

# Katalóg požiadaviek

## Vizualizácia senzorov mobilného robota

# Obsah

## 1. Úvod

- 1.1. Účel katalógu požiadaviek
- 1.2. Rozsah využitia systému
- 1.3. Slovník pojmov
- 1.4. Odkazy a referencie
- 1.5. Prehľad nasledujúcich kapitol

## 2. Všeobecný popis

- 2.1. Perspektíva systému
- 2.2. Funkcie systému
- 2.3. Charakteristika používateľov
- 2.4. Všeobecné obmedzenia
- 2.5. Predpoklady a závislosti

## 3. Špecifické požiadavky

## 1. Úvod

### 1.1. Účel katalógu požiadaviek

Tento dokument vznikol v rámci predmetu "Tvorba informačných systémov" v akademickom roku 2024/2025 na Fakulte matematiky, fyziky a informatiky Univerzity Komenského v Bratislave. Dokument predstavuje katalóg požiadaviek pre softvérové rozhranie (GUI) na monitorovanie a riadenie robota. Účelom tohto katalógu je jasne definovať požiadavky na nový GUI manažér, ktorý bude spracovávať a zobrazovať informácie z rôznych senzorov a modulov robota v jednotnom používateľskom rozhraní. Tento dokument poskytuje základ pre návrh, implementáciu a testovanie systému, aby sa zabezpečilo, že výsledné riešenie splní očakávania používateľov a zároveň je záväznou dohodou medzi zadávateľmi a vývojovým tímom o rozsahu projektu.

### 1.2. Rozsah využitia systému

Systém bude využívaný na monitorovanie a riadenie robota, ktorý obsahuje rôzne senzory (lidar, kamera s neurónovou sieťou, kompas, GPS, ultrazvukové senzory). Systém bude zhromažďovať informácie z jednotlivých modulov robota a zobrazovať ich v prehľadnom grafickom rozhraní.

### 1.3. Slovník pojmov

- 1.3.1. **Lidar:** Technológia merania vzdialeností pomocou laserových lúčov.
- 1.3.2. **GNSS:** Globálne navigačné satelitné systémy.
- 1.3.3. **Ultrazvukové senzory:** Senzory na meranie vzdialeností prostredníctvom zvukových vĺn.
- 1.3.4. **Modul:** Samostatný program alebo súčasť systému, ktorý vykonáva špecifickú funkciu.
- 1.3.5. **GUI:** Grafické používateľské rozhranie.

#### **1.4. Odkazy a referencie**

- 1.4.1. Git repozitár projektu: [Vizualizácia senzorov mobilného robot](#)
- 1.4.2. Stránka predmetu: [Tvorba informačných systémov](#)
- 1.4.3. Git repozitár zdrojového kódu pôvodného robota: [Smelý Zajko](#)
- 1.4.4. Git repozitár zdrojového kódu aktuálneho robota: [Smelý Zajko](#)
- 1.4.5. Git repozitár zdrojového kódu DEROS: [DEROS](#)
- 1.4.6. Záznam behu robota Smelý Zajko: [Link](#)
- 1.4.7. Kamerový záznam robota Smelý Zajko: [Link](#)

#### **1.5. Prehľad nasledujúcich kapitol**

Následujúce kapitoly detailne popisujú systém z hľadiska jeho funkcií, používateľských skupín, závislostí a obmedzení. Ďalej je uvedený podrobný zoznam funkcionalít, ktoré softvér musí poskytovať, aby splnil očakávania užívateľov.

## **2. Všeobecný popis**

### **2.1. Perspektíva systému**

Tento softvér bude slúžiť na zobrazenie informácií z viacerých modulov robota v jedinom prehľadnom okne. Riadiaci systém robota je modulárny a každý modul spracováva údaje z konkrétneho senzora alebo systému, alebo implementuje nejakú časť funkcionality celého riadiaceho algoritmu. Niektoré moduly zobrazujú informácie v textovej alebo grafickej podobe na obrazovke ale väčšinou v rôznych oknách, kvôli čomu je zobrazenie neprehľadné. Softvér umožní, aby tieto údaje boli zobrazené v rámci jednej obrazovky s možnosťou prispôsobovania veľkosti každého obrazového výstupu a prepínania medzi nimi.

### **2.2. Funkcie systému**

Systém bude komunikovať s nezávislými modulmi zodpovednými za spracovanie a vytváranie vizuálnych a textových dát. Bude komunikovať s riadiacim systémom prostredníctvom svojich dedikovaných modulov alebo pomocou komunikačného frameworku DEROS, pričom obidva spôsoby ponúkajú rovnakú funkcionality. Softvér bude možné kompilovať bez závislosti na frameworku DEROS. Systém načíta zoznam modulov z konfiguračného súboru a po svojej inicializácii zavolá inicializačné funkcie modulov, ktoré oznámia, aké grafické a vstupné prvky môžu spravovať.

GUI bude maximalizované na celú obrazovku, pričom proporcie prvkov sa automaticky prispôbia rôznym rozlíšeniam. Rozmery grafických prvkov budú nastavené absolútne a prepočítané na relatívne hodnoty, ktoré budú odosielané modulom na vykreslenie obsahu. Pri zmene zobrazenia informuje GUI manažér moduly o viditeľných prvkoch a posiela im údaje o rozmeroch grafických objektov, do ktorých majú vykresľovať. Používatelia budú môcť interagovať s ovládacími prvkami ako tlačidlá, dropboxy, checkboxy, posuvníky a textové vstupy.

Moduly budú môcť kedykoľvek požiadať GUI o zobrazenie notifikácie a zobrazovať viacero položiek v grafickom rozhraní. Softvér podporí textové oblasti a štítky, ktorých písmo a farbu bude možné konfigurovať. Textové oblasti budú mať dynamické scrollbar a možnosť autoscrollu.

Softvér poskytne používateľské API s funkciami na výber súboru alebo priečinka. Bude možné spustiť GUI program viackrát na jednom počítači, alebo naraz na rôznych počítačoch. Moduly budú môcť registrovať funkcie na klávesové skratky.

Konfiguračný súbor bude obsahovať šablóny ktoré majú byť v danom behu k dispozícii, zoznam modulov a zoznam prvkov na zaznamenávanie a prehrávanie. Softvér podporí viaceré konfiguračné súbory, ktoré budú vybrané pri spustení programu ako command-line parameter. Systém bude pracovať v troch módoch: konfiguračnom, prevádzkovom a prehrávacom.

V konfiguračnom móde bude možné upravovať rozloženie a veľkosti prvkov, ukladať šablóny a vybrať prvky na zaznamenávanie. Prevádzkový mód bude zobrazovať vybrané šablóny, umožní prepínanie šablón a zaznamenávanie označených prvkov. Prehrávací mód umožní prehrávať záznamy minulých behov.

### **2.3. Charakteristika používateľov**

Pri návrhu systému nebude rozlišovať typ používateľa, čo znamená, že všetci používatelia budú mať prístup k rovnakým funkciám a možnostiam

### **2.4. Všeobecné obmedzenia**

Systém bude kompatibilný s operačným systémom Linux a musí byť schopný komunikovať s frameworkom DEROS.

Obmedzenia môžu byť spôsobené hardvérovými kapacitami robota a priepustnosťou dátových spojení medzi modulmi a GUI.

## **2.5. Predpoklady a závislosti**

Predpokladom je, že softvér bude integrovaný s už existujúcim riadiacim systémom robota a že každý modul bude schopný poskytovať dáta v reálnom čase. Závislosťou je správne fungovanie jednotlivých modulov robota, ktoré poskytujú informácie GUI systému.

### **3. Špecifické požiadavky**

Voliteľné požiadavky sú označené \*.

- 3.1.** GUI manažér bude komunikovať s nezávislými modulmi, ktoré sú zodpovedné za spracovanie a vytvorenie vizuálnych a textových dát.
- 3.2.** GUI manažér bude s riadiacim systémom komunikovať buď pomocou dedikovaných prilinkovaných modulov využívajúcich jeho GUI manažer API v rámci toho istého GUI procesu, alebo priamo pomocou komunikačného frameworku DEROS, kde obe rozhrania poskytnú ekvivalentnú funkcionálnu funkcionálnu.
- 3.3.** GUI manažér bude možné prepnutím jedného prepínača skompilovať aj bez jeho modulu zabezpečujúceho komunikáciu s riadiacim systémom pomocou frameworku DEROS - to znamená závislosť tohto softvéru na frameworku DEROS nebude pevná.
- 3.4.** GUI manažér načíta zoznam modulov z konfiguračného súboru
- 3.5.** GUI manažér po vlastnej inicializácii postupne zavolá inicializačnú funkciu každého modulu, ktoré oznámia aké grafické prvky dokážu vykresliť a cez ktoré vstupné prvky vedú prijímať informácie.
- 3.6.** GUI manažér je vždy maximalizovaný na celú obrazovku (alebo prípadne celkom minimalizovaný) a pozície a veľkosti jednotlivých zobrazovaných prvkov sa na inom rozlíšení prepočítajú, aby relatívne sedeli, pričom je na zodpovednosti používateľa, aby obsah textových prvkov nepretiekal cez vymedzenú plochu na všetkých rozlíšeniach, na ktorých používateľ systém púšťa.
- 3.7.** Proporcionalita používateľom zvolených prvkov by sa mala dodržiavať.
- 3.8.** \* Spôsob prispôsobenia veľkosti jednotlivých prvkov pri inej proporcionalite celého displeja (napr. 4:3 vs 16:9) sa dá nastaviť konfiguračným parametrom: 1) čierne okraje, 2) čierne okraje v jednotlivých prvkoch, 3) prekrytie časti pozadia, 4) zmena proporcionality prvku.



- 3.9.** Rozmery grafických prvkov budú nastavené absolútne. Vtedy sa tieto hodnoty prepočítajú na relatívne a prepočítané hodnoty sa odosielať komponentom, keď sa očakáva, že do nich nakreslia svoj obsah.
- 3.10.** GUI manažér pri inicializácii, zmene alebo prepnutí zobrazenia (šablóny) informuje všetky moduly, ktoré prvky sú viditeľné v novom zobrazení a pri najbližšej príležitosti, keď modulu poslela grafický objekt, do ktorého má nové dáta nakresliť (buď explicitne alebo implicitne) o rozmeroch grafického objektu, do ktorého sa má vykresliť.
- 3.11.** Používatelia budú môcť interagovať prostredníctvom rôznych ovládacích prvkov, ako sú tlačidlá, dropboxy, checkboxy, posuvníky (vertikálne aj horizontálne) a textové jednoriadkové vstupné boxy. Takisto budú môcť interagovať pomocou ťahania a klikania myši.
- 3.12.** Moduly môžu kedykoľvek požiadať GUI pomocou jeho API o zobrazenie notifikácie.
- 3.13.** Softvér bude generovať výstupy vo forme grafických a textových prvkov, ktoré zabezpečia prehľadnú a efektívnu prezentáciu informácií. Medzi podporované textové prvky budú patriť:
  - 3.13.1.** Textová oblasť (text area)
    - 3.13.1.1.** Jednoriadkový a viacriadkový textový štítok, ktoré sa môžu zobrazovať priamo v pozadí hlavného okna. Pre oba typy štítkov bude možné nastaviť veľkosť písma a farbu textu.
    - 3.13.1.2.** bude mať dynamické scrollbar, ktoré sa zobrazia, ak sa obsah nezmestí do okna textovej oblasti.
    - 3.13.1.3.** Používatelia budú môcť aktivovať autoscroll kliknutím, čo zabezpečí, že pri každom pridaní nového riadku sa okno textovej oblasti automaticky zameria na začiatok alebo koniec, aby bol nový riadok viditeľný. Ak autoscroll nie je aktívny, okno textovej oblasti zostane na pozícii, na ktorej ho používateľ vyscrolloval.

3.13.1.4. \* API umožní pridať riadok, vložiť riadok na konkrétnu pozíciu J, odstrániť J-ty riadok, alebo zadať kompletne nový obsah.

- 3.14. Jeden modul bude môcť zobrazovať v grafickom rozhraní viacero položiek.
- 3.15. Moduly budú s GUI komunikovať na základe mechanizmu, ktorý zabezpečí efektívnu výmenu informácií. Keď modul získa novú informáciu, oznámi to GUI manažérovi, ktorý následne požiada príslušný modul, aby novú informáciu dokreslil do príslušného grafického prvku.
- 3.16. Pri textových aktualizáciách bude modul posilať novú informáciu priamo do GUI manažéra v momente, keď ju má.
- 3.17. Softvér by mal poskytovať vo svojom API aj dialóg pre výber súboru na otvorenie, výber súboru na uloženie a výber priečinka, napríklad pre možnosť načítania mapy.
- 3.18. Systém umožní spustenie GUI aj viackrát na jednom počítači.
- 3.19. Systém umožní spustenie GUI naraz na rôznych počítačoch nezávisle.
- 3.20. Moduly si môžu zaregistrovať svoje volania funkcií na klávesové skratky z množiny klávesových skratiek (napr. CTRL-A a CTRL-D).

### **3.21. Konfiguračný súbor**

- 3.21.1. Konfiguračný súbor obsahuje šablóny ktoré majú byť v danom behu k dispozícií, zoznam modulov a pre každý modul zoznam grafických prvkov a zoznam unikátnych názvov tých prvkov, ktoré v prípade prevádzkového režimu sa majú zaznamenávať a frekvencie ich zaznamenávania a v prípade prehrávacieho režimu sa bude prehrávať záznam. Taktiež obsahuje cestu k priečinku kde sa bude vytvárať podpriečinkov podľa dátumu a času so všetkými logmi z daného behu.
- 3.21.2. Podľa používateľských potrieb môže byť viacero konfiguračných súborov.
- 3.21.3. Konfiguračný súbor sa vyberá pri spustení programu v konfiguračnom móde ako command-line parameter, pričom ak

nebude zadaný vyberie sa prvý v príslušnom adresári pre konfiguračné súbory.

- 3.22.** Softvér bude pracovať v troch módoch: konfiguračný, prevádzkový a prehrávací mód

### **3.23. Konfiguračný mód**

- 3.23.1.** Konfiguračný mód načíta zoznam modulov z konfiguračného súboru.
- 3.23.2.** Konfiguračný mód poskytne možnosť uloženia aktuálneho rozloženia zobrazovaných prvkov vo forme šablóny do zvoleného súboru (save-as).
- 3.23.3.** Konfiguračný mód poskytne možnosť odstránenia vybraného prvku.
- 3.23.4.** V konfiguračnom móde bude prístup k “menu” cez vyskakovací/schovávacie panel, vďaka čomu budú všetky prvky v šablóne maximalizovanejšie na celú obrazovku editovateľné.
- 3.23.5.** V rámci konfiguračného módu bude dostupná aj možnosť výberu šablóny z už uložených šablón.
- 3.23.6.** Hlavnou funkcionalitou konfiguračného módu bude možnosť nastavenia prítomnosti, veľkosti a polohy zobrazovaných prvkov.
- 3.23.7.** V konfiguračnom móde bude možnosť zaznačiť pomocou checkboxou ktoré grafické prvky sa budú v prevádzkovom móde zaznamenávať.

### **3.24. Prevádzkový mód**

- 3.24.1.** Prevádzkový mód bude zobrazovať aktuálnu vybranú šablónu, čo používateľovi umožní pracovať s grafickým rozhraním na základe vopred definovaných šablón.
- 3.24.2.** Prevádzkový mód umožní používateľovi prepínať medzi šablónami pomocou klávesových skratiek, alebo kliknutím na nenápadnú kontrolku niekde v GUI rozhraní.

- 3.24.3.** V prevádzkovom režime by sa mali zaznamenávať všetky prvky, ktoré sú v konfiguračnom súbore označené na logovanie, nezávisle od aktívnej šablóny.
- 3.24.4.** GUI manažér podľa údajov v konfiguračnom súbore odošle konkrétnym modulom zoznam dvojíc názov a frekvencia záznamu grafických prvkov, ktoré sa majú zaznamenávať.
- 3.24.5.** Ak je aspoň jeden grafický prvok označený na zaznamenávanie, GUI manažér vytvorí priečinok podľa aktuálneho dátumu a času a odošle cestu k tomuto priečinku tým modulom, ktoré majú zaznamenávať.
- 3.24.6.** Prevádzkový mód umožní odosielanie používateľom zadáných informácií do príslušných modulov prostredníctvom textových vstupov a tlačidiel a ostatných prvkov menovaných v 3.11.

### **3.25. Prehrávací mód**

- 3.25.1.** GUI manažér sa dá spustiť v režime prehrávania záznamu, ktorý "simuluje" zaznamenaný beh prevádzkového módu.
- 3.25.2.** Prehrávanie podľa záznamu by malo vedieť preškálovať obrázky do inej veľkosti.
- 3.25.3.** GUI manažér by mal volať callback pre moduly, ktorý bude kresliť grafické prvky, prípadne logovať.
- 3.25.4.** GUI manažér poskytne priečinok cez konfiguračný súbor, do ktorého boli dáta zaznamenané a zoznam dvojíc názov a frekvencia záznamu grafických prvkov, ktoré modul následne načíta a spracuje.

# Návrh

Smelý zajko GUI

# Obsah

Obsah	2
Úvod	4
Účel dokumentu	4
Podrobná špecifikácia vonkajších interfejsov	5
Používateľské rozhranie	6
Konfiguračný mód	6
Prevádzkový mód	7
Prehrávací mód	8
UML diagramy	9
Use-case diagramy	9
UML sequence diagram	11
UML component diagram	13
Konfiguračný mód (Configuration Mode)	13
Hlavné komponenty:	13
Princíp fungovania:	13
Prevádzkový a prehrávací mód (Operating/Replay Mode)	13
Hlavné komponenty:	13
Princíp fungovania:	13
Správa systémových zdrojov	13
UML class diagram	15
Návrh implementácie	16
Konfiguračný mód	16
Menubar	17
Templates	17
Options	18
Configuration	18
Add Text Label	19
Test Notifications	19
Add Modules	19
Add Input Elements	20
Prevádzkový mód	20
Moduly a ich grafické reprezentácie	21
Counter Module	21
Map Module	22
Ultrasonic Module	23
Prehrávací mód	24
Výber záznamu	25

Menubar	26
Templates	26
Ovládacie prvky	26
Config	27
<b>Dokumentácia API</b>	<b>28</b>
<b>Formát súborov</b>	<b>29</b>
Konfiguračné súbory (.yaml)	29
Šablóny GUI (.json)	30
Logovacie súbory (.json)	31
<b>Knižnice</b>	<b>34</b>
GLFW	34
Dear ImGui	34
ImGuiFileDialog	34
yaml-cpp	34
nlohmann/json	34
<b>Plán implementácie</b>	<b>35</b>
Fáza 1: Analýza a príprava	35
Fáza 2: Návrh používateľského rozhrania	35
Fáza 3: Tvorba kódu GUI manažéra a modulov	35
Fáza 4: Testovanie a ladenie	36
Fáza 5: Dokumentácia a finalizácia	36
<b>Testovacie scenáre</b>	<b>37</b>
Scenár 1: Spustenie aplikácie a úprava rozhrania	37
Scenár 2: Interakcia s prvkami a notifikácie	37
Scenár 3: Zaznamenávanie a prehrávanie dát	38
Scenár 4: Práca s konfiguračnými súbormi	38
Scenár 5: Viacnásobné spustenie a stabilita systému	38

# Kapitola 1

## Úvod

### Účel dokumentu

Tento dokument poskytuje návrh pre vývoj softvéru pre vizualizáciu a správu dát z mobilného robota. Slúži ako osnova, podľa ktorej sa budeme riadiť počas implementácie. Hlavným cieľom je zabezpečiť efektívne grafické používateľské rozhranie (GUI), ktoré bude spracovávať a vizualizovať dáta z rôznych senzorov robota (teplomer, počítadlo, GPS a ultrazvukové senzory). Zahŕňa návrhy častí používateľského rozhrania, špecifikáciu vonkajších interfejsov, UML diagramy, testovacie scenáre a plán a rozdelenie implementácie medzi členov tímu.



# Kapitola 2

## Podrobná špecifikácia vonkajších interfejsov

### **ImGui knižnica:**

Cesta: libs/imgui

Popis: Používa sa na vytváranie grafických užívateľských rozhraní.

### **GLFW knižnica:**

Cesta: libs/glfw

Popis: Používa sa na vytváranie okien, kontextov a spracovanie vstupu.

### **YAML-CPP knižnica:**

Cesta: libs/yaml-cpp

Popis: Používa sa na parsovanie a generovanie YAML.

### **OpenGL:**

Popis: Používa sa na renderovanie grafiky.

### **Konfiguračné súbory:**

Cesta: config-files/

Formát: YAML

Popis: Tento súbor obsahuje konfiguráciu templátov a modulov.

### **Šablóny:**

Cesta: templates/

Formát: JSON

Popis: Tento súbor obsahuje konfiguráciu grafických modulov, vrátane ich ID, frekvencií, nastavení logovania, pozícií a veľkostí. Takisto obsahuje konfiguráciu elementov ako sú checkbox, slider a iné.

### **Vstupné zariadenia:**

Popis: Aplikácia používa GLFW na spracovanie vstupu z klávesnice a myši.

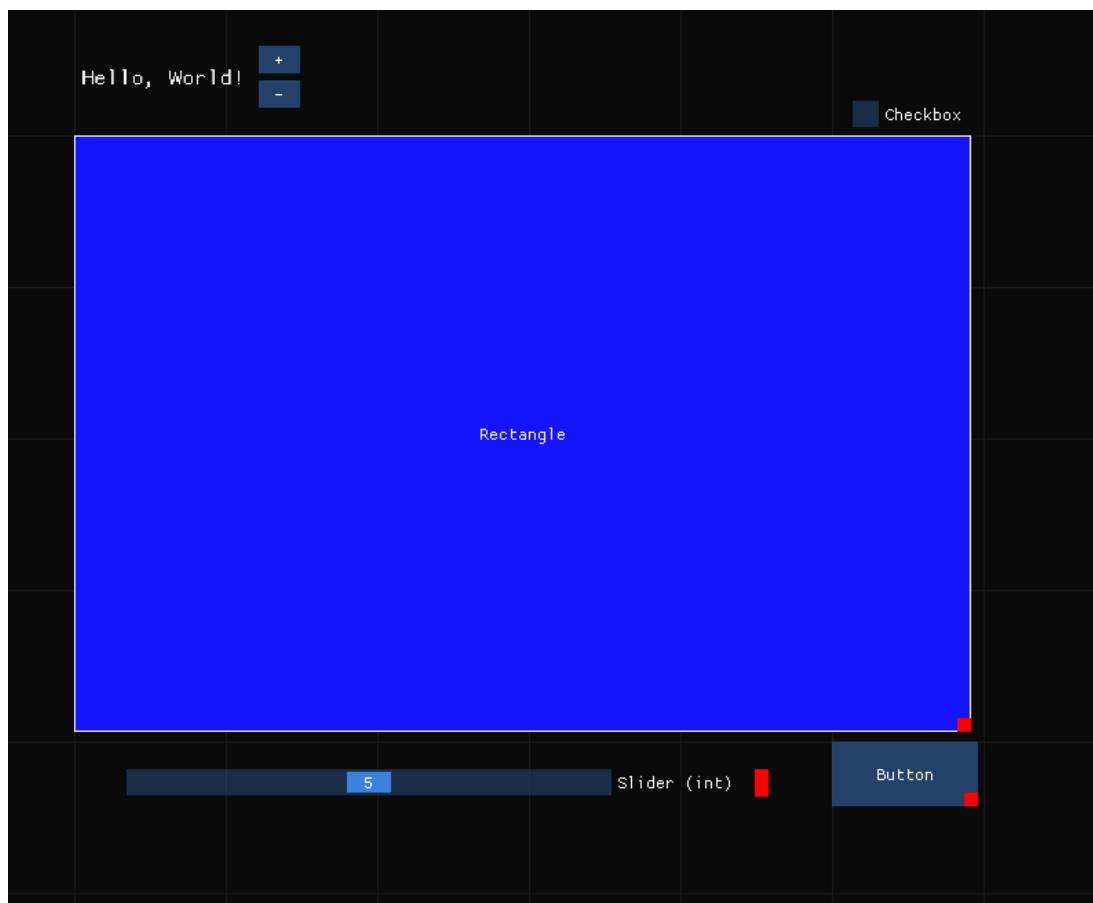
# Kapitola 3

## Používateľské rozhranie

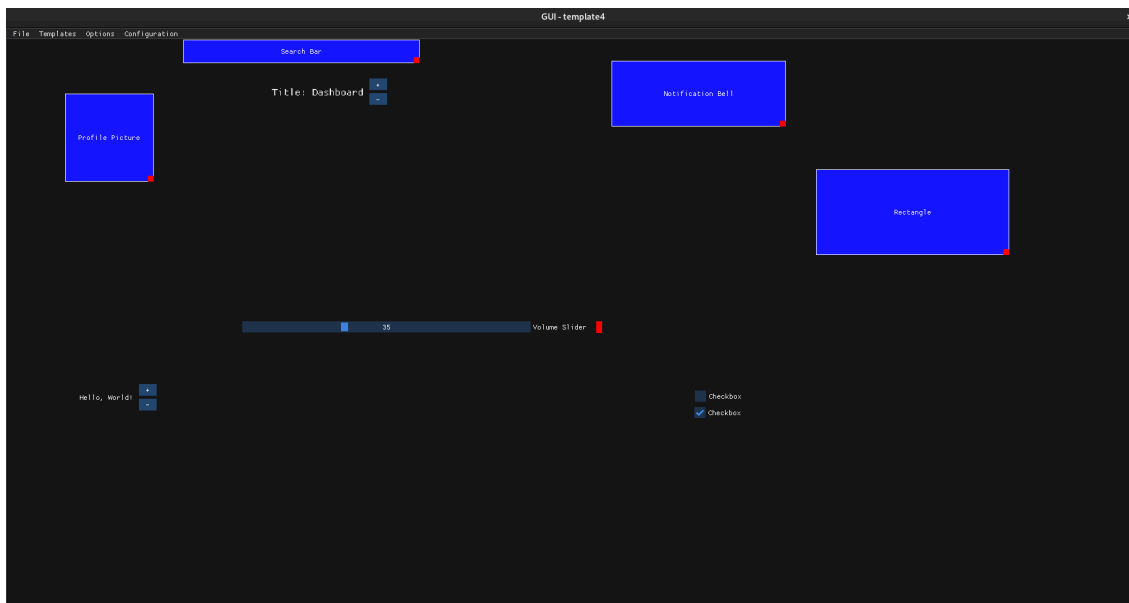
Používateľské rozhranie je navrhnuté ako centrálna súčasť systému, ktorá poskytuje používateľom priamy prístup k riadeniu a monitorovaniu robota. Systém sa vyznačuje tromi hlavnými režimami: konfiguračným, prevádzkovým a prehrávacím. Každý z nich má osobitné využitie a umožňuje používateľovi vykonávať konkrétne operácie potrebné pre riadenie robota.

### Konfiguračný mód

Konfiguračný mód umožňuje používateľovi upravovať vzhľad a nastavenia používateľského rozhrania. V tomto režime môže pridávať alebo odstraňovať prvky, meniť ich rozloženie a upravovať parametre jednotlivých modulov. Hlavným cieľom tohto módu je poskytnúť flexibilné prostredie na prispôsobenie GUI pred samotným používaním v reálnom čase.



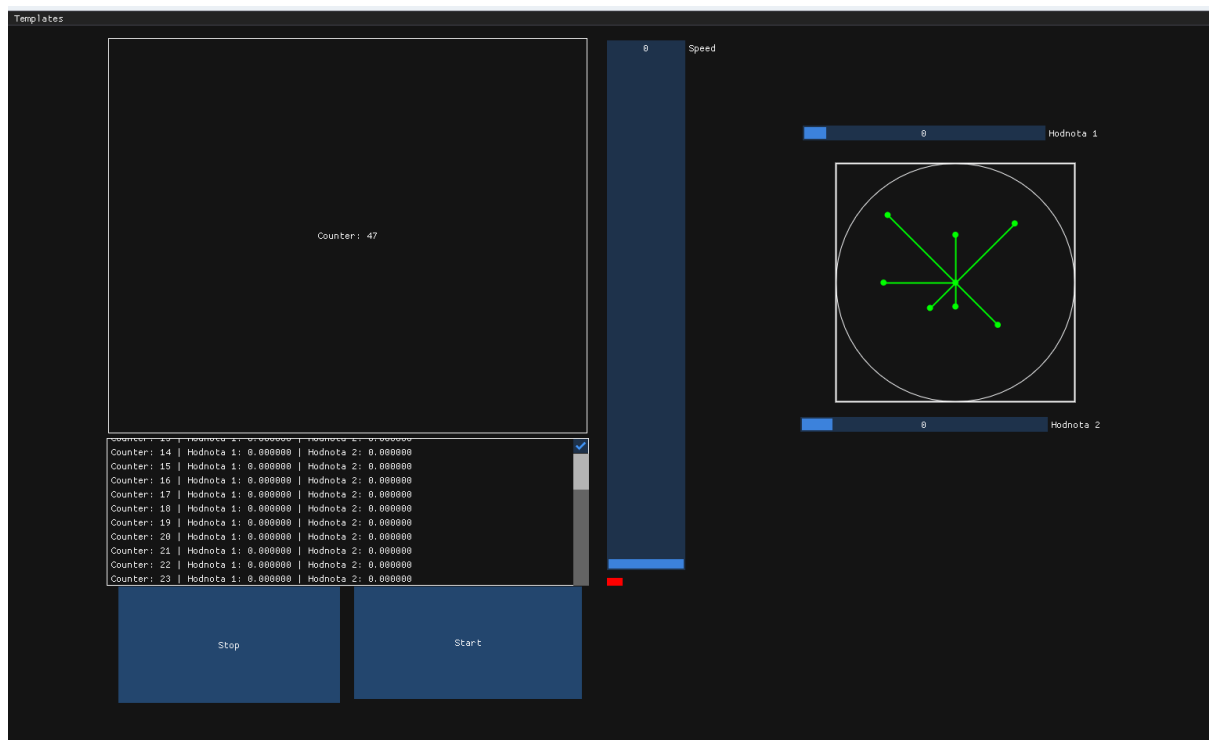
Obr. 3.1: Náhľad konfiguračného módu



Obr. 3.2: Náhľad konfiguračného módu

## Prevádzkový mód

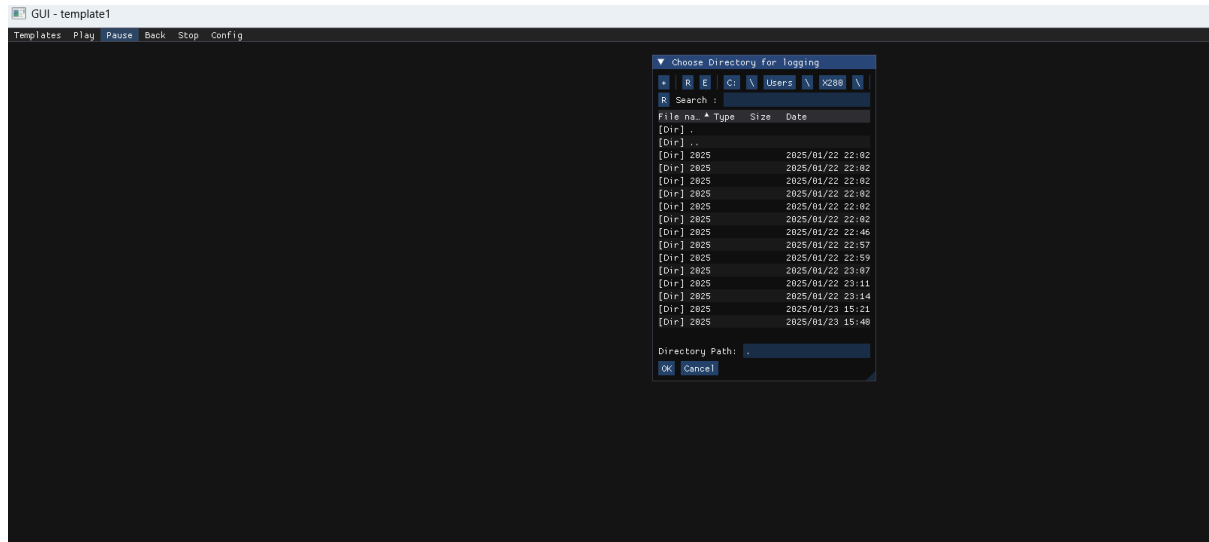
V prevádzkovom móde prebieha reálne monitorovanie a riadenie systému. Používateľ vidí všetky dôležité dáta z pripojených modulov v reálnom čase a môže interagovať s ovládacími prvkami. Tento mód umožňuje dynamickú vizualizáciu dát, automatickú aktualizáciu hodnôt a interaktívne ovládanie jednotlivých komponentov.



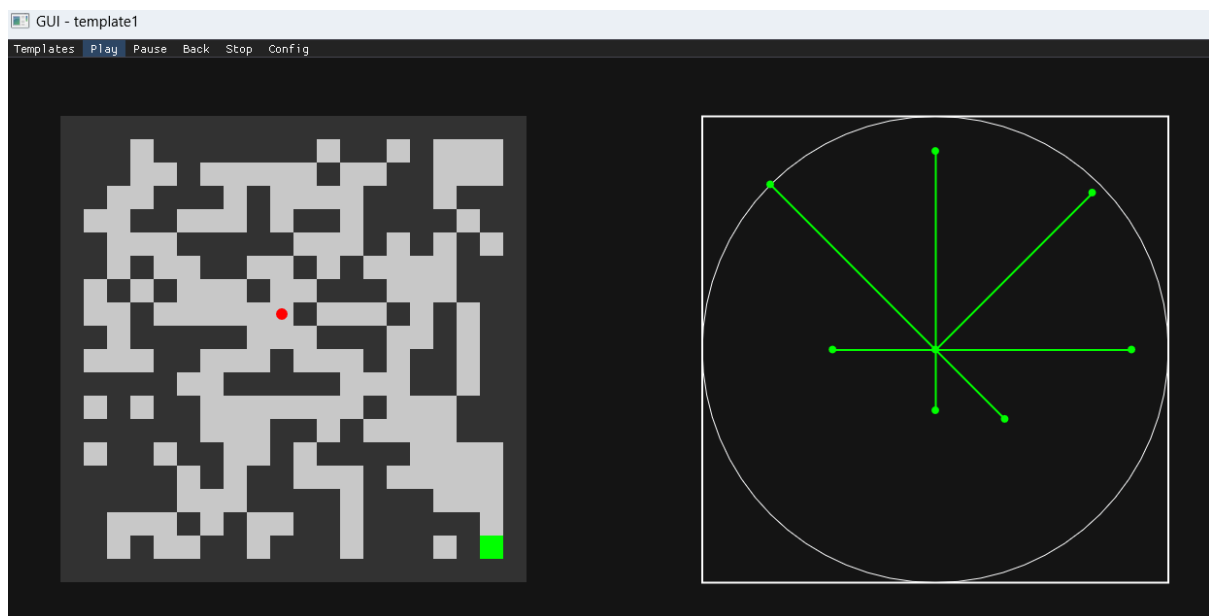
Obr. 3.3: Náhľad prevádzkového módu

## Prehrávací mód

Prehrávací mód slúži na analýzu zaznamenaných dát. Používateľ si môže vybrať konkrétny záznam, ktorý bol vytvorený počas prevádzkového režimu, a prehrať ho v GUI. Tento režim umožňuje spätne analyzovať správanie systému, skontrolovať históriu udalostí a testovať rôzne scenáre bez potreby priameho zapojenia systému do prevádzky.



Obr. 3.4: Náhľad prehrávacieho módu pred vybratím záznamu

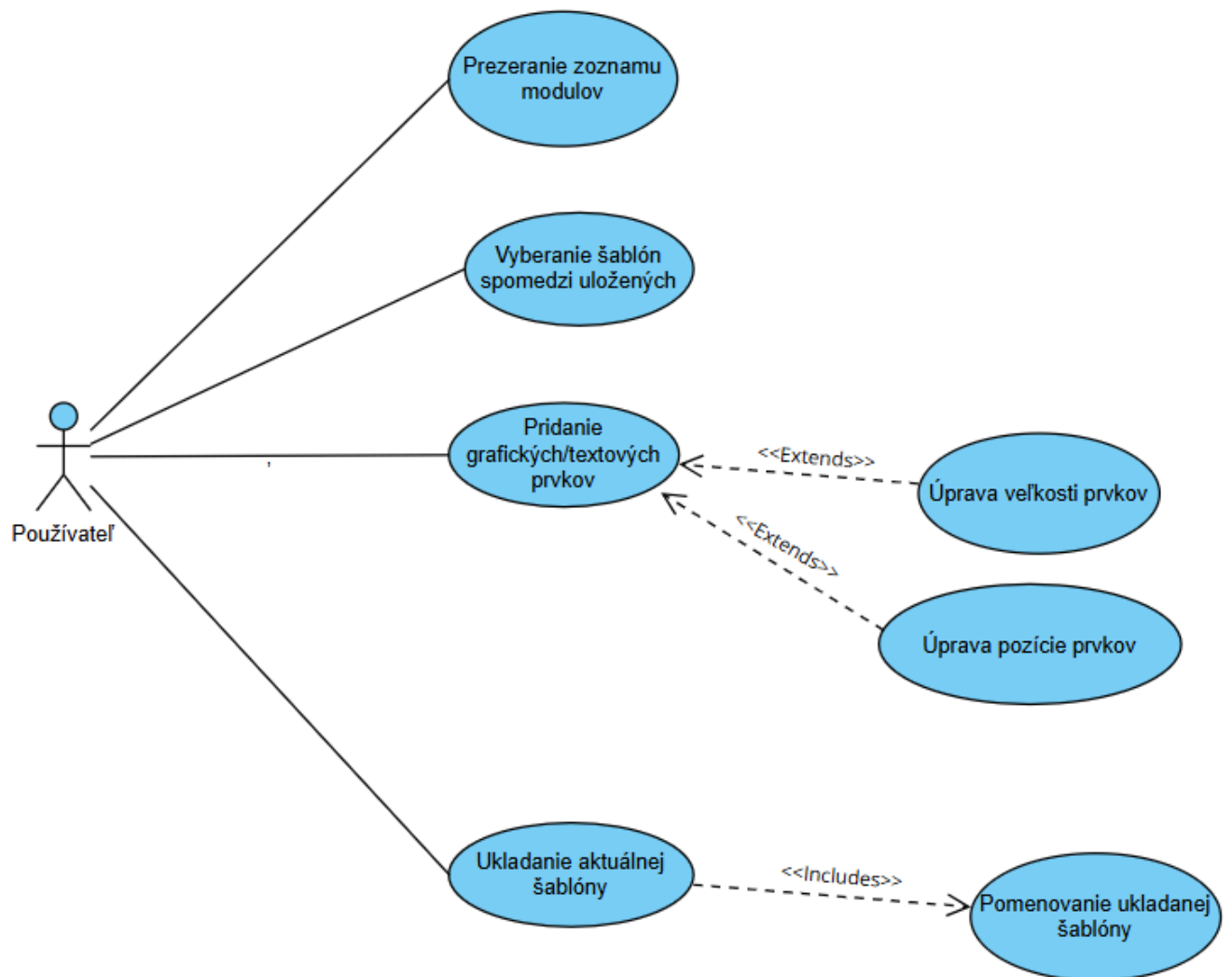


Obr. 3.5: Náhľad prehrávacieho módu po vybratí záznamu

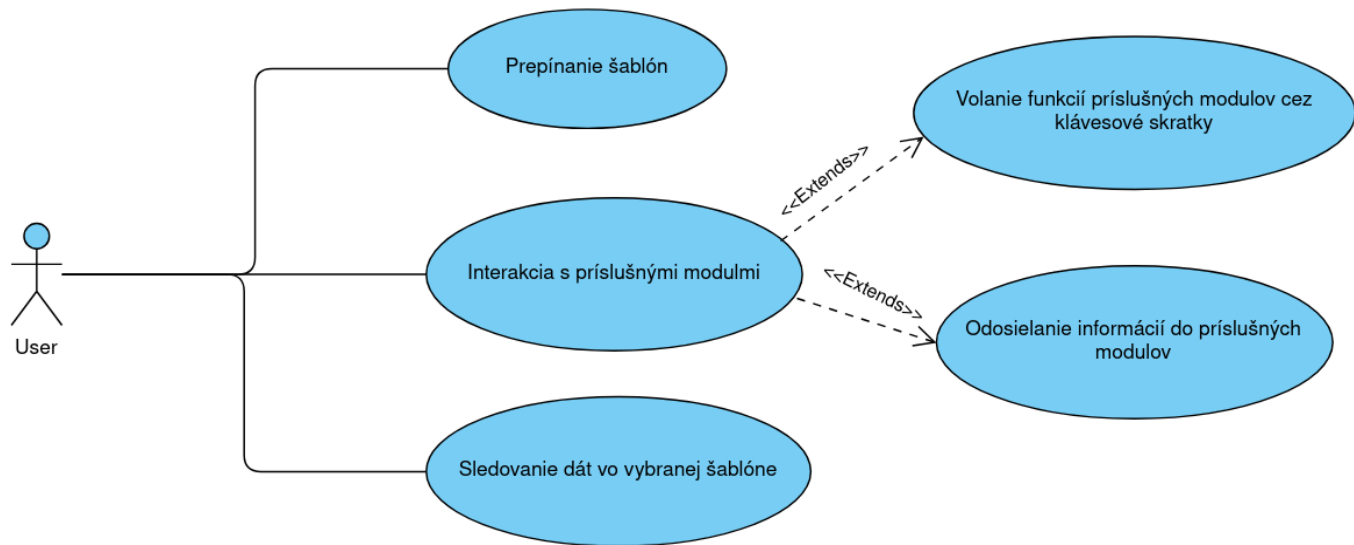
# Kapitola 4

## UML diagramy

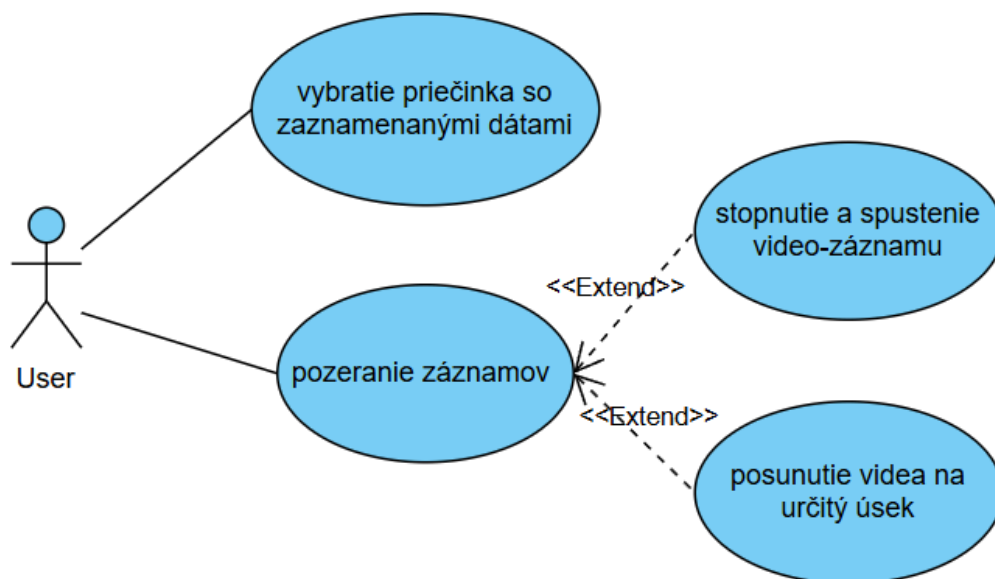
### Use-case diagramy



Obr. 4.1: Use-case diagram konfiguračného módu

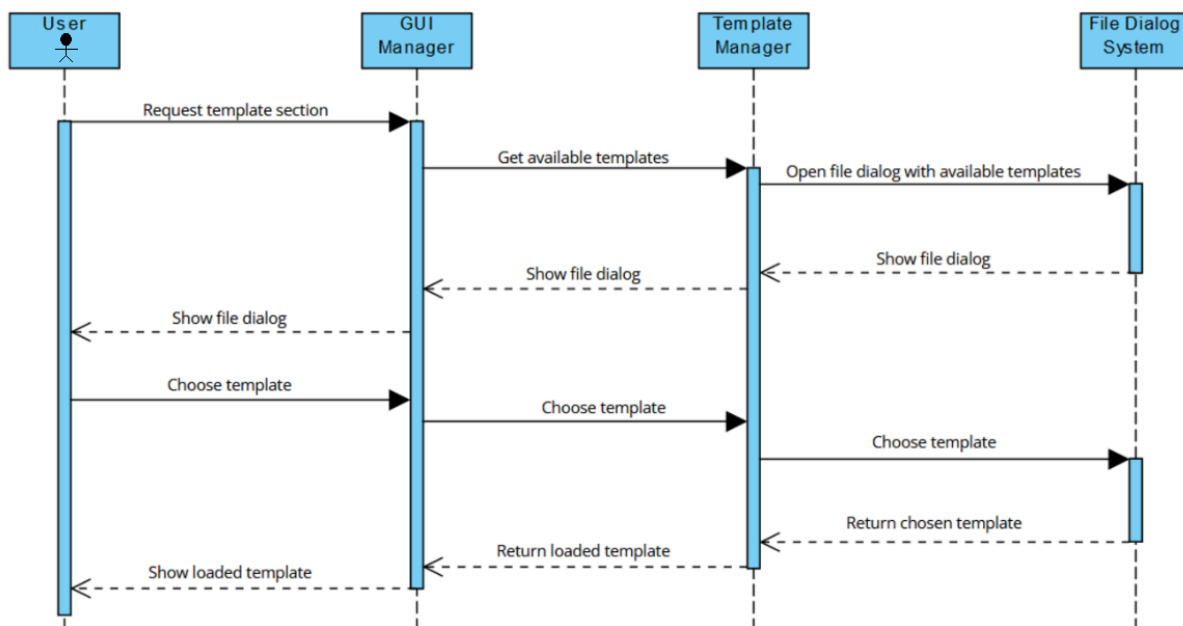


Obr. 4.2: Use-case diagram prevádzkového módu

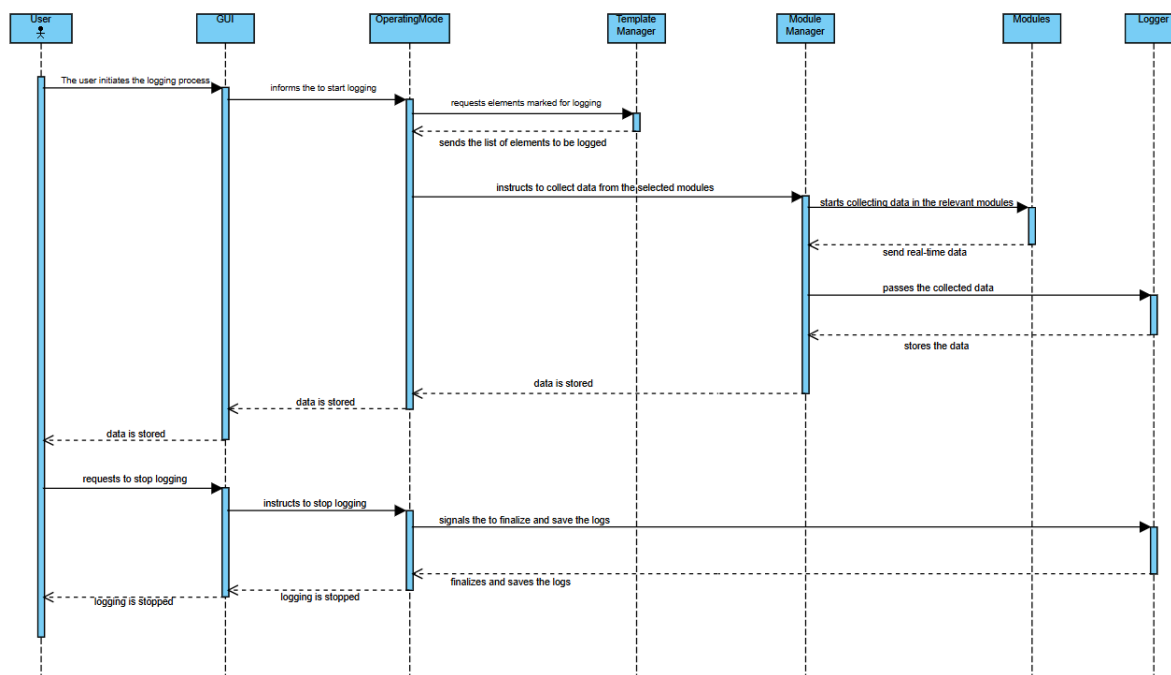


Obr. 4.3: Use-case diagram prehrávacieho módu

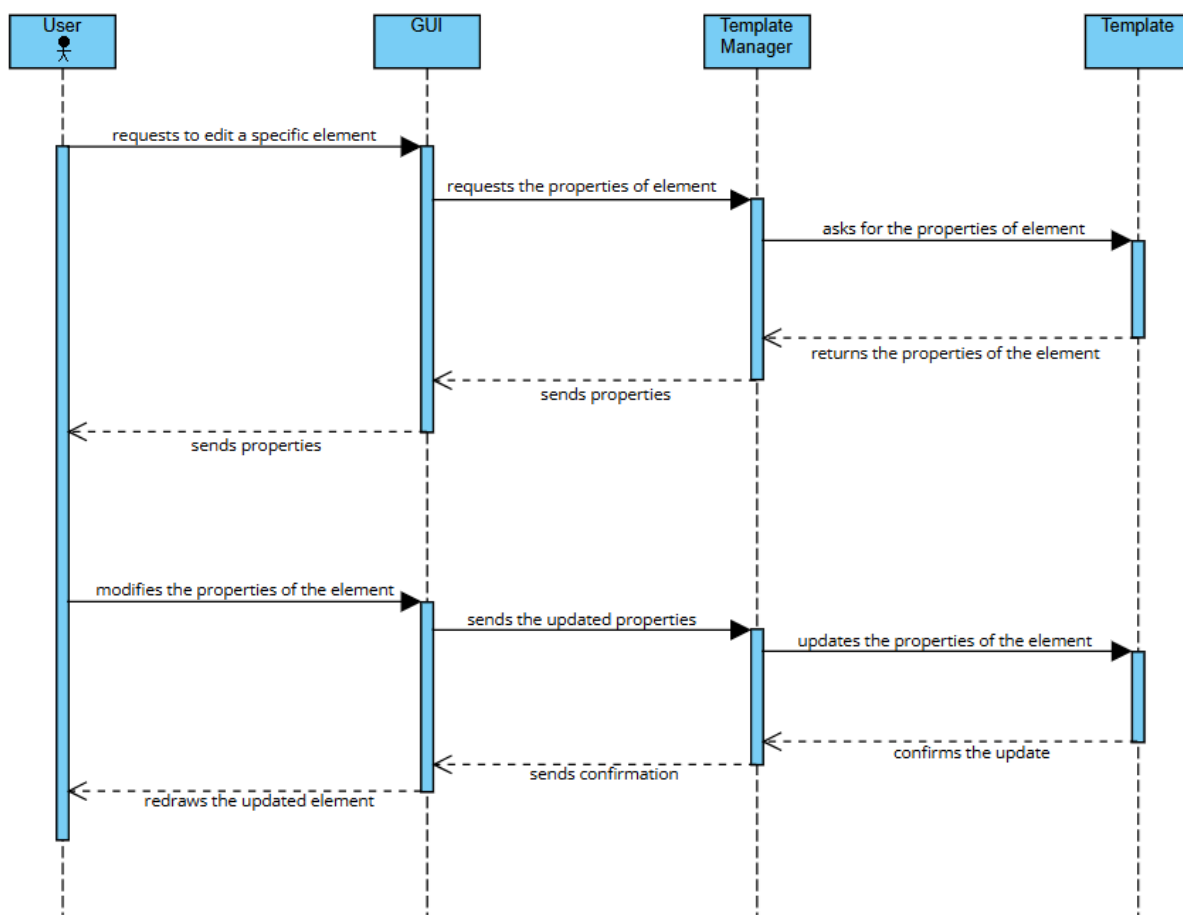
## UML sequence diagram



Obr. 4.4: Sekvenčný diagram pre spoluprácu medzi systémovými komponentmi na zabezpečenie výberu a načítania šablóny.



Obr. 4.5: Sekvenčný diagram pre proces zaznamenávania údajov v prevádzkovom režime.



Obr. 4.6: Sekvenčný diagram pre úpravu prvku v konfiguračnom režime



# UML component diagram

Tento komponentný diagram znázorňuje architektúru systému a jeho hlavné časti:

1. **Konfiguračný mód (Configuration Mode)** – Používateľ nastavuje šablóny, moduly a prvky GUI pred spustením systému.
2. **Prevádzkový a prehrávací mód (Operating/Replay Mode)** – Zabezpečuje vizualizáciu dát v reálnom čase alebo prehrávanie zaznamenaných údajov.
3. **Správa systémových zdrojov** – Načítava konfigurácie a inicializuje moduly.

## Konfiguračný mód (Configuration Mode)

Slúži na nastavenie a správu modulov pred spustením systému.

Hlavné komponenty:

- **ModuleManager** – Riadi pridávanie a konfiguráciu modulov.
- **TemplateManager** – Spravuje vizuálne šablóny GUI.
- **Template & Element** – Definujú rozloženie GUI prvkov.
- **Module** – Reprezentuje senzory a zariadenia.

Princíp fungovania:

1. Používateľ pridá a nastaví moduly.
2. Definuje rozloženie GUI pomocou šablón.
3. Uloží konfiguráciu, ktorá sa neskôr načíta v prevádzkovom režime.

## Prevádzkový a prehrávací mód (Operating/Replay Mode)

Tento režim spracováva a vizualizuje dáta v reálnom čase alebo prehráva uložené záznamy.

Hlavné komponenty:

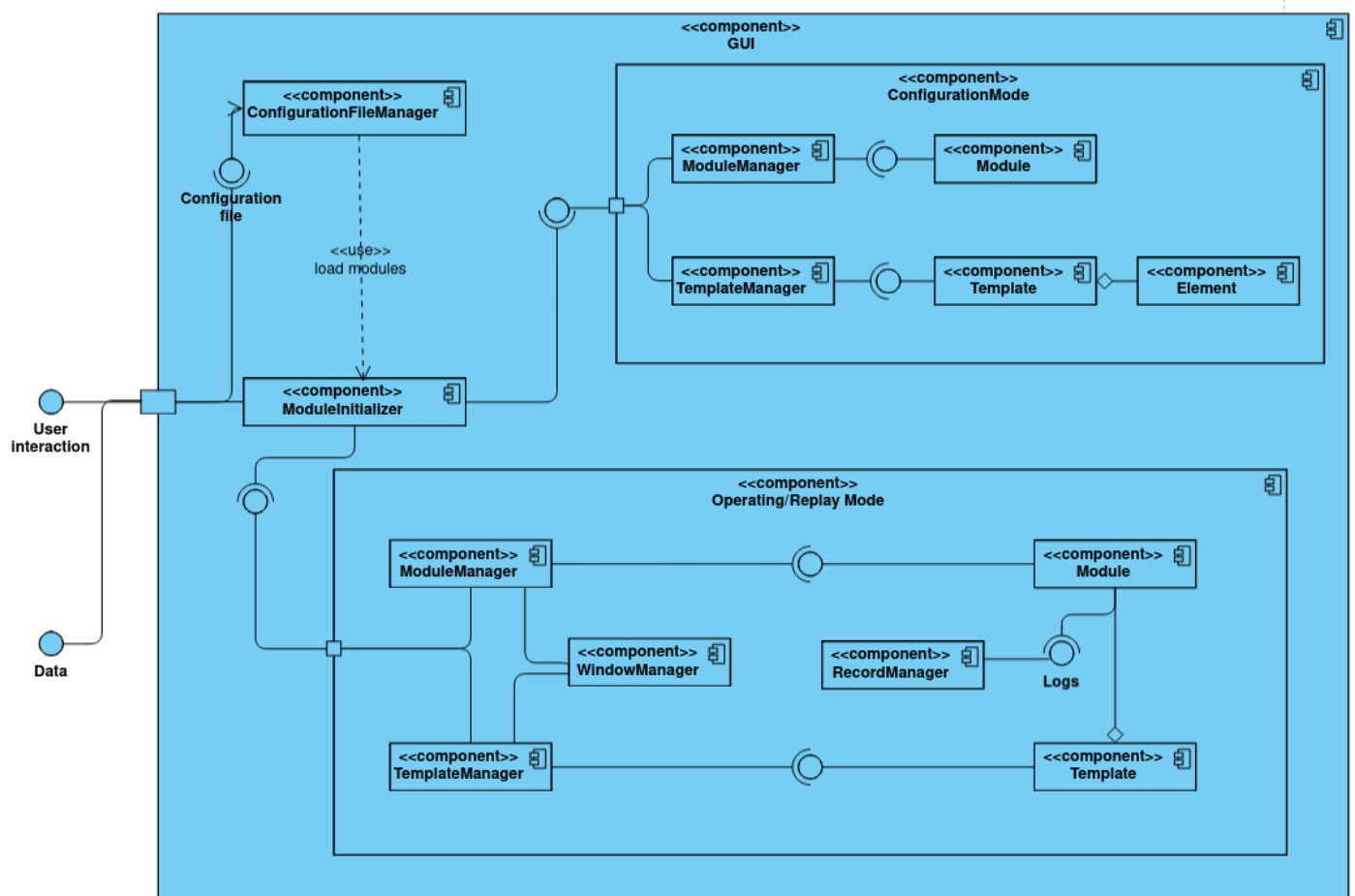
- **ModuleManager** – Spravuje moduly počas prevádzky.
- **TemplateManager** – Používa uložené šablóny na zobrazenie dát.
- **WindowManager** – Spravuje vizualizáciu v GUI.
- **RecordManager & Logs** – Uchováva zaznamenané dáta.

Princíp fungovania:

1. Načítajú sa moduly a šablóny.
2. Dáta sa zobrazujú v reálnom čase.
3. Používateľ môže interagovať s prvkami alebo prehrávať zaznamenané údaje.

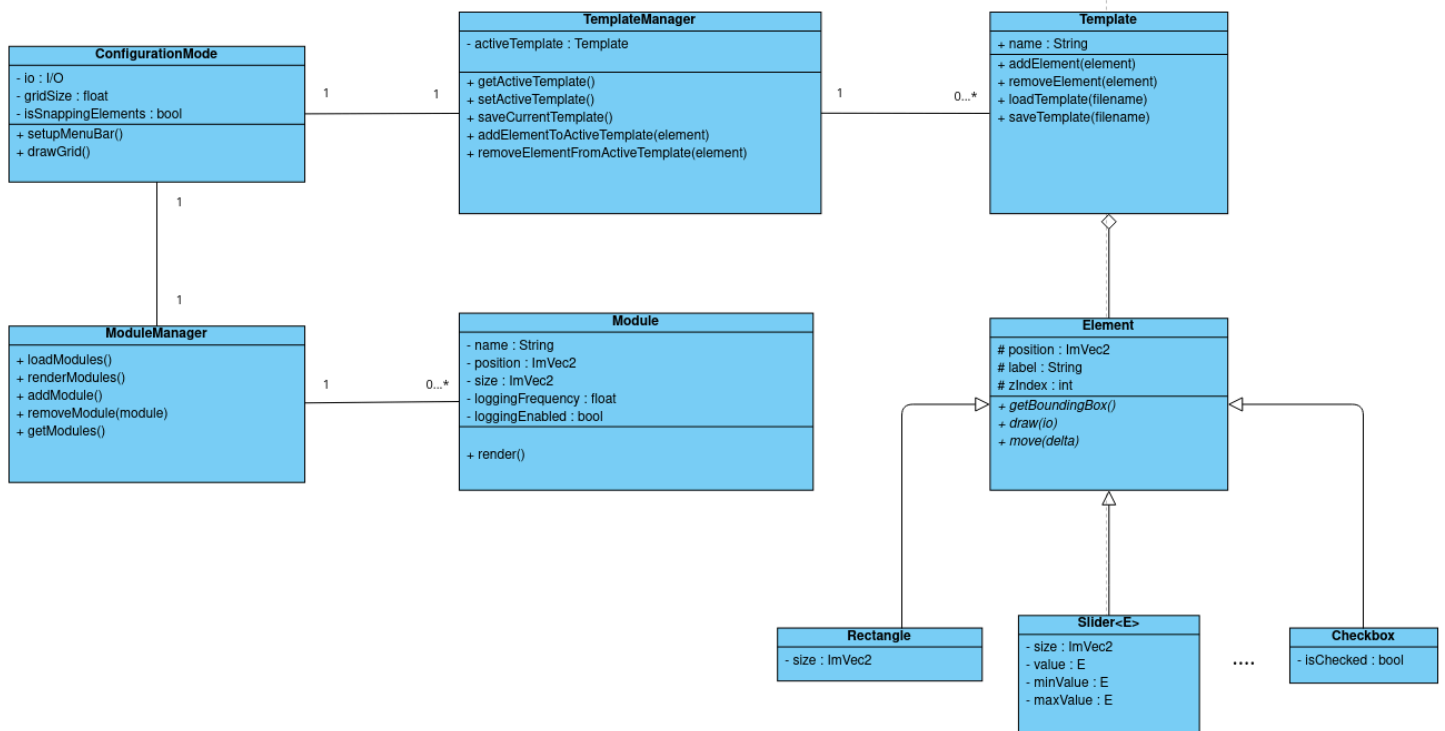
## Správa systémových zdrojov

- **ConfigurationFileManager** – Načítava konfigurácie.
- **ModuleInitializer** – Inicializuje moduly pred spustením systému.



Obr. 4.7: UML component diagram

# UML class diagram



Obr. 4.8: UML class diagram

# Kapitola 5

## Návrh implementácie

### Konfiguračný mód

**Konfiguračný mód** sa spúšťa pomocou argumentu `--config` v príkazovom riadku. Tento režim slúži na konfiguráciu a správu nastavení programu.

#### Spustenie konfiguračného režimu:

##### 1. Argumenty príkazového riadku:

- **--config:** Tento argument aktivuje konfiguračný režim programu.
- **[cesta\_k\_súboru]:** Môžete zadať voliteľnú cestu k YAML konfiguračnému súboru. Ak súbor nešpecifikujete, program automaticky vyhľadá prvý `.yaml` súbor v predvolenom adresári, zvyčajne označenom ako `config-files`.

##### 2. Príklad príkazového riadku

- `program --config /cesta/k/config_súboru.yaml`

Trieda `ConfigurationMode` predstavuje jadro logiky konfiguračného režimu GUI aplikácie. Rozširuje triedu `GUI` a poskytuje funkcie na správu šablón, vytváranie prvkov používateľského rozhrania a umožnenie interakcie používateľov s prispôsobiteľnými komponentmi.

#### Správa šablón:

- Umožňuje načítanie a správu šablón špecifikovaných v konfiguračnom súbore.
- Poskytuje možnosť vytvárať, ukladať a prepínať medzi rôznymi šablónami pomocou triedy `TemplateManager`.

#### Skratky v konfiguračnom móde:

- Uloženie šablóny - `CTRL+S`
- Ukončenie programu - `CTRL+Q`

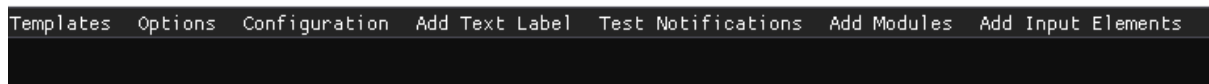
#### Dynamické vytváranie prvkov:

- Umožňuje pridávanie rôznych interaktívnych widgetov, ako sú prvky modulov, zaškrŕtávacie políčka, tlačidlá, posuvníky (slidery) a popisky (labels), do aktívnej šablóny.
- Obsahuje funkciu „prichytávania“ prvkov na mriežku pre presné zarovnanie.

## Prispôsobenie používateľského rozhrania:

- Obsahuje hlavnú ponuku (menu bar) s možnosťami na správu šablón, konfiguráciu nastavení a pridávanie prvkov.

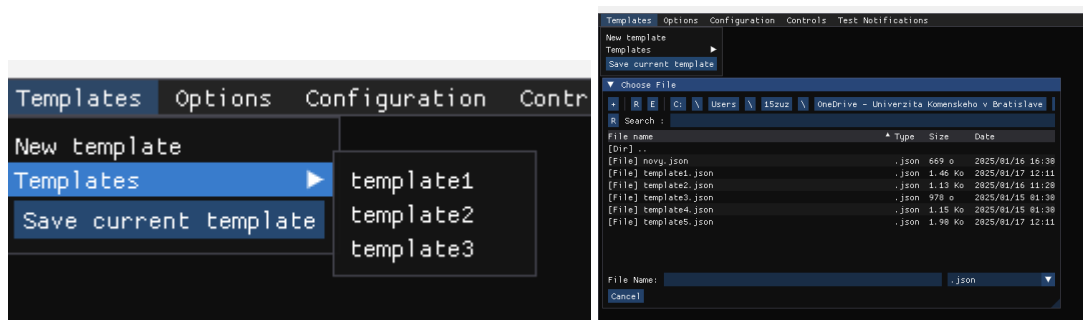
## Menubar



Obr. 5.1: Náhľad menubaru v konfiguračnom móde

Metóda `setupMenuBar` vytvára hlavnú ponuku aplikácie pomocou knižnice ImGui. Menu sa skladá z niekoľkých sekcií, pričom každá ponúka špecifickú funkcionality. Nižšie sú detailné popisy jednotlivých častí menu.

## Templates

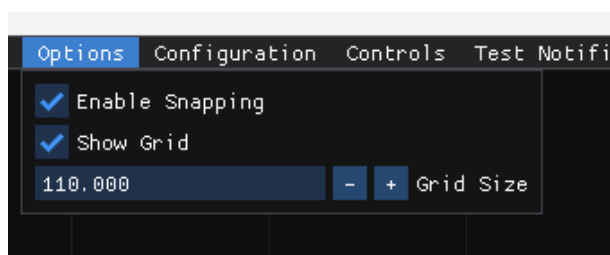


Obr. 5.2: Náhľad šablón a ukladanie šablón v konfiguračnom móde

Táto sekcia umožňuje užívateľovi spravovať šablóny aplikácie:

1. **New Template** - Vytvorí novú prázdnu šablónu a nastaví ju ako aktívnu. Názov okna aplikácie sa aktualizuje na "GUI".
2. **Templates** (podmenu) - Zoznam existujúcich šablón načítaných zo správcu šablón. Umožňuje prepínať medzi šablónami. Pri výbere sa názov okna aktualizuje na "GUI - [názov šablóny]".
3. **Save Current Template** - Umožňuje uložiť aktuálnu šablónu do súboru. Používa dialógové okno na výber súboru a podporuje ukladanie do formátu JSON. Pri ukladaní novej šablóny sa táto pridá do zoznamu všetkých šablón.

## Options

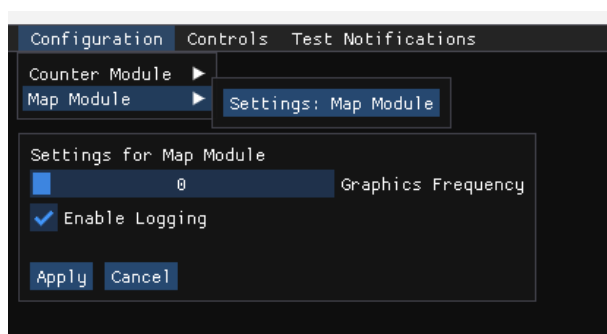


Obr. 5.3: Náhl'ad nastavenia mriežky a automatického vkladania do mriežky v konfiguračnom móde

Táto sekcia ponúka možnosti konfigurácie pracovného prostredia:

1. **Enable Snapping** - Zapne alebo vypne "snapping", čo je funkcia zarovnávaní prvkov na mriežku.
2. **Show Grid** - Zobrazuje alebo skrýva mriežku na pracovnej ploche.
3. **Grid Size** - Nastavuje veľkosť mriežky. Hodnota je obmedzená minimálnymi a maximálnymi hodnotami, aby sa zabránilo chybám (napr. delenie nulou).

## Configuration

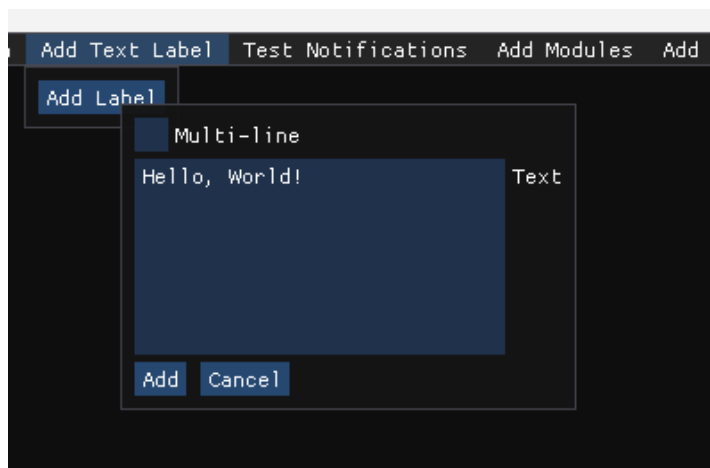


Obr. 5.4: Náhl'ad dynamickej konfigurácie modulov s nastaveniami

Sekcia pre dynamickú konfiguráciu modulov aplikácie:

Dynamicky generované menu podľa konfigurácie načítanej z konfiguračného súboru. Pre každý modul je dostupné tlačidlo **Settings**, ktoré otvára konfiguračné okno pre daný modul, kde sa dá nastaviť logovanie a frekvencia.

## Add Text Label

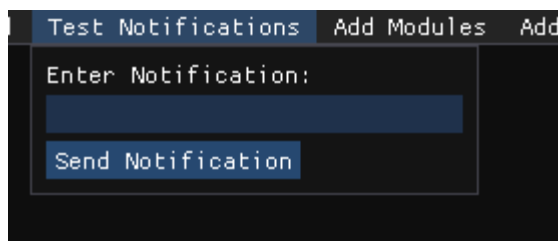


Obr. 5.5: Náhľad pridávania text label

Sekcia pre pridávanie text label do šablóny

- Poskytuje možnosť vytvoriť a nastaviť nový **textový štítok**.
- Po otvorení menu sa volá funkcia `createLabelSettings()`, ktorá zabezpečuje konfiguráciu nastavení pre štítky.

## Test Notifications

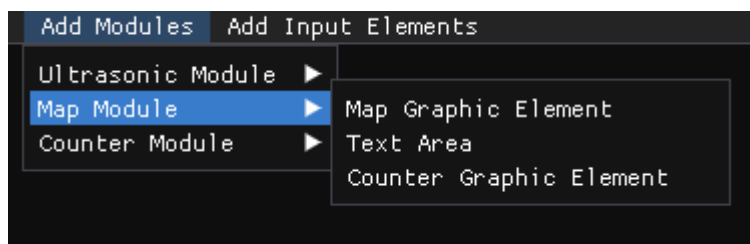


Obr. 5.6: Náhľad pridávania notifikácie

- Umožňuje používateľom odosielať notifikácie cez vlastný správcu `toastManager`.

## Add Modules

Toto menu umožňuje používateľom pridávať prvky modulov, ktoré sú definované jednotlivými modulmi. Každý modul obsahuje špecifické prvky, ktoré môžu byť vložené do aktuálneho rozloženia šablóny.

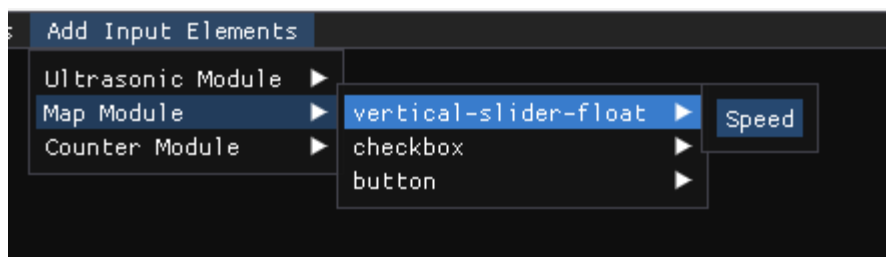


Obr. 5.7: Náhľad pridávania prvku modulu

## Add Input Elements

Toto menu poskytuje používateľovi možnosť pridávať **interaktívne vstupné prvky**, ktoré slúžia na ovládanie alebo modulov. Tieto prvky môžu zahŕňať posúvače, tlačidlá, zaškrŕavacie políčka a textové vstupy.

Každý modul má svoje vlastné podmenu, v ktorom sú zobrazené dostupné vstupné prvky. Po výbere konkrétneho prvku sa tento pridá do aktuálnej šablóny a je pripravený na použitie.



Obr. 5.8: Náhľad pridávania vstupných elementov modulu

## Prevádzkový mód

**Prevádzkový mód** sa spúšťa pomocou argumentu `--operate` v príkazovom riadku. Tento režim slúži na monitorovanie a riadenie systému v reálnom čase pomocou preddefinovaných šablón a modulov, ktoré sú špecifikované v konfiguračnom súbore.

Trieda `OperatingMode` predstavuje základ logiky prevádzkového režimu GUI aplikácie.

Rozširuje triedu `GUI` a zabezpečuje riadenie šablón, záznamov a interakcií s modulmi počas aktívnej prevádzky robota.

- **Riadenie šablón:**

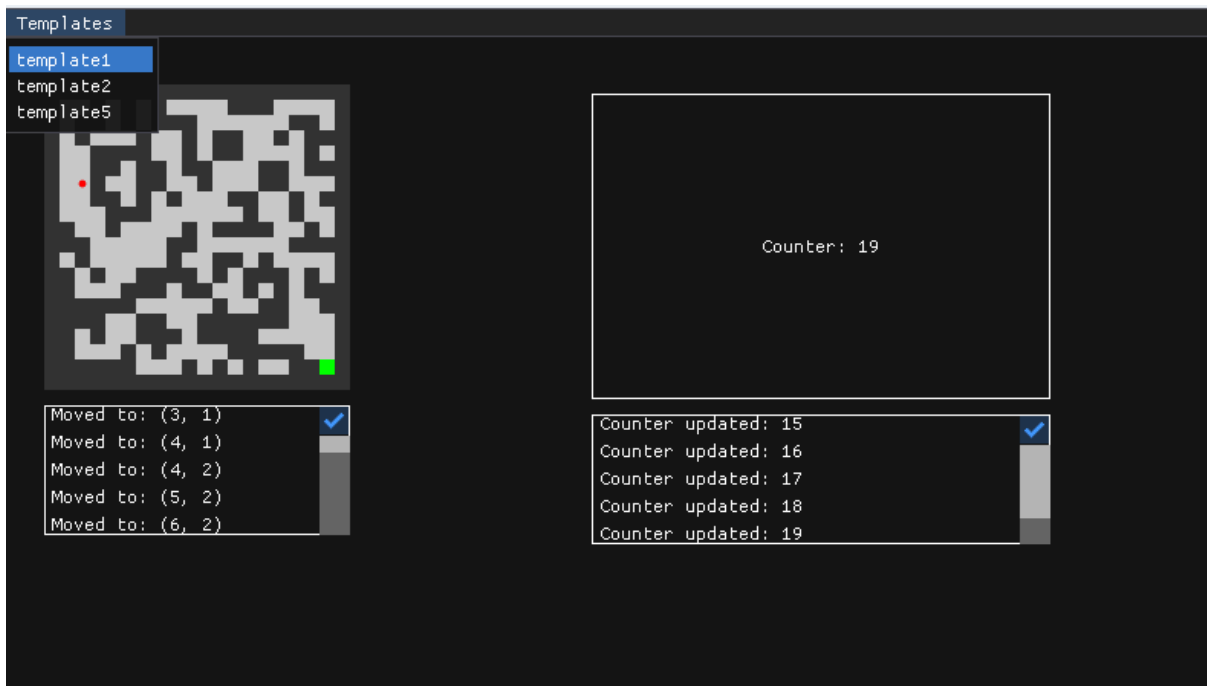
Trieda `OperatingMode` spravuje zoznam šablón načítaných z konfiguračného súboru pomocou triedy `TemplateManager`. Používateľ môže prepínať medzi dostupnými šablónami a aktívna šablóna je vždy synchronizovaná so stavom GUI.

- **Zobrazenie a aktualizácia prvkov:**

GUI vykresľuje grafické a textové prvky na základe aktívnej šablóny a aktuálnych údajov z modulov. Trieda obsahuje metódy na dynamické prispôbovanie prvkov rozlíšeniu obrazovky a ich proporciám.



- **Záznam údajov:**  
`OperatingMode` spolupracuje s modulmi na zázname údajov podľa nastavení definovaných v konfiguračnom súbore. Logovanie prebieha v reálnom čase a zahŕňa vytváranie štruktúrovaných priečinkov na uchovávanie záznamov.
- **Interakcia a ovládanie:**  
Trieda umožňuje používateľom ovládať systém prostredníctvom rôznych interaktívnych prvkov, ako sú tlačidlá, posuvníky alebo textové vstupy. Tieto prvky sú aktualizované v reálnom čase a reagujú na zmeny v systéme.
- **Skratky a prechod medzi šablónami:**  
`OperatingMode` implementuje klávesové skratky, ktoré umožňujú rýchle prepínanie medzi šablónami (CTRL + Šípka doprava, CTRL + Šípka doľava).
- **Inicializácia okna a modulov:**  
Pri spustení režimu trieda inicializuje GUI a načíta všetky potrebné moduly a šablóny. Názov okna sa dynamicky prispôsobuje aktuálne aktívnej šablóne, čo uľahčuje orientáciu používateľa.



Obr. 5.9: Náhľad výberu šablóny

## Moduly a ich grafické reprezentácie

Každý modul implementuje špecifickú logiku spracovania a komunikácie a má priradený zodpovedajúci grafický modul, ktorý zabezpečuje jeho vizualizáciu v GUI. Nižšie je prehľad hlavných modulov a ich grafických reprezentácií:

### Counter Module

- **Hlavná trieda:** `CounterModule`  
Tento modul generuje narastajúcu hodnotu v pravidelných intervaloch (500 ms) a zaznamenáva ju do logu.
- **Grafický modul:** `CounterModuleGraphics`
  - **Funkcia** `draw`: Vykresľuje aktuálnu hodnotu počítadla spolu s dynamicky aktualizovanou textovou oblasťou (`TextArea`) pre záznam logov.
  - **Funkcia** `updateValueOfModule`: Prijíma nové hodnoty z modulu a aktualizuje zobrazené dáta aj logy.
- **Skratky:**
  - `Ctrl+S`: Spustí počítanie.
  - `Ctrl+P`: Zastaví počítanie.
- **Riadenie behu modulu:**
  - **Spustenie a zastavenie:** Funkcie `setValueFromInputElement` umožňujú spúšťať (`Start`) alebo zastavovať (`Stop`) simuláciu.
  - **Interval aktualizácie:** Interval medzi generovaním hodnôt je možné nastaviť prostredníctvom posuvníka `Speed`. Hodnota intervalu je uvedená v milisekundách a určuje, ako často sa generujú nové hodnoty.



Obr. 5.10: Náhľad modulu počítadla

## Map Module

- **Hlavná trieda:** `MapModule`  
Generuje mapu s priechodnými cestami a vizualizuje dynamickú trasu pohybu medzi počiatočným a cieľovým bodom.
- **Grafický modul:** `MapModuleGraphics`
  - Funkcia `draw`: Vykresľuje mapu ako mriežku s rôznymi farbami pre steny, cesty a cieľ. Pohyb po trase sa vizualizuje červenou guľičkou.
  - **Funkcia** `updateValueOfModule`: Aktualizuje polohu guľičky na základe prijatých dát a zaznamenáva pohyb do logov.
- **Skratky:**
  - `Ctrl+R`: Reset mapy.
  - `Ctrl+M`: Zastavenie pohybu.
  - `Ctrl+N`: Spustenie pohybu.

- **Riadenie behu modulu:**
  - **Spustenie a zastavenie:** Funkcie `setValueFromInputElement` umožňujú spúšťať alebo zastavovať pohyb po vygenerovanej trase pomocou zaškrtnávacieho políčka `Running`. Ak je hodnota `true`, pohyb sa spustí, a ak je `false`, pohyb sa zastaví.
  - **Resetovanie mapy:** Funkcia `resetMap` umožňuje obnoviť mapu do počiatočného stavu, vrátane polohy na začiatku trasy, čo je možné aktivovať pomocou tlačidla `Reset`.
  - **Rýchlosť pohybu:** Rýchlosť pohybu po trase je možné upraviť pomocou posuvníka `Speed`. Nastavená hodnota určuje multiplikátor rýchlosti, kde väčšia hodnota znamená rýchlejší pohyb.



Obr. 5.11: Náhľad modulu mapy

## Ultrasonic Module

- **Hlavná trieda:** `UltrasonicModule`  
Simuluje sadu ultrazvukových senzorov, ktoré pravidelne merajú vzdialenosti objektov od robota a ukladajú ich do logu.
- **Grafický modul:** `UltrasonicModuleGraphics`
  - **Funkcia `draw`:** Vizualizuje senzory ako kruhové usporiadanie čiar, ktoré ukazujú vzdialenosti. Textová oblasť pod grafickým prvkom zobrazuje záznam zmien v meraniach.
  - **Funkcia `updateDynamicSensors`:** Dynamicky aktualizuje hodnoty senzorov a zaznamenáva zmeny.
- **Riadenie behu modulu:**
  - **Spustenie a zastavenie:** Funkcie `setValueFromInputElement` umožňujú spúšťať (`Start`) alebo zastavovať (`Stop`) simuláciu.



Obr. 5.12: Náhľad Ultrasonic modulu

## Prehrávací mód

Trieda `ReplayMode` predstavuje základ logiky prehrávacieho režimu GUI aplikácie. Rozširuje triedu `GUI` a zabezpečuje prehrávanie zaznamenaných údajov, vizualizáciu týchto údajov pomocou grafických modulov a interakciu s používateľom počas spätnej analýzy historických dát.

### Spustenie prehrávacieho módu:

- **Argument príkazového riadku:** `--replay` (napríklad: `program --replay /cesta/k/config_súboru.yaml`).
- **Načítanie konfiguračného súboru:** Prehrávací mód využíva YAML konfiguračný súbor, ktorý obsahuje cestu k adresáru s logmi a nastavenia pre šablóny a moduly.

### Funkcionalita prehrávacieho módu:

1. **Prehrávanie dát:**
  - Modul `ReplayMode` načítava logy z adresára špecifikovaného v konfiguračnom súbore.
  - Prehrávanie logov je vizualizované v GUI pomocou grafických modulov, ktoré simulujú predchádzajúce záznamy.
2. **Šablóny:**
  - Používateľ môže prepínať medzi rôznymi šablónami (uloženými konfiguráciami rozloženia grafických prvkov) počas prehrávania.
  - GUI dynamicky upravuje vzhľad podľa vybranej šablóny.
3. **Riadenie prehrávania:**
  - **Play/Pause/Stop:** Používateľ môže prehrávanie spustiť, pozastaviť alebo zastaviť.
  - **Spätné prehrávanie:** Funkcia „Back“ umožňuje preskočiť späť na predchádzajúce logy.
  - **Zmena šablón:** Šablóny je možné meniť pomocou klávesových skratiek.
  - **Úprava rýchlosti:** Možnosť zmeniť pevný časový krok („Fixed Time Step“) cez menu v GUI.

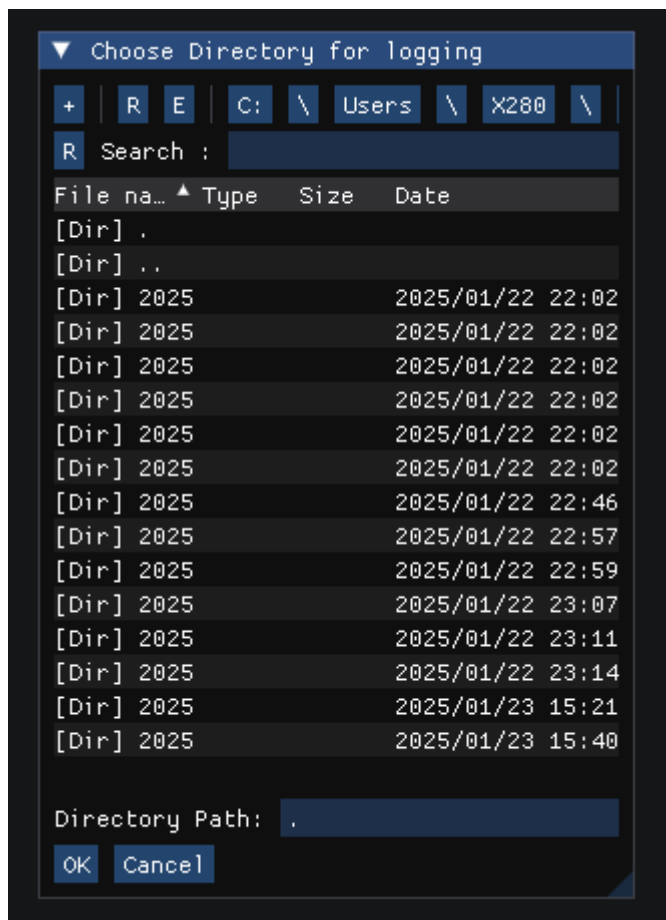
#### 4. Notifikácie a skratky:

- Oznámenia sú zobrazované cez „Toast Manager“.
- Klávesové skratky sú preddefinované na ovládanie šablón a prehrávania (CTRL + Šípka doprava, CTRL + Šípka doľava).

#### 5. Kontrola a načítanie logov:

- Aplikácia overuje, či adresár logov zodpovedá požadovanému formátu (napr. **YYYY-MM-DD\_HH-MM-SS**).
- Používatelia môžu vybrať adresár logov manuálne cez grafický dialóg.

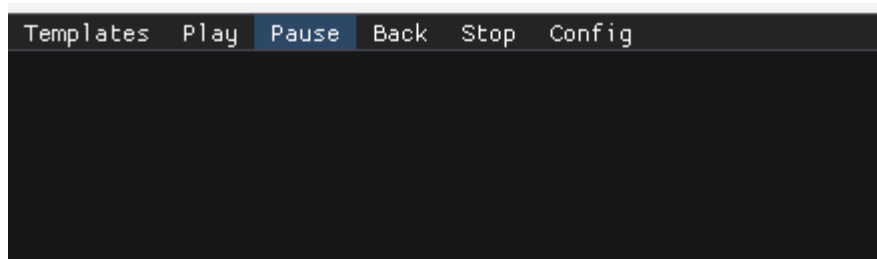
### Výber záznamu



Obr. 5.13: Náhľad okna na výber záznamov

Dialógové okno „**Choose Directory for Logging**“ slúži na výber adresára pre logy v režime prehrávania. Používateľ môže prechádzať adresárovú štruktúru, filtrovať obsah podľa názvu alebo manuálne zadať cestu. Tlačidlá **OK** a **Cancel** umožňujú potvrdenie alebo zrušenie výberu. Po výbere aplikácia overí validitu adresára a načíta logy. Medzi adresármi sa nachádza **.** (**bodka**): Označuje aktuálny adresár, v ktorom sa práve nachádzate. Používa sa na referencovanie aktuálnej pozície v súborovom systéme, a **..** (**dve bodky**): Slúžia na prechod o úroveň vyššie, teda do nadradeného adresára aktuálneho adresára.

## Menubar



Obr. 5.14: Náhl'ad menubaru v prehrávacom móde

Metóda `drawMenuBar()` vytvára hlavnú ponuku aplikácie v režime prehrávania. Hlavné menu je rozdelené do niekoľkých sekcií: **Templates**, **Play**, **Pause**, **Back**, **Stop** a **Config**, ktoré poskytujú funkcionality na ovládanie prehrávacieho režimu.

## Templates



Obr. 5.15: Náhl'ad výberu šablón v prehrávacom móde

Používateľ si môže vybrať konkrétnu šablónu, ktorá upraví rozloženie a vizualizáciu prvkov v GUI. Aktívna šablóna sa automaticky aplikuje na zobrazenie.

## Ovládacie prvky



Obr. 5.16: Náhl'ad ovládacích prvkov v prehrávacom móde

### Play

Možnosť „**Play**“ automaticky spustí prehrávanie po výbere logov a inicializácii systému. Prehráva zaznamenané údaje podľa aktívnej šablóny a frekvencií jednotlivých modulov. Ak je prehrávanie pozastavené (pomocou „Pause“), výberom „Play“ sa prehrávanie opäť spustí od aktuálnej pozície.

## Pause

Možnosť „**Pause**“ dočasne zastaví prehrávanie, pričom všetky moduly zostanú v aktuálnom stave.

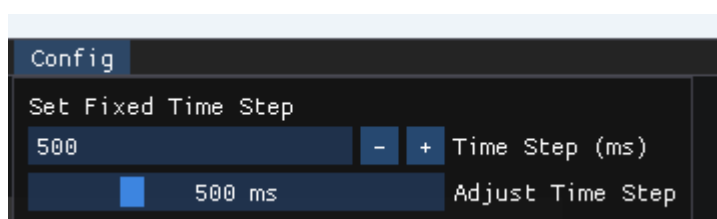
## Stop

Možnosť „**Stop**“ zastaví prehrávanie a ukončí všetky vlákna grafických modulov.

## Back

Možnosť „**Back**“ umožňuje posúvať prehrávanie o krok späť. Tento krok sa synchronizuje s časovými intervalmi prehrávania, aby sa údaje zobrazovali presne podľa ich záznamu.

## Config



Obr. 5.17: Náhľad nastavenia času pre spätný krok v prehrávacom móde

**Config** umožňuje nastaviť pevný časový krok (**Fixed Time Step**) v milisekundách, ktorý sa používa pri spätnom posune prehrávania pomocou funkcie „Back“. Nastavenie ovplyvňuje, o aký časový interval sa prehrávanie vráti späť. Krok sa dá nastavovať klikaním alebo posuvníkom.

# Kapitola 6

## Dokumentácia API

Systém obsahuje vlastné API pre správu modulov, ktoré umožňuje registráciu, aktualizáciu a vizualizáciu modulov v grafickom rozhraní. Trieda `ModuleManager` slúži na centrálné riadenie modulov a ich grafických reprezentácií v systéme. API poskytuje metódy na registráciu modulov, správu ich hodnôt a prepojenie s grafickými modulmi.

Dokumentácia:

<https://github.com/TIS2024-FMFI/smelyzajko-gui/blob/API/TestModules/README.md>



# Kapitola 7

## Formát súborov

Systém pracuje s tromi hlavnými typmi súborov:

1. **Konfiguračné súbory (.yaml)** – obsahujú nastavenia modulov a systému.
2. **Šablóny GUI (.json)** – definujú rozloženie prvkov v používateľskom rozhraní.
3. **Logovacie súbory (.json)** – ukladajú zaznamenané údaje z prevádzky.

### Konfiguračné súbory (.yaml)

Účel:

- Uchovávať konfigurácie jednotlivých modulov, frekvenciu záznamov a povolenie logovania.
- Používajú sa na načítanie nastavení pri spustení systému.

Vysvetlenie:

- **modules** – Obsahuje zoznam dostupných modulov.
- **graphicsLogEnabled** – Určuje, či sa majú zaznamenávať grafické logy.
- **graphicsFrequency** – Frekvencia záznamu vizuálnych údajov (v Hz).
- **textLogEnabled** – Povolenie textového logovania.
- **textFrequency** – Frekvencia zápisu textových logov.
- **logDirectory** – Adresár, kde sa ukladajú logy (hodnota môže byť prázdna, ak nie je špecifikovaná).

```
1  templates:
2  modules:
3    - UltrasonicModule:
4      graphicsLogEnabled: true
5      graphicsFrequency: 120
6      textLogEnabled: true
7      textFrequency: 60
8    - MapModule:
9      graphicsLogEnabled: true
10     graphicsFrequency: 120
11     textLogEnabled: true
12     textFrequency: 60
13   - CounterModule:
14     graphicsLogEnabled: true
15     graphicsFrequency: 120
16     textLogEnabled: true
17     textFrequency: 60
18  logDirectory:
```

Obr. 7.1: Náhľad konfiguračného súboru

## Šablóny GUI (.json)

Účel:

- Definujú vizuálne prvky rozhrania a ich parametre.
- Ukladajú pozície, veľkosti a typy prvkov GUI.

Vysvetlenie:

- **elements** – Zoznam grafických prvkov v GUI.
- **label** – Názov prvku, ktorý sa zobrazí používateľovi.
- **moduleName** – Modul, ku ktorému prvok patrí.
- **position** – Umiestnenie prvku na obrazovke [x, y].
- **size** – Veľkosť prvku [šírka, výška].
- **type** – Typ prvku (napr. posuvník, tlačidlo).
- **value** – Prednastavená hodnota prvku.

```
1  {
2    "elements": [
3      {
4        "label": "Reset",
5        "moduleName": "MapModule",
6        "position": [
7          455.0,
8          229.0
9        ],
10       "size": [
11         100.0,
12         25.0
13       ],
14       "type": "button"
15     },
16     {
17       "label": "Interval",
18       "maxValue": 10,
19       "minValue": 0,
20       "moduleName": "UltrasonicModule",
21       "position": [
22         560.0,
23         369.0
24       ],
25       "size": [
26         200.0,
27         20.0
28       ],
29       "type": "horizontal-slider-int",
30       "value": 5
31     },
32   ]
33 }
```

Obr. 7.2: Náhľad šablóny

## Logovacie súbory (.json)

Účel:

- Zaznamenávajú historické údaje z prevádzky.
- Ukladajú sa pre neskoršie prehrávanie alebo analýzu.

```
{  
  "counter_values": [  
    0,  
    2,  
    5,  
    7,  
    10,  
    12,  
    15,  
    17,  
    20,  
    22,  
    25,  
    27,  
    30,  
  ],  
}
```

Obr. 7.3: Náhľad logu pre counter modul

```
{  
  "graphicsFrequency": 120.0,  
  "sensor_data": [  
    {  
      "angle": 0.0,  
      "distance": 5.0  
    },  
    {  
      "angle": 45.0,  
      "distance": 7.0  
    },  
    {  
      "angle": 90.0,  
      "distance": 4.0  
    },  
    {  
      "angle": 135.0,  
      "distance": 8.0  
    },  
  ],  
}
```

Obr. 7.4: Náhľad logu pre ultrasonic modul

```
{  
  "logs": [  
    "EMPTY",  
    "Counter: 5 | Hodnota 1: 5.000000 | Hodnota 2: 0.000000",  
    "Counter: 10 | Hodnota 1: 5.000000 | Hodnota 2: 0.000000",  
    "Counter: 15 | Hodnota 1: 5.000000 | Hodnota 2: 0.000000",  
    "Counter: 20 | Hodnota 1: 5.000000 | Hodnota 2: 0.000000",  
    "Counter: 25 | Hodnota 1: 5.000000 | Hodnota 2: 0.000000",  
    "Counter: 29 | Hodnota 1: 5.000000 | Hodnota 2: 0.000000",  
    "Counter: 34 | Hodnota 1: 5.000000 | Hodnota 2: 0.000000",  
    "Counter: 39 | Hodnota 1: 5.000000 | Hodnota 2: 0.000000",  
    "Counter: 44 | Hodnota 1: 5.000000 | Hodnota 2: 0.000000",  
  ],  
}
```

Obr. 7.5: Náhľad logu pre textovú oblasť

```
{  
  "ball_positions": [  
    {  
      "col": 0,  
      "row": 0  
    },  
    {  
      "col": 2,  
      "row": 1  
    },  
    {  
      "col": 2,  
      "row": 2  
    },  
    {  
      "col": 2,  
      "row": 3  
    },  
  ],  
}
```

Obr. 7.6: Náhl'ad logu pre modul mapy

# Kapitola 8

## Knižnice

V tomto systéme sú použité nasledujúce knižnice, ktoré poskytujú API na interakciu s rôznymi časťami aplikácie.

### GLFW

Na vytvorenie okna, spracovanie vstupov (klávesnica, myš) a renderovanie OpenGL obsahu.

Dokumentácia: <https://github.com/glwf/glwf/blob/master/README.md>

### Dear ImGui

Vytváranie grafického používateľského rozhrania (GUI) ako je napríklad kreslenie tlačidiel, posuvníkov, textov a iných GUI prvkov.

Dokumentácia: <https://github.com/ocornut/imgui>

### ImGuiFileDialog

Zobrazenie dialógového okna na výber a ukladanie súborov v aplikáciách používajúcich ImGui.

Dokumentácia: <https://github.com/aiekick/ImGuiFileDialog>

### yaml-cpp

Načítanie konfiguračných súborov YAML do aplikácie a ich spracovanie.

Dokumentácia: <https://github.com/jbeder/yaml-cpp>

### nlohmann/json

Načítanie a spracovanie JSON súborov so šablónami GUI a logovacími údajmi.

Dokumentácia: <https://github.com/nlohmann/json>

# Kapitola 9

## Plán implementácie

### Fáza 1: Analýza a príprava

- **Zber a analýza požiadaviek:** Overiť a doplniť všetky požiadavky na systém podľa katalógu požiadaviek.
- **Špecifikácia a návrh architektúry:** Definovať modulárnu architektúru s GUI manažérom ako centrálnou súčasťou, ktorá bude riadiť komunikáciu s jednotlivými senzormi a modulmi robota.
- **Definícia API:** Vytvoriť špecifikáciu API pre komunikáciu medzi GUI manažérom a ostatnými modulmi.

### Fáza 2: Návrh používateľského rozhrania

- **Vytvorenie prototypu GUI:** Vypracovať prvotný návrh rozhrania pre konfiguračný, prevádzkový a prehrávací mód, s ohľadom na usporiadanie ovládacích prvkov.
- **Prispôsobenie UI prvkov:** Nastaviť a otestovať prispôsobenie UI prvkov pre rôzne rozlíšenia a zobrazovacie režimy.
- **Dizajn šablón:** Implementovať základné šablóny zobrazenia údajov z modulov, ktoré sa budú dať upravovať v konfiguračnom móde.

### Fáza 3: Tvorba kódu GUI manažéra a modulov

- **Vytvorenie prostredia konfiguračného módu** (Černák, Neupauerová)
  - Načítanie zoznamu modulov z konfiguračného súboru
  - Nastavenie prítomnosti, veľkosti a polohy zobrazovaných prvkov
  - Uloženie aktuálneho rozloženia zobrazovaných prvkov vo forme šablóny
  - Odstránenie vybraného prvku
  - Výber uloženej šablóny
  - Štruktúra šablóny
  - Posúvanie grafických/textových prvkov v mriežke
- **Vytvorenie prostredia prevádzkového módu** (Beluško, Krajčovič)
  - Načítanie uložených šablón
  - Prepínanie medzi šablónami
- **Vytvorenie prostredia prehrávacieho módu** (Krajčovič, Beluško)
  - Spustenie prehrávania jednotlivých zaznamenaných modulov
- **Vytvorenie grafických modulov + štruktúra konfiguračného súboru** (Krajčovič)
  - Vytvorenie rôznych modulov pre každý mód
  - Nastavenie logovania pre jednotlivé moduly.
  - vytvorenie YAML súboru
- **Definícia API pre komunikáciu s DEROS** (Beluško)
  - Moduly budú komunikovať s GUI prostredníctvom ModuleManagera, ktorý bude zodpovedný za vykresľovanie jednotlivých modulov a zároveň za

aktualizáciu hodnôt, ktoré od týchto modulov dostane. Moduly budú môcť vyvolať funkciu `ModuleManagera updateModule(moduleName, newValue)`, čím zabezpečia aktualizáciu svojich hodnôt v GUI.

- Táto funkcia bude mať návratovú hodnotu typu string, čo umožní grafickým prvkom (napr. sliderom alebo checkboxom) spätnú komunikáciu a posielanie aktualizovaných hodnôt modulom.
- Tento prístup zároveň umožní jednoduché pripojenie na DEROS. Do `ModuleManagera` bude implementovaná funkcia na prijímanie správ od robota Zajka, ktorá zabezpečí synchronizáciu medzi DEROS a GUI.

## Fáza 4: Testovanie a ladenie

- **Testovanie jednotlivých režimov:** Vykonať testovanie pre každý režim (konfiguračný, prevádzkový, prehrávací) podľa definovaných testovacích scenárov.

## Fáza 5: Dokumentácia a finalizácia

- **Dokumentácia API a GUI manažéra:** Pripraviť podrobnú dokumentáciu API a funkčnosti GUI manažéra, vrátane návodov pre používateľov.
- **Ukončenie a odovzdanie projektu:** Zabezpečiť, aby finálna verzia softvéru bola pripravená na odovzdanie, skontrolovať plnenie všetkých požiadaviek a pridať finálne úpravy podľa spätnej väzby.

# Kapitola 10

## Testovacie scenáre

### Scenár 1: Spustenie aplikácie a úprava rozhrania

1. **Akcia používateľa:** Používateľ spustí aplikáciu v konfiguračnom móde.
  - **Očakávaný výsledok:** Aplikácia sa otvorí. (3.4)
2. **Akcia používateľa:** Používateľ vyberie šablónu zo zoznamu uložených šablón.
  - **Očakávaný výsledok:** Systém zobrazí vybranú šablónu na úpravu. (3.4, 3.21.3)
3. **Akcia používateľa:** Klikne na tlačidlo inicializácie.
  - **Očakávaný výsledok:** Moduly začnú vykresľovať svoje prvky. (3.5)
4. **Akcia používateľa:** Používateľ edituje prvky.
  - **Očakávaný výsledok:** Prvky sa presunú a zmenia veľkosť. (3.23.4, 3.23.6)
5. **Akcia používateľa:** Upravené rozloženie uloží ako novú šablónu.
  - **Očakávaný výsledok:** Systém uloží šablónu do konfiguračného súboru. (3.23.2)
6. **Akcia používateľa:** Prepne sa do prevádzkového režimu.
  - **Očakávaný výsledok:** Nové rozloženie je aplikované. (3.24.1)
7. **Akcia používateľa:** Vyberie iné rozlíšenie obrazovky.
  - **Očakávaný výsledok:** Systém proporčne prispôsobí veľkosť prvkov. (3.6, 3.7)

### Scenár 2: Interakcia s prvkami a notifikácie

1. **Akcia používateľa:** Používateľ spustí aplikáciu.
  - **Očakávaný výsledok:** Aplikácia sa maximalizuje a zobrazuje zvolenú šablónu. (3.6, 3.24.1)
2. **Akcia používateľa:** Klikne na tlačidlá v rozhraní.
  - **Očakávaný výsledok:** Tlačidlá vykonávajú príslušné akcie. (3.11)
3. **Akcia používateľa:** Posunie slider na novú hodnotu.
  - **Očakávaný výsledok:** Nová hodnota sa prenesie do modulu. (3.11)
4. **Akcia používateľa:** Vyberie možnosť z dropboxu.
  - **Očakávaný výsledok:** Rozhranie zobrazuje zvolenú možnosť. (3.11)
5. **Akcia používateľa:** Spustí textové vstupy v rozhraní.
  - **Očakávaný výsledok:** Zadaný text je správne zobrazený a odoslaný. (3.11)
6. **Akcia používateľa:** Aktivuje autoscroll v textovej oblasti.
  - **Očakávaný výsledok:** Nové riadky sú automaticky viditeľné bez posúvania. (3.13.1.3)
7. **Akcia používateľa:** Použije scrollbar v textovej oblasti.
  - **Očakávaný výsledok:** Obsah sa posúva podľa používateľských akcií. (3.13.1.2)
8. **Akcia používateľa:** Zmení farebné schémy textových štítkov.



- **Očakávaný výsledok:** Nové nastavenia písma a pozadia sa okamžite aplikujú. (3.13.1.1)

## Scenár 3: Zaznamenávanie a prehrávanie dát

1. **Akcia používateľa:** Používateľ aktivuje prevádzkový režim.
  - **Očakávaný výsledok:** Systém načíta šablónu a zobrazuje prvky podľa konfigurácie. (3.24.1)
2. **Akcia používateľa:** Označí prvky na zaznamenávanie a spustí logovanie.
  - **Očakávaný výsledok:** Systém začne ukladať dáta do logovacieho priečinka. (3.24.3, 3.24.5)
3. **Akcia používateľa:** Interaguje s ovládacími prvkami počas logovania.
  - **Očakávaný výsledok:** Zaznamenané sú všetky zmeny a interakcie. (3.24.4)
4. **Akcia používateľa:** Zastaví logovanie.
  - **Očakávaný výsledok:** Logovanie sa ukončí a uloží všetky dáta. (3.24.5)
5. **Akcia používateľa:** Prepne na prehrávací režim.
  - **Očakávaný výsledok:** Systém načíta uložené dáta a spustí simuláciu. (3.25.1, 3.25.2)
6. **Akcia používateľa:** Prispôsobí veľkosť grafických prvkov počas prehrávania.
  - **Očakávaný výsledok:** Vizualizácia sa upraví podľa nových nastavení. (3.25.3)

## Scenár 4: Práca s konfiguračnými súbormi

1. **Akcia používateľa:** Používateľ otvorí aplikáciu v editačnom režime a upraví existujúcu šablónu.
  - **Očakávaný výsledok:** Systém načíta šablónu z konfiguračného súboru, ktorú je možné upravovať. (3.21.2)
2. **Akcia používateľa:** Upravený konfiguračný súbor uloží na zvolené miesto.
  - **Očakávaný výsledok:** Systém uloží súbor správne. (3.21.1)
3. **Akcia používateľa:** Pri spustení aplikácie zadá parameter pre výber konfiguračného súboru.
  - **Očakávaný výsledok:** Aplikácia načíta zadaný konfiguračný súbor. (3.21.3)
4. **Akcia používateľa:** Použije API na výber súboru na načítanie dát.
  - **Očakávaný výsledok:** Systém zobrazí štandardné okno na výber súboru. (3.17)
5. **Akcia používateľa:** Upraví a uloží šablónu pre ďalšie spustenia.
  - **Očakávaný výsledok:** Upravená šablóna sa uloží do konfigurácie. (3.23.2)

## Scenár 5: Viacnásobné spustenie a stabilita systému

1. **Akcia používateľa:** Spustí aplikáciu na dvoch rôznych počítačoch.
  - **Očakávaný výsledok:** Aplikácie pracujú nezávisle bez rušenia. (3.19)
2. **Akcia používateľa:** Spustí viacero inštancií aplikácie na jednom počítači.
  - **Očakávaný výsledok:** Inštancie fungujú stabilne. (3.18)
3. **Akcia používateľa:** Prepne medzi spustenými inštanciami.

- **Očakávaný výsledok:** Všetky inštancie fungujú bez straty dát alebo vizualizácie. (3.18, 3.19)
- 4. **Akcia používateľa:** Použije API na komunikáciu medzi modulmi.
  - **Očakávaný výsledok:** Moduly odosielajú a prijímajú informácie správne. (3.15)
- 5. **Akcia používateľa:** Skontroluje stabilitu systému pri simultánnom vykresľovaní dát.
  - **Očakávaný výsledok:** GUI nezamrzne a údaje sú správne zobrazené. (3.13, 3.15)