

# Návrh

## 3D HRA

Projekt z predmetu Tvorba informačných systémov

Vypracovali: *Liliana Hodoňová, Tomáš Magula, Jakub Šmihul'a, Lukáš Podskalka*

<b>Návrh</b>	<b>1</b>
<b>3D HRA</b>	<b>1</b>
1. Podrobná špecifikácia vonkajších interfejsov	3
2. Návrh používateľského rozhrania	5
3. Plán implementácie	17
Prvá časť	17
Druhá časť	17
Tretia časť	18
4. Testovacie scenáre	19
Scenáre pre jedného hráča	19
Scenáre pre dvoch hráčov	19
Všeobecné herné pravidlá	19
5. UML diagramy	20

# 1. Podrobná špecifikácia vonkajších interfejsov

## Komunikácia s RealSense kamerou:

- Kamera slúži na snímanie priestoru pred stenou a na detekciu pohybu lopty.
- Kalibračný proces je povinný pred začatím detekcie hodov. Po kalibračnom procese kamera poskytuje súradnice miesta dotyku lopty so stenou, ktoré sú mapované na projekčnú plochu(pôvodná verzia aplikácie)

## Projekcia a interakcia s kalibrovanou plochou:

- Projekčný systém zobrazuje plochu na stene, kde kamera detekuje zásahy lopty.
- Aplikácia emuluje funkciu myši – pri zásahu lopty na stenu "klikne" na príslušné miesto na ploche v reálnom čase, čím simuluje pohyb a kliknutie myšou na kalibrovanom mieste(pôvodná verzia aplikácie)

## Rozhranie pre aktualizáciu rýchlosti a smeru:

- Aktualizácia aplikácie umožní pomocou kombinácie kláves odosielať údaje o rýchlosti a smere lopty pri zásahu.
- Tieto údaje budú získané na základe spracovania dát z RealSense kamery a následne odoslané do webového rozhrania. Rýchlosť a smer budú ovplyvňovať simulácie v hre (rýchlosť, trajektória).
- Na získavanie rýchlosti a smeru sa používa už existujúca funkcia v aplikácii na získanie všetkých bodov trajektórie pred dotykom lopty so stenou
- Ak existujú minimálne 2 body:
  - **Rýchlosť:** Vypočíta sa euklidovská vzdialenosť medzi posledným a pred posledným bodom trajektórie.
  - **Zmena X:** Získa sa rozdiel x-súradníc medzi posledným a prvým bodom trajektórie.
- Ak existujú minimálne 3 body:
  - **Predpis paraboly:** Bodmi sa vedie parabolická funkcia pomocou metódy riešenia lineárnych rovníc.
- Tieto informácie odošleme na web kde sa budú dať vypočítať ďalšie body letu lopty

## Komunikačný protokol:

- Aplikácia komunikuje s webovým rozhraním pomocou vytvorenia klávesnice
- Aplikácia vytvorí virtuálnu klávesnicu pomocou ktorej emuluje znaky ktoré webové rozhranie číta:

pr. #844,441,0.1243766,0.373329,-1.392079,0.764278,0.015763\*

- # - štartovací znak
- 1. argument - x súradnica (px)
- 2. argument - y súradnica (px)
- 3. argument - rýchlosť
- 4. argument - koeficient a (parabola)

- 5. argument - koeficient b (parabola)
- 6. argument - koeficient c (parabola)
- 7. argument - zmena súradnice x
- \* - koncový znak

**Komunikácia so systémom:**

- Aplikácia komunikuje so systémom tým že simuluje novú myš a vytvára jej klikanie.
- Informácie o rýchlosti a smere sú zasielané ako doplnkový vstup v kombinácii s polohou kliku, pričom systém interpretuje kombináciu kláves ako parameter pre tieto vlastnosti.

## 2. Návrh používateľského rozhrania



Obr. 2.1.1: Konfigurácia hry - jazyk. Po kliknutí na ľavú stranu sa jazyk hry nastaví na slovenský (poprípade anglický pri kliknutí na pravú polovicu). Požiadavka 3.2.1 v katalógu požiadaviek.



Obr. 2.1.2.1: Konfigurácia hry-počet hráčov slovenský jazyk. Po kliknutí na ľavú stranu sa počet hráčov nastaví na 1(poprípade 2 pri kliknutí na pravú polovicu). Požiadavka 3.2.2.1 v katalógu požiadaviek.



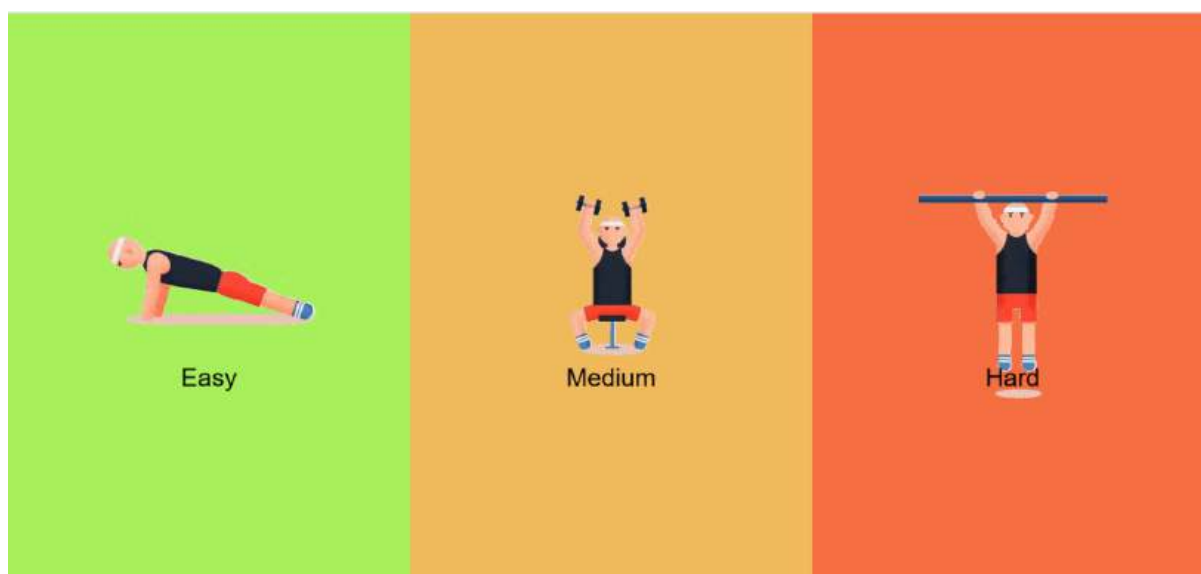
Obr: 2.1.2.2: Konfigurácia hry-počet hráčov anglický jazyk. Po kliknutí na ľavú stranu sa počet hráčov nastaví na 1(poprípade 2 pri kliknutí na pravú polovicu). Požiadavka 3.2.2.2 v katalógu požiadaviek.



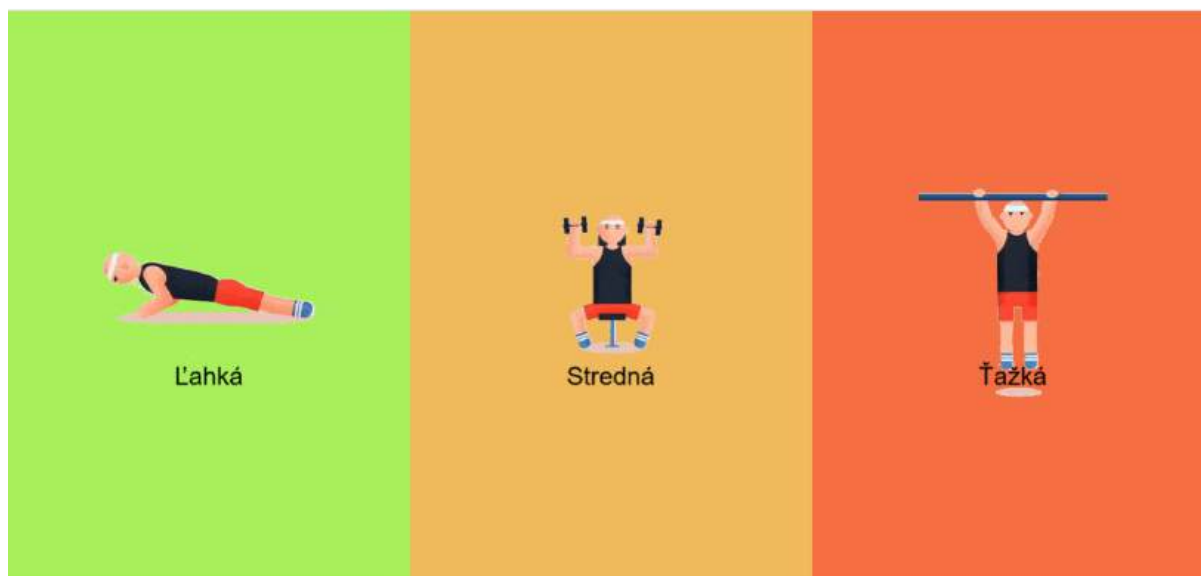
Obr: 2.1.4.1: Konfigurácia hry- náročnosť hry pre dvoch hráčov po anglicky. Po kliknutí na ľavú stranu sa nastaví hra s ľahkou náročnosťou (poprípade s ťažkou pri kliknutí na pravú polovicu). Požiadavka 3.2.4.2 v katalógu požiadaviek.



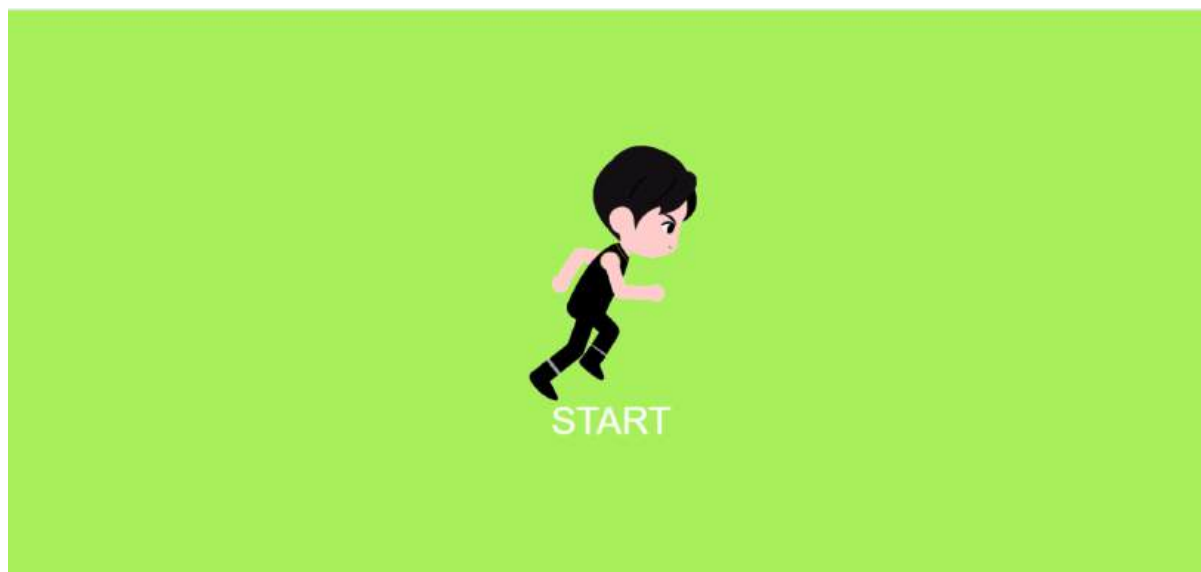
Obr: 2.1.4.2: Konfigurácia hry- náročnosť hry pre dvoch hráčov po slovensky. Po kliknutí na ľavú stranu sa nastaví hra s ľahkou náročnosťou (poprípade s ťažkou pri kliknutí na pravú polovicu). Požiadavka 3.2.4.2 v katalógu požiadaviek.



Obr: 2.1.4.3: Konfigurácia hry- náročnosť hry pre jedného hráča po anglicky. Po kliknutí na ľavú stranu sa nastaví hra s ľahkou náročnosťou (poprípade so strednou pri kliknutí na strednú časť obrazovky alebo s ťažkou pri kliknutí na pravú časť). Požiadavka 3.2.4.1 v katalógu požiadaviek.



Obr. 2.1.4.4: Konfigurácia hry- náročnosť hry pre jedného hráča po slovensky. Po kliknutí na ľavú stranu sa nastaví hra s ľahkou náročnosťou (poprípade so strednou pri kliknutí na strednú časť obrazovky alebo s ťažkou pri kliknutí na pravú časť). Požiadavka 3.2.4.1 v katalógu požiadaviek.



Obr. 2.1.4: Konfigurácia hry-štart. Po kliknutí kamkoľvek na obrazovku sa hra spustí. Požiadavka 3.2.4 v katalógu požiadaviek.





Obr. 2.2.1.1: Hra - slovenská verzia hry pre jedného hráča, jednoduchá náročnosť. Požiadavka 3.4 v katalógu požiadaviek.



Obr. 2.2.1.2: Hra - slovenská verzia hry pre jedného hráča, stredná náročnosť. Požiadavka 3.4 v katalógu požiadaviek.



Obr. 2.2.1.3: Hra - slovenská verzia hry pre jedného hráča, ťažká náročnosť. Požiadavka 3.4 v katalógu požiadaviek.



Obr. 2.2.2.1: Hra - slovenská verzia hry pre dvoch hráčov, jednoduchá náročnosť. Požiadavka 3.4 v katalógu požiadaviek.





Obr. 2.2.2.2: Hra - slovenská verzia hry pre dvoch hráčov, ťažká náročnosť. Požiadavka 3.4 v katalógu požiadaviek.



Obr. 2.2.3.1: Hra - anglická verzia hry pre jedného hráča, jednoduchá náročnosť. Požiadavka 3.4 v katalógu požiadaviek.



Obr. 2.2.3.2: Hra - anglická verzia hry pre jedného hráča, stredná náročnosť. Požiadavka 3.4 v katalógu požiadaviek.



Obr. 2.2.3.2: Hra - anglická verzia hry pre jedného hráča, ťažká náročnosť. Požiadavka 3.4 v katalógu požiadaviek.





Obr. 2.2.4.1: Hra - anglická verzia hry pre dvoch hráčov, ľahká náročnosť. Požiadavka 3.4 v katalógu požiadaviek.



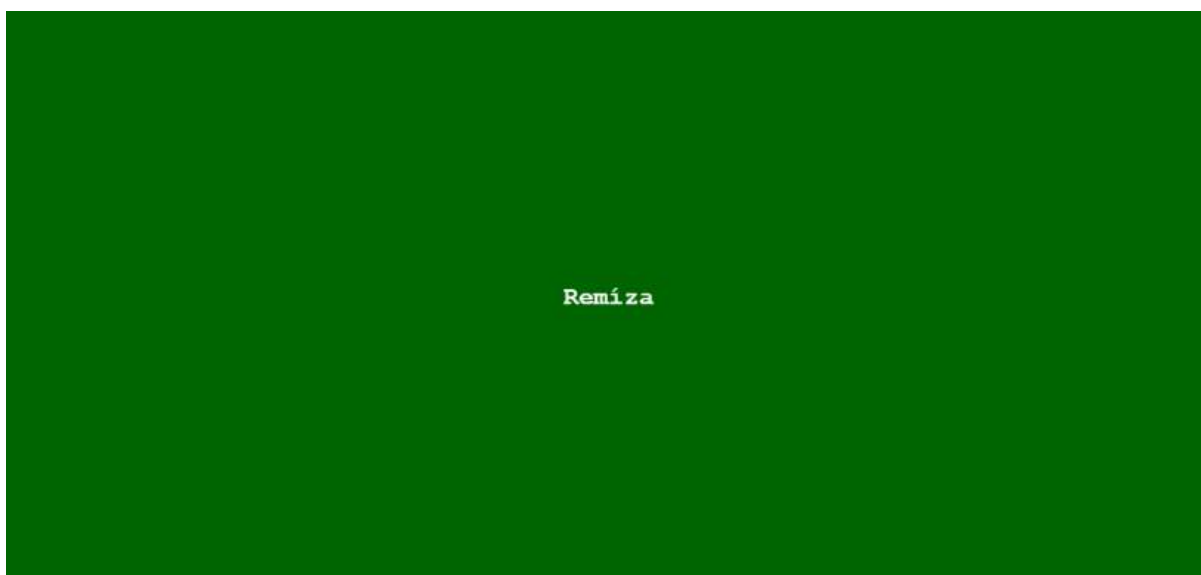
Obr. 2.2.4.2: Hra - anglická verzia hry pre dvoch hráčov, ťažká náročnosť. Požiadavka 3.4 v katalógu požiadaviek.



Obr. 2.3.1.1: Hra - verzia hry pre jedného hráča. Výpis nahraného skóre na obrazovke po uplynutí času. Požiadavka 3.4 v katalógu požiadaviek.



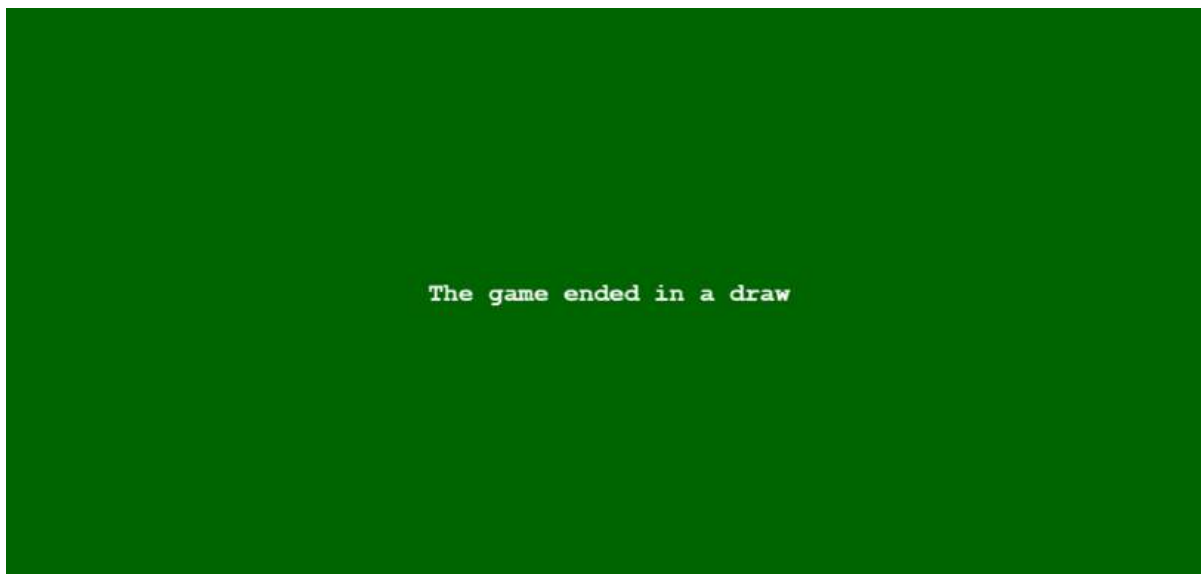
Obr. 2.3.1.2: Hra - slovenská verzia hry pre dvoch hráčov. Výpis víťaza a jeho nahraného skóre na obrazovke po uplynutí času. Požiadavka 3.4 v katalógu požiadaviek.



Obr. 2.3.1.3: Hra - slovenská verzia hry pre dvoch hráčov. Hra skončená remízou po uplynutí času. Požiadavka 3.4 v katalógu požiadaviek.



Obr. 2.3.2.1: Hra - anglická verzia hry pre dvoch hráčov. Výpis víťaza a jeho nahraného skóre na obrazovke po uplynutí času. Požiadavka 3.4 v katalógu požiadaviek.



Obr. 2.3.2.2: Hra - anglická verzia hry pre dvoch hráčov. Hra skončená remízou po uplynutí času. Požiadavka 3.4 v katalógu požiadaviek.



### 3. Plán implementácie

Implementácia sa rozdelí na tri na sebe nezávislé časti, kde sa bude pracovať na jednotlivých modeloch, ktoré sa nakoniec spoja.

#### Prvá časť

Táto časť implementácie sa zaoberá prácou s kamerou. Cieľom je dostať novú aplikáciu, ktorá bude vychádzať z pôvodnej aplikácie na snímanie odrazu loptičky od steny, avšak ju obohatiť o novú funkčnosť a to o snímanie pohybu loptičky (vektor) a rýchlosť loptičky. Táto časť je kľúčová, preto je rozdelená do jednotlivých fáz.

##### Fáza 1.

- v tejto časti je potrebné spojiť pôvodnú aplikáciu, nakonfigurovať kameru a spustiť aplikáciu
- **zodpovedný: Tomáš Magula**

##### Fáza 2.

- táto časť sa zaoberá naštudovaním kódu a potrebných materiálov ako sú napríklad rovnice a vzorce na rátanie rýchlosti a uhla
- **zodpovední: Tomáš Magula, Jakub Šmihul'a**

##### Fáza 3.

- prerobenie aplikácie s pridanou funkčnosťou
- **zodpovední: Tomáš Magula, Jakub Šmihul'a**

#### Druhá časť

Táto časť implementácie sa zaoberá funkčnosťou hry, teda berie do úvahy hru ako celok, nehľadiac na aplikáciu pre kameru.

##### Fáza 1.

- v tejto fáze sa implementuje prostredie hry
- **zodpovedná: Liliana Hodoňová**

##### Fáza 2.

- Pohyb lopty po kliknutí myšou do priestoru hry
- Lopta letí pod istým uhlom a rýchlosťou na správne miesto určené ľavým klikom myši
- **zodpovední: Jakub Šmihul'a**

##### Fáza 3.

- v tomto kroku sa implementujú základné objekty na triedenie.
- **zodpovední: Liliana Hodoňová, Jakub Šmihul'a, Lukáš Podskalka**

## Tretia časť

Časť sa zaoberá konfiguráciou hry, teda ako hru nakonfigurovať a vybrať si druh, zložitosť a podobne.

### Faza 1.

- Konfigurácia hry, ktorá sa dá vybrať aj pomocou hodenia lopty (v našom prípade na kliknutie myši)
- zodpovedný: **Lukáš Podskalka**

V poslednom kroku sa budú jednotlivé moduly spájať.

## 4. Testovacie scenáre

### Scenáre pre jedného hráča

1. **Klikanie** (slovenský/anglický jazyk, 1 alebo 2 hráči):
  - **Jazyk a konfigurácia:** Užívateľ vyberie jazyk (SK alebo EN), počet hráčov (1 alebo 2) a náročnosť hry.
  - **Herné prvky:** Zobrazí sa hra, čas začne plynúť a vykreslí sa odpad.
  - **Interakcia:** Hodená lopta sa po dotyku so stenou vykreslí vo virtuálnom priestore, má predom nastavené hodnoty rýchlosti aj smeru.
  - **Správanie lopty:** Lopta sa realisticky odráža od objektov.
  - **Bodovanie:** Trafenie nesprávneho terča odpočíta 5 bodov, zatiaľ čo trafenie správneho pridá 10 bodov, čím sa vykreslí nový odpad.
  - **Ukončenie:** Po vypršaní času sa zobrazí skóre, a kliknutím do priestoru sa hra reštartuje.
2. **S kamerou** (slovenský/anglický jazyk, 1 alebo 2 hráči):
  - **Jazyk a konfigurácia:** Užívateľ hádže loptu na oblasť s označením jazyka a výberu hráčov. Hodením lopty volí medzi náročnosťou hry.
  - **Interakcia:** Hodená lopta sa po dotyku so stenou vykreslí vo virtuálnom priestore a napodobňuje rýchlosť a smer reálnej lopty.
  - **Bodovanie a správanie lopty:** Bodovanie a správanie lopty zostáva rovnaké ako pri klikaní.
  - **Ukončenie:** Po hodení lopty do priestoru sa hra reštartuje.

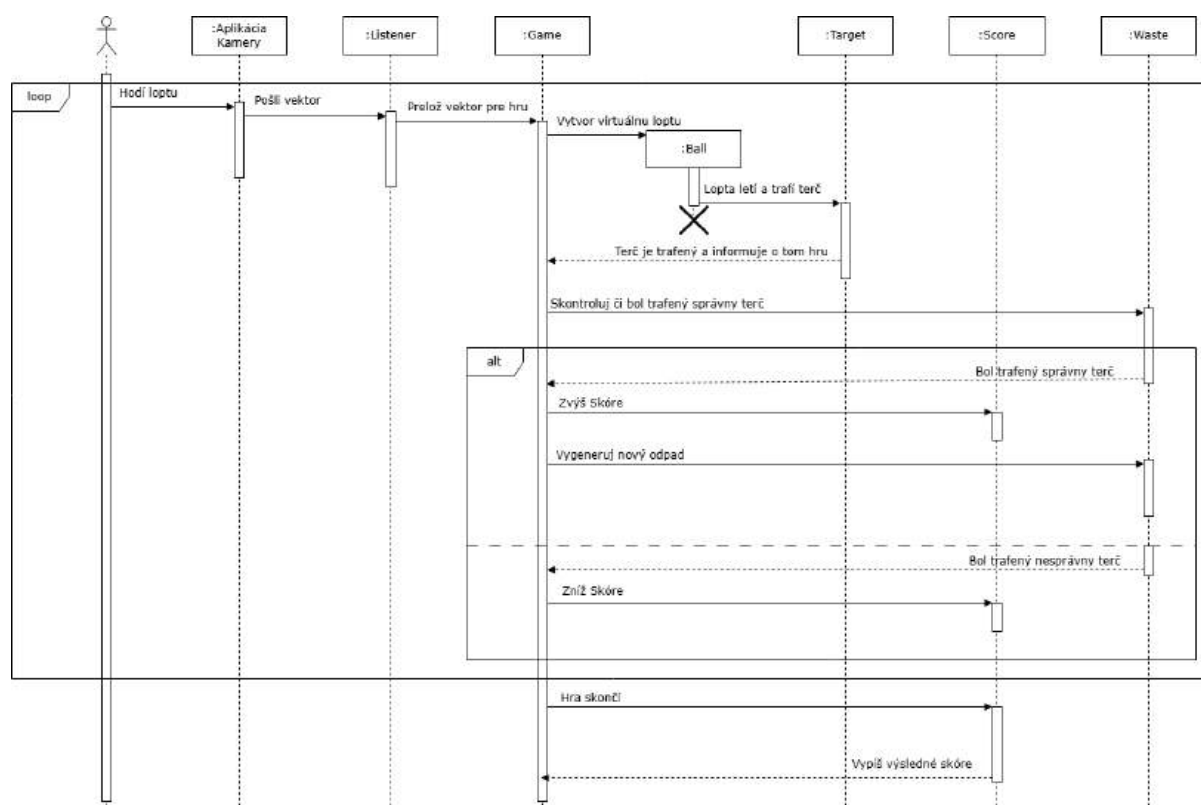
### Scenáre pre dvoch hráčov

3. **S kamerou** (slovenský/anglický jazyk, 1 alebo 2 hráči):
  - **Jazyk a konfigurácia:** Užívateľ hádže loptu pre výber jazyka (SK alebo EN), počtu hráčov (1 alebo 2) a náročnosť.
  - **Interakcia pre dvoch hráčov:** Odpad je na oboch stranách nezávislý. Pri súčasnem dotyku steny oboma loptami sa vo virtuálnom priestore vykreslia dve lopty.
  - **Bodovanie:** Trafenie nesprávneho terča odpočíta 5 bodov. Trafenie správneho pridá 10 bodov a vytvorí nový odpad pre konkrétneho hráča.
  - **Ukončenie:** Po vypršaní času sa zobrazí skóre a víťaz. Po hodení lopty do priestoru sa hra reštartuje.

### Všeobecné herné pravidlá

- **Vykreslenie odpadu:** Pri každej hre sa vykreslí odpad, ktorý hráč "upratuje" loptičkami do príslušných kontajnerov.
- **Správanie lopty:** Lopta sa realisticky odrážajú a ich rýchlosť a smer závisí od hádzania.
- **Bodovanie a ukončenie:** Trafenie správneho/nesprávneho terča má vplyv na skóre. Hra končí vypršaním času s možnosťou reštartu.

## 5. UML diagramy

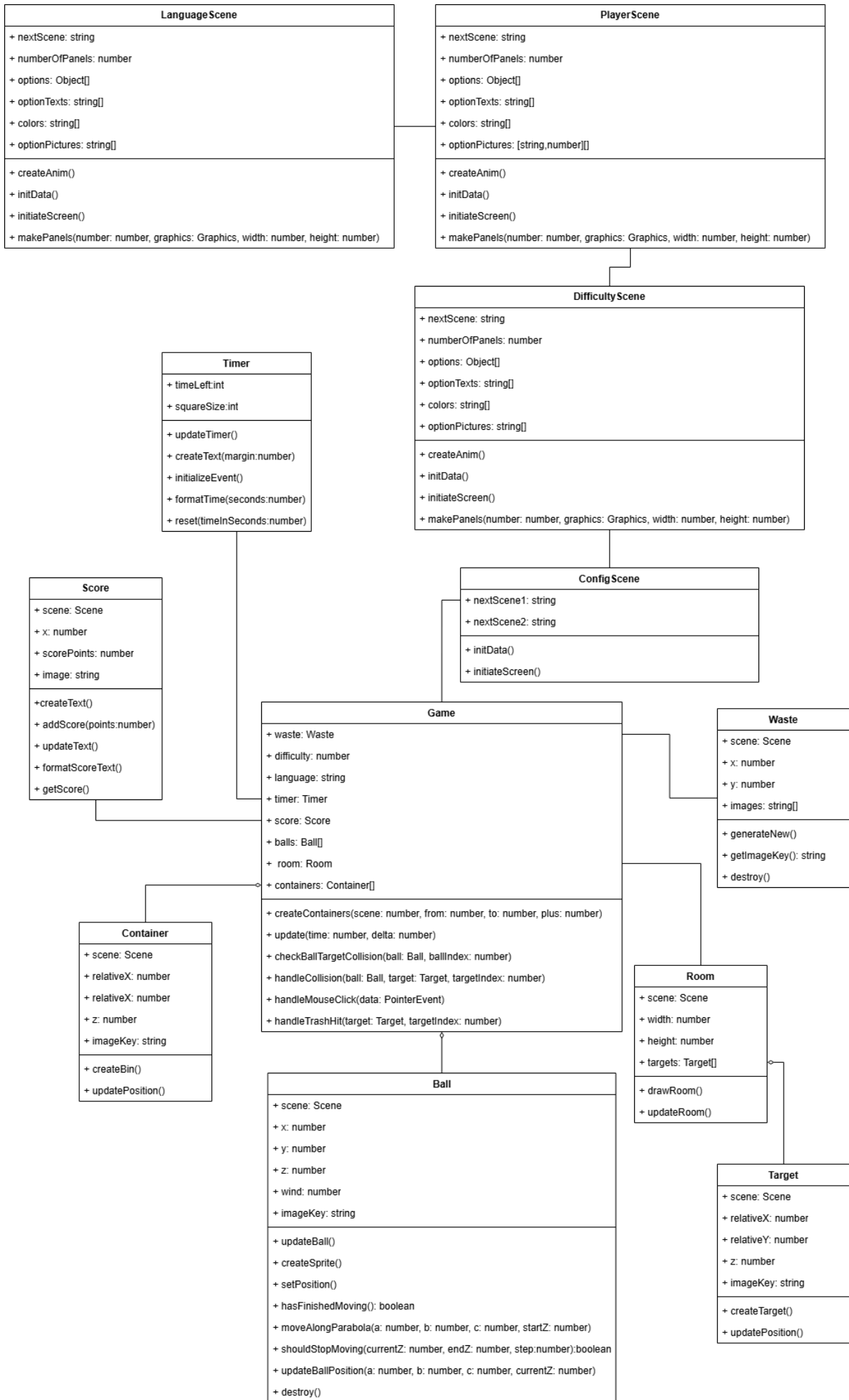


Obr. 5.1: Sekvenčný diagram

### Popis procesu:

1. **Začiatok interakcie** – Hráč hodí loptu (akcia používateľa).
2. **Spracovanie vstupu:**
  - Aplikácia kamery zaznamená pohyb a pošle **vektor hodu** do **Listenera**.
  - **Listener** preloží vektor na dáta zrozumiteľné pre hru a odošle ich do komponentu **Game**.
3. **Simulácia hodu:**
  - **Game** vytvorí virtuálnu loptu (**:Ball**).
  - Lopta letí a zasiahne **terč**.
  - **Terč (Target)** spracuje zásah a odošle informáciu späť do hry.
4. **Vyhodnotenie správnosti:**
  - **Game** kontroluje, či bol trafiť správny terč.
  - Ak áno, vyvolá sa sekvencia:
    - **Zvýši skóre (Score)**.
    - **Vygeneruje nový odpad (Waste)**.
  - Ak bol terč nesprávny:
    - **Zníži skóre (Score)**.
5. **Ukončenie hry:**
  - Po uplynutí času alebo splnení podmienky konca hra skončí.
  - Skóre sa zobrazí hráčovi.





## Hlavná trieda: Game

- **Game** je centrálny prvok, ktorý riadi vytváranie herných objektov.
  - Atribúty:
    - **difficulty** - obtiažnosť hry
    - **language** - jazyk (sk/en)
    - **waste, timer, score** - grafické objekty hry
    - **balls** - zoznam virtuálnych lôpt aktuálne na ploche
    - **room** - pozadie scény, obsahuje aj terče
  - Metódy:
    - **createContainers(scene, from, to, plus)** – vytvára kontajnery v hre.
    - **update(time, delta)** - aktualizuje pozíciu lôpt
    - **checkBallTargetCollision(ball, ballIndex)** - kontroluje kolízie lôpt s terčami

## Herné objekty a ich vlastnosti:

### 1. Target (Terče)

- Atribúty:
  - **relativeX, relativeY** – relatívne súradnice terča
  - **imageKey** - string ktorý cez ktorý Phaser referuje na obrázok terča
- Metódy:
  - **createTarget()** – vytvorí cieľ.
  - **updatePosition()** – aktualizuje polohu cieľa.

### 2. Room (Miestnosť)

- Atribúty:
  - **width, height** – rozmery miestnosti
  - **targets** – zoznam terčov
- Metóda:
  - **drawRoom()** – vykreslí izbu
  - **updateRoom()** – prekreslí izbu pri zmene veľkosti okna

### 3. Container (Kontajnery na odpad)

- Atribúty:
  - **relativeX, relativeY** – relatívne súradnice kontajnera.
  - **z** - vzdialenosť od zdroja lopty
  - **imageKey** – string ktorý cez ktorý Phaser referuje na obrázok kontajnera
- Metóda:
  - **createBin()** – vytvorí kontajner.
  - **updatePosition()** – upraví pozíciu kontajnera pri zmene veľkosti okna

### 4. Ball (Lopta)

- Atribúty:
  - **x, y, z** – súradnice lopty.
  - **wind** – sila vetra (upravuje ako letí lopta).
  - **imageKey** – string ktorý cez ktorý Phaser referuje na obrázok lopty
- Metódy:

- `updateBall()` - aktualizuje pozíciu a stav lopty
- `createSprite()` - vytvorí obrázok pre lopty
- `moveAlongParabola(a: number, b: number, c: number, startZ: number)` - zodpovedá za pozície kam po ktorých sa lopta posúva
- `shouldStopMoving(currentZ: number, endZ: number, step: number)` - zodpovedá za ukončenie pohybu lopty keď buď odletí z dohľadu alebo trafi terč

#### 5. Waste (Odpad)

- Atribúty:
  - `x, y` – súradnice odpadu.
- Metódy:
  - `generateNew()` – vytvorí nový odpad.
  - `destroy()` – zničí odpad.

#### 6. Score (Skóre)

- Atribúty:
  - `scorePoints` – počet bodov.
  - `x, y` – súradnice zobrazenia skóre.
  - `image` – string ktorý cez ktorý Phaser referuje na pozadie ukazovateľa skóre

#### 7. Timer (Časovač)

- Atribúty:
  - `timeLeft` – zostávajúci čas.
  - `squareSize` – veľkosť prvku.
- Metódy:
  - `updateTimer()` - aktualizuje časovač
  - `createText(margin)` - vypíše čas na obrazovku
  - `initializeEvent()` - zapne časovač
  - `formatTime(seconds)` - naformátuje čas do mm:ss

### Scény konfigurácie:

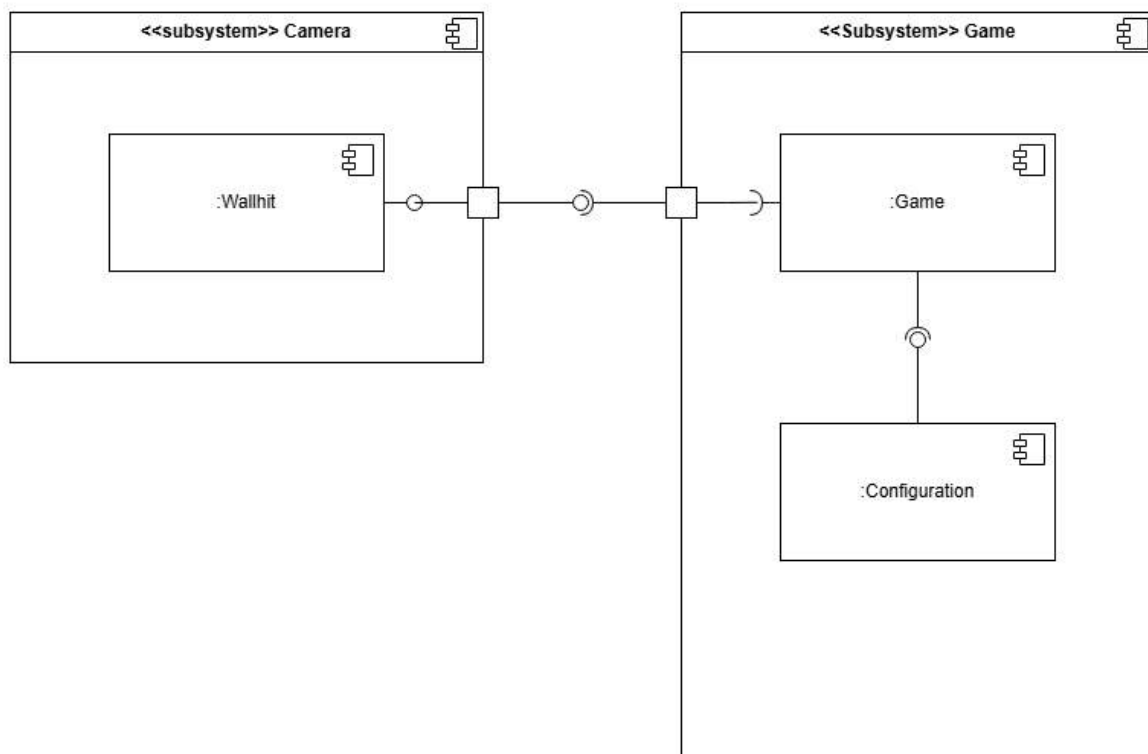
- **LanguageScene** – výber jazyka.
- **PlayerScene** – výber počtu hráčov.
- **DifficultyScene** – výber obtiažnosti.

Toto sú scény v ktorých si hráč dokáže nastaviť podrobnosti nasledujúcej hry. Majú z praktického hľadiska totožnú štruktúru.

- Atribúty:
  - `nextScene` - scéna ktorá nasleduje po aktuálnej scéne
  - `numberOfPanels` – počet vertikálnych panelov na ktoré je scéna rozdelená.
  - `colors` - zoznam farieb jednotlivých panelov v smere zľava
  - `options` - možnosti náležiacie k jednotlivým panelom. Možnosť zvolená kliknutím na panel sa odošle do ďalšej scény v slovníku data spoločne so zvolenými možnosťami z predchádzajúcich scén.
  - `optionTexts` - zoznam textov napísaných na paneloch



- `optionPictures` - zoznam kľúčov k animáciám zobrazených na paneloch
- Metódy:
  - `initData()` - slúži do veľkej miery ako náhrada konštruktora (kvôli anatomickej Phaseru to bolo potrebné), tu sú zadané atribúty.
  - `createAnim()` - vytvorí animáciu zo spriteSheetov načítaných na základe `optionPictures`
  - `initiateScreen()` - vykreslí panely, obrázky, texty a pripojí funkcionality kliknutí
  - `makePanels(number, graphics, width, height)` - pomocná funkcia čo vytvára panely



Obr. 5.3 Component diagram

## Hlavné subsystémy:

1. **Camera Subsystem**
  - Obsahuje komponent **Wallhit**, ktorý pravdepodobne detekuje kolízie s múrom.
2. **Game Subsystem**
  - Obsahuje komponenty:
    - **Game** – hlavný komponent hry.
    - **Configuration** – konfiguračný modul hry.

---

## Spojenia medzi subsystémami:

- **Prepojenie medzi kamerou a hrou:**
  - **Wallhit** z kamery komunikuje s **Game** komponentom.
  - Pravdepodobne signalizuje, keď objekt (napr. lopta) narazí na múr.
- **Prepojenie medzi komponentmi v hre:**
  - **Game** je prepojený s **Configuration**, čo naznačuje, že konfigurácia ovplyvňuje hernú logiku.