Univerzita Komenského v Bratislave Fakulta matematiky, fyziky a informatiky

Návrh

Smelý zajko GUI

2024
Tibor Černák, Zuzana Neupauerová, Štefan Beluško, Matej Krajčovič

Obsah

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, 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 templatov 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 4

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 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

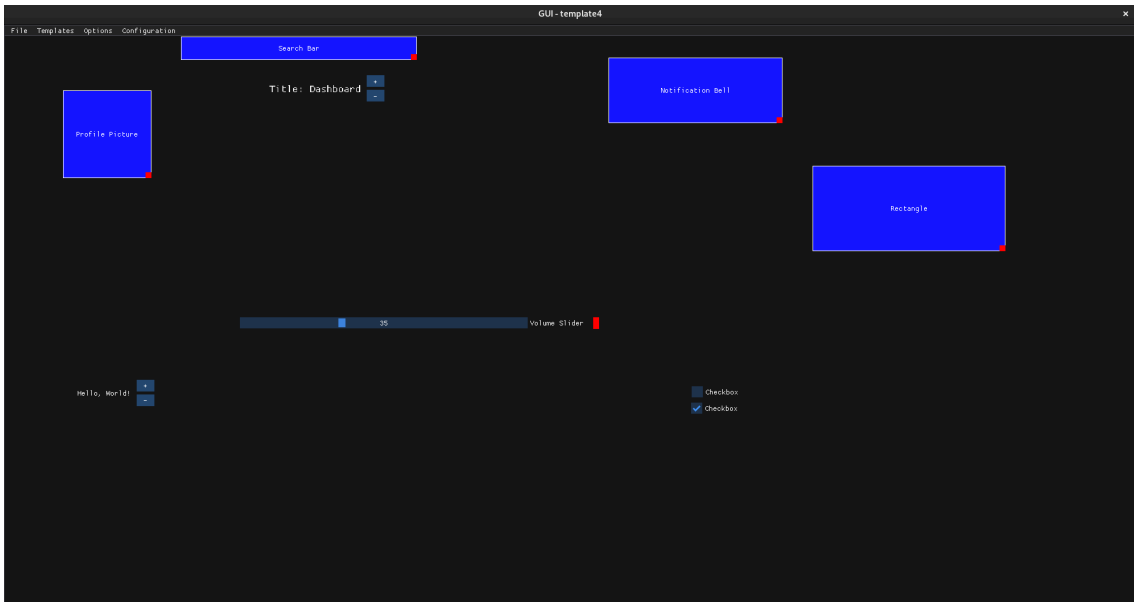
Dynamické vytváranie prvkov:

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

Prispôsobenie používateľského rozhrania:

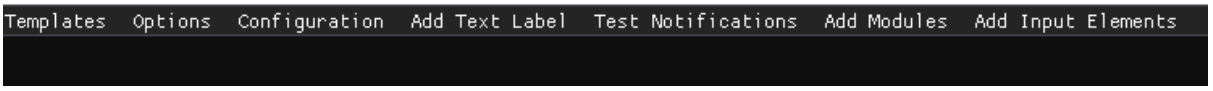
- Obsahuje hlavnú ponuku (menu bar) s možnosťami na správu šablón, konfiguráciu nastavení a pridávanie prvkov.

Obr. 4.1: Náhľad konfiguračného módu



Obr. 4.2: Náhľad konfiguračného módu

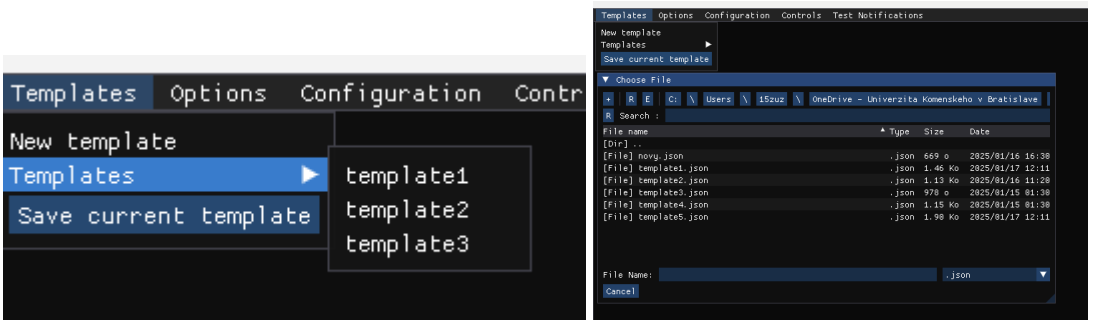
Menubar



Obr. 4.3: Náhľad menubaru v konfiguračnom móde

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

Templates

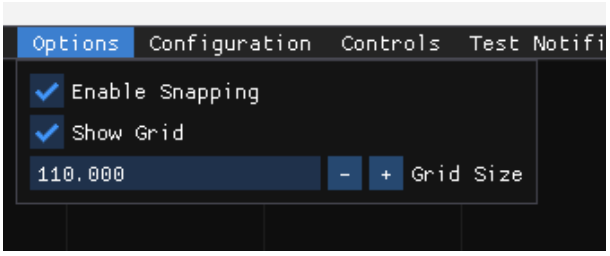


Obr. 4.4: Náhľad šablón a ukladanie šablón v konfiguračnom móde

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

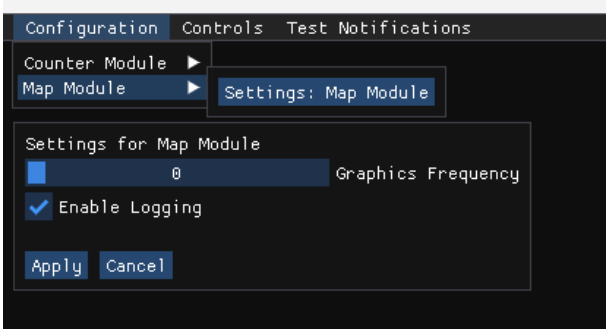


Obr. 4.5: Náhľad nastavenia mriežky a automatického vkladania do mriežky v konfiguračnom móde

Táto sekcia ponúka možnosti konfigurácie pracovného prostredia:

1. **Enable Snapping** - Zapne alebo vypne "snapping", čo je funkcia zarovnávania 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).

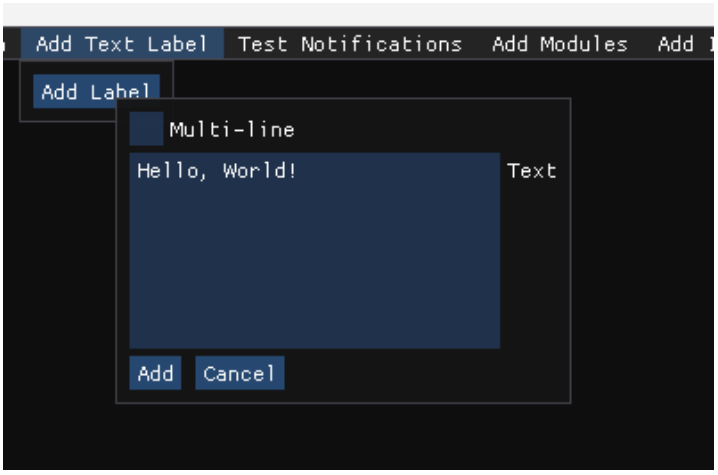
Configuration



Obr. 4.6: Náhľad 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

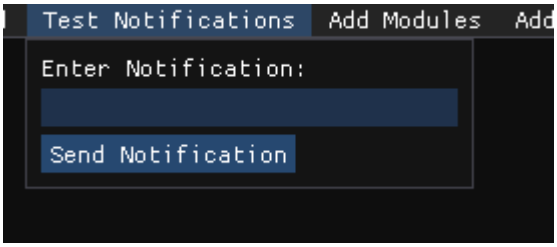


Obr. 4.7: Náhľad pridávania text label

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

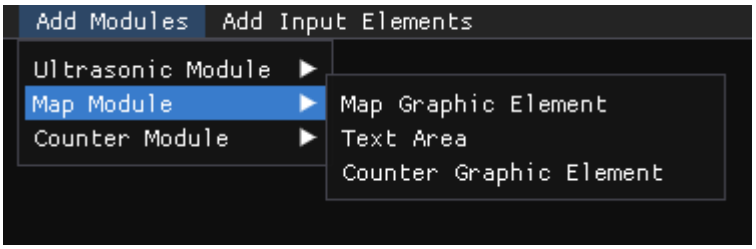


Obr. 4.8: Náhľad pridávania notifikácie

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

Add Modules

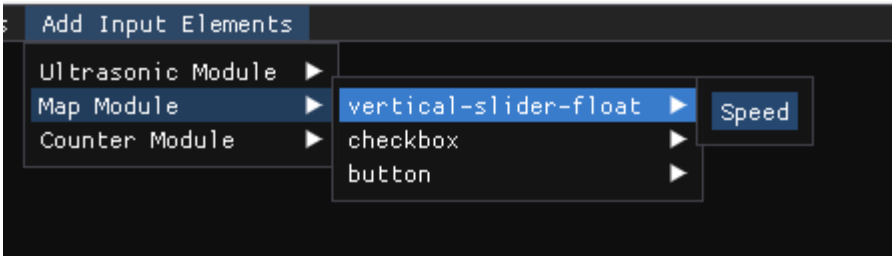
Toto menu umožňuje používateľom pridávať prvky modulov, ktoré sú definované jednotlivými modulmi. Každý modul obsahuje špecifické prvky, ktoré môžu byť vložené do aktuálneho rozloženia šablóny.



Obr. 4.8: 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, zaškrtávacie 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.



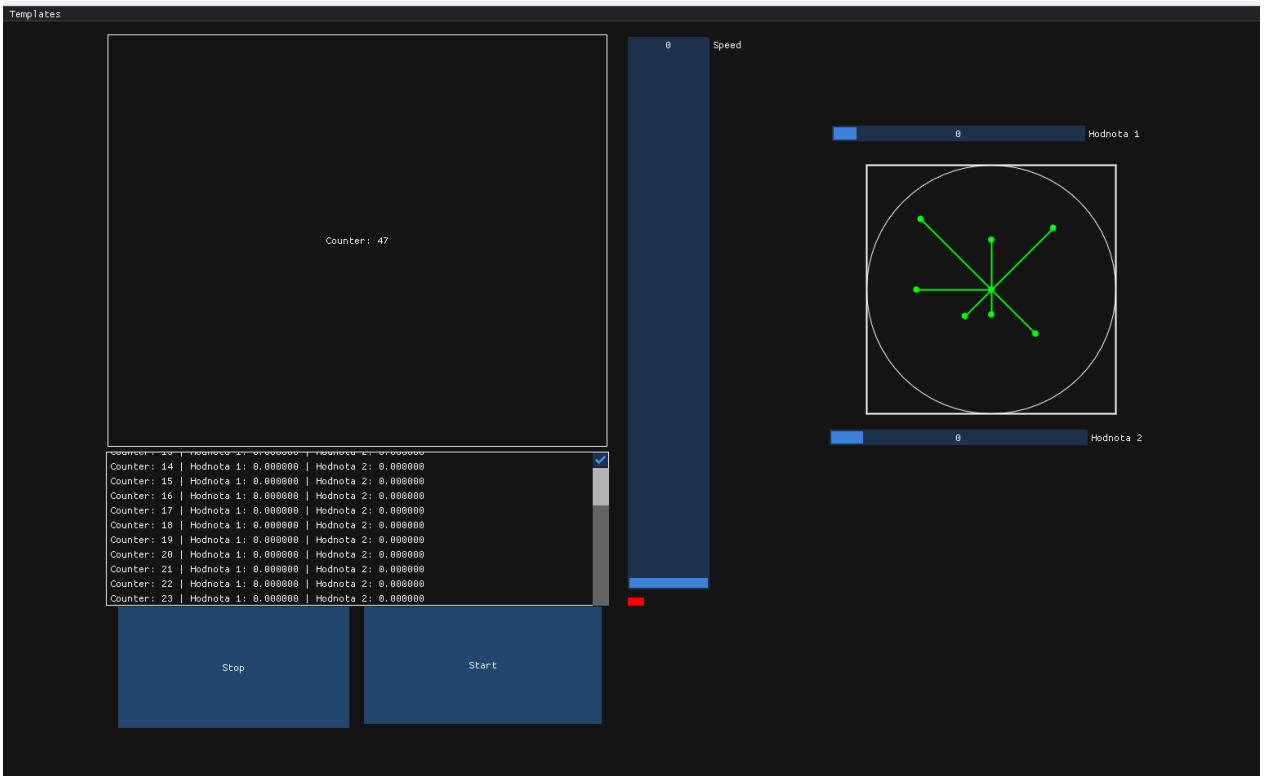
Obr. 4.9: Náhľad pridávania vstupných elementov modulu

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 prispôsobovanie 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. 4.9: Náhľad prevádzkového módu

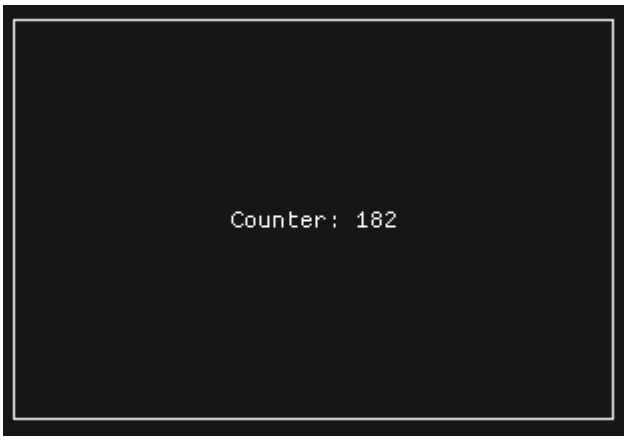
Obr. 4.10: Náhľad výberu šablóny

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 `setValueFromInputElements` 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. 4.11: 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 guličkou.
  - **Funkcia** `updateValueOfModule`**:** Aktualizuje polohu guličky 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 `setValueFromInputElements` umožňujú spúšťať alebo zastavovať pohyb po vygenerovanej trase pomocou zaškrtávacieho 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. 4.12: Náhľad modulu mapy

Ultrasonic Module

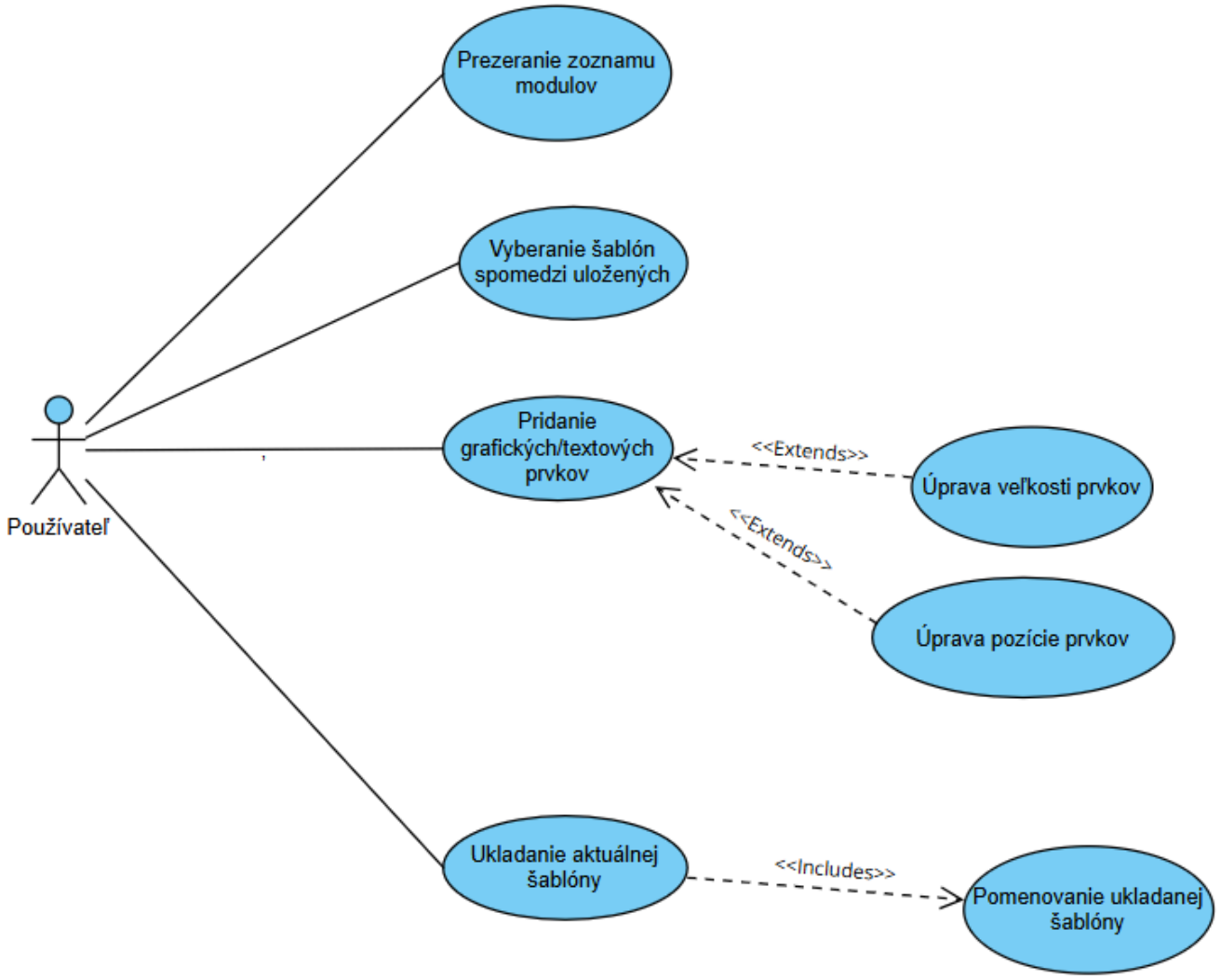
- **Hlavná trieda:** `UltrasonicModule`
  Simuluje sadu ultrazvukových senzorov, ktoré pravidelne merajú vzdialenosti objektov od robota a ukladajú ich do logu.
- **Grafický modul:** `UltrasonicModuleGraphics`
    - **Funkcia** `draw`**:** Vizualizuje senzory ako kruhové usporiadanie čiar, ktoré ukazujú 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 `setValueFromInputElements` umožňujú spúšťať (`Start`) alebo zastavovať (`Stop`) simuláciu.

Obr. 4.13: Náhľad Ultrasonic modulu

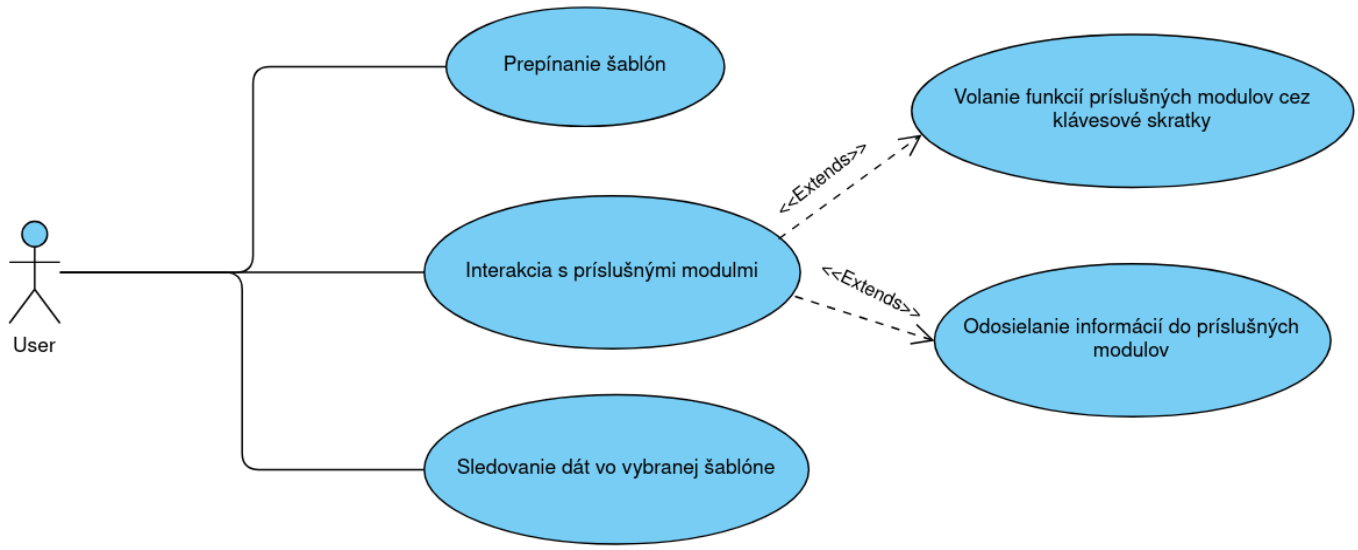
Kapitola 5

UML diagramy

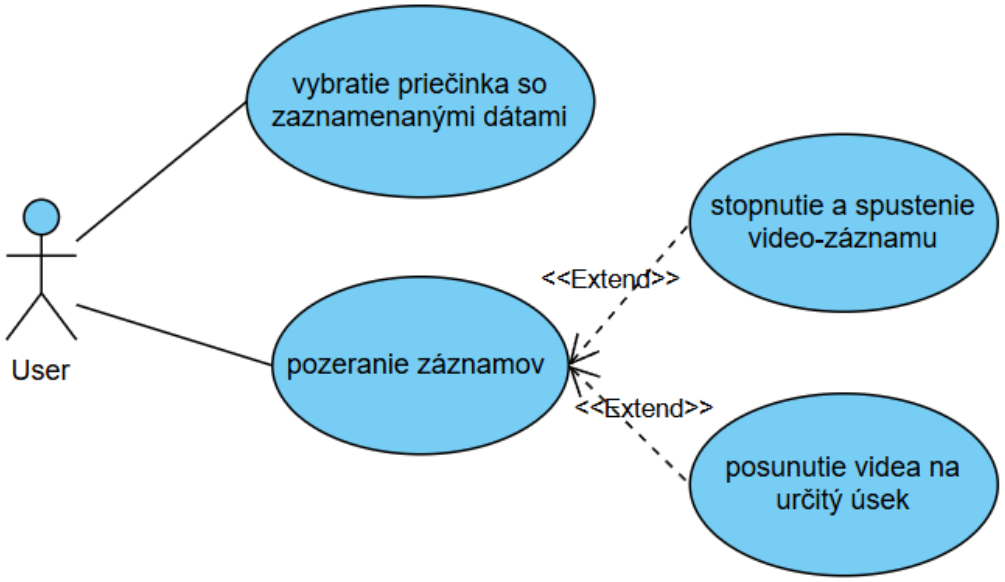
Use-case diagramy



Obr. 5.1: Use-case diagram konfiguračného módu

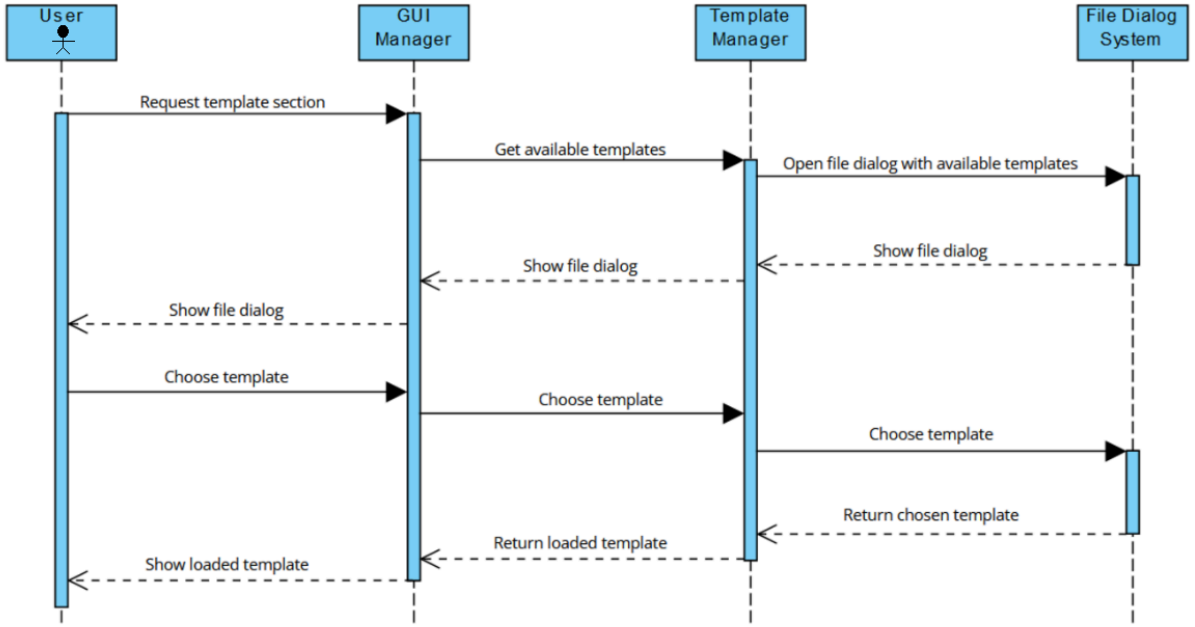


Obr. 5.2: Use-case diagram prevádzkového módu

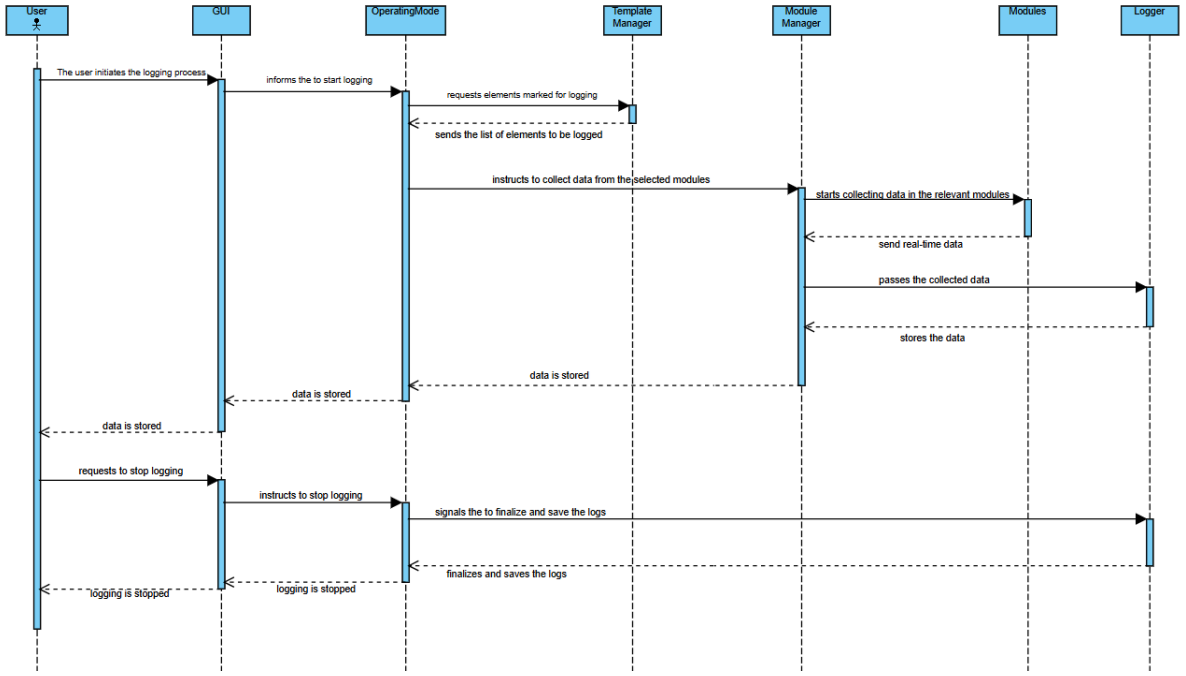


Obr. 5.3: Use-case diagram prehrávacieho módu

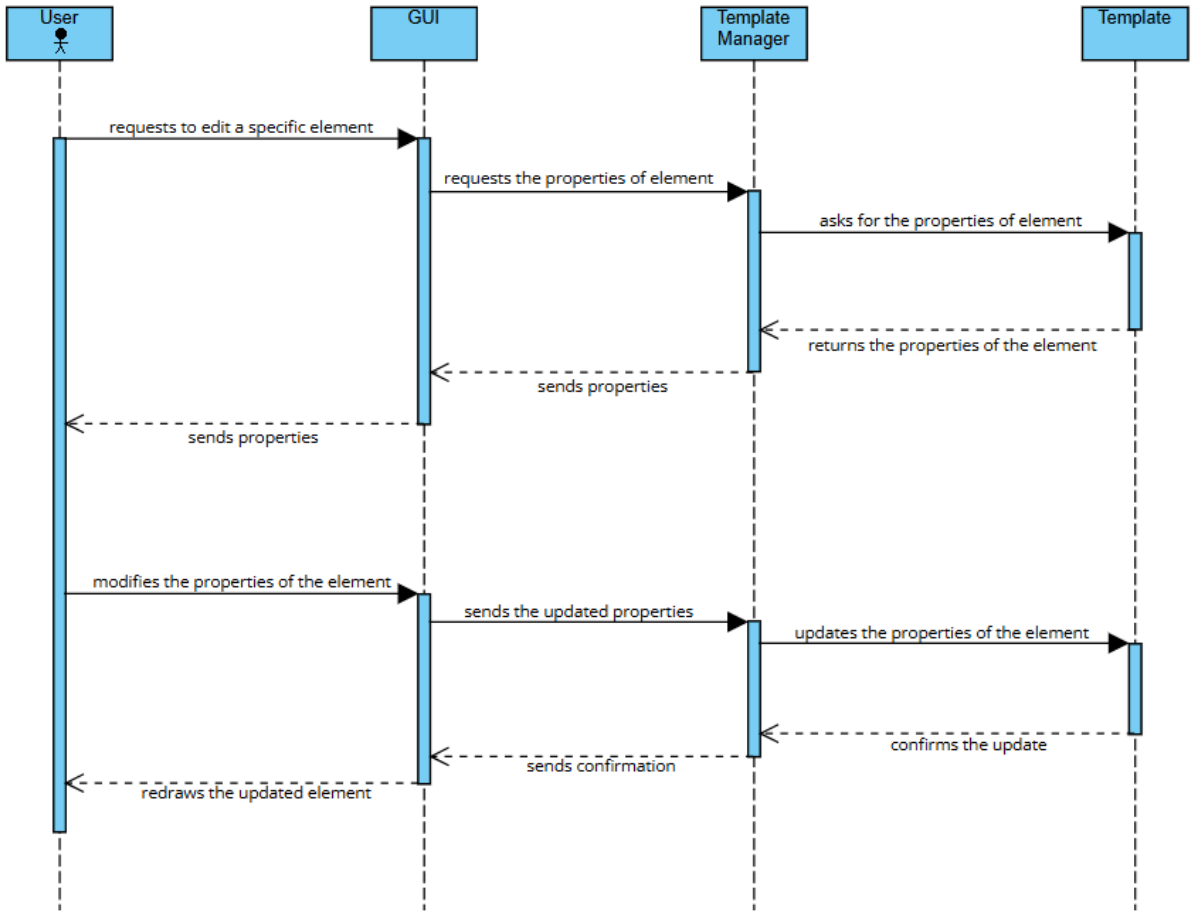
UML sequence diagram



Obr. 5.4: Sekvenčný diagram pre spoluprácu medzi systémovými komponentmi na zabezpečenie výberu a načítania šablóny.

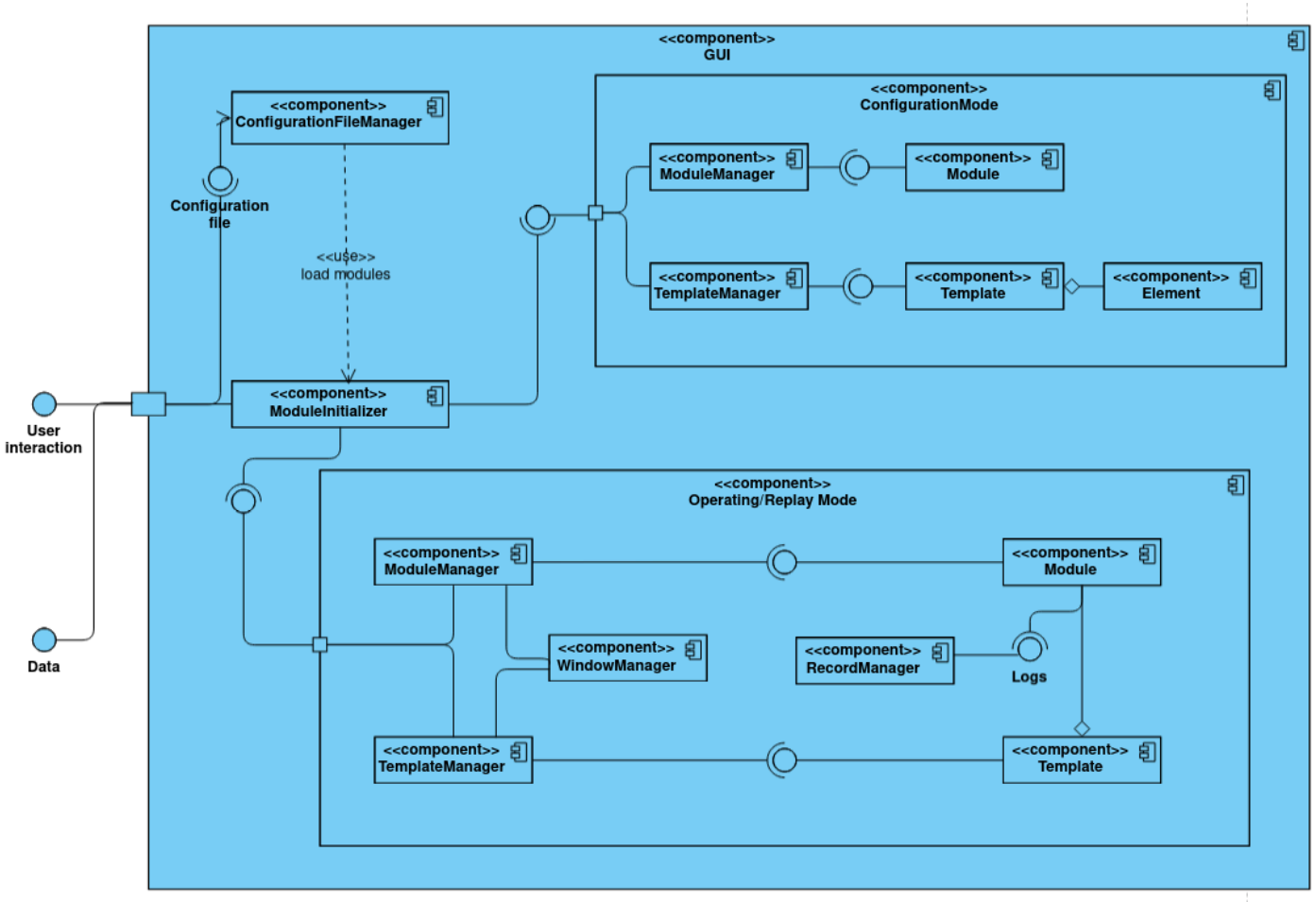


Obr. 5.5: Sekvenčný diagram pre proces zaznamenávania údajov v prevádzkovom režime.



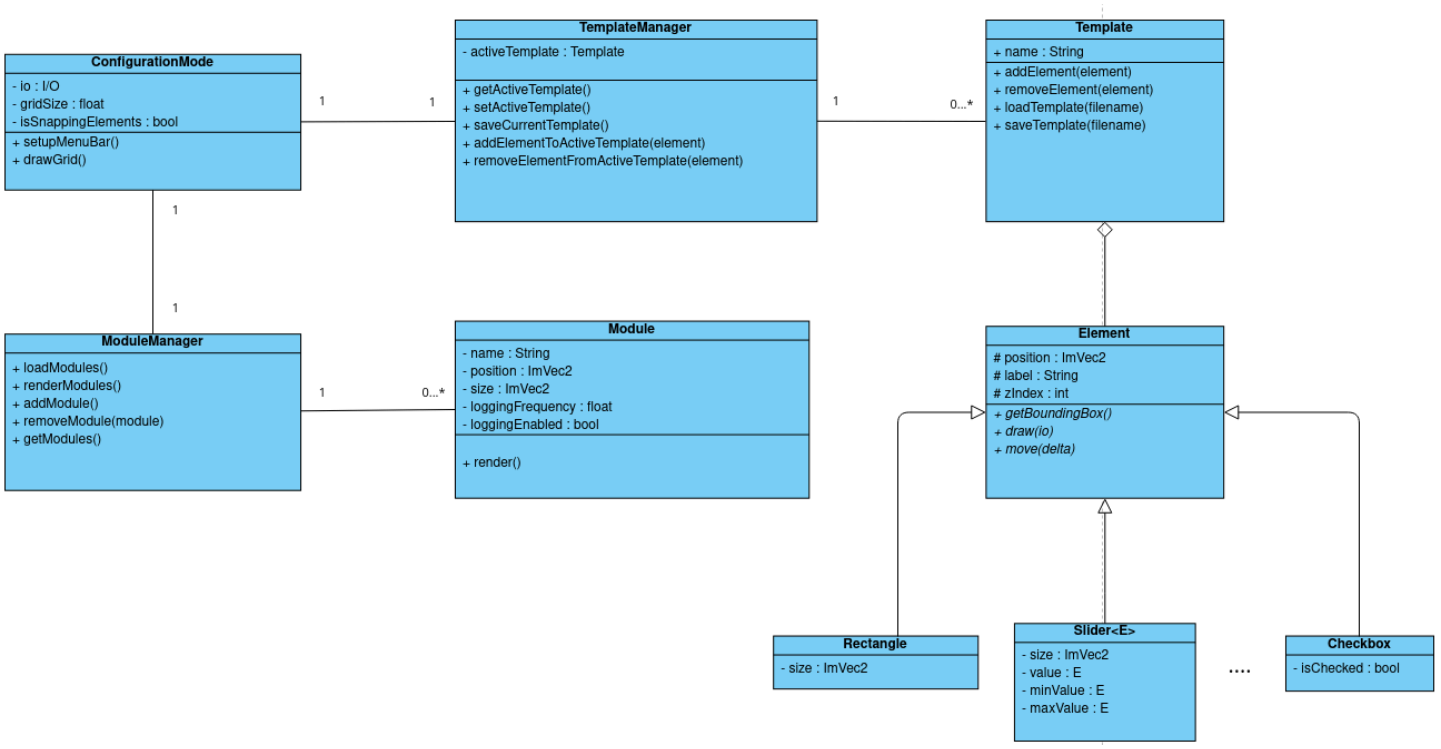
Obr. 5.6: Sekvenčný diagram pre úpravu prvku v konfiguračnom režime

UML component diagram



Obr. 5.7: UML component diagram

UML class diagram



Obr. 5.8: UML class diagram

Kapitola 6

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 7

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)*