

# Tímový projekt

## Dokumentácia

---

Číslo tímu: 12

Názov tímu: 3DSpaceGen

Členovia: Ján Antal, Marek Drahoš, Matej Mikuš, Dominik Mazák, Peter Pápay, Andrej Pisarčík, Adam poperník

Vedúci tímu: Ing. Karol Rástočný, PhD.

Ak. rok: 2018/19

# Obsah

---

|   |    |
|---|----|
| <b>1. Riadenie projektu</b> .....         | 3  |
| 1.1 Big Picture .....                     | 3  |
| 1.1.1 Role členov .....                   | 3  |
| 1.1.2 Aplikácie manažmentov .....         | 4  |
| 1.1.3 Sumarizácia šprintov .....          | 4  |
| 1.2 Motivačný dokument .....              | 5  |
| 1.2.1 Tím .....                           | 5  |
| 1.2.2 Motivácia .....                     | 5  |
| 1.2.2 Zoradenie tém podľa priority .....  | 6  |
| 1.2.3 Rozvrh tímu .....                   | 7  |
| 1.3 Metodiky .....                        | 7  |
| 1.3.1 Metodika komunikácie .....          | 7  |
| 1.3.2 Metodika manažmentu tímu .....      | 8  |
| 1.3.3 Metodika verziovania projektu ..... | 8  |
| 1.4 Webové sídlo .....                    | 10 |
| <b>2. Inžinierske dielo</b> .....         | 11 |
| 2.1 Big picture .....                     | 11 |
| 2.1.1 Globálne ciele projektu .....       | 11 |
| 2.1.2 Celkový pohľad na systém .....      | 11 |
| 2.2 Moduly systému .....                  | 12 |
| 2.2.1 Modul virtuálnej reality .....      | 12 |
| 2.2.2 Importér .....                      | 13 |
| 2.2.3 Rest API Servis .....               | 13 |
| 2.2.4 Moduly inferencie .....             | 14 |
| Prílohy                                   |    |
| A: Export evidencie úloh .....            | 15 |

# 1. Riadenie projektu

---

## 1.1 Big Picture

Dokument vznikol ako výsledok práce tímového projektu 3DSpaceGen po prvom semestri, za účelom zdokumentovania celkovej práce na projekte. Primárnym cieľom tohto dokumentu je opísať manažment tímu, detailnú špecifikáciu softvérového produktu bližšie popisuje dokument inžinierske dielo.

Tento dokument popisuje podrobný priebeh riadenia tímového projektu 3DSpaceGen po prvých troch šprintoch s využitím metodiky Scrum. V dokumente sa nachádzajú podrobne popísané manažérske role jednotlivých členov tímu, vzhľadom na ich úlohy. Dokument ďalej obsahuje popis šprintov, podiel práce každého z členov tímu, metodiky práce a záverečný pohľad na jednotlivé retrospektívy. Súčasťou dokumentu je aj export úloh získaný z nástroja TFS, ktorý bol použitý pre jednoduché a rýchle riadenie tímu.

### 1.1.1 Role členov

Tím tvorí 7 členov, product owner Vladimír Hlaváček a vedúci tímu Karol Rástočný. Pre manažment tímu používame nástroj Microsoft Team Foundation Server (TFS). S týmto nástrojom interaguje každý člen tímu a každý má aj isté zodpovednosti. Špecifikácia metodiky práce je definovaná v sekcii Metodiky tohto dokumentu.

Adam Poperník: Frontend programmer, C# programemer, UI/UX designer, webpage admin  
Adam je zodpovedný za webovú stránku projektu. Okrem toho sa podieľa na vývoji VR aplikácie.  
V dokumentácii vypracoval: Implementácia

Ján Antal: Frontend programmer, C# programmer, VR developer  
Ján sa hlavne venuje primárne vývoju VR aplikácie v HTC Vive a pracuje najmä s Unity.  
V dokumentácii vypracoval: Analýza

Andrej Pisarčík: Backend python programmer, research specialist, Git master  
Andrej sa venuje testovaniu existujúcich riešení a open source programov na testovanie GANiek.  
Okrem toho sa venuje aj implementácii Flask serveru v pythone.  
V dokumentácii vypracoval: Úvod, Návrh

Dominik Mazák: Backend programmer, Server Maintenance, DevOps  
Dominik je zodpovedný za administráciu školského serveru a Docker containerov. Zároveň je zodpovedný aj za DevOps ako CI (Continuous Integration ... ).  
V dokumentácii vypracoval: Aplikácie manažmentov, Globálna retrospektíva

Matej Mikuš: Backend python programmer, research specialist  
Matej je hlavný researcher tímu a venuje sa vyhľadávaniu nových riešení a rôznych open source programov v oblasti GAN sietí. Okrem toho sa podieľa pri vývoji Flask serveru.  
V dokumentácii vypracoval: Testovanie, Motivačný dokument

Peter Pápay: SCRUM Master, python programmer  
Peter je SCRUM master a je zodpovedný za celkovú prácu s nástrojom pre manažment tímu. Okrem toho programuje v pythone a venuje sa implementácii GAN siete.  
V dokumentácii vypracoval: Role členov, Sumarizácia Šprintov, Celkový pohľad na systém

Marek Drahoš: Backend programmer, Data pipeline developer

Marek je backend programátor a je zodpovedný za tok dát v projekte.

V dokumentácii vypracoval: Úvod, Globálne ciele projektu, Export úloh a Review

## 1.1.2 Aplikácie manažmentov

### Slack

Slack je sada kolaboračných služieb a nástrojov uložených na cloude. Medzi hlavné funkcie Slacku patria:

- Perzistentné chatové miestnosti
- Súkromné chatové miestnosti
- Súkromné správy
- Vytváranie tímov
- Zdieľanie súborov
- Vyhľadávanie

V rámci nášho projektu je Slack našim hlavným spôsobom komunikácie a organizácie v rámci tímu. Využili sme funkciu vytvárania tímov, ktorú Slack poskytuje, a vytvorili sme si vlastný tím, kde sa nachádzajú všetci členovia tímu vrátane vedúceho projektu a product owner-a. Ďalej Slack používame aj na príjem notifikácií. Tak isto máme vytvorených viacero špecifických komunikačných kanálov určených pre užšiu podskupinu členov tímu, ktorí pracujú na rovnakej časti projektu. V týchto špecifických kanáloch sa členovia Tieto kanály sú bližšie popísané v metodike komunikácie.

### Team Foundation Server

Team Foundation Server (ďalej už len TFS) je produkt od spoločnosti Microsoft, ktorý poskytuje služby ako manažovanie zdrojového kódu, verziovanie, reporting, automatické buildovanie a deployment, manažment projektu. Celkovo je TFS našim základným prvkom agilného vývoja softvéru. Činnosti, ktoré v TFS vykonávame v rámci nášho projektu sú vytváranie user stories, kde zdefinujeme určitú časť funkcionality alebo určitý cieľ, ktorý chceme dosiahnuť. Ďalej si rozbijeme tieto user stories na jednotlivé úlohy, ktoré sa priradia členom tímu. Následne si vytvoríme plán šprintu. Na základe ohodnotenia zložitosti jednotlivých user stories a tieto si následne pridelieme do šprintu. Toto nám umožňuje efektívne plánovať jednotlivé šprinty a sledovať priebeh práce jednotlivých členov tímu na svojich úlohách.

Na serveri TFS sme si založili našu vlastnú Wiki, ktorú počas práce na projekte neustále rozširujeme. Služi nám ako jednotné miesto pre zbieranie všetkých relevantných znalostí k téme projektu. TFS sme taktiež použili aj na vytvorenie automatického releasovania softvéru na cieľové platformy. K nástroju TFS sme napísali aj metodiku, ktorá sa nachádza tu.

### Google kalendár

Nástroj google kalendár používame na plánovanie. Náš tím má založený jednotný kalendár, v ktorom si vyhradíme presný čas určený na prácu na jednotlivých úlohách

## 1.1.3 Sumarizácia šprintov

V projekte sme absolvovali troje dvojtýždňové šprinty. Po prvom šprinte sme v retrospektívne identifikovali malé problémy v tíme (spôsob komunikácie, práca s TFS), ktoré sa nám v druhom šprinte podarilo opraviť.

### Šprint 1 - Londýn

Prvotný šprint v ktorom sme si stanovili naše ciele, premysleli možné riešenia a pripravili si vývojové prostredie.

### Šprint 2 - Paríž

Vytvorenie VR aplikácie v Unity s možnosťou pridania objektov. Príprava flask serveru a REST Api. Študovanie rôznych vedeckých prác na spracovanie 3D objektov pomocou GAN sietí.

## Šprint 3 - Berlín

Príprava architektúry celkového systému. Manipulácia s objektami vo VR (premiestňovanie, rotovanie, zmena veľkosti). Testovanie rôznych ďalších GAN sietí a kontrola ich výsledkov.

### 1.2 Motivačný dokument

**Kontaktný mail:** [timovyprojekt@centrum.sk](mailto:timovyprojekt@centrum.sk)

#### 1.2.1 Tím

Náš tím pozostáva z členov: Ján Antal, Marek Drahoš, Dominik Mazák, Matej Mikuš, Peter Pápay, Andrej Pisarčík a Adam Poperník.

Za veľké pozitívum považujeme vzájomné spoluprácu členov nášho tímu, kedy sme mali možnosť priamo počas štúdia pracovať na viacerých projektoch. Na druhej strane udržiavame spoločné kontakty aj mimo školy, čo vo veľkej miere upevňuje súdržnosť nášho tímu.

Väčšina členov nášho tímu prišla počas bakalárskeho štúdia do kontaktu s oblasťou strojového učenia, či už išlo o samotnú bakalársku prácu alebo predmet inteligentná analýza údajov. Konkrétne sme v projektoch využívali neuronové siete za pomoci frameworkov, ako napríklad - CNTK (Matej), Keras (Marek), TensorFlow (Peter). Vieme pracovať s relačnými (MySQL, PostgreSQL) a nerelačnými (MongoDB, Firebase) databázami. Okrem toho máme skúsenosti so správou sietí (Dominik), spracovaním obrazu pomocou knižnice OpenCV (Ján) a SimpleITK (Matej), paralelným či asynchrónnym programovaním, JEE klient-server architektúrou a programovaním elektronických zariadení s využitím hardvéru Arduino (Adam).

Medzi predmety, ktoré by nám mohli pomôcť k úspešnému vypracovaniu tímového projektu patria objavovanie znalostí, pokročilé databázové technológie, vyhľadávanie informácií, neuronové siete, počítačové videnie, distribúcia obsahu v internete, štatistické metódy vyhodnocovania experimentov.

V rámci mimoškolskej aktivity nám bolo umožnené nadobudnúť dôležité skúsenosti aj v praxi. Predovšetkým v oblasti vývoja webových aplikácií, kde sme mali možnosť vyvíjať ako frontend (Angular, React, HTML, CSS, Javascript, JQuery, Bootstrap), tak aj backend (Java Spring, PHP, Python Flask, ASP.NET Core, Node.js). Pracovali sme agilným vývojom a princípy SCRUM-u nám vôbec nie sú cudzie (Marek, Matej, Andrej, Peter, Adam). Matej a Peter sa podieľali na vývoji interaktívnych chatbotov integrovaných do aplikácii Slack a Skype. Ovládame prácu s verziovacím systémom Git, systémom pre deployment Docker (Andrej a Peter) a Linux serverom (Andrej). Navyše máme skúsenosť s project management systémami JIRA (Andrej, Adam), VSTS (Matej). Jeden člen nášho tímu riešil taktiež štátne zákazky, napríklad pre Ministerstvo dopravy, kde sa podieľal na vytváraní informačného systému pre cestnú dopravu (Adam).

Vo voľnom čase sa radi venujeme aktuálnym technológiami, ktoré nám umožňujú rozširovať si svoje obzory. Ako tím máme radi výzvy, s čím priamo súvisí naše nadšenie pre objavovanie a prehľbovanie znalostí.

#### 1.2.2 Motivácia

##### Monitoring antisociálneho správania [MonAnt]

Sociálne siete sú súčasťou dnešného každodenného života. Niet pochýb, že sú veľkým pomocníkom, no podobne ako iné médiá aj sociálne siete sú zneužívané v prospech tretej osoby. Príkladom je šírenie dezinformácií, hoaxov alebo "fake news". Dôvodom na ich publikáciu je zvädzanie čitateľa s falošnými informáciami a teda neférovovo ovplyvňovať verejnú mienku a snažiť sa pretlačiť svoju agendu. Takto sa

to stalo aj v roku 2016 pri prezidentských voľbách v US. Veľké spoločnosti ako Facebook a Google boli prvotne kritizované za nečinnosť, no obe začali vynakladať obrovské peniaze na boj proti fake news.

Myslíme si, že šírenie fake news je veľkým problémom demokratických krajín, vrátane Slovenska. Príkladom môžu byť aj často nepodložené výroky politikov, ktoré sa dostávajú do médií a rozdeľujú našu spoločnosť. Články nemožno len tak cenzurovať kvôli slobody tlače. Čo však môžeme urobiť je upozorniť na to, že nie všetko v nich môže byť pravda. Prvým krokom je detekcia takýchto článkov. Ale tým to nekončí, rovnako dôležitou súčasťou je aj ich prezentovanie verejnosti čo možno najjednoduchšou formou.

Túto tému sme si zvolili nielen z dôvodu, že si vážime a chceme zachovávať demokratické hodnoty na Slovensku, ale zároveň by sme chceli pomocou detekcie a upozorňovania na zavádzajúce a nepravdivé informácie aj pozdvihnúť úroveň mediálneho obsahu na internete. Veríme, že prácou na tomto projekte môžeme prispieť k zníženiu spoločenského napätia ktoré tu v súčasnosti panuje.

Z technického hľadiska máme v tíme niekoľko ľudí ktorí sa venujú front-endu a majú skvelé nápady ako realizovať front-endovú časť projektu. Zároveň je projekt zaujímavý aj z back-end časti, kde je potrebné crawlovať web a budovať dataset. Táto časť zaujala práve tých z nás, ktorí sa venujú umelej inteligencii a veľkým dátam.

## **Vyhľadávanie pomocou obrázkov [ImageSearch]**

Častokrát sa stane, že si vo voľnom čase niečo odfotíme, či už v prírode, v obchode, skrátka všade. V dnešnej dobe má už skoro každý človek fotoaparát alebo mobil, pomocou ktorého vie zaznamenať obraz a uložiť ho na svoje úložisko, prípadne do databázy. Obraz podobne ako aj text zobrazuje (poskytuje) človekovi informáciu. Niekedy sa však stane, že človek nemá dostatok znalostí o danom objekte - predstavme si hríb v lese - nevieme ako sa volá, no chceme zistiť či je jedlý. Keby sme teda vedeli jeho názov, zadáme túto informáciu do Google alebo iného vyhľadávacieho nástroja. Keďže ale nevieme slovné identifikovať daný objekt, môžeme využiť práve vyhľadávanie pomocou obrazu/obrázok.

Niekedy však nechceme predmet iba vyhľadať a zistiť čo to je. Vyhľadávanie podľa obrázka nám už v súčasnosti poskytuje napríklad Google. Určite sa už každý z nás aspoň raz ocitol v situácii, keď narazil na nejaký zaujímavý predmet a povedal si “toto by som chcel mať!”. A tu prichádza na rad otázka, kde ten predmet kúpiť. Vyhľadávanie v e-shopoch v drvivej väčšine zahŕňa vyhľadávanie na základe textu alebo kľúčových parametrov. Preto sme sa ako tím rozhodli, že by sme chceli toto vyhľadávanie vylepšiť a využiť vyhľadávanie pomocou obrázkov na vytvorenie systému schopného tento spôsob vyhľadávania aplikovať na tovar v e-shopoch. Vidíme v tom skvelú príležitosť, ako vytvoriť niečo zaujímavé a zároveň praktické do života. Človek by si mohol čokoľvek čo sa mu zapáči jednoducho odfotiť a behom pár sekúnd zistiť, v ktorom e-shope to ponúkajú.

V našom tíme sa už viacerí zaoberali prácou s obrazom, najmä počas bakalárskych prác (identifikácia tumoru z obrazu) ale aj počas predmetov, ktoré sme počas štúdia absolvovali, či plánujeme absolvovať (napr. počítačové videnie, princípy počítačovej grafiky a spracovania obrazu). Aj to bol jeden z dôvodov, prečo sme sa rozhodli pre danú tému. Práca s obrazom má určite v budúcnosti veľkú perspektívu a dá sa na ňu nahliadať z viacerých aspektov.

### **1.2.2 Zoradenie tém podľa priority**

Po prekonzultovaní sme zmenili priority tém. Našou najväčšou prioritou je projekt **MonAnt**, ktorý sme zvolili aj v systéme YonBan.

Odovzdané poradie v dotazníku:

1. Importer
2. MonAnt
3. ImageSearch
4. SmartMobility

5. TxtEnv
6. GridBox
7. In-memory-DB
8. Breyslet
9. Behamatrix
10. SK-CZ-TEXT
11. VReduction
12. FIIT-DU

**Finálne poradie priorit tém:**

1. MonAnt
2. ImageSearch
3. Importer
4. SmartMobility
5. TxtEnv
6. GridBox
7. In-memory-DB
8. Breyslet
9. Behamatrix
10. SK-CZ-TEXT
11. VReduction
12. FIIT-DU
13. Holographic Eyes
14. WFuntoro
15. 3DSpaceGen
16. VizReal
17. 3D-UML
18. BioMonitor
19. IoTesting
20. 3D futbal

### **1.2.3 Rozvrh tímu**

<https://docs.google.com/spreadsheets/d/1ZCgnhetO0VVVJn3e0Jv9qCxUQdWshURk8J8V6ZgAk8o/edit#gid=0>

## **1.3 Metodiky**

### **1.3.1 Metodika komunikácie**

All communication must be done in Slack.

No private messages of important information, which are information all of team could benefit.

Description of channels:

- *Deploy-webpage* channel just for notifications of webpage deployment
- *Events* channel just for notification about upcoming events created in google calendar
- *Garbage* Just chill and off topic talk.
- *Neural networks* Only communication about NN is discussed in this channel.
- *Stand-up* All virtual stand ups is written here. Good for history.
- *Virtualreality* Only communication about VR and unity is discussed in this channel.

## 1.3.2 Metodika manažmentu tímu

### Definition of Done

User stories are created at weekly tuesdays. The user stories then must be assigned a story points, by using planning poker. When the sprint starts, the product owner chooses which user stories will be inserted into sprint. Those user stories, which are inserted into sprint, must have assigned a team member, who is responsible for its completion. When the work on the user story starts, the responsible team member must set its state to Active. After all tasks in the user stories are completed, and the review is done, and the user story passes the acceptance criteria, the user story can be moved into the Resolved column. At the end of the sprint, the product owner may decide if the user story is finished. If he agrees, the user story is closed. If not, the user story must be splitted into two parts and taken into the next sprint.

### Task lifecycle

1. After a team member is assigned an user story, he is obliged to create tasks for the user story as soon as possible (< 2 days after assignment).
2. Those tasks then can be assigned to any team member, who will work on those tasks.
3. The team member who is responsible for task, must fill in the time (Estimated, completed, remaining) in TFS
4. When the team member starts the work on the task, he moves the task into Active column.
5. After he finishes his task, he moves it into Closed column.

### User Story

- Produced code for presumed functionalities
- Assumptions of User Story met
- Project builds without errors
- Feature ok-ed by UX designer
- QA performed & issues resolved (what are criteria for quality?)
- Feature is tested against acceptance criteria
- Feature ok-ed by Product Owner
- Documentation updated
- Peer Code Review performed

In future:

- Unit tests written and passing
- Refactoring completed

### Sprint

- DoD of each single User story, included in the Sprint are met
- “to do’s” are completed
- All unit tests passing
- Product backlog updated
- All bugs fixed
- Sprint marked as ready for the production deployment by the Product Owner

In future: The performance tests passed

## 1.3.3 Metodika verziovania projektu

### 1.3.3.1 Git rules

Every repository must have at least two branches created by default

- Master
  - This symbolises production version. Merges can be made only on version release.
- Develop



- All tasks that were accepted by reviewers in pull request must be merged into develop branch

Every task must have its own branch.

This branch will be identified by specific format.

- Format = [TASK-XXXX]
  - XXXX - unique value generated by TFS

### 1.3.3.2 Workflow

#### If I don't have installed Git Bash

1. Download the Git Bash program from <https://git-scm.com/downloads>
2. Install Git Bash

#### If I have installed Git Bash (Clone repository - first start)

1. Open Git Bash program (in Windows called MINGW64)
2. Go to required folder, where you want to have your local git with command `cd` (Change directory)

Examples:

#### Change to the required directory with absolute path:

```
C:\School> cd C:\School\3DSpaceGen
```

#### Change to the parent directory:

```
C:\School> cd ..
```

#### Change to the grand-parent directory:

```
C:\School\backup\January> cd ....
```

#### Change to the ROOT directory:

```
_C:\School\backup\January> cd _
```

#### Display the current directory in the specified drive:

```
C:> cd D:
```

3. Go to the [TFS Git page](#) and choose your repository (folder/file...) that you want to clone in left side on webpage
4. After choose the repository click to the Clone button
5. Copy the displayed url, type `git clone` yourcopiedurl (example: `git clone http://yourwebpage.com`) into the Git Bash and press Enter button
6. Now you have actual version of git repository in your PC

### 1.3.3.3 Creating branches

1. Change your current branch to the develop - type **git checkout develop** (more about [checkout](#) )
2. Now you need to REBASE your local repository with command **git rebase origin/develop** (more about [rebase](#) )
3. Create new branch in TFS according to rules specified above (Code -> Branches -> **New branch**)
4. In Git Bash type **git fetch --all --prune** (more about [fetch](#) )
5. Checkout to required (created) branch with **git checkout yourCreatedBranch**
6. Type **git rebase origin/develop**
7. Type **git push --force**
8. Now you can add new folder(s)/file(s) with **git add** command

Examples:

#### Stages all changes:

```
git add -A
```

#### Stages new files and modifications, without deletions

```
git add .
```

#### Stages modifications and deletions, without new files

```
git add -u
```

#### Adding the concrete file/folder:

*git add myfile.txt*

With **git status** command you can verify the current state of your git branch

9. When you want to commit your changes type **git commit -m "TASK\_ID define\_your\_changes"**
10. Type git status and check if you have committed all changes
11. Use **git push** to upload the changes into TFS

#### **1.3.3.4 Pull request**

All pull requests must be reviewed by at least one other team member. The reviewer might reject the pull request but must specify the reason. All pull request of task branches must go into develop branch.

In future tasks will be merged into user story branches. The user story branches will be merged into develop.

#### **1.3.3.5 Commits**

General rule, commit often.

Commits must follow this format: [verb noun ...]

For example:

- "add styling for buttons"
- "modify support for voice commands"
- "remove unnecessary code"
- "refactor python module"

### **1.4 Webové sídlo**

Okaz na zip sùbor, obsahující webovú stránku tohto tímu:

<https://uloz.to!/K4CsXWhGI8WK/webpage-zip>

## 2. Inžinierske dielo

---

### 2.1 Big picture

In this section is presented the project created by 3DSpaceGen team. The goals of the project are presented in this document. Document describes the whole architecture, but also goes into details of each module.

#### 2.1.1 Globálne ciele projektu

##### 2.1.1.1 Zimný semester

Output of our project in winter term is an application programmed in Unity for HTC Vive device. This application will support dynamic insertion of objects into scene and also manipulation of those objects. The objects will be generated with neural networks, specifically using generative adversarial networks. Inference will work in real time after accepting a request from client through provided REST Api.

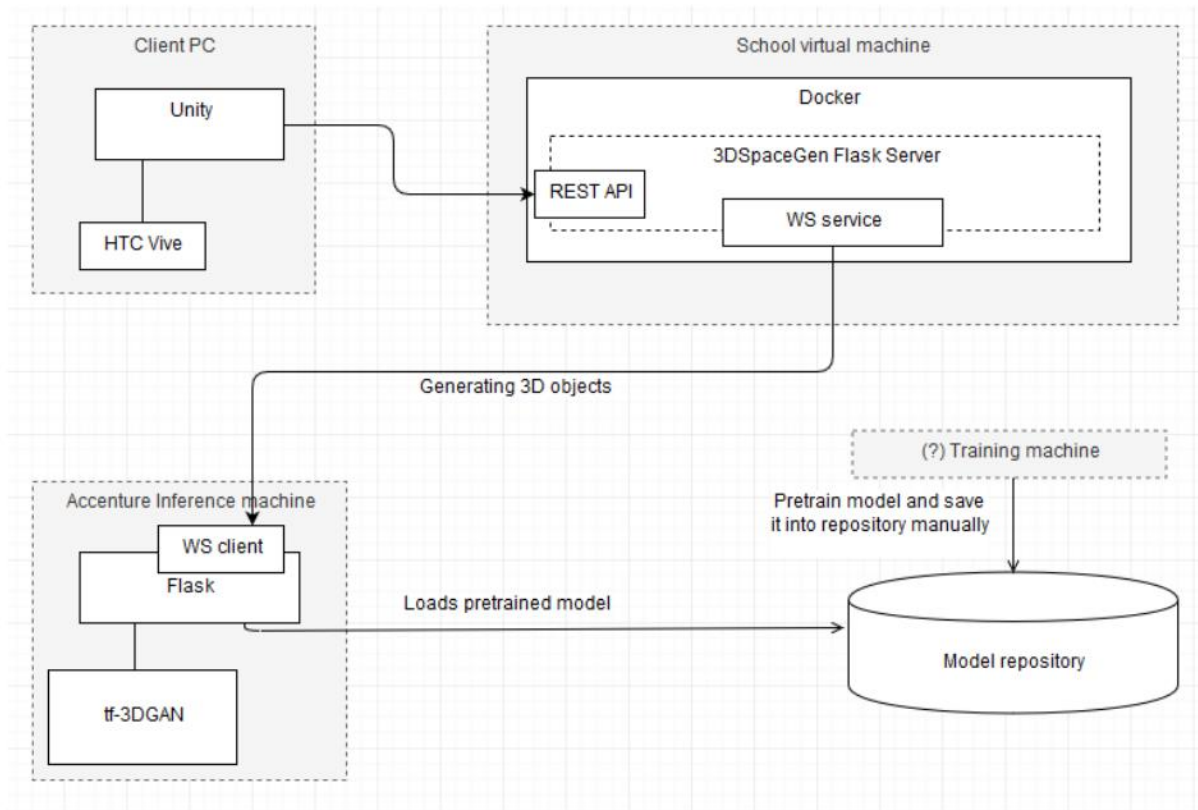
##### 2.1.1.2 Letný semester

The goal of summer term is to improve existing application. Main goal is to implement a feature, that an user can modify the structure of generated objects. Voice control of application will also be added, to increase the user experience.

#### 2.1.2 Celkový pohľad na systém

After the first three sprints we have identified and described the following architecture. The client is a basic personal computer with Unity installed and HTC Vive connected. The user interacts with the world using the HTC Vive controllers. This client sends requests for 3D models to REST Api, which is provided on our school virtual machine. This REST Api is provided by Flask, which runs in Docker container. Flask communicates with Inference machine (accenture PC) using the web socket, which can invoke the request for generating specific 3D model. Inference machine is used to generate 3D models from pre-trained model of GAN. This pre-trained model is loaded from shared git repository (or local storage). An agent will watch the repository for changes. When the change happens, agent updates the model on inference machine. New models will be inserted into repository by hand and they will be trained on external computers.

Communication between inference machine (IM) and school virtual machine (SVM) is described below. SVM represents a server role and IM represents client. The reason is that the IM is provided by Accenture company, which has strict security rules. After the connection is made, the roles "switches" and SVM becomes a client and sends requests to IM.



Obr. 1: System architecture

## 2.2 Moduly systému

### 2.2.1 Modul virtuálnej reality

#### Analýza

For the space visualisation, we decided that the virtual reality will be more suitable for our solution unlike the augmented reality, because it allows to change the whole space and the user is not limited by the real objects in the scene.

**Unity 3D** is a multi-platform gaming engine developed by Unity Technologies. Unity provides development of 2D and 3D games also in virtual reality using SteamVR plugin. In addition to the graphical creation environment, it also supports the creation of scripts primarily in C #.

**HTC Vive** is a virtual reality head-mounted display. It uses "room scale" tracking technology to allow the user to move in 3D virtual space and motion-tracked handheld controllers to interact with the environment.

**VRTK** is a collection of useful scripts and concepts for building VR solutions rapidly and easily in Unity3D.

#### Návrh

We suggest creating a simple menu to allow the user to select the category of the desired object which will be spawned in the scene.

When the object is spawned, the user will be able to manipulate the object (change the scale, position and rotation) via the HTC Vive controllers and delete it.

## **Implementácia**

The interaction with the virtual reality is implemented via the Virtual Reality Toolkit. We use implemented VRTK solutions such as:

- Interactions like touching, grabbing and using objects
- Interacting with Unity3d UI elements through pointers or touch.
- Head-set menu with 2D buttons.

The menu with the object category will appear when the user presses the button on the controller. The category of the object can be selected by the VR laser pointer and confirmed by the trigger button. When the menu category is selected the importer will take care to spawn the object in the scene. The user can grab the object with the controllers to change its scale, position, and rotation.

### **2.2.2 Importér**

#### **Analýza**

This module is responsible for importing new objects into Unity game engine created at runtime by the end user. Being able to create and import new 3D models at runtime is crucial for our project. We want to allow the user to create new objects while he's inside VR environment. We could just bundle the final product with 3D GAN neural network, pretrained models and all other modules, but running all these services like VR application, object generation and object processing all at once at one machine would put the client machine under huge load and consume a lot of hardware resources. Therefore we've decided that the best solution will be to implement object creating and importing as a service running on a dedicated server (or servers) and the client machine just calls this service when needed.

#### **Návrh**

The client sends a request using our implemented REST API requesting an object from a specific category. After receiving a successful reply, the reply contains the 3D object data, which are the processed by object imported and transformed to a GameObject. After that the imported GameObject can be added to the scene, where the user can interact with the objects.

All of this functionality is being done outside of Unity game engine - that is as a service provided by our server. With this design we aim to reduce the load on the end user equipment and achieve a faster processing speed as opposed to doing everything on the client machine.

## **Implementácia**

The client sends a request with the specification of the object requested to the 3DSpaceGen Flask Server. The Flask Server forwards the request to the Accenture Inference machine, which generates a 3D object using a pre-trained model. After the object generation phase, post-processing follows. The generated object is converted from the output .MAT file to wavefront .OBJ file, and a second pass of post-processing minimizes the face and vertex count. The object is then sent back to the 3DSpaceGen Flask Server, which forwards the object to the client machine, running the VR setup. Client accepts the object file and adds it into the scene.

### **2.2.3 Rest API Servis**

#### **Návrh**

We designed a Flask server as REST service that is responsible for responding to a user request, where a user specifies kind of object to generate. The request is a message, which contains various information, such as requested object type (e.g. chair, table..) or id. Other parameters will be specified later in the project. After the message is parsed, the tf-3DGan module is called.

Since our server is not able to generate an object (missing graphics card), we propose to use a websocket server in order to communicate with the external machine, that has an appropriate driver to perform this task. When the request from the client comes, the server determines which type of object has to be generated and sends the request via web socket to inference server. The server is then waiting for the response from the inference machine.. If the inference server returned the generated object as the response, the server sends this object to the end user, otherwise sends an HTTP error message.

### **2.2.4 Moduly inferencie**

Main module on the inference machine will be a simple websocket client. On the startup, this WS client will try to establish a connection with our specified school server (IP address and port will be given in config file). After the connection is established, the 'client' will listen to requests made from 'server'.

The requests are specified in the section above. On the request arrival, tf-3DGAN module is called.

The tf-3DGan module is a neural network implemented in tensorflow. On function call, correct pre-trained neural network model is loaded into memory from local storage (this will be optimised later in the project). Then the inference (feedforward phase in neural network of random noise) is made. Results (generated 3D object) are returned in a function.

Afterwards the 3D object pre-process module is called. This module will contain various graphical algorithms to transform the output of neural network (which is a tensor of rank 3) into a standard '.obj' file (list of vertices, faces, normals...), which can be easily imported into unity. Also, other methods for reducing the size of object might be used.

After the generating and preprocessing events, the generated model is returned as a response to origin of a request (school server). Then the module listens for another request in endless loop.

The pre-trained models are loaded into local storage manually. Later in the project we plan to implement a module, which will train, deploy, and test the models independently and automatically.

All modules will be implemented in python, using various frameworks (tensorflow, sacred, ... ) to improve efficiency.

## Príloha A – export úloh

### Export úloh prvého šprintu

| ID                   | Title  | Assigned To       | State  | Description  |
|----------------------|--|-------------------|--------|--|
| <a href="#">8918</a> | Continuous deployment of the web page  | Dominik Mazak     | Closed | As the product owner, I would like to have the webpage automatically deployed into Apache root after commit to master branch. TODO: * Download build agent from TFS * Deploy build agent to server * Set-up build definition in TFS                  |
| <a href="#">8920</a> | Build result notification - web page   | Andrej Pisarcik   | Closed | We want to receive automated notifications when build has finished in a separate Slack channel. TODO: * Create appropriate Slack channel. * Get channel authorization key from Slack. * Set-up notification rule in TFS with the authorization key.  |
| <a href="#">8921</a> | Task status change   | Andrej Pisarcik   | Closed | We want to receive automated notifications when task status changed in a separate Slack channel. TODO: * Create appropriate Slack channel. * Get channel authorization key from Slack. * Set-up notification rule in TFS with the authorization key. |
| <a href="#">8924</a> | As a user, I want to see empty 3D space in HTC Vive because I want to fill it later  | Jan Antal         | Closed | TODO: * Set-up empty Unity project for HTC Vive. * Enclose the scene into a big black box. * Pair with HTC Vive. * Run.  |
| <a href="#">8925</a> | As a user, I want to see a chair in the middle of the scene because I want to have implementation ready for 3D model visualization | Bc. Adam Popernik | Closed | TODO: * Find free online 3D model repository with appropriate model file format. * Download 3D model of a chair. * Import the 3D model into Unity project. * Place the chair into the middle of the scene, scale the chair appropriately. * Run.     |
| <a href="#">8927</a> | As a developer, I want to have a manual for neural networks because I want to use them in the future development                   | Matej Mikus       | Closed | TODO: * Set-up Wiki in TFS. * Create a Wiki page with neural networks overview. * FFNN * RNN * CNN * GAN * Create a simple neural network demonstration.   |
| <a href="#">8928</a> | As a developer, I want to replicate the implementation from 3D GAN research paper because I want to try it out                     | Peter Papay       | Closed | TODO: * Read the paper. * Clone the repository. * Set-up the environment. * Run the experiment. * Verify results.  |

### Export úloh druhého šprintu

| ID                   | Title   | Assigned To       | State  | Description  |
|----------------------|---|-------------------|--------|--|
| <a href="#">9214</a> | As a user, I want to see progress of 3DSpaceGen team on website, because I am interested. | Bc. Adam Popernik | Closed | * názov témy * riešitelia a základné informácie o nich (napr. z ponuky) * odkazy na doteraz vypracovanú dokumentáciu vrátane záznamov zo stretnutí (záznamy zo stretnutí budú priamo čitateľné na webe, t.j. najlepšie vo formáte pdf, prípadne HTML)<br>Remaining: * plán projektu (na semester) * aktuálny stav plnenia plánu (t.j., úlohy, ktoré vyplynuli zo stretnutí, ich plnenie a vzťah k plánu) roly jednotlivých členov tímu (aj dočasné) * všetko zaujímavé v súvislosti s projektom a postupom prác na projekte, napr. odkazy na iné zdroje súvisiace s témou projektu |
| <a href="#">9215</a> | As a user, I want to have service to trigger  | Dominik Mazak     | Closed | Setup FLASK service on OUR server. It will contain pretrained model. Model generates an object. (off to obj, command line tool   |

|                      |   |                   |        |  |
|----------------------|---|-------------------|--------|--|
|                      | generate 3D model, so I can place it in the scene.  |                   |        | exists) Object needs to be converted to correct format. Service returns the object in correct format. Risks: Converter problems  |
| <a href="#">9216</a> | As a user, I want regenerate 3D model in a middle of scene, because I want to see a new object.                               | Jan Antal         | Closed | When button is pressed on controller, request is made to service and returns an object. The object is spawned in the scene. * learn how to interact with the VR controllers in unity. * create the menu using the game objects (2D plane with text or image) to specify the object which we want to generate. * for each menu item assign the specific text (name of the object). * implement the VR pointer to see on which menu item am I pointing with the controller. * add the action to the controller button - find the position on which the controller is pointing and print the object name in the console.  |
| <a href="#">9220</a> | As a developers, we want to get a TATRA bank grant, so we can run models on perfect machine.                                  | Bc. Adam Popernik | Closed | Write a request (2-3 pages) based on Tatra banka requirements. Try to build compatible hardware and get the prices. Deadline in 2nd of November. <a href="http://www.nadaciatatrabanky.sk/index.php/grantove-programy/e-talent/">http://www.nadaciatatrabanky.sk/index.php/grantove-programy/e-talent/</a> Pridat <a href="https://www.fiit.stuba.sk/sk/fakulta-informatiky-a-informacnych-technologii/prihovor-dekana-fiit-stu/zriadovacia-listina.html?page_id=416">https://www.fiit.stuba.sk/sk/fakulta-informatiky-a-informacnych-technologii/prihovor-dekana-fiit-stu/zriadovacia-listina.html?page_id=416</a>  |
| <a href="#">9221</a> | As a team, we want to participate in TP cup, to get honor and glory in FIIT STUBA.  | Peter Papay       | Closed | <a href="http://www2.fiit.stuba.sk/tp-cup/">http://www2.fiit.stuba.sk/tp-cup/</a> * Tím sa prihlási do súťaže vypracovaním a odovzdaním prihlášky v rozsahu max. 2 strán A4 (text fontom Times New Roman, 11pt). * Prihláška predstavuje prvotný návrh riešenia projektu * Obsahuje * kontakt na tím (minimálne mail), * stručné predstavenie tímu, * základnú motiváciu, * opis kontextu a náplne projektu, * stanovenie cieľov projektu, požiadaviek na výsledný produkt, ako aj spôsob a postup akým budú dosiahnuté. * Cieľom je presvedčiť, že: * ste identifikovali zaujímavý problém, * zamýšľané riešenie rieši tento problém, * ste tí správni pre riešenie projektu, * viete aké technológie a postupy využijete na jeho realizáciu. |
| <a href="#">9384</a> | As a developer, I want to replicate the implementation in python3.6 from 3D GAN research paper, because I want to try it out. | Peter Papay       | Closed | TODO: * Set-up the environment. * Reimplement to python 3.6 * Write wiki * Run the experiment. * Verify results. * Commit * Push * Pull request * Ask friend for review * Merge  |
| <a href="#">9386</a> | As a customer, I want to see analysis of state of the art GAN, because I want to be sure that we select best solution.        | Matej Mikus       | Closed | Read about recent GAN papers from github repository in wiki. Write about it to Wiki. Try to find interesting survey papers. What dataset was used, what models, parameters and short summary for every paper.  |
| <a href="#">9387</a> | As a developers, we should contact google/aws/azure to get better machines, because we want to use them.                      | Andrej Pisarcik   | Closed | * Write the general application (can be derived from TP Cup) * We should learn the requirements for applications for every vendor * Write the application * send the mail from <a href="mailto:@stuba.sk">@stuba.sk</a> domain   |



## Export úloh tretieho šprintu

| ID                   | Title  | Assigned To       | State  | Description   |
|----------------------|--|-------------------|--------|---|
| <a href="#">9600</a> | As a user, I want to move an object in my scene, because I want to customize the scene.  | Bc. Adam Popernik | Closed | The created object will be spawned in middle of scene The user then can grab him a move the object, and put the object down   |
| <a href="#">9601</a> | As a user, I want to rotate an object in my scene, because I want to customize the scene.                                      | Jan Antal         | Closed | User can selected object in scene User then can rotate the object User then can place it down   |
| <a href="#">9602</a> | As a user, I want to scale an object in my scene, because I want to customize the scene.                                       | Jan Antal         | Closed | The created object will be spawned in middle of scene The user then can grab him a scale the object, and put the object down  |
| <a href="#">9765</a> | As a user, I want regenerate 3D model in a middle of scene, because I want to see a new object.                                | Jan Antal         | Closed | When button is pressed on controller, request is made to service and returns an object. The object is spawned in the scene. * learn how to interact with the VR controllers in unity. * create the menu using the game objects (2D plane with text or image) to specify the object which we want to generate. * for each menu item assign the specific text (name of the object). * implement the VR pointer to see on which menu item am I pointing with the controller. * add the action to the controller button - find the position on which the controller is pointing and print the object name in the console.   |
| <a href="#">9766</a> | As a developer, I want to have working implementation of tf-3dgan on windows machine, because I probably have no other choice. | Peter Papay       | Closed | Make tf-3dgan run on windows machine. <a href="https://docs.microsoft.com/en-us/windows-hardware/drivers/display/timeout-detection-and-recovery">https://docs.microsoft.com/en-us/windows-hardware/drivers/display/timeout-detection-and-recovery</a> <a href="https://stackoverflow.com/questions/17186638/modify-ing-registry-to-increase-gpu-timeout-windows-7">https://stackoverflow.com/questions/17186638/modify-ing-registry-to-increase-gpu-timeout-windows-7</a> <a href="https://stackoverflow.com/questions/16317505/cublas-failed-to-synchronize-stop-event">https://stackoverflow.com/questions/16317505/cublas-failed-to-synchronize-stop-event</a> |
| <a href="#">9767</a> | As a user, I want to see implementation of other 3dgans, because I want to see the results compared.                           | Matej Mikus       | Closed | One more implementation is used.  |
| <a href="#">9768</a> | As a team, check out external box, because we might buy one.   | Dominik Mazak     | Closed | * Extern GPU through USB-C / USB3/ Thunderbolt * So we can try and run it on our notebooks / labtops * Find more alternatives of boxes and GPU's  |
| <a href="#">9769</a> | As a team, we want to have methodics (git, tfs, communication) and guidelines, because we want to be more organised.           | Peter Papay       | Closed | * Write down methods for GIT /etc (more in Sprint 02 Retrospective)   |
| <a href="#">9770</a> | As a team, we must have sprint 1-3 documented, because we are obliged to for subject.  | Marek Drahos      | Closed | Inzinierske dielo <a href="https://docs.google.com/document/d/1eLUq6wszjWFBjZw_5UAGH2njD4jZa8jNh3fhg8XUvvl/edit">https://docs.google.com/document/d/1eLUq6wszjWFBjZw_5UAGH2njD4jZa8jNh3fhg8XUvvl/edit</a> Riadenie projektu <a href="https://docs.google.com/document/d/1khojBT5EtdC7H6UezrEjvTCNEhnAEmZuigYh-l7a8XY/edit">https://docs.google.com/document/d/1khojBT5EtdC7H6UezrEjvTCNEhnAEmZuigYh-l7a8XY/edit</a> for more information see: <a href="http://www2.fiit.stuba.sk/~bielik/courses/tp-slov/materialy/dokumentacia2017-18.pdf">http://www2.fiit.stuba.sk/~bielik/courses/tp-slov/materialy/dokumentacia2017-18.pdf</a> deviaty týždeň                |

|  |  |  |  |
|--|--|--|--|
|  |  |  | <p>semestra (podľa rozvrhu stretnutí, pracovný deň nasledujúci po dni v týždni, kedy má tím plánované stretnutia), najneskôr však 23.11.2018, 12:00 (tí čo máte stretnutia v piatok, naplánujte si to tak, aby ste to zvládli) odovzdanie dokumentácie po prvých troch šprintoch (vrátane celkového obrázku projektu) plus dokumentácia k riadeniu - elektronicky * Úvod - o čom je tento dokument, ciele, ohraničenia. * Globálne ciele projektu na zimný semester * Celkový pohľad na systém (architektúra, dátový model, diagramy tried, moduly plus referencie/zoznam priložených e-dokumentov) Retrospektiva a aj vyjadrenie ako sprint prebiehal (čo sa spravilo, čo sa nedstihlo, pridať nejaký burndown chart, spraviť vyjadrenie či sme spravili toho viac alebo menej, info k organizácii tímového projektu atd) <a href="http://www2.fiit.stuba.sk/~bielik/courses/tp-slov/tp-main.html#odovzd-zs">http://www2.fiit.stuba.sk/~bielik/courses/tp-slov/tp-main.html#odovzd-zs</a></p> |
|--|--|--|--|