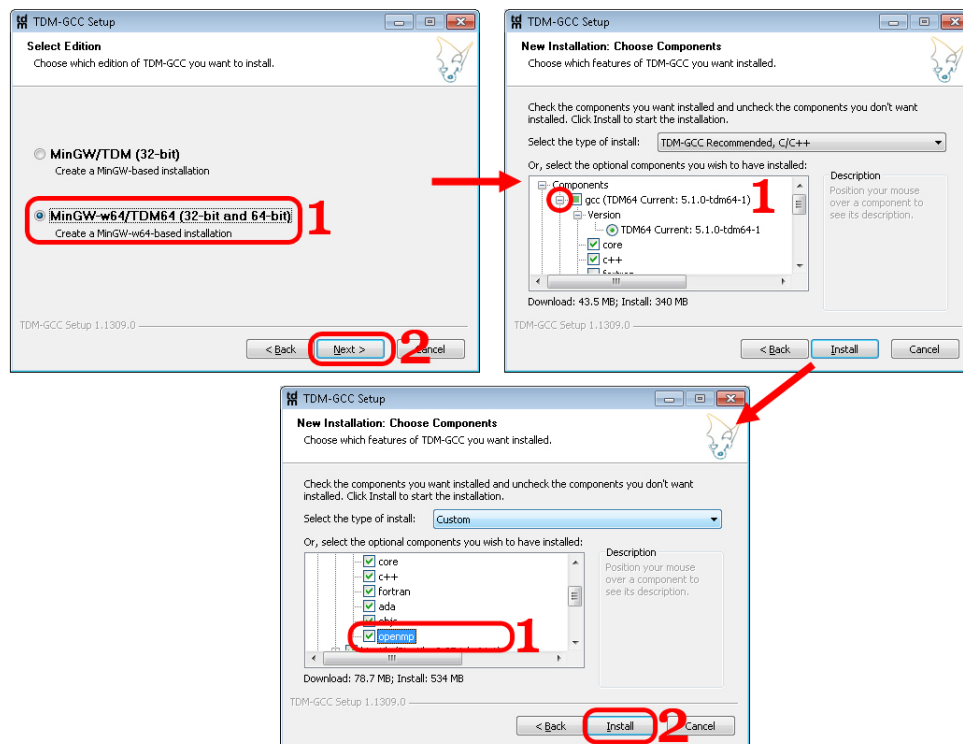
V prípade požiadavky na inštaláciu nového softvéru alebo požiadavky na iný úkon vyžadujúci administrátorské oprávnenia napíše člen tímu požiadavku na komunikačný kanál #labak na tímovom Slacku. Taktiež vytvorí novú úlohu v Jire, ktorú si jeden z administrátorov pridelí. Po uzatvorení úlohy administrátor informuje zadávateľa o výsledku na kanáli #labak, ako aj komentárom k úlohe v Jire.

Pri plánovanej údržbe počítača je administrátor povinný takúto údržbu ohlásiť aspoň 24 hodín vopred pomocou komunikačného kanála #labak. Počas ohlásenej údržby je administrátor oprávnený reštartovať počítač. 10 minút pred reštartom je však povinný upozorniť na túto skutočnosť všetkých pripojených členov tímu, a to pomocou príkazu `shutdown /r /t 60 /c „Krátka správa vysvetľujúca dôvod reštartu“`.

A.6 Metodika prenosu projektu

Podľa nasledovných krokov je možné spustiť projekt pod operačným systémom Windows 7 64-bit a vyššie:

1. *Inštalácia kompilátora TDM GCC.* Pre spustenie projektu pod operačným systémom Windows budete potrebovať kompilátor TDM GCC. Ten je dostupný na prevzatie na adrese ². Typ inštalácie treba zvoliť podľa typu operačného systému. Dôležité je počas inštalácie označiť komponent OpenMP v *Components* → *gcc*.



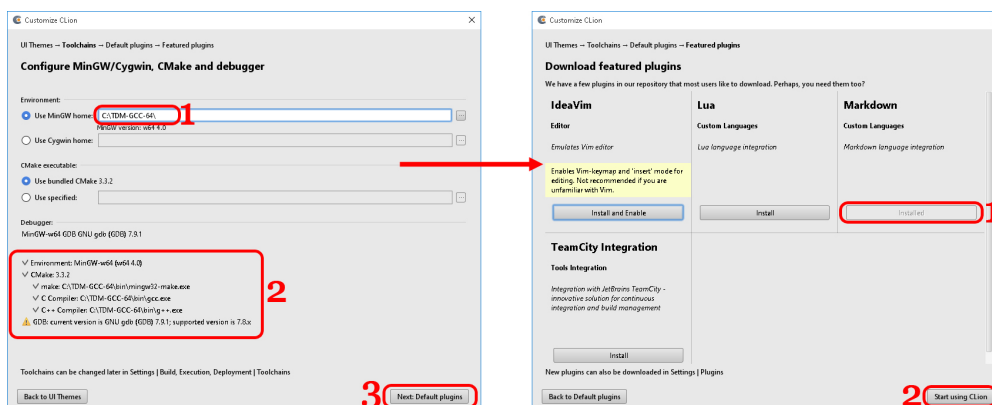
Obr. 3: Inštalácia TDM GCC.

2. *Inštalácia prostredia CLion.* Inštalácia IDE prostredia CLion prebieha klasicky, prostredníctvom grafického rozhrania. Inštalačný súbor je dostupný na adrese ³.
3. *Konfigurácia prostredia CLion.* Po nainštalovaní prostredia a po jeho následnom spustení sa začne prvotná konfigurácia prostredia. Pri konfigurácii je potrebné zadať cestu ku nainštalovanému kompilátoru. Po zadaní správ-

²<http://sourceforge.net/projects/tdm-gcc/files/TDM-GCC%20Installer/tdm64-gcc-5.1.0-2.exe/download>

³<http://download.jetbrains.com/cpp/clion-1.2.1.exe>

nej cesty by mali byť identifikované jednotlivé komponenty. Odporúča sa nainštalovať Markdown a Lua.



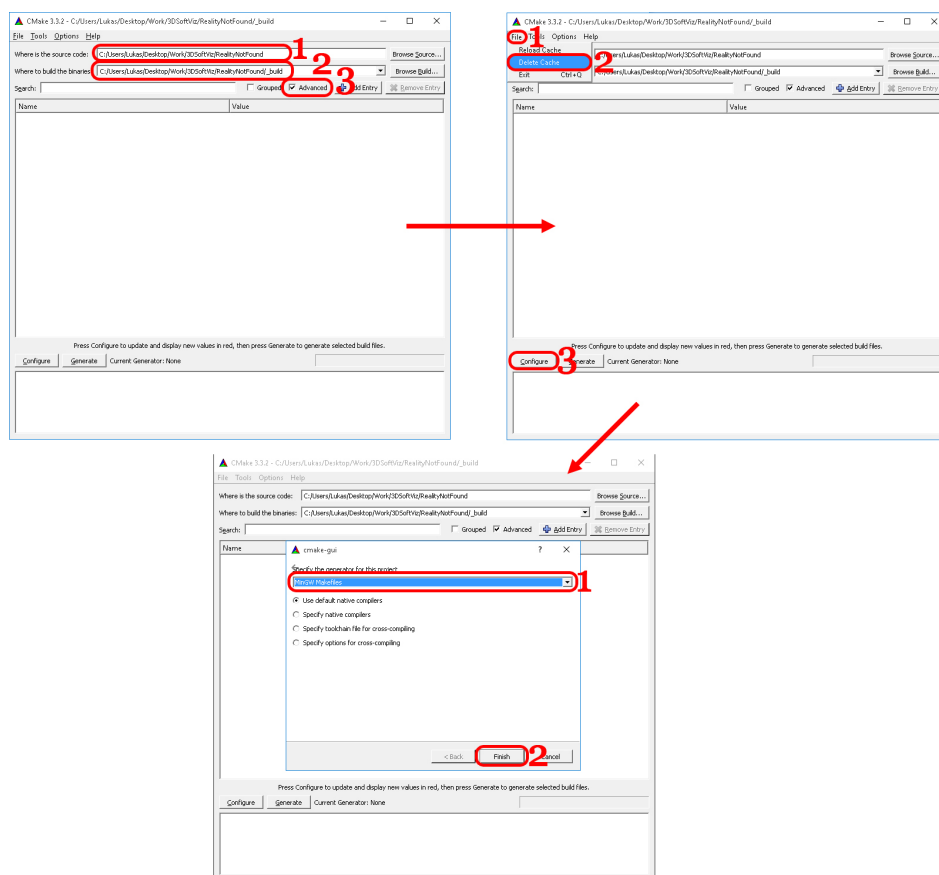
Obr. 4: Konfigurácia prostredia CLion.

4. *Rozbalenie projektu 3DSsoftviz.* Pri rozbalovaní treba dbať na to, aby sa vo výslednej ceste *nenachádzali* medzery. Projekt je potrebné rozbaľiť, napríklad, do priečinka *C:/ (vaša cesta)/3DSsoftviz/*. Vytvorte podpriečinok *_build* a *_install* v *C:/ (vaša cesta)/3DSsoftviz/RealityNotFound*.

5. *Konfigurácia v Cmake.* Otvorte nainštalovaný *Cmake.exe*. Je potrebné zadať priečinok, v ktorom sa nachádzajú zdrojové kódy pomocou *Browse Source*. Je potrebné zadať aj priečinok, do ktorého sa budú ukladať vytvorené súbory počas procesu kompilácie pomocou *Browse Build*. Označte políčko *Advanced*.

Pokiaľ je toto vaša prvá konfigurácia vojdite do *File* → *Delete Cache* a následne kliknite na tlačidlo *Configure*. Malo by sa vám otvoriť nasledujúce okno. V ňom vyberte MinGW kompilátor a kliknite na *Finish*. Podrobný postup je znázornený na obrázku 5.

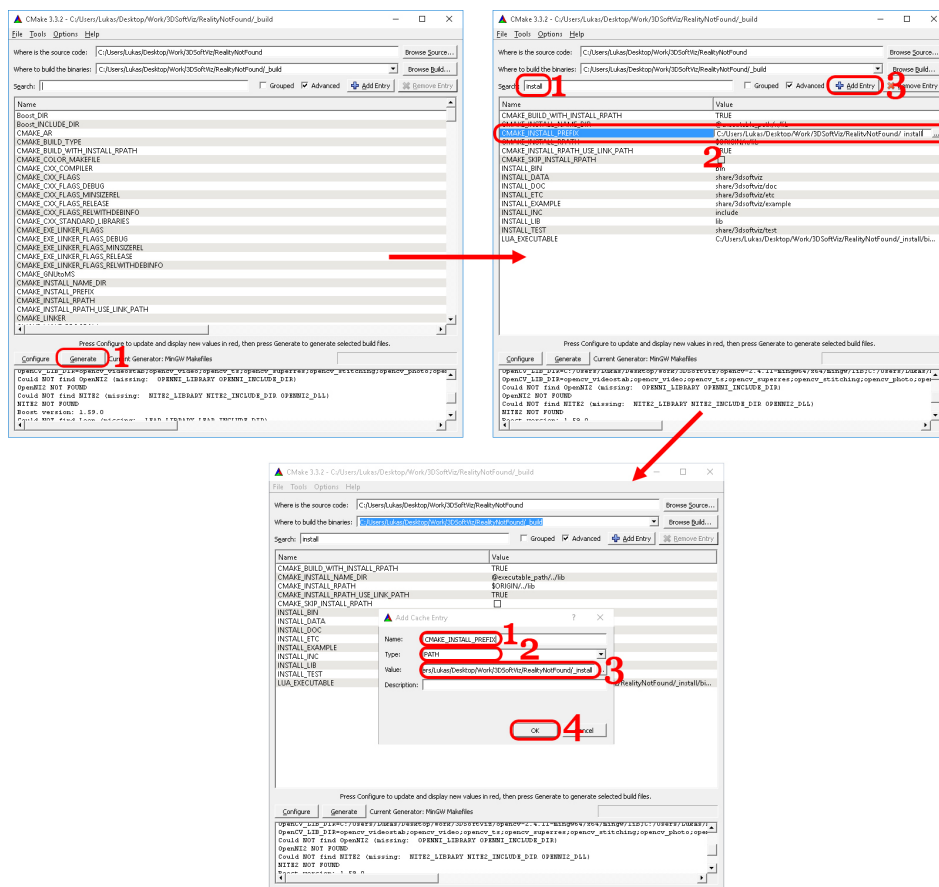
6. *Generovanie v Cmake.* Po vygenerovaní zadajte do vyhľadávania *install* a skontrolujte, či sa tam nachádza položka s názvom *CMAKE_INSTALL_PREFIX* so správnou cestou k priečinku *_install*. Pokiaľ sa tam nenachádza, položku pridajte pomocou tlačidla *Add Entry*.



Obr. 5: Konfigurácia v Cmake.

Otvorí sa dialógové okno. V ňom zadajte názov *CMAKE_INSTALL_PREFIX*, typ zvolíte *PATH* a vyberte cestu k správnej priečinke *_install*. Kliknite OK. Skontrolujte, či v spodnej konzole ide celá kompilácia do vami požadovaných priečinkov. Podrobný postup je opäť znázornený na obrázku 6.

7. *Zostavenie projektu.* Otvorte konzolu cmd. Vojdite do priečinka, ktorý ste si zvolili na kompiláciu. Zadajte príkaz *cmake ..*. Tým otestujete, či vám funguje príkaz *cmake* a spustíte konfiguráciu, ktorú ste vykonali v predošlých krokoch. Následne zadajte príkaz *cmake --build . -- -j12*. Parameter *-- -j12* spustí kompiláciu na 12 logických jadrách procesora v laboratóriu. Po pár minútach by sa mala kompilácia dokončiť bez chýb, ale s výstrahami. Príkaz

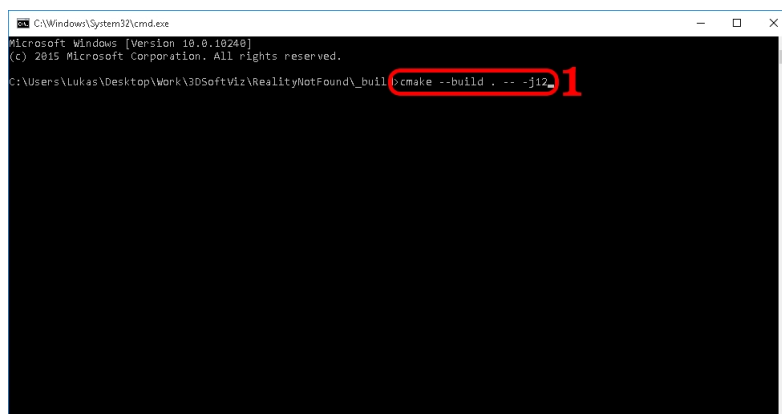


Obr. 6: Generovanie v Cmake.

je znázornený na obrázku 7.

8. *Inštalácia projektu.* Po skompilovaní zadajte do konzoly príkaz `cmake --build . --target install`, ktorý vytvorí výsledný spustiteľný súbor `3dsoftviz.exe` v priečinku `C:/((vaša cesta))/3DSoftviz/RealityNotFound/_install/bin/`, ako vidno na obrázku 8.

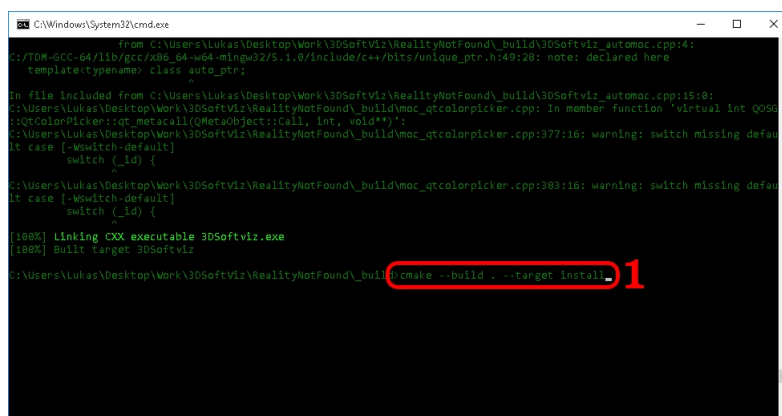
9. *Kopírovanie potrebných knižníc.* Pre spustenie vytvoreného súboru budete potrebovať nakopírovať potrebné knižnice do priečinku `_install/bin`. Nakopírovanie sa dá urobiť viacerými spôsobmi. Buď sa skopírujú všetky knižnice *.dll okrem version.dll do priečinku `C:/((vaša cesta))/3DSoftviz/RealityNotFound/_install/bin/`. Pokiaľ ale nechcete duplikovať 1 GB knižníc, dajú sa skopírovať iba potrebné knižnice pre spustenie 3DSoftvizu. Je vytvorený balíček, ktorý je dostupný



```
C:\Windows\System32\cmd.exe
Microsoft Windows [Version 10.0.10240]
(c) 2015 Microsoft Corporation. All rights reserved.

C:\Users\Lukas\Desktop\Work\3DSoftViz\RealityNotFound\_build>cmake --build . -- -j12 1
```

Obr. 7: Spustenie kompilácie cez príkazový riadok.



```
C:\Windows\System32\cmd.exe
from C:\Users\Lukas\Desktop\Work\3DSoftViz\RealityNotFound\_build\3DSoftViz_automoc.cpp:4:
  C:/TDM-GCC-64/lib/gcc/x86_64-w64-mingw32/5.1.0/include/c++/bits/unique_ptr.h:49:20: note: declared here
    template<typename> class auto_ptr;

In file included from C:\Users\Lukas\Desktop\Work\3DSoftViz\RealityNotFound\_build\3DSoftViz_automoc.cpp:15:0:
  C:\Users\Lukas\Desktop\Work\3DSoftViz\RealityNotFound\_build\moc_qtcolorpicker.cpp: In member function 'virtual int QO
  QtColorPicker::qt_metacall(QMetaObject::Call, int, void**)':
  C:\Users\Lukas\Desktop\Work\3DSoftViz\RealityNotFound\_build\moc_qtcolorpicker.cpp:377:16: warning: switch missing defau
  lt case [-Wswitch-default]
    switch (_id) {
      ^
  C:\Users\Lukas\Desktop\Work\3DSoftViz\RealityNotFound\_build\moc_qtcolorpicker.cpp:383:16: warning: switch missing defau
  lt case [-Wswitch-default]
    switch (_id) {
      ^
[100%] Linking CXX executable 3DSoftViz.exe
[100%] Built target 3DSoftViz

C:\Users\Lukas\Desktop\Work\3DSoftViz\RealityNotFound\_build>cmake --build . --target install 1
```

Obr. 8: Inštalácia projektu cez príkazový riadok.

na PC v laboratóriu v priečinku *C:/Work/ddl.zip* a na tímovom DropBoxe pod názvom *Kniznice pre 3dsoftviz.zip*.

10. *Kompilácia pomocou Clion.* Po kompilácii otvorte rozbalený projekt 3DSoftviz a skontrolujte, či je správne nakonfigurovaný GCC kompilátor vo *File* → *Settings* → *Build* → *Toolchain*. Ak nebol dobre nakonfigurovaný kompilátor, nastavte ho a po kliknutí na OK by sa vám mala začať vytvárať tabuľka symbolov v projekte. To prvotne potrvá zhruba 2 minúty. V pravom hornom rohu sa nachádza položka *Build All*. Z možností vyberte 3DSfotviz a kliknite na *build*. Postup je znázornený na obrázku 9.
11. *Známe nezrovnalosti.* V dodanej verzii existujú problémy pri kompilácii

A.8 Metodika vytvárania úloh v nástroji na evidenciu úloh

Na vytváranie a pridelenie úloh je potrebné otvoriť nástroj Jira⁴. Každý člen tímu sa musí po získaní úlohy zaznačiť a vyplniť popis krokov, ktoré je na danej úlohe potrebné vykonať. Manažér úloh je zodpovedný za správny priebeh šprintu a rozdeľovanie úloh medzi jednotlivých členov.

Slovník pojmov

- *Šprint*. Časový úsek dlhý dva týždne, počas ktorých vývojový tím pracuje na zadaných úlohách a určitej fáze projektu. Cieľom šprintu je dokončiť naplánované úlohy pred koncom šprintu.
- *Scrum*. Agilná vývojová metóda, kde sa na úlohách pracuje iteratívne a úlohy sú rozdelené do viacerých časových úsekov, šprintov.
- *Backlog*. Zoznam úloh, ktoré je potrebné splniť, ale zatiaľ neboli zaradené do šprintu.
- *Epic*. Väčšia sekcia alebo časť úlohy, ktorá je rozdelená do viacerých podúloh. Splnenie *epicu* môže trvať aj viacero šprintov.
- *Issue*. Úloha v Jire.

Pridanie novej úlohy

Pre pridanie novej úlohy je potrebné prihlásiť sa do systému Jira. V hornej lište je potrebné kliknúť na tlačítko *Create*, ktoré otvorí dialógové okno, v ktorom je potrebné vypísať nasledovné údaje o úlohe. Polia označené hviezdičkou sú povinné.

- *Project**. Projekt, ku ktorému patrí daná úloha.
- *Issue Type**. Na výber sú štyri možnosti: Task, Epic, Bug a Story. Pri pridávaní úlohy sa vyberá len z dvoch:
 - *Task*. Nová úloha alebo funkcionálna, na ktorej treba pracovať.

⁴<http://jira.fiit.stuba.sk/secure/Dashboard.jspa>

- *Bug*. Problém, ktorý zabráňuje správne fungovaniu produktu.
- *Summary**. Pomenovanie úlohy. Úlohy sa pomenúvajú v slovenskom jazyku, s úplnou diakritikou, vecne a najviac 5 slovami. Z názvu úloh musí byť zrejmé všetkým členom tímu, o čo v úlohe ide. Pri nesprávnom pomenovaní sa zadávateľ úlohy upozorní na tímovom Slacku v kanáli #jira.
- *Priority*. Priorita úlohy. Na výber je päť možností: Critical, Minor, Major, Trivial a Blocker.
 - *Critical*. Produkt nefunguje, strata dôležitých dát, najvyššia priorita.
 - *Major*. Dôležité dokončiť pre správny chod produktu, vysoká priorita, štandardná hodnota.
 - *Minor*. Problém, prípadne úloha, od ktorej nezávisí úspech šprintu.
 - *Trivial*. Kozmetické problémy, preklepy.
 - *Blocker*. Bez dokončenia tejto úlohy nie je možné začať inú úlohu.
- *Assignee*. Výber člena alebo viacerých členov tímu, ktorým sa úloha prideli.
- *Description*. Krátky popis úlohy. Zdôvodenie motivácie k tejto úlohe.
- *Attachment*. Ak je potrebné k úlohe pridať doplňujúce súbory, pridajú sa sem.
- *Linked Issues*. Ak je úloha prepojená s inou úlohou, prípadne ju blokuje, vyberie sa ich vzťah.
- *Issue*. Tu sa vyberie úloha, na ktorú je vytváraná úloha prepojená.
- *Epic Link*. Epic, ku ktorému sa vzťahuje daná úloha.
- *Sprint*. Šprint, do ktorého má byť vytváraná úloha zaradená. Úlohy sa priraďujú len do budúcich šprintov, aby sa neovplyvnil beh aktuálneho šprintu.

Práca na úlohe a vyriešenie úlohy

Keď ide člen tímu pracovať na pridelenej úlohe, úlohu si otvorí. Do komentáru v krátkosti napíše, čo ide robiť, čo je cieľom a kedy je úloha dokončená. Stav úlohy zmení na *In progress*.

Po ukončení úlohy člen tímu, ktorý na úlohe pracoval, napíše ďalší komentár. Obsahom komentáru bude krátke zhrnutie toho, čo spravil, aby ostatní členovia mohli nájsť výsledok jeho práce. Ak sa vyskytli nejaké problémy počas plnenia úlohy, treba opísať, ako boli vyriešené. Stav úlohy je po pridaní komentára potrebné zmeniť na *Done*.

A.9 Metodika komunikácie v tíme

Touto metodikou sa riadia všetci členovia tímu. Cieľom je určiť jasné pravidlá pre komunikáciu v tíme, aby sa uľahčila spolupráca na projekte.

Slovník pojmov

- *Slack*. On-line komunikačný nástroj.
- *Jira*. Nástroj na manažment v tíme.
- *Google Groups*. Skupiny spoločnosti Google.
- *Dropbox*. Úložisko dát.

Komunikačné kanály

- *Slack*. Umožňuje chat, zdieľanie dokumentov a vytvorenie rôznych kanálov. V súčasnosti je vytvorených 14 kanálov. Každý kanál je zameraný na určitú tému. Prehľad kanálov je zobrazený v tabuľkách [4](#). Pravidlá:
 - Každý člen tímu si každý večer prečíta nové správy, aby zostal informovaný.
 - Pokiaľ chce notifikovať všetkých členov tímu, použije funkciu @channel.

- Pokiaľ chce notifikovať len niektorých členov tímu, použije @menoClenaTímu.
- Ak chce vytvoriť kanál, prejedná to s ostatnými členmi tímu v kanále general alebo na tímovom stretnutí.
- *Dropbox.* Priečink Dropbox slúži na zdieľanie dokumentov v tíme. Pravidlá:
 - Ak niektorý člen tímu nájde niečo užitočné, nahrá to do úložiska a informuje o tom ostatných členov prostredníctvom kanálu #technologie alebo #general na Slacku.
 - Každý člen vopred skontroluje, či tam rovnaký súbor už nie je, aby nevznikali duplicity.
- *E-mailová komunikácia.* E-mailová komunikácia je prevažne používaná na komunikáciu s vedúcim tímu. Pravidlá:
 - Je zakázané používať skrytú kópiu.
 - Na kontaktovanie celého tímu je potrebné použiť emailovú adresu *tim_16@googlegroups.com*.
 - Medzi adresátov e-mailu treba zahrnúť všetkých členov, ktorých sa téma e-mailu týka.
 - Je potrebné uviesť predmet mailu, vhodné je do hranatých zátvoriek uviesť oblasť, ktorej sa to týka, napr. [kinect] a v stručnosti objasniť problém.
 - K prílohám je vždy nutné napísať komentár.
 - Dlhší e-mail je nutné deliť na odseky.
- *Kalendár.* Do kalendára treba zaznamenávať dôležité termíny (odovzdania, kontrolné body), časy prednášok a neprítomnosť členov tímu. Rovnako je možné zaznačiť tam dôležité termíny z iných predmetov. Keďže členovia tímu sú z rôznych odborov, takto vzniká možnosť vedieť, kedy je kto vyťažený viac. Prispievať môže každý člen tímu.

- *Jira*. Komunikácia v Jire pozostáva z informovania tímových kolegov prostredníctvom opisov a komentárov k úlohám. Pravidlá:
 - Pri pridaní úlohy do Jiry vždy vyplniť pole „Description“.
 - Pri každej zmene stavu pridať komentár.
 - Pri ukončení úlohy pridať komentár.
 - Komentáre obsahujú informáciu o tom, kde výsledný produkt úlohy nájdeme. Príkladom je odkaz na *Github.com*.
 - K prílohám je vždy nutné napísať komentár.
 - Dlhší e-mail je nutné deliť na odseky.
- *Tímové stretnutia*. Komunikácia je nevyhnutnou súčasťou tímových stretnutí. Pravidlá:
 - Komunikácia prebieha podľa vopred určeného programu.
 - Komunikáciu vedie vždy jeden člen tímu.
 - Stretnutí sa zúčastňujú všetci členovia tímu. V prípade, že niekto nemôže prísť, je potrebné o tom informovať Scrum mastera.
 - Výstupy zo stretnutia sa zaznamenávajú v zápisnici.
 - K prílohám je vždy nutné napísať komentár.
 - Dlhší e-mail je nutné deliť na odseky.

Číslo	Názov	Popis
1	3dsoftviz	Tento kanál slúži na komunikáciu týkajúcu sa projektu 3Dsoftviz, napríklad problémy s kompiláciou projektu.
2	dokumentacia	Tento kanál slúži na komunikáciu ohľadom dokumentácie, napríklad, dohadovanie sa, kto napíše ktorú časť, čomu sa budeme venovať v dokumentácii, upozorňovanie či diskutovanie pravidiel.
3	general	Základný kanál, ktorý slúži na komunikáciu, pre ktorú ešte nebol vytvorený vlastný kanál. Jednоразové otázky, pre ktoré nie je potrebné zakladať nový kanál. Využíva sa rovnako na dohodu, či vytvoríme nový kanál alebo hlasovania, ktoré neboli uskutočnené na tímovom stretnutí. Sem prichádzajú upozornenia na udalosti v tímovom kalendári.
4	Jira	Tento kanál slúži na komunikáciu týkajúcu sa nástroja na manažment, Jiry. Riešia sa problémy, ako napríklad, ako zaradiť úlohu ako sub-task k nejakej story.
5	labak	Tento kanál slúži na dohadovanie, kedy a kto navštívi laboratórium a bude pracovať na projekte. Rovnako sa tu hlasuje o softvéri, ktorý je potrebný nainštalovať na PC v laboratóriu.
6	programovanie	Kanál, kde sa preberajú pokyny ku programovaniu.
7	random	Základný kanál, ktorý aktuálne nie je nevyužívaný.

Tabuľka 4: Zoznam kanálov v Slacku.

Číslo	Názov	Popis
8	retrospektívy	Tento kanál slúži na diskusiu retrospektív po ukončení šprintu.
9	rozdelenie-uloh	Kanál sa využíva na rozdelenie úloh, ktoré sa neprerokovali na stretnutí, pretože vznikli po ňom a je potrebné ich vyriešiť rýchlo. Takisto sa tu zverujú všetky úlohy, ktoré boli odsúhlasené na stretnutí.
10	technologie	Tento kanál slúži na zdieľanie rôznych užitočných zdrojov, v ktorých sú spomínané technológie potrebné k projektu.
11	webpage	Kanál určený na komunikáciu ohľadom webovej stránky. Slúži napríklad na pripomienky, ak chýbajú aktuálne informácie alebo nahranie súborov, ktoré je potrebné pridať na stránku.
12	zapisnice	Tento kanál slúži na overenie zápisníc.
13	poster	Kanál, ktorý sme využili na začiatku tímového projektu. Nachádzali sa tu návrhy plagátov, ku ktorým prebiehalo hlasovanie. V súčasnosti bol tento kanál archivovaný.
14	english	Kanál slúži na pomoc s angličtinou medzi sebou, keďže dokumentácia k produktu je písaná v anglickom jazyku.

Tabuľka 5: Zoznam kanálov v Slacku.

B Zápisky

Príloha obsahuje zápisy zo strenutí počas prvých troch šprintov. Tieto zápisy obsahujú opis strenutia, zúčastnených, dátum a čas strenutia, a úlohy na ďalší týždeň.

Zápis zo stretnutia č. 1

Tím č. 16

Dátum a čas stretnutia: 21.9.2015, 16:30

Miesto stretnutia: FIIT STU, U120

Prítomní: Erik Bujna, Mário Csaplár, Michal Dobai, Lukáš Doubravský, Lenka Kutlíková, Martin Petráš, Ondrej Vlček

Zapisovateľ: Lenka Kutlíková

Téma stretnutia:

Zoznámenie sa, úvod do problematiky

Opis stretnutia:

1. Stretnutie prebehlo po prednáške z tímového projektu. Stretli sme sa ako tím s vedúcim, ktorý nás uviedol do danej problematiky, oboznámil nás s organizáciou tohto predmetu. Taktiež sme si dohodli miesto a čas ďalších stretnutí (streda 17:00).
2. Na stretnutí prebehla voľba SCRUM mastra. Stala sa ním Lenka Kutlíková.
3. Na konci stretnutia nám vedúci tímového projektu odporučil technológie a knižnice, s ktorými sa máme oboznámiť.

Rozdelenie úloh:

Číslo úlohy	Popis úlohy	Vypracuje
1.1	Navrhnutie plagátu	Erik Bujna, Lukáš Doubravský, Ondrej Vlček
1.2	Vytlačenie plagátu	Ondrej Vlček
1.3	Vytvorenie webovej stránky	Martin Petráš
1.4	Zozbieranie termínov	Lenka Kutlíková
1.5	Oboznámenie sa s technológiami	Všetci

Zápis zo stretnutia č. 2

Tím č. 16

Dátum a čas stretnutia: 30.9.2015, 17:00

Miesto stretnutia: FIIT STU, 1.27

Prítomní: Erik Bujna, Mário Csaplár, Michal Dobai, Lukáš Doubravský, Lenka Kutlíková, Martin Petráš, Ondrej Vlček

Zapisovateľ: Lenka Kutlíková

Téma stretnutia:

Dohoda o organizácii stretnutí, oboznámenie sa s cieľmi na zimný semester

Opis stretnutia:

1. Na začiatku stretnutia nás vedúci oboznámil s termínom príchodu hardvéru potrebného k našemu projektu. Odporučil nám nástroje na manažment v tíme: Jira (Confluence), Redmine, GitHub.
2. Dohodli sme sa na dokumentácii – budeme ju tvoriť v LateXu a bude písaná v angličtine.
3. Oboznámili sme vedúceho s úlohami, ktoré sa nám od minulého týždňa podarilo splniť: začali sme vytvárať webovú stránku, vytvorili sme plagát.
4. Stretli sme sa s tímom 4, keďže s nimi budeme spolupracovať.
5. Určili sme si termín začiatku implementácie – 5. týždeň.
6. Určili sme si ciele na zimný a letný semester. V zimnom semestri by sme mali rozbehať záznam z kinectu a rozbehať modul pre stereo zobrazenie – oculus. V letnom semestri musíme integrovať (GUI) a implementovať protokol spolu s tímom č.4.
7. Určili sme si koordinátora GitHubu – Erik Bujna. Na odporúčanie vedúceho tímu sme si zvolili jednotného Git klienta – Source tree.

Rozdelenie úloh:

Číslo úlohy	Popis úlohy	Vypracuje
1.1	Rozhodnúť sa o účasti na TP, IIT.src, SCCG	Všetci
1.2	Oboznámiť sa so štandardom C++11	Všetci
1.3	Zvoliť Document Mastra	Všetci
1.4	Vytvoriť web a doplniť obsah (zadanie + členov tímu)	Martin Petráš
1.5	Skompilovať existujúci projekt	Všetci
1.6	Vytvoriť LaTeX template	Lenka Kutlíková

1.7	Pozrieť si introduction k Cmake	Ondrej Vlček
1.8	Pozrieť OpenSceneGraph 3.2.1. + plugin na Oculus rift	Mário Csaplár
1.9	Nájsť multiplatformovú knižnicu pre Kinect	Erik Bujna
1.10	Vybrať a nainštalovať si nástroje pre manažment v tíme	Všetci

Zápis zo stretnutia č. 3

Tím č. 16

Dátum a čas stretnutia: 7.10.2015, 18:00

Miesto stretnutia: FIIT STU, 1.27

Prítomní: Erik Bujna, Mário Csaplár, Michal Dobai, Lukáš Doubravský, Lenka Kutlíková, Martin Petráš, Ondrej Vlček

Zapisovateľ: Lenka Kutlíková

Téma stretnutia:

Príprava na začiatok šprintov

Opis stretnutia:

1. Počas tohto stretnutia sme sa pripravovali na začiatok prvého šprintu. Dohodli sme sa, že cieľom bude rozbehať SDK Oculus Riftu aj za predpokladu, že nepríde nový hardvér.
2. Dohodli sme sa na nástroji na manažment v tíme – Jira.
3. Vedúci tímu nám odporučil preštudovať si odporúčania ohľadom Git Flow.
4. Na konci stretnutia nás upozornil, že v intervale každé 2 týždne by mal byť aktualizovaný hlavný Git repozitár.

Rozdelenie úloh:

Číslo úlohy	Popis úlohy	Vypracuje
1.1	Rozbehať Jiru	Všetci
1.2	Skompilovať existujúci projekt	Všetci
1.3	Preštudovať si Git Flow	Erik Bujna

Zápis zo stretnutia č. 4

Tím č. 16

Dátum a čas stretnutia: 14.10.2015, 17:00

Miesto stretnutia: FIIT STU, 1.31, Jobsovo softvérové štúdio

Prítomní: Erik Bujna, Mário Csaplár, Michal Dobai, Lukáš Doubravský, Lenka Kutlíková, Martin Petráš, Ondrej Vlček

Zapisovateľ: Lenka Kutlíková

Téma stretnutia:

Problémy s kompiláciou existujúceho projektu

Opis stretnutia:

1. Na začiatku stretnutia sme si zvolili cieľ prvého šprintu – rozbehať SDK Oculus Riftu.
2. V ďalšej časti stretnutia sme prebrali čo všetko je potrebné pripraviť v laboratóriu v škole – nainštalovať Windows a potrebný softvér, rozbehať existujúci projekt.
3. V poslednej časti stretnutia sme riešili kompilačné problémy, ktoré nastali v existujúcom projekte.

Rozdelenie úloh:

Číslo úlohy	Popis úlohy	Vypracuje
4.1	Kontaktovať tím č.4 s problémami s kompiláciou	Lenka Kutlíková
4.2	Nainštalovať Windows a potrebný softvér na školský PC	Ondrej Vlček
4.3	Rozbehať SDK	

Zápis zo stretnutia č. 5

Tím č. 16

Dátum a čas stretnutia: 21.10.2015, 18:00

Miesto stretnutia: FIIT STU, 1.31, Jobsovo softvérové štúdio

Prítomní: Erik Bujna, Mário Csaplár, Michal Dobai, Lukáš Doubravský, Lenka Kutlíková, Martin Petráš, Ondrej Vlček

Zapisovateľ: Lenka Kutlíková

Téma stretnutia:

Ukončenie prvého šprintu

Opis stretnutia:

1. Na začiatku stretnutia sme si zvolili ciele druhého šprintu – nájsť miesto, kde sa azapojíme v existujúcom projekte, vyskúšať si Oculus, zoznámiť sa s projektom.
2. V ďalšej časti stretnutia sme sa dohodli, že sa TP cupu nezúčastníme.
3. Ďalej sme rozbehali existujúci projekt za pomoci vedúceho tímu.

Rozdelenie úloh:

Číslo úlohy	Popis úlohy	Vypracuje
5.1	Naučiť sa pracovať s knižnicou pre oculus	Všetci
5.2	Naučiť sa pracovať s knižnicou pre kinect	Všetci
5.3	Vytvoriť stereoskopický pohľad projektu pre oculus	Všetci

Zápis zo stretnutia č. 6

Tím č. 16

Dátum a čas stretnutia: 28.10.2015, 17:00

Miesto stretnutia: FIIT STU, 1.31, Jobsovo softvérové štúdio

Prítomní: Erik Bujna, Mário Csaplár, Michal Dobai, Lukáš Doubravský, Lenka Kutlíková, Martin Petráš

Zapisovateľ: Lenka Kutlíková

Téma stretnutia:

Priebeh druhého šprintu, nedokončené úlohy

Opis stretnutia:

1. Na stretnutí sme boli poučení vedúcim projektu o nedostatkoch v nástroji na manažment tímu – v Jire. Do budúceho stretnutia je potrebné ich napraviť (pridať komentáre na začiatku a konci tasku, vytvoriť stories).
2. Do budúceho týždňa je potrebné vyhodnotiť prvý šprint.
3. Dohodli sme sa na cieľoch druhého šprintu – zoznámiť sa s existujúcim projektom.
4. Neskôr sme riešili problémy pri buildovaní.
5. Na konci sme si rozdelili tasky a pridali ich do Jiry.

Rozdelenie úloh:

Číslo úlohy	Popis úlohy	Vypracuje
6.1	Zistiť obsah dokumentácie	Lenka Kutlíková
6.2	Vyhodnotiť prvý šprint	Všetci
6.3	Vytvoriť exporty z Jiry a pridať ich na stránku	Lenka Kutlíková, Michal Dobai
6.4	Spraviť prototyp v OpenSceneGraph	Mário Csaplár
6.5	Prispôsobiť build systému na účely projektu	Ing. Peter Drahoš, PhD.

Zápis zo stretnutia č. 7

Tím č. 16

Dátum a čas stretnutia: 4.11.2015, 17:30

Miesto stretnutia: FIIT STU, 1.31, Jobsovo softvérové štúdio

Prítomní: Erik Bujna, Mário Csaplár, Michal Dobai, Lukáš Doubravský, Lenka Kutlíková, Martin Petráš, Ondrej Vlček

Zapisovateľ: Lenka Kutlíková

Téma stretnutia:

Plánovanie tretieho šprintu, pridelenie úloh

Opis stretnutia:

1. Na začiatku stretnutia sme si vyhodnotili čo sme stihli a nestihli v druhom šprinte. V tomto šprinte prebehlo zoznamovanie sa s projektom 3Dsofviz a OpenSceneGraph-om. Vzhľadom na niektoré nedostatky v Jire, task vytvoriť exporthy sa posunul do ďalšieho šprintu.
2. Oboznámili sme sa s tým, ktoré náležitosti nemôžu chýbať v dokumentácii.
3. Na stretnutí prebehlo hlasovanie o spôsobe zdieľania potrebných dokumentov na štúdium – zvolil sa Dropbox.
4. Dohodli sme sa ako budeme postupovať pri vytváraní kanálov na Slacku.
5. Odhlasovali sme jazyk dokumentácie, dokumentácia riadenia – slovensky, dokumentácia produktu – anglicky. Vedúcim tímu nám bolo odporúčené, aby čo najviac dokumentácie produktu bolo generovanej z kódu.
6. V ďalšej časti stretnutia sme si rozdelili tasky na ďalší šprint a určovali prislúchajúce storypointy.

Rozdelenie úloh:

Číslo úlohy	Popis úlohy	Vypracuje
7.1	Vytvoriť zápisnicu zo stretnutia	Lenka Kutlíková
7.2	Nahodiť zápisnice na web	Michal Dobai
7.3	Vytvoriť exporthy z Jiry a pridať ich na stránku	Lenka Kutlíková, Michal Dobai
7.4	Vytvoriť základ manažérskej dokumentácie	Mário Csaplár
7.5	Vytvoriť základ dokumentácie k inžinierskemu dielu	Mário Csaplár

7.6	Zriadiť remote prístup	Michal Dobai, Lukáš Doubravský
7.7	Vytvoriť retrospektívy za šprint 1 a 2	Lenka Kutlíková
7.8	Kontaktovať ownera Jiry so žiadosťou o admin prístup	Lenka Kutlíková

Zápis zo stretnutia č. 8

Tím č. 16

Dátum a čas stretnutia: 11.11.2015, 17:00

Miesto stretnutia: FIIT STU, 1.31, Jobsovo softvérové štúdio

Prítomní: Erik Bujna, Mário Csaplár, Michal Dobai, Lukáš Doubravský, Lenka Kutlíková, Martin Petráš, Ondrej Vlček

Zapisovateľ: Lenka Kutlíková

Téma stretnutia:

Priebeh tretieho šprintu, tvorba dokumentácie

Opis stretnutia:

1. Na začiatku stretnutia sme si vyhodnotili stav úloh po jednom týždni – Erikovi sa podarilo rozbehnúť knižnicu aj kostrový model v Kinecte, Martin naďalej pracuje s Oculusom. Ondrej nainštaloval Windows v laboratóriu. Michal a Lukáš zriadili vzdialený prístup do laboratória a vytvorili každému vlastný účet. Rozbehali projekt na školskom počítači. Lenka vytvorila retrospektívy za prvé dva šprinty.
2. Vytvorili sme hodnotenie tímu, kde sme spomenuli kto čo spravil za prvé dva šprinty, koľko vynaložil námahy a poslali sme to vedúcemu tímu.
3. Rozdelili sme si úlohy pri písaní dokumentácie.
4. V ďalšej časti stretnutia prišiel vedúci tímu, predstavili sme mu dosiahnuté výsledky, opýtali sme sa ho na nejasnosti v Jire – od ďalšieho šprintu používame epic miesto story, nesplnené úlohy dať archivovať.

Rozdelenie úloh:

Číslo úlohy	Popis úlohy	Vypracuje
8.1	Vytvoriť zápisnicu zo stretnutia	Lenka Kutlíková
8.2	Nahodiť zápisnice na web	Michal Dobai
8.3	Vytvoriť exporty z Jiry a pridať ich na stránku	Lenka Kutlíková, Michal Dobai
8.4	Napísať dokumentáciu k produktu	Všetci
8.5	Napísať dokumentáciu k riadeniu	Všetci

C Exporty z Jiry

Táto príloha obsahuje export z nástroja Jira po 9. týždni.

Key	Summary	Status	Priority	Assignee	Description	Story
VRCOLLAB-53	Vytvoriť základ manažérskej dokumentácie	Done	Major	Mário Csaplár	Na základe požiadaviek na obsah dokumentácie treba vytvoriť základný obsah.	Dokumentácia
VRCOLLAB-54	Vytvoriť základ dokumentácie k inžinierskemu dielu	Done	Major	Mário Csaplár	Na základe požiadaviek na obsah dokumentácie treba vytvoriť základný obsah.	Dokumentácia
VRCOLLAB-57	Tvorba zápisníc	In progress	Major	Lenka Kutlíková	Task skončí napísaním a nahodením všetkých zápisníc zo stretutí počas šprintu 3.	Dokumentácia
VRCOLLAB-51	Vytvoriť Jira exporthy	In progress	Major	Lenka Kutlíková	Podľa podmienok predmetu je potrebné mať na stránke exporthy z nástroj ana manažment tímu - v našom prípade Jiry.	Dokumentácia
VRCOLLAB-63	Vytvoriť retrospektívu za šprint 1 a 2	Done	Major	Lenka Kutlíková	Retrospektívy spísané a pridané na web.	Dokumentácia
VRCOLLAB-68	Napísať metodiku ohodnotenia taskov	Done	Major	Ondrej Vlček	Cieľom tasku je vytvoriť metodiku opísať proces ako sa dohadujeme na storypointoch.	Dokumentácia
VRCOLLAB-69	Napísať metodiku tvorby zápisníc	Done	Major	Lenka Kutlíková	Cieľom je napísať presný postup ako sa tvoria zápisnice - spísať metodiku.	Dokumentácia
VRCOLLAB-70	Spísať metodiku práce s JIROU	Done	Major	Ondrej Vlček		Dokumentácia
VRCOLLAB-74	Napísať metodiku komunikácie	Done	Major	Lenka Kutlíková		

VRCOLLAB-76	Napísať úvod k dokumentácii riadenia a inžinierskemu dielu	Done	Major	Prvá kapitola v oboch dokumentáciách.	Mário Csaplár	Dokumentácia
VRCOLLAB-77	Metodika písania dokumentácie	Done	Major	Napísať metodiku, podľa ktorej budú všetci členovia tímu schopní písať dokumentáciu.	Mário Csaplár	Dokumentácia
VRCOLLAB-78	Ciele pre zimný semester	Done	Major	Tretia kapitola v dokumentácii k produktu.	Mário Csaplár	Dokumentácia
VRCOLLAB-79	Napísať metodiku pre komunikáciu v tíme	Done	Major		Ondrej Vlček	Dokumentácia
VRCOLLAB-81	Vytvoriť metodiku písania kodu			Cielom je napísať metodiku k písaniu kodu (konvencie pre názvy premenných, funkcií, tried, parametrov funkcií)	Martin Petráš	Dokumentácia
VRCOLLAB-59	Vytvoriť kostrový model využitím Kinectu SDK	Done	Major		Erik Bujna	Práca s kinectom
VRCOLLAB-60	Naučiť sa pracovať s Kinect SDK for Windows	Done	Major	Rozbehať SDK, pozrieť príklady	Erik Bujna	Práca s kinectom
VRCOLLAB-62	Vytvoriť hĺbkovú reprezentáciu využitím Kinect SDK	Done	Major		Erik Bujna	Práca s kinectom
VRCOLLAB-66	Kontaktovať ownera Jiry so žiadosťou o	Closed	Major		Lenka Kutlíková	

	adminsý prístup					
VRCOLLAB-52	Nahodiť Jira exporty na stránku	To do	Major		Michal Dobai	Webová stránka
VRCOLLAB-64	Nahodiť zápisnicu na web	To do	Major	Nahrať na web vytvorené záisnice zo stretnutí	Michal Dobai	Webová stránka
VRCOLLAB-65	Zriadiť remote prístup na PC v laboratóriu	Done	Major	Cieľom je vymyslieť a zrealizovať najlepší spôsob ako umožniť vzdialený prístup k počítaču dostupným u v laboratóriu aby mohli jednotliví členovia tímu pracovať na implementačných úlohách aj na diaľku.	Lukáš Doubravský	
VRCOLLAB-67	Práca s oculusom	Done	Major	cieľom v tomto sprinte je vytoriť prototyp integrujúci OSG a Oculus	Martin Petráš	Práca s oculusom
VRCOLLAB-75	Napísať report o 3DSOFTVIZE	Done	Major	Projekt sa nedá skompilovať pod MS Windows. Je potrebné odskúšať kompiláciu a napísať	Mário Csaplár	

				správu o zisteniach.		
--	--	--	--	-------------------------	--	--