**CityScape Logic**

Dokumentace

**Obsah**

[1. Analýza požadavků 3](#_Toc170463056)

[1.1. Katalog obecných požadavků 3](#_Toc170463057)

[1.1.1. Funkční stůl 3](#_Toc170463058)

[1.1.2. Funkční „Back end“ 3](#_Toc170463059)

[1.1.3. Dostatek TUIO kartiček 3](#_Toc170463060)

[1.1.4. Cílová platforma 3](#_Toc170463061)

[1.2. Diagram případů užití 4](#_Toc170463062)

[1.3. Krátké popisy případů užití 4](#_Toc170463063)

[1.3.1. Upravovat objekty 4](#_Toc170463064)

[1.3.2. Upravovat podmínky pro objekt 4](#_Toc170463065)

[1.3.3. Projektovat 4](#_Toc170463066)

[2. Návod k použití 5](#_Toc170463067)

[2.1. Dependence – knihovny 5](#_Toc170463068)

[2.2. Tracker 5](#_Toc170463069)

[2.3. Server 5](#_Toc170463070)

[2.4. Stůl 5](#_Toc170463071)

# Analýza požadavků

## Katalog obecných požadavků

### Funkční stůl

U stolu bude funkční kamera snímající obraz, který půjde přes prodloužený USB kabel do počítače. Zároveň bude funkční projektor, který pomocí HDMI kabelu bude získávat obraz z počítače a správně ho interpretovat na stůl.

### Funkční „Back end“

Tracker bude správně snímat na stole ležící kartičky s ArUco kódy a přeposílat je na Python TUIO server, který data bude zpracovávat. Pythonovský server pak následně bude fungovat jako „černá krabička“ s vyhodnocovací logikou, jejíž výstup pak půjde do projektoru.

### Dostatek TUIO kartiček

Tracker nezvládá snímat dva objekty se stejným kódem zároveň. Více objektů bude snímáno za pomocí více různých ArUco kódů určených pro jeden objekt. Tímto způsobem bude přesně omezený počet jednotlivých kartiček pro každý objekt (v implementaci rovnoměrně).

### Cílová platforma

Jedna z mála smysluplných platforem, kde tento projekt bude fungovat, je Tactile Matrix Box. Implementace bude potřebovat, aby kódy na kartičkách byly dobře snímatelné a aby šlo dobře vidět vyhodnocení kartiček.

## Diagram případů užití

A diagram of a diagram

Description automatically generated

## Krátké popisy případů užití

### Upravovat objekty

Případ užití umožňuje uživateli přidat, odebírat nebo upravovat objekty, které pak budou snímány a vyhodnocovány. Jednoduše si tak uživatel může upravit CityScape Logic vstup, aby odpovídal jeho požadavkům na téma simulace a projektování.

### Upravovat podmínky pro objekt

Tento případ užití poskytne uživateli možnost přidávat, odebírat nebo upravovat podmínky u už existujících objektů. Uživatel si sám může zvolit kritéria, podle kterých budou objekty vyhodnocovány – podle požadavků na sousední objekty a minimální vzdálenosti pro tyto objekty.

### Projektovat

Uživatel může pomocí manipulace s kartičkami simulovat projektování města. Každá kartička představuje nějakou stavbu nebo objekt. K tomu, aby byl vyhodnocen jako správně postaven, tak musí být funkční. To znamená, že ve svém požadovaném okolí potřebuje mít všechny potřebné jiné objekty (požární stanice potřebuje rybník, bytovky potřebují park apod.). Pokud má ve svém okolí požadované objekty, ale tyto objekty nejsou funkční, pak se nebudou počítat jako že se na ně naše stavba může spolehnout a tudíž tato nefunkční stavba nebude splňovat naše podmínky.

# Návod k použití

Ke spuštění naší implementace je doporučeno být na Linuxovém prostředí. Zároveň pro tracker bude potřeba, aby Váš počítač měl přístup ke kameře pro snímání.

## Dependence – knihovny

Potřebné příkazy pro stažení Python knihoven:

* ***pip3 install python-tuio***
* ***pip3 install python-osc***
* ***sudo apt install python3-tk***

## Tracker

* Stáhněte si zkomprimovaný **thesis-tracker** ze složky **Trackers** z našeho repozitáře. Je v něm už zbuilděný tracker, který byl použit pro naší implementaci
* Rozbalte ho
* V repozitáři, kde je rozbalen, ho spusťte pomocí:
  + ***./tracker***
* Tracker má přepínač zajišťující výběr jiného vstupu obrazu pro snímání:
  + ***./tracker -v 2***

## Server

* Stáhněte si složku **server** z našeho repozitáře
* V této složce jsou:
  + **PyServer\_GridVisualization.py** – naše „černá skříňka“ a zároveň interpretace výstupu
  + helper\_files složka – config.json se zadanými objekty a ConfigParser, který json zpracuje do classes
  + images složka – obrázky použité při streamování
* Program spusťte jednoduše pomocí
  + ***./PyServer\_GridVisualization.py***
* Program lze vypnout zmáčknutím klávesy *q*

## Stůl

* Přes HDMI daný k projektoru se k němu připojte. Projektor potřebuje být ještě připojen do zásuvky pomocí kabelu k němu danému (koncová část je kulatá a port na ní je blízko portu na HDMI)
* Zapněte projektor (zepředu je tlačítko)
* Dole pod skleněnou deskou je kamerka, která má USB
* Připojte si ji do počítače
* Zapněte tracker, případně s přepínačem, pokud má váš počítač k dispozici více kamer
* Zapněte PyServer, klikněte a podržte lištu Canvas okna a přetáhněte ho přes pravou stranu obrazovky, kde pak dvakrát klikněte (chcete maximalizovat to Canvas okno přes dvojitý kliknutí na lištu)

Pomocí kamery byste teď měli být schopni snímat kartičky s ArUco kódy a zároveň na tkinter plátně na stole vidět jejich pozici a jak byly vyhodnoceny.