



Převod pohybu objektů do prostředí hry pomocí UHF RFID

Diplomová práce

Studijní program: N2612 – Elektrotechnika a informatika
Studijní obor: 1802T007 – Informační technologie

Autor práce: Martin Votýpka
Vedoucí práce: Ing. Jana Kolaja Ehlerová Ph.D.
Ústav nových technologií a aplikované informatiky





Zadání diplomové práce

Převod pohybu objektů do prostředí hry pomocí UHF RFID

Jméno a příjmení: **Bc. Martin Votýpka**
Osobní číslo: M20000178
Studijní program: N2612 Elektrotechnika a informatika
Studijní obor: Informační technologie
Zadávací katedra: Ústav nových technologií a aplikované informatiky
Akademický rok: **2021/2022**

Zásady pro vypracování:

1. Provedte rešerši UHF RFID systémů pro sledování pohybu objektů.
2. Seznamte se s UHF RFID čtečkou a implementujte systém pro čtení štítků se softwarovou filtrací.
3. Otestujte systém pro převod pohybu válcového předmětu do prostředí aplikace.
4. Ověřte výsledky modelu vůči fyzikálním výpočtům.
5. Pokuste se o převod pohybu kulového předmětu do prostředí aplikace.
6. Využijte zjištění pohybu objektů pro hru pro mobilní zařízení.

Rozsah grafických prací: dle potřeby dokumentace
Rozsah pracovní zprávy: 40-50 stran
Forma zpracování práce: tištěná/elektronická
Jazyk práce: Čeština



Seznam odborné literatury:

- [1] CURTY, Jari-Pascal, Michel DECLERCQ, Catherine DEHOLLAIN a Norbert JOEHL. Design and optimization of passive UHF RFID systems. New York: Springer, [2010]. ISBN 978-1-4419-4199-2.
- [2] DOBKIN, Daniel Mark. The RF in RFID: UHF RFID in practice. Second edition. Amsterdam: Newnes, Newnes is an imprint of Elsevier, 2013. ISBN 978-0-12-394583-9.

Vedoucí práce: Ing. Jana Kolaja Ehlerová, Ph.D.
Ústav nových technologií a aplikované informatiky

Datum zadání práce: 12. října 2021
Předpokládaný termín odevzdání: 16. května 2022

prof. Ing. Zdeněk Plíva, Ph.D.
děkan

L.S.

Ing. Josef Novák, Ph.D.
vedoucí ústavu

Prohlášení

Prohlašuji, že svou bakalářskou práci jsem vypracoval samostatně jako původní dílo s použitím uvedené literatury a na základě konzultací s vedoucím mé bakalářské práce a konzultantem.

Jsem si vědom toho, že na mou bakalářskou práci se plně vztahuje zákon č. 121/2000 Sb., o právu autorském, zejména § 60 – školní dílo.

Beru na vědomí, že Technická univerzita v Liberci nezasahuje do mých autorských práv užitím mé bakalářské práce pro vnitřní potřebu Technické univerzity v Liberci.

Užiji-li bakalářskou práci nebo poskytnu-li licenci k jejímu využití, jsem si vědom povinnosti informovat o této skutečnosti Technickou univerzitu v Liberci; v tomto případě má Technická univerzita v Liberci právo ode mne požadovat úhradu nákladů, které vynaložila na vytvoření díla, až do jejich skutečné výše.

Současně čestně prohlašuji, že text elektronické podoby práce vložený do IS/STAG se shoduje s textem tištěné podoby práce.

Beru na vědomí, že má bakalářská práce bude zveřejněna Technickou univerzitou v Liberci v souladu s § 47b zákona č. 111/1998 Sb., o vysokých školách a o změně a doplnění dalších zákonů (zákon o vysokých školách), ve znění pozdějších předpisů.

Jsem si vědom následků, které podle zákona o vysokých školách mohou vyplývat z porušení tohoto prohlášení.

12. května 2022

Bc. Martin Votýpka

Abstrakt

RFID je technologie pro bezdrátovou identifikace objektů, která se v současné době těší čím dál větší popularitě. Je používána napříč velkou škálou odvětví a v posledních letech se můžeme setkat se snahou o použití této technologie ke snímání pohybu.

Tato práce se věnuje návrhu, tvorbě a testování systému pro snímání pohybu založeném na technologii RFID a současně vývoji her, které budou pomocí vzniklých systémů ovládány.

V teoretické části této práce je uveden základní popis technologie RFID, možné rozdělení této technologie a zároveň jsou zde uvedeny příklady použití RFID ke snímání pohybu. Dále se práce věnuje popisu použitých nástrojů a prací s nimi.

V části praktické je popsán princip vytvářených systému a současně je uveden postup při jejich návrhu, tvorbě a testování. Následně je uveden proces vývoje jednotlivých her, které budou pomocí vzniklých systémů ovládány.

V závěru jsou uvedeny výsledky, jichž bylo během této práce dosaženo současně s možným zdokonalením vytvořených systémů.

Klíčová slova: RFID, snímání pohybu, RFID hry

Abstract

RFID is a wireless object identification technology that is currently gaining in popularity. It is used across a wide range of industries and in recent years we may see efforts to use this technology for motion detection.

This work deals with the design, creation and testing of a system for motion detection based on RFID technology and at the same time the development of games that will be controlled by the created systems.

The theoretical part of this work provides a basic description of RFID technology, the possible division of this technology and also provides examples of the use of RFID for motion detection. Furthermore, the work describes the tools used and how to work with them.

The practical part describes the principle of the created systems and at the same time the procedure for their design, creation and testing is given. Subsequently, the process of development of individual games, which will be controlled by the resulting systems, is presented.

In the conclusion, the results achieved during this work are presented, together with a possible improvement of the created systems.

Key words: RFID, motion detection, RFID games

Obsah

Úvod.....	12
1 Technologie RFID.....	13
1.1 Co je to RFID?.....	13
1.2 Nízká frekvence – LF.....	13
1.3 Vysoká frekvence – HF.....	13
1.4 Ultravysoká frekvence – UHF.....	14
1.5 RFID štítky.....	15
1.5.1. Pasivní štítky.....	15
1.5.2. Polopasivní štítky.....	15
1.5.3. Aktivní štítky.....	15
1.6 Antény.....	16
1.7 Čtečky.....	16
1.8 Použití RFID ke snímání pohybu.....	16
1.9 Princip vyhodnocování polohy v této práci.....	17
2 Použité nástroje a práce s nimi.....	19
2.1 Čtečka.....	19
2.2 Antény.....	19
2.3 Štítky.....	20
2.4 Vývojové prostředí.....	20
2.5 Připojení RFID čtečky k mobilnímu zařízení.....	20
2.6 Získávání hodnot ze čtečky.....	22
3 Poloha předmětu v ploše.....	24
3.1 Princip určování polohy.....	24
3.2 Filtrace hodnot získaných ze čtečky.....	25
3.3 Vytvoření konstrukce pro určování polohy předmětu.....	26
3.4 Vyhodnocení polohy předmětu.....	29
4 Rotační pohyb válcového předmětu.....	31

4.1 Princip určování orientace předmětu.....	31
4.2 Vytvoření konstrukce snímaného předmětu.....	33
4.3 Algoritmus vyhodnocení rotačního pohybu.....	35
4.4 Testování vytvořených konstrukcí.....	38
5 Rotační pohyb sférického předmětu.....	39
5.1 Princip určování orientace sférického předmětu.....	39
5.2 Vytvoření konstrukce rotujícího předmětu.....	40
5.3 Algoritmus vyhodnocení rotačního pohybu.....	42
5.4 Testování vytvořených konstrukcí.....	46
6 Tvorba her využívajících snímání pohybu pomocí RFID.....	48
6.1 Breakout.....	48
6.1.1. Vytvoření a vykreslení herní plochy.....	49
6.1.2. Vyhodnocování kolizí.....	50
6.1.3. Reakce na vyhodnocené kolize.....	52
6.1.4. Provázání hry se systémem vyhodnocení polohy.....	53
6.2 Spoj 4.....	54
6.2.1. Vytvoření a vykreslení herní plochy.....	54
6.2.2. Vyhodnocení optimální pozice pro položení žetonu.....	55
6.2.3. Provázání hry se systémem vyhodnocení polohy.....	59
6.3 Curling.....	60
6.3.1. Vytvoření a vykreslení herní plochy.....	60
6.3.2. Určování rychlosti a směru pohybu kamenu.....	61
6.3.3. Provázání hry se systémem vyhodnocení pohybu.....	61
7 Závěr.....	62
Použitá literatura.....	63
Obsah příloženého CD.....	65

Seznam obrázků

Obrázek 1: Vývojový přípravek čtečky Hadron R4320C.....	19
Obrázek 2: Vyhodnocování polohy předmětu.....	24
Obrázek 3: Schéma použité sítě štítků.....	26
Obrázek 4: Nákres posuvného dílu.....	27
Obrázek 5: Modely dílů krytu.....	27
Obrázek 6: Modely dílů konstrukce.....	27
Obrázek 7: Konstrukce ovládacího prvku.....	28
Obrázek 8: Nákres tlačítka.....	28
Obrázek 9: Modely podložky a krytu.....	28
Obrázek 10: Sestavené tlačítko.....	29
Obrázek 11: Princip určení orientace předmětu.....	31
Obrázek 12: První navržená konstrukce.....	33
Obrázek 13: Modely dílů první konstrukce.....	33
Obrázek 14: Částečně složená první konstrukce.....	34
Obrázek 15: Druhá navržená konstrukce.....	34
Obrázek 16: Modely dílů druhé konstrukce.....	35
Obrázek 17: Vnitřní část druhé konstrukce.....	35
Obrázek 18: Princip určení orientace předmětu.....	39
Obrázek 19: Návrh první konstrukce.....	40
Obrázek 20: Modely tištěných dílů první konstrukce.....	41
Obrázek 21: První vytvořená konstrukce.....	41
Obrázek 22: Návrh druhé konstrukce.....	41
Obrázek 23: Modely dílů druhé konstrukce.....	42
Obrázek 24: Vnitřní část druhé konstrukce.....	42
Obrázek 25: Původní hra Breakout.....	48
Obrázek 26: Vzhled hry Breakout.....	50
Obrázek 27: Ohraničení n-úhelníku kružnicí.....	50

Obrázek 28: Desková verze hry Spoj 4.....	54
Obrázek 29: Vzhled hry Spoj 4.....	55
Obrázek 30: Princip algoritmu minimax a alfa-beta ořezání.....	57
Obrázek 31: Zakódování herní pozice.....	58
Obrázek 32: Vzhled hry Curling.....	60

Seznam použitých zkratek

HF	High Frequency
Hz	hertz
kHz	kilohertz (1 kHz = 1 000 Hz)
LF	Low Frequency
MHz	megahertz (1MHz = 1 000 000 Hz)
ms	milisekunda (1 s = 1000 ms)
RFID	Radio Frequency Identification
RSSI	Received Signal Strength Indication
UART	Universal Asynchronous Receiver-Transmitter
UHF	Ultra High Frequency
USB	Universal Serial Bus

Úvod

RFID (Radio Frequency Identification) technologie je technologie nejčastěji používaná pro bezdrátovou identifikaci objektů opatřených speciálními štítky. V současné době se těší čím dál větší popularitě a do kontaktu s ní přicházíme každý den. Důvodem její oblíbenosti je stále lepší dostupnost a současně velké výhody plynoucí z jejího použití. Hlavní výhodou použití této technologie při identifikaci objektů je možnost jejich identifikace na velké vzdálenosti a odstranění nutnosti přímé viditelnosti identifikovaného objektu. V posledních letech se můžeme setkat také se snahou o použití této technologie pro snímání pohybu, což s sebou nese značné výhody oproti konvenčně používaným systémům, ale zároveň i nevýhody.

Cílem této práce bude vytvořit systém pro vyhodnocování polohy a pohybu předmětů založený na technologii RFID. Tento systém bude následně použit k ovládání během této práce vytvořených her pro zařízení s operačním systémem Android.

V teoretické části dojde nejprve k základnímu seznámení s technologií RFID. Bude popsáno, co tato technologie představuje a zároveň uvedeno její možné rozdělení. Následně dojde k popisu jednotlivých částí, ze kterých se RFID systém sestává a budou uvedeny důvody vedoucí ke snaze o použití této technologie ke snímání pohybu, ale současně i úskalí, která z použití RFID k tomuto účelu plynou. Nakonec budou popsány jednotlivé nástroje, použité v rámci této práce a práce s nimi.

Praktická část této práce bude věnována vytvoření systémů založených na technologii RFID pro snímání polohy a pohybu předmětů. Nejprve dojde k popisu principu, na kterém budou jednotlivé systémy pracovat, postupu při vytváření systémů a nakonec jejich testování. Dále se bude práce v praktické části věnovat tvorbě her, pro jejichž ovládání budou jednotlivé vytvořené systémy použity.

V závěru budou uvedeny výsledky jichž bylo v rámci této práce dosaženo.

1 Technologie RFID

1.1 Co je to RFID?

RFID je zkratka anglického názvu Radio Frequency Identification neboli identifikace pomocí rádiových vln. S RFID technologií přicházíme do styku prakticky denně, byť si to většina z nás ani neuvědomuje. Tato technologie je rozšířena napříč velkou škálou různých odvětví a nejčastěji je využívána k bezdrátové identifikaci výrobků a osob. Jedno ze současných využití RFID technologie je například zabezpečení výrobků v obchodních domech proti krádeži nebo například v bezkontaktních platebních kartách. Jak již z názvu vyplývá technologie RFID je založena na principu komunikace pomocí rádiových vln. V základu se skládá ze tří částí a těmito částmi jsou čtečky, antény a štítky. Čtečky pomocí antén vysílají signál ke štítku. Štítek poté vysílá zpět informace, nejčastěji své identifikační číslo, které čtečka pomocí antén přijímá a je možné je následně zpracovat. Rozdělit lze tuto technologii do několika skupin v závislosti na použité frekvenci a také na technologii štítků. Informace ohledně RFID technologie použité v této kapitole byly čerpány z [2] a [3].

1.2 Nízká frekvence – LF

Jedná se o RFID systémy pracující na frekvencích 125 – 134 kHz. RFID systémy pracující na nízké frekvenci jsou často používány pro čipování zvířat, neboť jsou v porovnání s vysokou a hlavně ultravysokou frekvencí poměrně málo ovlivněny přítomností vody a zvířecích tkání. Jejich nevýhodou je v porovnání s vysokou a ultravysokou frekvencí nižší komunikační rychlost a hlavně nižší vzdálenost, na kterou je čtečka schopná štítek číst a také absence antikolizních algoritmů, které by umožňovaly číst větší množství štítků zároveň. Tato vzdálenost se pohybuje v řádech maximálně několika desítek centimetrů. Při označování zvířat mohou být štítky připevněny k uším zvířat (například při označování dobytka), nebo mohou být ve skleněné kapsli zavedeny pod kůži označovaného zvířete (například při označování psů). Velkou výhodou systému RFID pracujících na nízké frekvenci je relativně nízká cena s porovnání se systémy pracujícími na frekvencích vyšších. Své uplatnění, kromě čipování zvířat, našly tyto systémy například v identifikačních kartách zajišťujících vstup do hlídaných prostor, kde nízká čtecí vzdálenost, nízká rychlost a nemožnost čtení několika štítků zároveň nečiní velký problém.

1.3 Vysoká frekvence – HF

Jedná se o RFID systémy pracující na frekvenci 13,56 MHz. V porovnání s nízkou frekvencí se HF RFID systémy vyznačují vyšší rychlostí a také delšími vzdálenostmi, na které je možné štítky číst. Ty se pohybují v řádech desítek centimetrů až kolem jednoho metru. Tyto systémy se používají například při inventarizaci. Při použití HF RFID pro inventarizaci ve firmách je nutné řešit relativně krátké vzdálenosti, na které lze štítek spolehlivě přečíst. Tento problém je často řešen použitím velkých antén a štítků, umístěním antény co nejbližší místu, kde se budou předměty opatřené štítkem pohybovat, nebo využitím přenosných čteček pomocí kterých mohou štítky načíst přímo zaměstnanci. Další využití RFID systému pracujících na vysoké

frekvenci je například v kreditních kartách. Při snížení čtecí vzdálenosti je čtečka schopna dodávat štítku dostatečnou energii potřebnou pro kryptografické operace zajišťující bezpečnou komunikaci mezi čtečkou a štítkem.

1.4 Ultravysoká frekvence – UHF

Jedná se o RFID systémy pracující na frekvenci 860 – 960 MHz a 2,4GHz. Pro UHF RFID systémy je charakteristická vysoká rychlost komunikace a velká vzdálenost, na kterou je možné pomocí antén získat informace ze štítků. Tato vzdálenost se v závislosti na použité technologii štítků pohybuje od jednotek metrů (u pasivních štítků) až k jednomu kilometru (u aktivních štítků). Další velkou výhodou je možnost detekce většího množství štítků najednou. Díky relativní jednoduchosti pasivních UHF štítků jsou tyto štítky poměrně levné (ty nejlevnější se prodávají v přepočtu kolem jedné koruny za kus), což značně dopomáhá rychlému rozšíření UHF RFID systémů. V současné době jsou UHF RFID systémy používány ve spoustě různých aplikací od inventarizace, sledování lodních kontejnerů či vlakových vagónů po automatické zaznamenávání časů závodníků při běžeckých a dalších závodech. Velkou nevýhodou UHF systémů je jejich velká náchylnost na rušení způsobené vodivými předměty v okolí čtečky a štítků. Například v případě vody, zatím co je elektromagnetická vlna na nízké frekvenci schopna projít do hloubky několika metrů, je na ultravysoké frekvenci schopna projít pouze do hloubky pár centimetrů. Hloubka, do které je vlna schopna projít skrz materiál, závisí jednak na frekvenci a také na materiálu. Hodnoty hloubky penetrace vlny skrz materiál je pro různé frekvence a materiály možné najít v tabulce 1.

Tabulka 1: Hloubka průchodu vln o daných frekvencích skrz vybrané materiály [2]

Materiál	Hloubka průchodu vlny při frekvenci:			
	125 kHz	13,56 MHz	900 MHz	2,4 GHz
Voda	8 m	2 m	4 cm	8mm
Zvířecí tkáň	2 m	60 cm	2 cm	8 mm
Hliník	0,23 mm	71 μ m	2,7 μ m	1,6 μ m
Měď	0,18 mm	55 μ m	2,1 μ m	1,3 μ m

1.5 RFID štítky

RFID štítky jsou zařízení, která odesílají informace, nejčastěji své identifikační číslo, do čtečky, která tyto informace dále zpracovává. RFID štítky je možné rozdělit do tří skupin na pasivní, polopasivní a aktivní.

1.5.1. Pasivní štítky

Jedná se o štítky, které nedisponují žádným vlastním zdrojem energie a veškerou potřebnou energii pro komunikaci a chod obvodů štítku získávají z elektromagnetického pole generovaného čtečkou. Tento typ štítků funguje na principu zpětně odraženého signálu, který je vysílán z antén. Signál, který se od štítků odráží, je pomocí obvodů štítků změněn a pozměněný odražený signál je následně přijímán pomocí antén čtečkou, která signál zpracuje. Velkou výhodou pasivních štítků je jejich relativní jednoduchost a z toho vyplývající nízká cena. Další velkou výhodou pasivních štítků je jejich dlouhá životnost plynoucí z absence baterie, která by se postupně vybíjela. Nevýhodou tohoto typu štítků je v porovnání s polopasivními a aktivními štítky nízká vzdálenost, na kterou je čtečka schopná získat informace ze štítku. Ta se u UHF pohybuje v řádech jednotek metrů.

1.5.2. Polopasivní štítky

Polopasivní štítky na rozdíl od pasivních již obsahují zdroj energie, který napájí obvody štítků a případně čidla ke štítku připojená. Princip komunikace mezi čtečkou a štítkem zůstává stejný jako u pasivních štítků. Čtečka vysílá signál, který se od štítku odráží a obvody štítků signál pozmění, čímž předají potřebné informace. Výhodou polopasivních štítků je větší vzdálenost, na kterou je možné se štítky komunikovat a také vyšší spolehlivost. Vzdálenost, na kterou je čtečka schopná získat informace ze štítků, se pohybuje v řádech desítek metrů. Nevýhodou těchto štítků je vyšší cena, větší velikost a požadavky na údržbu. Své uplatnění tento typ štítků našel především při sledování cenných výrobků, jako jsou například díly letadel.

1.5.3. Aktivní štítky

Aktivní štítky obsahují zdroj energie, který zajišťuje veškerou potřebnou energii pro chod štítku a také pro komunikaci mezi štítkem a čtečkou. Aktivní štítky jsou již plnohodnotnými rádiovými vysílači. Díky vnitřními oscilátoru jsou schopné komunikovat v rozmezí určitého frekvenčního pásma a mohou tím pádem v přítomnosti jiných štítků měnit frekvenční kanál. Vzdálenost, na kterou jsou aktivní štítky schopné komunikovat, se pohybuje v rozmezí od stovek metrů do jednotek kilometrů. Nevýhodami aktivních štítků je jejich velikost, požadavky na údržbu a složitost a z ní vyplývající vysoká cena. Aktivní štítky jsou používány například při označování lodních kontejnerů.

1.6 Antény

Antény jsou nezbytné prvky RFID systémů, pomocí kterých je čtečka schopná odesílat signál do štítků a přijímat informace odesílané štítky. RFID antény lze rozdělit do dvou základních skupin v závislosti na jejich polarizaci. Polarizace antény udává, v jakém směru se šíří rádiové vlny generované anténou. Polarizace antény může mít velký dopad na vzdálenost, na kterou je čtečka schopná přečíst informace ze štítků. Aby byl dosah co největší, musí se polarizace antény shodovat s polarizací štítku. První skupinou jsou lineárně polarizované antény, které generují vlny šířící se pouze podél vertikální nebo horizontální roviny. Lineárně polarizované antény se používají v případě, kdy je možné zajistit pevnou orientaci štítku. Druhou skupinou jsou kruhově polarizované antény, kde rovina, ve které jsou elektromagnetické vlny generovány, postupně rotuje a je tak možné číst štítky v jakékoliv orientaci. Kruhově polarizované antény jsou používány v případě, kdy orientace štítků není známá. Cenou za možnost číst štítky v jakékoliv orientaci je nižší zisk kruhově polarizované antény oproti lineárně polarizované.

1.7 Čtečky

RFID čtečky jsou prvky RFID systémů sloužící ke sběru informací ze štítků a posílání těchto informací k dalšímu zpracování. Pokud jsou použity pasivní a polopasivní štítky, čtečky slouží zároveň jako zdroj energie potřebné ke komunikaci mezi štítkem a čtečkou. Pomocí čteček je zároveň možné štítky ovládat a ukládat do nich nové hodnoty identifikátorů, štítky uzamknout proti dalším modifikacím, případně uložit nastavení zabraňující štítkům v další komunikaci. Rozdělit lze čtečky v závislosti na použité frekvenci (LF, HF, UHF) a také podle toho, zda se jedná o čtečku s pevným umístěním či čtečku mobilní.

1.8 Použití RFID ke snímání pohybu

V současné době jsou k vyhodnocování pohybu nejčastěji používány kamerové systémy. Ty sebou však nesou jednu velkou nevýhodou a tou je, že sledovaný předmět musí být dobře viditelný kamerou. Při použití RFID tato nutnost odpadá a není tedy divu, že se čím dál častěji můžeme setkat se snahou o použití RFID systémů k tomuto účelu.

V práci [11] byl RFID systém použit ke zjišťování polohy předmětů za účelem navádění robota v místnosti. Robot pomocí RFID antény dvakrát za sekundu zjišťuje přítomnost štítků v okolí a snaží se vyhodnotit jejich polohu v prostoru. Pro zpřesnění vyhodnocování polohy je robot doplněn o laserový skener.

Další aplikaci RFID systému pro vyhodnocování pohybu předmětů můžeme vidět v práci [12], kde bylo snahou vyhodnocovat činnosti prováděné na pracovišti. Ke sledovaným předmětům byly připevněny RFID štítky a bylo zjišťováno, zda je předmět používán či ne. Štítky byly čteny pomocí dvou antén, jedné umístěné na stěně místnosti a druhé umístěné na stropě. To, zda je předmět v pohybu, je vyhodnocováno pomocí změn hodnot parametru RSSI, který udává sílu přijatého signálu ze štítku.

Posledním příkladem použití RFID k vyhodnocování pohybu, který bude uveden v této práci, je práce [13], kde se pokoušeli o snímání pohybu nohy běžce běžícího na běžeckém páse. Pomocí dvojité polarizované antény byl snímán úhel natočení štítků, které byly běžci připevněny na lýtko a stehno a ze získaných dat se autoři práce pokoušeli vyhodnocovat pohyb nohy běžce.

Byť využití RFID systémů ke sledování pohybu předmětů skýtá výhodu, že sledované předměty nemusí být přímo viditelné, pořád je nutné počítat s nevýhodami, které jsou s použitím této technologie spojené. Hlavní nevýhodou je přesnost snímání parametru RSSI. Na ten má vliv jednak prostředí a předměty, které se nacházejí v blízkosti štítků, ale také orientace štítků vůči anténě. Další nevýhodou je relativně nízká snímkovací frekvence, která se v závislosti na použité čtečce a také na množství štítků, které čtečka snímá, pohybuje od jednotek až k nízkým desítkám hertzů. Z tohoto důvodu se můžeme setkat s buď s relativně velkou nepřesností systému pro snímání pohybu čistě založených na RFID technologii, či odebrání některých ze stupňů volnosti snímaných předmětů, anebo doplnění systémů o další technologie pro zpřesnění vyhodnocení. Z těchto důvodů se se systémy snímání pohybu využívající RFID technologii stále nesetkáváme v komerční sféře.

1.9 Princip vyhodnocování polohy v této práci

Jak již bylo zmíněno v kapitole 1.8 vyhodnocování polohy RFID štítků pomocí parametru RSSI je relativně nepřesné a to hlavně kvůli náchylnosti síly signálu na okolí štítku a také jeho orientaci. Hlavní cílem této práce bude vytvořit systém vyhodnocující polohu předmětů a tímto způsobem ovládat hru pro mobilní zařízení s operačním systémem Android. Aby hráč mohl hru pomocí prvků s RFID štítky ovládat, bude nutné zajistit poměrně přesné vyhodnocování polohy ovládacích prvků a to by bylo s využitím RSSI velmi problematické, neboť i hráčova ruka, která by byla k manipulaci s ovládacími prvky pravděpodobně použita, jej značně ovlivní. Z tohoto důvodu bude vyhodnocování polohy v této práci založeno na jiném principu. Stejně jako v práci [1], na kterou tato práce navazuje, budou do blízkosti RFID štítků umístěny hliníkové segmenty, které zajistí stínění daných štítků takovým způsobem, že z nich bude posléze možné vyhodnotit polohu nebo orientaci sledovaného předmětu.

Nenačítání hodnot štítků v přítomnosti kovových (v tomto případě hliníkových) předmětů je způsobeno ze dvou důvodů. Prvním důvodem je skutečnost, že elektromagnetická vlna prochází skrz vodivé předměty pouze do určité hloubky. Tato hloubka závisí jednak na materiálu a jednak na použité frekvenci. V případě UHF RFID systému pracujícím na frekvenci 900 MHz a hliníkovém materiálu, které jsou použity v tomto případě, je hloubka průchodu vlny skrz materiál pouhých 2,7 μm . Hodnoty hloubky průchodu vlny skrz materiál pro další materiály a frekvence je možné najít v tabulce 1. Pokud se tedy bude štítek nacházet za překážkou z hliníku, nebude čtečkou nabit a bude možné vyhodnotit, že překážka se nachází nad daným štítkem.

Vliv na zastínění štítku má ovšem také difrakce elektromagnetické vlny, způsobenou kovovou překážkou. To, jak dobře bude štítek zastíněn při započítání difrakce elektromagnetických vln, lze zjistit z fázového posunu vlny vzniklé difrakcí oproti vlně, která by procházela přímo skrz kovovou překážku. Fázový posun lze tedy vyjádřit jako

$$\phi = \frac{2\pi f(L_2 - L_1)}{c}, \quad (1)$$

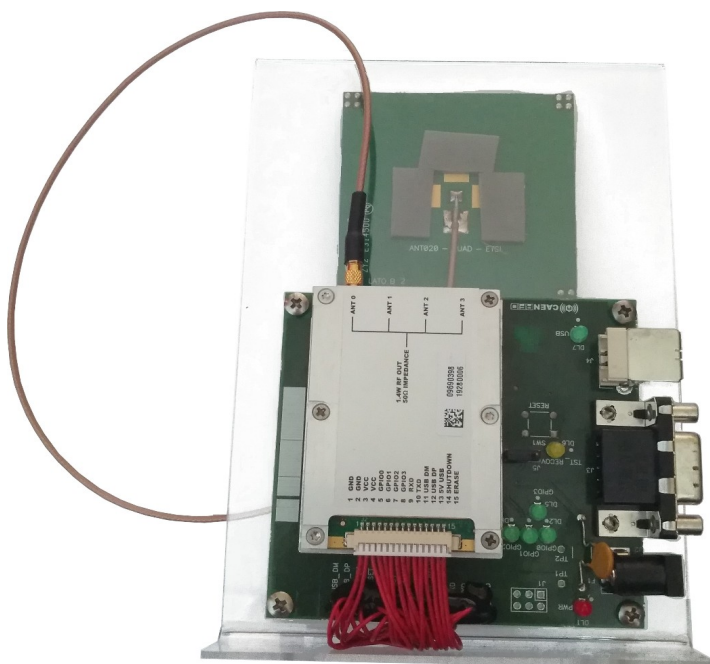
kde f je frekvence, L_1 je přímá vzdálenost jdoucí skrz překážku, L_2 je vzdálenost, po které jde vlna vzniklá difrakcí a c je rychlost šíření vlny. Čím větší je hodnota fázového posunu, tím menší má difrakce vliv na zastínění štítku a tím lépe je štítek stíněný [2].

Druhým důvodem nepřijetí energie emitované skrz antény čtečkou štítkem je vznik takzvané inverzní antény. Inverzní anténa vzniká ve chvíli, když se štítky nacházejí v blízkosti vodivého materiálu. Ten odráží elektromagnetické vlny, čímž vzniká rušení, které má za následek nemožnost štítku přijmout dostatečné množství energie pro komunikaci se čtečkou. V případě, že se štítky nacházejí velmi blízko kovovému materiálu, snížení síly přijatého signálu odpovídá vztahu $(h/\lambda)^2$, kde h je vzdálenost štítku od kovového materiálu a λ je vlnová délka. Ve chvíli, kdy je vzdálenost štítku od kovového materiálu menší než přibližně 1/8 vlnové délky (v případě UHF RFID na 865 MHz se jedná přibližně o 4,5 cm), jde síla přijatého signálu velmi rychle k nule [2]. Aby bylo možné tento efekt kompenzovat a štítky byly stále čitelné, je třeba dostatečně přiblížit anténu čtečky ke štítkům.

2 Použité nástroje a práce s nimi

2.1 Čtečka

Základním nástrojem použitým v této práci byla UHF RFID čtečka Hadron R4320C od výrobce CAEN. Jedná se o stejnou čtečku, která byla použita v práci [1], na kterou tato práce navazuje. Tato čtečka pracuje na frekvenci 865 – 867 MHz a 902 – 928 MHz. Ke čtečce je možné připojit zároveň až čtyři antény. Komunikace mezi čtečkou a ovládacím zařízením (v tomto případě zařízením s operačním systémem Android) může probíhat pomocí USB a nebo pomocí sériového portu UART. Hlavními výhodami této čtečky je její malá velikost, možnost nastavení emitované energie a nižší cena v porovnání s jinými modely RFID čteček. Použitá RFID čtečka je připojena ve vývojovém přípravku, který je možné vidět na obrázku 1. Další informace ohledně použité čtečky je možné najít na v [16].



Obrázek 1: Vývojový přípravek čtečky Hadron R4320C

2.2 Antény

Pro komunikaci mezi čtečkou a štítky byly použity antény WANT020 od firmy CEAN RFID. Jedná se o velmi kompaktní RFID antény pracující ve frekvenčním pásmu 865 – 867 MHz. Antény jsou kruhově polarizované, což dovoluje libovolnou orientaci štítku vůči anténě. Zisk antény udávaný výrobcem je 0,2 dBi. Anténu WANT020 připojenou ke čtečce je možné vidět na zadní straně vývojového přípravku na obrázku 1. Další technické parametry použité antény je možné najít v [17].

2.3 Štítky

Štítky použité v této práci byly štítky ALN-9629 od výrobce Alien. Jedná se o UHF RFID štítky s 64 bitovým identifikátorem pracující na frekvenci 840 – 960 MHz. Hlavním důvodem pro volbu těchto štítků byla jejich malá velikost (25mm na 25mm), díky které bylo možné zmenšit velikost prvků, u kterých byl vyhodnocován jejich pohyb.

2.4 Vývojové prostředí

Aplikace zpracovávající informace přijaté ze čtečky byly vyvíjeny ve vývojovém prostředí Android Studio. Jedná se o vývojové prostředí vyvíjené firmou Google, které disponuje velkým množstvím funkcí, které značně zjednoduší vývoj vytvářených aplikací.

Android studio podporuje vývoj aplikací v programovacích jazycích Java, Kotlin a C++. Aplikace, které vznikly v rámci této práce, byly psány v programovacím jazyce Java.

2.5 Připojení RFID čtečky k mobilnímu zařízení

Informace v kapitolách 2.5 a 2.6 byly čerpány z [14], [15] a [18].

Pro připojení RFID čtečky k mobilnímu zařízení je nejprve nutné vyžádat si od uživatele povolení k použití USB portu. Pro získání povolení od uživatele je nejprve třeba vytvořit objekt implementující rozhraní *BroadcastReceiver*, které je součástí vývojového balíčku prostředí Android Studio.

```
public static final String ACTION_USB_PERMISSION = "com.android.example.USB_PERMISSION";

private final BroadcastReceiver usbReceiver = new BroadcastReceiver() {

    public void onReceive(Context context, Intent intent) {

        String action = intent.getAction();
        if (ACTION_USB_PERMISSION.equals(action)) {
            synchronized (this) {
                UsbDevice device = (UsbDevice)intent.getParcelableExtra(UsbManager.EXTRA_DEVICE);
                if (intent.getBooleanExtra(UsbManager.EXTRA_PERMISSION_GRANTED, false)) {
                    if(device != null){
                        // zde je již možné připojit se ke čtečce
                    }
                }
            }
        }
    }
};
```

Kód uvnitř objektu *BroadcastReceiver* bude vyvolán ve chvíli, když uživatel přístup k USB portu odsouhlasí a nebo zamítne. Pokud uživatel povolení odsouhlasil, je již možné připojit se k RFID čtečce.

Aby byla tato část programu provedena, je nutné *BroadcastReceiver* zaregistrovat. K zaregistrování *BroadcastReceiveru* je nejprve potřeba vytvořit proměnnou typu *IntentFilter*, která definuje typ záměru *BroadcastReceiveru*. Po vytvoření této proměnné je již možné

BroadcastReceiver zaregistrovat. Tato akce bude probíhat pouze při startu aplikace a je tudíž možné ji provést v metodě *onCreate*.

```
@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    IntentFilter filter = new IntentFilter(ACTION_USB_PERMISSION);
    registerReceiver(usbReceiver,filter);
}
```

Pro zobrazení zobrazení žádosti o povolení bude následně třeba vytvořit metodu, která bude v patřičném okamžiku, například po stisku tlačítka, vyvolána. V ní bude nejprve pomocí instance třídy *UsbManager* získán list všech zařízení připojených pomocí USB k zařízení. Následně proběhne iterace všemi takto získanými zařízeními a bude provedena kontrola čísla výrobce, aby se zamezilo chybovým stavům vyplývajícím ze snahy připojit se k jinému zařízení než je daná RFID čtečka.

```
private void askForPermission(){
    UsbManager usbManager = (UsbManager) getSystemService(Context.USB_SERVICE);
    HashMap<String, UsbDevice> deviceList = usbManager.getDeviceList();
    Iterator<UsbDevice> deviceIterator = deviceList.values().iterator();
    while (deviceIterator.hasNext()){
        UsbDevice device = deviceIterator.next();
        if(device.getVendorId() == VENDOR_ID){
            usbManager.requestPermission(device, permissionIntent);
            return;
        }
    }
}
```

Po získání povolení k použití USB portu může proběhnout připojení k RFID čtečce. K připojení byly použity knihovny, které jsou dostupné na webových stránkách výrobce.

Po úspěšném importování knihoven je nejprve třeba vytvořit instanci třídy *CAENRFIDReader*. Pokud uživatel schválil použití USB portu, je možné pomocí metody *findVCPDevice* ve třídě *VCPSerialPort* získat list portů zařízení. Z listu je následně vybrán port, ke kterému je čtečka připojena. Jelikož většina mobilních zařízení disponuje pouze jedním USB portem, jedná se většinou o první objekt v listu, tedy objekt s indexem 0. Následně je možné se pomocí metody *Connect* ke čtečce připojit.

```
CAENRFIDReader reader = new CAENRFIDReader();
List<VCPSerialPort> ports = VCPSerialPort.findVCPDevice(getApplicationContext());
reader.Connect(ports.get(0));
```

Po připojení ke čtečce je dále nutné získat instance odpovídající jednotlivým anténám. Ty lze získat pomocí metody *GetSource*. Vstupním parametrem je označení portu ke kterému je anténa do čtečky zapojena. Čtečka obsahuje celkově čtyři porty jejichž označení je *Source_0*, *Source_1*, *Source_2* a *Source_3*.

```
CAENRFIDLogicalSource source = reader.GetSource("Source_0");
```

2.6 Získávání hodnot ze čtečky

Hodnoty načtených štítků je možné získávat dvěma způsoby. Prvním způsobem je použití metody *InventoryTag*. Pokud je potřeba hodnoty získané ze čtečky filtrovat, případně aktivovat či deaktivovat některé funkce čtečky, metoda přebírá až pět vstupních parametrů. Těmito parametry je hodnota udávající paměťovou banku štítku, hodnota udávající bitovou masku, která má být aplikována na paměťovou banku, délka bitové masky, hodnota reprezentující pozici prvního bitu identifikátoru štítku, kde bude vyhodnocována shoda s hodnotou masky a posledním parametrem je parametr příznaků.

Během vyhodnocování načtených štítků bude čtečka porovnávat hodnoty paměťové banky štítků začínající a končící na nastavené hodnotě s hodnotami v masce a uživateli bude vrátet pouze štítky, které vyhoví tomuto filtru.

Pomocí parametru příznaků je možné aktivovat další funkce čtení. Pomocí prvního bitu proměnné je možné aktivovat či deaktivovat vyhodnocování RSSI. Druhý bit ovládá takzvaný snímkovací režim. Pokud je tento režim deaktivován, jsou načtené štítky uloženy v paměti čtečky a všechny najednou vráceny na konci čtecího cyklu. Po aktivování tohoto režimu jsou štítky odesílány čtečkou do cílového zařízení okamžitě po jejich načtení. Pomocí třetího bitu je možné aktivovat kontinuální čtení, kdy čtečka provede definovaný počet čtecích cyklů. Počet cyklů, které má čtečka provést, je možné nastavit pomocí metody *SetReadCycle* třídy *CAENRFIDLogicalSource*. Pomocí čtvrtého bitu je možné aktivovat režim kompaktních dat, kdy čtečka bude do cílového zařízení odesílat pouze identifikátor štítku a parametry, jako je například údaj o čase čtení, zůstanou neodeslány, čímž lze zrychlit proces čtení.

Po zavolání metoda *InventoryTag* vrátí pole objektů *CAENRFIDTag*, které reprezentují čtečkou přečtené štítky.

```
CAENRFIDTag[] tags = source.InventoryTag(new byte[4], (short)0, (short)0, (short)8);
```

Druhým způsobem získávání hodnot ze čtečky je použití metody *EventInventoryTag*. Pomocí této metody se ve čtečce odstartuje kontinuální čtení a čtečka bude odesílat všechny štítky okamžitě poté, co je přečte.

Vstupní parametry metody *EventInventoryTag* jsou stejné jako v případě metody *InventoryTag*. Zároveň je nutné pomocí metody *SetReadCycle* nastavit počet čtecích cyklů, které má čtečka vykonat. Pokud je nastaveno nula čtecích cyklů, čtečka čte do té doby, než je čtení zastaveno pomocí metody *InventoryAbort*.

Aby bylo možné v aplikaci hodnoty získané ze čtečky zpracovat, je nutné současně pomocí metody *addCAENRFIDEventListener* zaregistrovat proměnnou, která implementuje rozhraní *CAENRFIDEventListener*. Toto rozhraní definuje metodu *CAENRFIDTagNotify*, která bude zavolána poté, co čtečka odešle data načteného štítku.

Využitím tohoto způsobu čtení lze dosáhnout vyšších čtecích rychlostí v porovnání s použitím předchozího způsobu. Nevýhodou je, že čtečka odesílá data každého štítku hned poté, co je načte a je tedy nutné ve vlastní aplikaci vyhodnotit, jaké štítky jsou v daném okamžiku čtené a jaké ne.

```
CAENRFIDEventListener listener = new CAENRFIDEventListener() {
    @Override
    public void CAENRFIDTagNotify(CAENRFIDEvent caenrfidEvent) {
        ArrayList<CAENRFIDNotify> data = caenrfidEvent.getData();
    }
};

reader.addCAENRFIDEventListener(listener);
source.SetReadCycle(0);
source.EventInventoryTag(new byte[4], (short) 0, (short) 0, (short) 6);

...

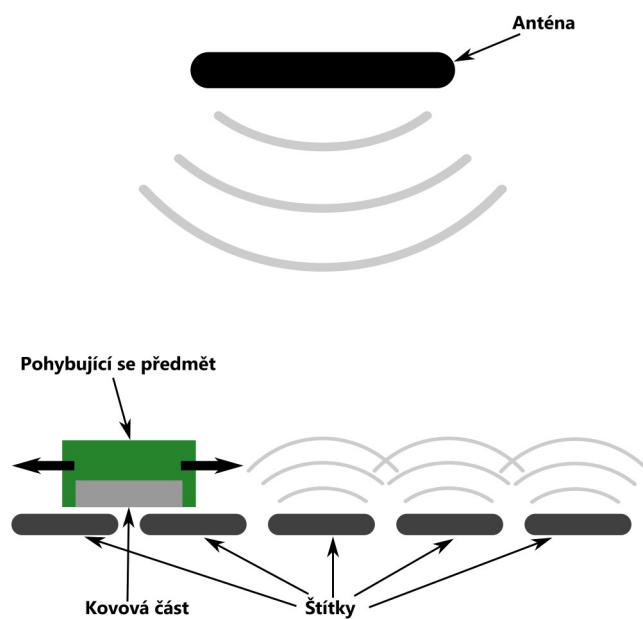
reader.InventoryAbort();
```

3 Poloha předmětu v ploše

Prvním typem vyhodnocování pohybu a polohy předmětu pomocí RFID technologie, jemuž se bude tato práce věnovat, je určování polohy předmětu v ploše vytvořené z RFID štítků.

3.1 Princip určování polohy

V případě vyhodnocování polohy předmětu bude vytvořena z RFID štítků síť, po které se bude pohybovat předmět s kovovou částí, která bude umístěna velmi blízko štítku. Jak již bylo v kapitole 1.9 popsáno, kovové materiály se v případě rádiových vln chovají jako zrcadla a vlnění odrážejí, čímž vzniká takzvaná inverzní anténa. Takto odražené elektromagnetické vlny následně částečně ruší elektromagnetické vlny ze skutečné antény a pokud je štítek dostatečně blízko kovovému materiálu, nepřijme dostatečné množství energie a nebude komunikovat se čtečkou. Schéma tohoto způsobu vyhodnocování polohy předmětu je možné vidět na obrázku 2. Stejný princip byl použit i v práci [1].



Obrázek 2: Vyhodnocování polohy předmětu

3.2 Filtrace hodnot získaných ze čtečky

Při vyhodnocování polohy předmětu pohybujícího se po síti z RFID štítků je zároveň nutné počítat s šumem. Šum se při vyhodnocování polohy projevuje čtením štítků, které by v daném okamžiku čtené být neměly anebo stíněním štítků, které by čtené být měly. Takovéto chování nastává především ve chvíli, kdy se v okolí sítě nachází kromě kovové části pozorovaného předmětu i další vodivé předměty.

Jelikož je cílem této práce vytvořit pomocí technologie RFID systém určování polohy a pohybu použitelný pro ovládání hry, je nutné s přítomností vodivých předmětů v okolí RFID štítků počítat, protože i lidské tělo je vodivé a do jisté míry odráží elektromagnetické vlny, což vede k výše zmíněnému šumu.

Za cílem snížení tohoto šumu je možné implementovat filtr na principu klouzavého průměru. Při použití tohoto způsobu filtrace je to, zda je štítek čtený nebo ne, vyhodnocováno jako průměr z aktuálně získané hodnoty a n hodnot předešlých. Při použití tohoto způsobu dolnoproustní filtrace je možné dosáhnout výrazného snížení vlivu šumu, cenou za to ale současně je vznik zpoždění. Jelikož by reakce hry na hráčem provedené akce měla být co nejrychlejší, je nutné volit délku klouzavého průměru takovým způsobem, aby jeho použití vedlo k dostatečnému zvýšení přesnosti vyhodnocování aktuálního stavu štítku, ale zároveň aby jeho použití nezpůsobilo přílišné zpoždění reakce aplikace na změnu polohy pozorovaného předmětu.

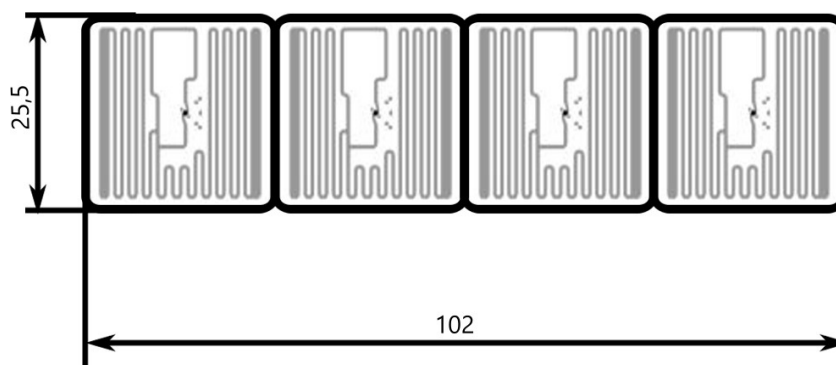
Důležitým faktorem se tedy stává také frekvence, se kterou je čtečka schopná vyhodnotit stav všech okolních štítků. V případě použité čtečky Hadron R4320C od výrobce CAEN se tato frekvence značně liší v závislosti na použitém způsobu čtení. Při použití funkce *InventoryTag*, kdy čtečka vrátí všechny aktuálně viděné štítky, se snímkovací frekvence v případě, kdy se v okolí čtečky nachází jediný štítek, pohybuje okolo 4 Hz. Při použití kontinuálního čtení pomocí funkce *EventInventoryTag* je tato frekvence výrazně vyšší a v případě přítomnosti jediného štítku se pohybuje okolo 25 Hz. Tato frekvence ale výrazně klesá se zvyšujícím se množstvím štítků v okolí antény. V případě čtyř štítků se tato frekvence pohybuje kolem 10Hz a s přidáváním dalších štítků dále klesá.

Závislost snímkovací frekvence na množství štítků tedy výrazně ovlivňuje velikost sítě, kterou je možné pro snímání pohybu použít, protože při použití příliš velké sítě by vyhodnocování polohy trvalo příliš dlouho a pro účely hry by bylo nepoužitelné.

3.3 Vytvoření konstrukce pro určování polohy předmětu

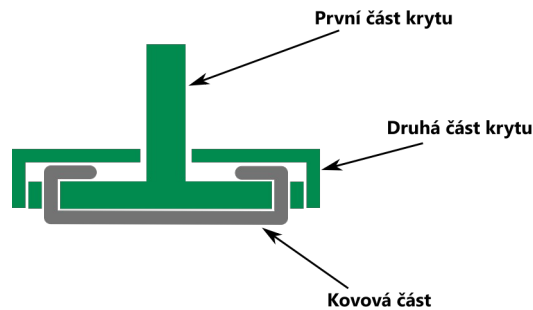
Při vytváření konstrukce, která bude použita pro sledování pohybu předmětu, je nutné brát v potaz to, co je cílem vyhodnocování polohy. V případě této práce bude určování polohy předmětu založené na principu popsaném v kapitole 3.1 použito pro realizace dvou her. První hrou bude obdoba arkádové hry Breakout. Jedná se o hru, ve které je cílem hráče pomocí koule a podložky zničit všechny objekty, které se nacházejí nad podložkou. Koule se pohybuje po herní ploše a odráží se od objektů a podložky a hráč musí zabránit tomu, aby se koule dostala pod podložku. Vyhodnocování polohy pomocí RFID bude v případě této hry použito k ovládní pohybu podložky. Druhou hrou pro jejíž ovládní bude použit stejný princip bude elektronická verze známé logické hry Spoj 4 často známější pod anglickým názvem Connect 4. V této hře je hráčovým cílem jako první vytvořit nepřerušenou linii čtyř svých žetonů. Žetony jsou umísťovány vždy na dno jednoho ze sedmi sloupců. Vyhodnocování polohy pomocí RFID bude v tomto případě použito k zadávání sloupce, do kterého hráč plánuje položit svůj žeton.

V případě hry Breakout bude třeba vytvořit konstrukci, pomocí níž bude možné dostatečně přesně ovládat podložku, od které se bude odrážet koule. První úkolem bude vytvořit síť z RFID štítků, po které se bude pohybovat předmět reprezentující podložku. Jak bylo v kapitole 3.2 řečeno, množství štítků nacházejících se ve čtecí oblasti čtečky má zásadní vliv na rychlost čtení. V případě použití sedmi štítků se frekvence čtení pohybuje kolem 7 Hz a nejdelší doba mezi hráčem provedenou akcí a reakcí aplikace je tedy necelých 150 ms. Zároveň je nutné započítat zpoždění vzniklé filtrací, které je závislé na délce použitého filtru. Je tedy vždy nutné volit kompromis mezi pomalým čtením vzniklým použitím velkého množství štítků a nebo nepřesným vyhodnocováním polohy, které je důsledkem malého množství štítků a tím pádem malého množství poloh, které je možné vyhodnotit. Vzhledem k výše zmíněným problémům byla síť pro vyhodnocování polohy předmětu vytvořena ze čtyř štítků. Takto velká síť dovoluje vyhodnotit celkem sedm poloh, což je v případě hry pro mobilní zařízení dostatečné množství. Frekvence, se kterou je čtečka schopná načíst všechny čtyři štítky se pohybuje kolem 10 Hz. Při použití filtrace hodnot s cílem zpřesnění vyhodnocení aktuální polohy je zpoždění mezi akcí hráče a reakcí aplikace sice znatelné, ale stále dovoluje hráči bezproblémově hru ovládat. Schéma použité sítě je možné vidět na obrázku 3. Rozměry jsou uvedeny v milimetrech.



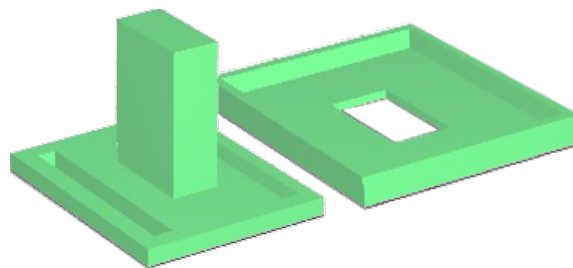
Obrázek 3: Schéma použité sítě štítků

Další úkolem při návrhu konstrukce bude vytvoření dílu, jehož pozice bude vyhodnocována. Hlavní částí tohoto dílu bude část kovová, která bude mít za úkol zamezit komunikaci štítků, které se nacházejí pod ní, se čtečkou. Aby bylo možné jednoduše kovový díl přemísťovat po ploše polepené RFID štítky, bude kovová část uchycena v plastovém krytu. Nákres celého dílu je možné vidět na obrázku 4.



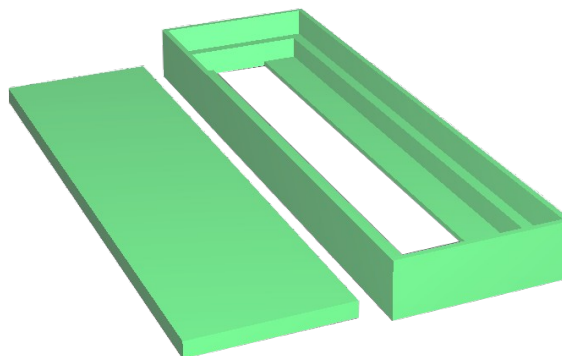
Obrázek 4: Nákres posuvného dílu

Kryt se skládá ze dvou částí. První část slouží ke snadnému přemísťování, druhá část slouží k uchycení kovové části k první. Modely obou částí krytu je možné vidět na obrázku 5.



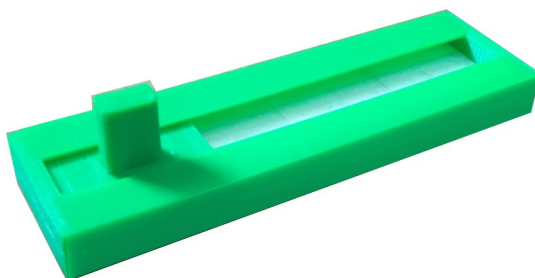
Obrázek 5: Modely dílů krytu

Poslední částí konstrukce bude díl zajišťující správnou polohu sledovaného předmětu vůči RFID štítkům. Druhou funkcí tohoto dílu bude zabránění přímého kontaktu hráčovy ruky s RFID štítky, což by mohlo mít za následek jednak nechtěné rušení štítků a jednak možné poškození štítků. Tento díl se skládá ze dvou částí, které je možné vidět na obrázku 6.



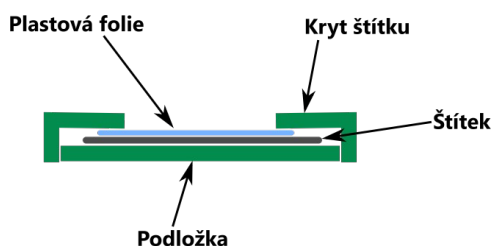
Obrázek 6: Modely dílů konstrukce

Složenou konstrukci ovládacího prvku pro hru Breakout je možné vidět na obrázku 7.



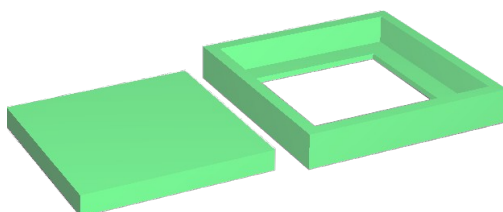
Obrázek 7: Konstrukce ovládacího prvku

V případě hry Spoj 4 bude vytvořena konstrukce sloužící jako bezdrátové tlačítko, pomocí něž bude možné zadávat pozici sloupce, do kterého hráč plánuje umístit svůj žeton. Na rozdíl od hry Breakout bude rušení komunikace mezi štítkem a čtečkou docíleno položením prstu na dané tlačítko. Štítek přestane být čtečkou viděn z důvodu vodivosti lidského těla a částečného odrážení elektromagnetického vlnění. Odraz vlnění není v případě lidského těla natolik dobrý jako v případě hliníku, ale v případě správného nastavení výstupního výkonu čtečky a pozice tlačítka vůči anténě bude dostatečné k spolehlivému zabránění štítku v nabití a tím pádem komunikaci se čtečkou. Náčrsek tlačítka je možné vidět na obrázku 8.

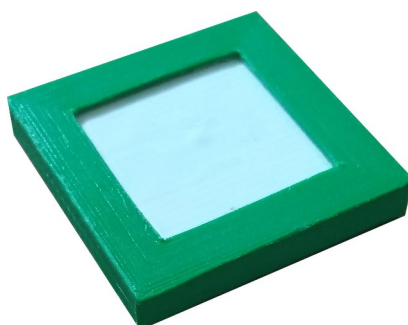


Obrázek 8: Náčrsek tlačítka

Konstrukce bezdrátového tlačítka se skládá celkem ze čtyř částí. Hlavní částí je štítek, pomocí něž bude vyhodnocen dotyk hráče. Druhou částí je podložka, na kterou bude štítek nalepen. Třetí částí konstrukce je plastová folie chránící štítek před poškozením při doteku. Poslední částí je kryt chránící štítek a fólii před nechtěným odlepením od podložky. Modely krytu a podložky je možné vidět na obrázku 9. Sestavené tlačítko je možné vidět na obrázku 10.



Obrázek 9: Modely podložky a krytu



Obrázek 10: Sestavené tlačítko

3.4 Vyhodnocení polohy předmětu

Princip vyhodnocování pozice bude v obou případech velmi podobný. Lišit se bude především v parametrech ovlivňující spolehlivost a rychlost vyhodnocení. V případě vyhodnocování nastavené pozice podložky ve hře Breakout bude kladen důraz hlavně na rychlost, aby prodleva mezi změnou polohy a reakcí aplikace byla co nejmenší. V případě hry Spoj 4 bude důležité vyhodnotit pozici s co nejvyšší spolehlivostí, aby nedošlo k položení žetonu hráče do sloupce, který hráč nevybral.

V případě hry Breakout bude vyhodnocování pozice probíhat ve dvou fázích. V první fázi budou hodnoty získané ze čtečky filtrovány pomocí filtru založeném na klouzavém průměru. Hodnota štítku je určena jako průměr hodnoty získané ze čtečky (jedná se o binární hodnotu, 1 pokud je štítek čtečkou viděn 0 pokud viděn není) s n předešlými hodnotami.

V druhé fázi vyhodnocování je ze všech štítků vybrán štítek s nejnižší hodnotou a pozice, které tento štítek odpovídá, je označena za právě nastavenou pozici. V případě, že nejnižší hodnotu má více než jeden štítek, je pozice vypočtena jako průměr mezi těmito pozicemi.

Výsledné maximální zpoždění mezi nastavením nové pozice a reakcí v závislosti na délce filtru n lze vyjádřit jako $(\lfloor n/2 \rfloor + 1) \cdot T$, kde T je perioda mezi jednotlivými čtení. V případě čtyř štítků, které byly k realizaci ovládacího prvku pro tuto hru použity, je průměrná hodnota periody $\bar{T} = 100 \text{ ms}$. Aby bylo ovládání hry dostatečně citlivé, nesmí být reakce hry na novou pozici příliš vysoká a tudíž délka filtru příliš velká. Zároveň musí být filtr patřičně dlouhý, aby dostatečně potlačil šum. Z testování vychází jako nejlepší hodnota $n=5$, kdy je délka filtru dostatečná k odfiltrování rychlých dějů a zároveň nejdelší doba mezi nastavením nové pozice a reakcí aplikace je 300 ms, což je prodleva znatelná avšak akceptovatelná.

Jak již bylo řečeno, v případě hry Spoj 4 bude algoritmus vyhodnocování pozice velmi podobný jako v případě hry Breakout. Rozdíl bude v délce použitého filtru. Aby bylo zajištěno i za cenu výrazného zvýšení doby mezi dotekem a reakcí aplikace, že nedojde k položení žetonu na pozici, která nebyla hráčem vybrána, bude v případě této hry zvolena délka filtru $n=7$. Dotyk hráče na dané tlačítko bude vyhodnocen v případě, že je hodnota štítku odpovídající tomuto tlačítku rovna 0. V případě, že v jeden okamžik bude mít více štítků nulovou hodnotu, bude vyhodnocen chybový stav a žeton hráče nebude položen do žádného ze sloupců.

Zpoždění v případě hry Spoj 4 bude zřejmě vyšší jednak z důvodu delšího filtru a jednak kvůli většímu množství štítků a tím pádem větší periodě T . Ta se v případě sedmi štítků, které jsou

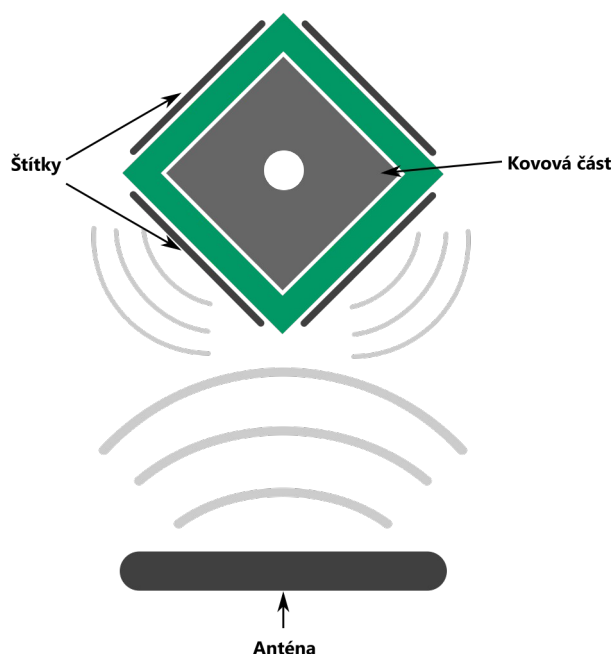
u této hry potřeba, pohybuje kolem 150 ms. Dobu, která trvá, než hodnota štítku přejde z 1, kdy je štítek čtečkou čten po celou délku filtru, do 0, kdy dojde k vyhodnocení doteku, je možné vyjádřit jako $n \cdot T$. Pro $n=7$ a $\bar{T}=150\text{ ms}$ je výsledné maximální zpoždění rovno 1050 ms.

4 Rotační pohyb válcového předmětu

Dalším typem pohybu, jehož snímání se tato práce bude věnovat, je rotační pohyb. Aby bylo snímání pohybu co nejpřesnější, bude se jednat o předmět s pevnou polohou v prostoru a jediný možný pohyb bude rotace kolem jedné osy. Cílem snímání tohoto předmětu s jedním stupněm volnosti bude ze získaných dat vypočítat průběh úhlové dráhy a tento průběh následně převést do mobilní aplikace. V rámci této práce bude pohyb předmětu zjednodušen na rovnoměrně zpomalený pohyb. Důvodem tohoto zjednodušení je nízká vzorkovací frekvence, se kterou je čtečka schopná snímat polohu předmětu.

4.1 Princip určování orientace předmětu

Aby bylo možné snímat úhel natočení předmětu pomocí RFID technologie, bude třeba předmět konstruovat takovým způsobem, aby štítky, které budou v daný okamžik čtečkou čtené, jednoznačně definovaly daný úhel. Princip, na kterém bude určování orientace probíhat, je možné vidět na obrázku 11. Informace v této kapitole byly čerpány z [2] a [3].



Obrázek 11: Princip určení orientace předmětu

Zatímco v případě určování polohy předmětu v prostoru popsaném v kapitole 3 byl hlavním důvodem možnosti vyhodnotit polohu sledovaného předmětu odraz elektromagnetických vln od kovového předmětu a následném rušení vlnění, které je emitované anténou čtečky takto odraženým vlněním, v případě určování natočení předmětu bude hlavním důvodem stín vznikající díky nemožnosti průchodu vln skrz kov. Se čtečkou poté budou komunikovat pouze štítky, které se nacházejí na přivrácené straně předmětu k anténě čtečky, což je možné vidět na obrázku 11. Aby štítky nebyly příliš rušené vlnami odraženými, je nutné štítek umístit

do dostatečné vzdálenosti od kovového předmětu. Z tohoto důvodu bude mezi štítek a kovový předmět umístěn díl, který bude tuto vzdálenost zajišťovat. Za účelem nalezení co možná nejlepší hodnoty vzdálenosti mezi štítkem a kovovým předmětem bylo provedeno několik testů, z nichž vyšla jako nejlepší vzdálenost 5 mm, při které byla úspěšnost vyhodnocení orientace nejvyšší (testy byly provedeny se vzdálenostmi 2, 3, 4, 5, 6 a 7 mm).

Na to, jak dobře budou štítky na odvrácené straně předmětu stíněny, bude mít vliv i difrakce. To, jak velký tento vliv bude, lze zjistit z fázového posunu vlny, která vznikne difrakcí, oproti vlně, která by prošla skrz kovovou překážku. Z toho vyplývá, že čím větší předmět stínící štítek je a čím blíže je štítek předmětu, tím lépe je štítek stíněn. Štítek však nemůže být příliš blízko kovovému materiálu, neboť odražené vlnění by mu nedovolilo dostatečně se nabít. I velikost předmětu stínícího štítek je omezena, protože příliš velký předmět by byl pro realizaci ovládacího prvku hry nepoužitelný. K velikosti ovládacího prvku je nutné připočítat velikost dílu tvořícího mezeru mezi štítky a kovovým předmětem a také velikost krytu, tvořícího mezeru mezi štítky a hráčovou rukou. Velikost hrany kovového předmětu byla stanovena na 35 mm. Fázový posun lze vypočítat pomocí vzorce (1). Pro tento případ jsou hodnoty parametrů $L_1=0,005\text{ m}$, $L_2\approx 0,018\text{ m}$, $f=865\,000\,000\text{ Hz}$ a $c=299\,792\,458\text{ ms}^{-1}$. Fázový posun je tedy roven

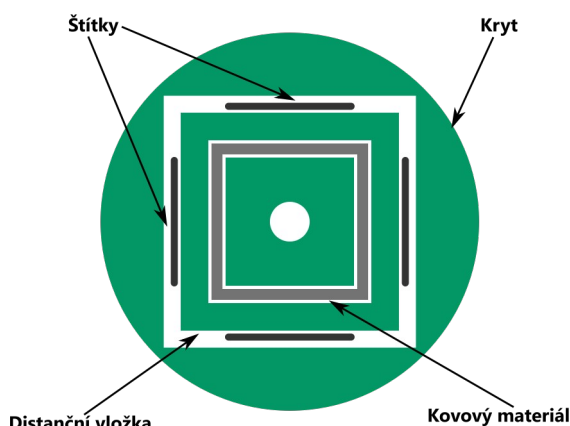
$$\phi = \frac{2\pi \cdot 865 \cdot 10^6 \cdot (0,018 - 0,005)}{299792458} \approx 0,236\text{ rad} \quad . \quad (2)$$

Pokud je tato hodnota nižší než 2π , což je dle literatury [2] hodnota, od které je stín vznikající díky překážce jasně definovaný, pak difrakce nemá při stínění velký vliv. V tomto případě je hodnota fázového posunu výrazně nižší než 2π a stín tedy není jasně definovaný a difrakce vlnění bude mít na spolehlivost stínění značný vliv. Při započítání snížení intenzity vlnění vlivem blízkosti kovového materiálu a zvolení správné pozice bude však vznikající stín dostatečný ke správnému vyhodnocení úhlu natočení rotujícího předmětu. Dalším faktorem, který značně ovlivňuje úspěšnost vyhodnocení pozice, je rychlost, se kterou čtečka bude schopna načíst všechny štítky, které se nacházejí na přivrácené straně předmětu. Tato rychlost je nepřímo úměrná množství přítomných štítků. Pro případ této práce byly k určení pozice použity čtyři štítky. V jeden okamžik by měly být čtečkou viděny maximálně tři štítky, které je čtečka schopná přečíst v průměru jednou za 80 ms. Z Nyquistova–Shannonova teoremu vyplývá, že aby bylo možné snímat rotační pohyb předmětu s vzorkovací periodou $\bar{T}=80\text{ ms}$, nesmí frekvence otáčení přesáhnout hodnotu $f=6,25\text{ Hz}$.

4.2 Vytvoření konstrukce snímaného předmětu

Jak bylo řečeno v kapitole 4.1 vzniklý stín nebude jasně definovaný a bude na něj mít velký vliv vlnění vzniklé difrakcí. Z tohoto důvodu byly vytvořeny dvě konstrukce, na kterých byly prováděny testy a následně byla vybrána konstrukce s vyšší spolehlivostí vyhodnocení nastavené pozice.

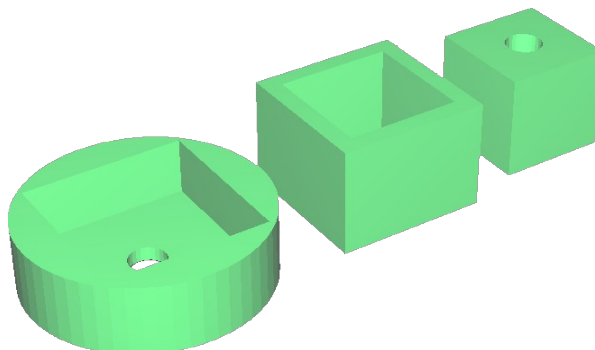
Nákres první konstrukce použité pro snímání úhlu natočení je možné vidět na obrázku 12.



Obrázek 12: První navržená konstrukce

Předmět, jehož natočení bude snímáno, se skládá celkem ze čtyř hlavních částí. První částí je kovový (v případě této práce hliníkový) materiál, jehož cílem je stínit štítky, které se nacházejí za ním. Jelikož elektromagnetická vlna projde skrz materiál pouze do velmi malé hloubky, není nutné, aby kovová část byla plná a je možné ji vyplnit jiným, v tomto případě plastovým, materiálem. Druhou částí je díl, jehož úkolem je vytvořit mezeru mezi kovovým materiálem a štítky. Na obrázku 12 je tato část nazvána jako distanční vložka. Třetí částí jsou štítky, které v závislosti na orientaci předmětu budou stíněné a nebo čtené a bude tímto způsobem možné vyhodnotit natočení předmětu. Poslední částí je kryt, který plní funkci ochrany proti poškození a také proti rušení vzniklé interakcí vodivých hráčkových rukou se sledovaným předmětem.

Modely jednotlivých částí tištěných na 3D tiskárně je možné vidět na obrázku 13.



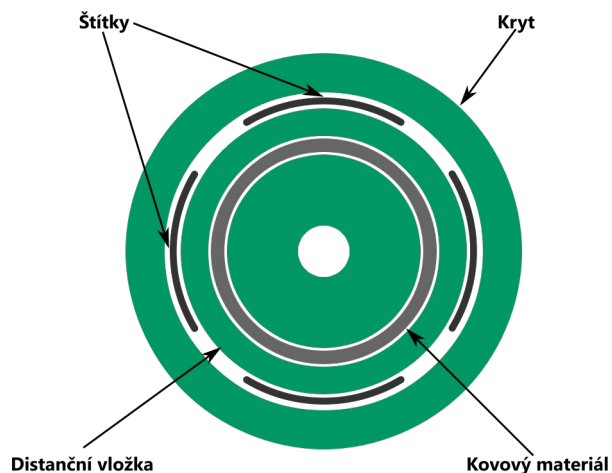
Obrázek 13: Modely dílů první konstrukce

Částečně složenou první konstrukci snímaného předmětu je možné vidět na obrázku 14.



Obrázek 14: Částečně složená první konstrukce

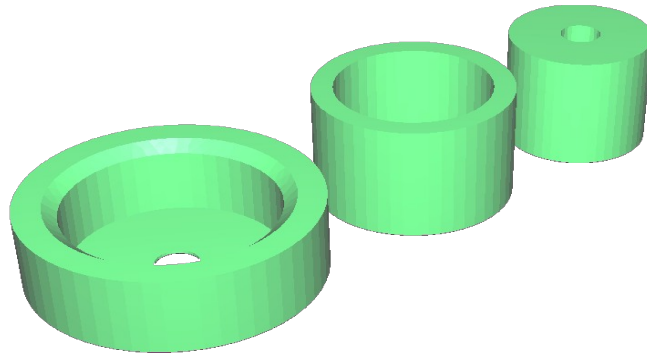
V případě druhé konstrukce použité ke snímání pohybu zůstal princip vyhodnocování pozice stejný. Oproti konstrukci první byl změněn tvar vnitřních částí, která slouží ke stínění štítků. Nákres druhé použité konstrukce je možné vidět na obrázku 15.



Obrázek 15: Druhá navržená konstrukce

Jak je z obrázku 15 patrné, v případě druhé konstrukce jsou vnitřní části namísto kvádrového tvaru, jak tomu bylo v případě konstrukce první, ve tvaru válce. Díky tomu bude vzniklý stín odlišný a bude možné vybrat konstrukci, u které bude vyhodnocení pozice spolehlivější.

Modely druhé konstrukce použité pro 3D tisk je možné vidět na obrázku 16.



Obrázek 16: Modely dílů druhé konstrukce

Vnitřní část druhé konstrukce snímaného předmětu je možné vidět na obrázku 17.



Obrázek 17: Vnitřní část druhé konstrukce

4.3 Algoritmus vyhodnocení rotačního pohybu

Po vytvoření konstrukcí bylo možné přistoupit k tvorbě a ladění algoritmu pro vyhodnocování rotačního pohybu. Při procesu vývoje tohoto algoritmu bylo vyzkoušeno více principů, z nichž některé ke kýženému výsledku vedly a jiné ne. Zásadní vliv na úspěšnost vyhodnocení měla také vzorkovací frekvence, se kterou čtečka dokázala přečíst všechny potřebné štítky a také vznikající šum v získaných datech.

V případě tří štítků, což je maximum štítků, které by měly být v jeden okamžik čtečkou čtené, se vzorkovací frekvence pohybuje kolem 12,5 Hz a dle Nyquistovova–Shannonova teorému by frekvence otáčení neměla přesáhnout polovinu této frekvence tedy 6,25 Hz. V případě, kdy by došlo k porušení této podmínky, došlo by také k chybnému vyhodnocení vzdálenosti mezi současnou a předchozí pozicí a tím pádem i k chybnému výpočtu průběhu úhlové vzdálenosti. Jelikož ale bude snímání pohybu použito ve hře, bude tato podmínka v naprosté většině

případů splněna, neboť hráč nebude schopen předmět rukou roztočit dostatečně rychle k jejímu porušení.

Z tohoto důvodu bylo snahou v případě prvního algoritmu pro vyhodnocení pohybu rotujícího předmětu vyhodnocovat a převádět do aplikace tento pohyb v reálném čase. Při použití tohoto algoritmu byla neustále vyhodnocována úhlová vzdálenost současné pozice od pozice předchozí. Pozice, ve které se válcový předmět nachází, byla vyhodnocena pomocí množiny štítků, které byly v daný okamžik čtečkou čtené. Každému štítku byla přidělena pozice na jednotkové kružnici a výslednou pozici bylo možné vyhodnotit jako jednotkový vektor ve směru vektoru vzniklého ze součtu všech bodů aktuálně čtených štítků. Ve chvíli, kdy tato vzdálenost byla větší než nula, byla z této vzdálenosti a časového rozdílu mezi pozicemi spočtena aktuální rychlost a z orientace úhlu směr pohybu.

První algoritmus je tedy velmi jednoduchý a také snadno implementovatelný. Problém je, že funguje pouze teoreticky, neboť počítá se stoprocentní úspěšností vyhodnocení aktuálního úhlu natočení předmětu. Ve skutečnosti však tato úspěšnost ani zdaleka stoprocentní není. V některých případech docházelo k nabíjení a čtení štítků, které měly být skryty za kovovým předmětem a vlivem toho docházelo k vyhodnocení pohybu i v případech, kdy k pohybu nedocházelo, ale hlavně k vyhodnocování chybných orientací úhlu v případech, když se předmět pohyboval. Díky tomu byl první algoritmus v aplikaci prakticky nepoužitelný a bylo nutné vytvořit algoritmus dostatečně odolný vůči chybnému vyhodnocení aktuální pozice.

V případě druhého algoritmu bylo snahou vytvořit takový algoritmus, který by se dokázal s chybnými hodnotami pozice vypořádat, zároveň ale zaváděl co možná nejmenší zpoždění mezi pohybem předmětu a reakcí aplikace.

Nejprve byla vytvořena fronta, do které byly ukládány vyhodnocené pozice. Ve chvíli, kdy se ve frontě nacházelo n hodnot, začalo vyhodnocování, zda se předmět pohybuje nebo ne. Předmět byl vyhodnocen jako pohybující se ve chvíli, kdy se ve frontě nacházely více než dvě různé pozice. Následně bylo získáno dalších m hodnot, ze kterých společně s hodnotami použitými k vyhodnocení začátku pohybu, bylo vyhodnoceno v jakém směru k pohybu dochází.

Poté byl seznam všech získaných pozic rozdělen na dvě poloviny. Z první poloviny byla vyhodnocena počáteční rychlost předmětu, z poloviny druhé byla vyhodnocena konečná rychlost předmětu. Počáteční i konečná rychlost byly vyhodnoceny jako maxima z daného intervalu v seznamu. Pokud došlo k tomu, že konečná rychlost je vyšší než počáteční, byla vybrána z intervalu výpočtu konečné rychlosti taková hodnota, která byla nižší než počáteční rychlost. K tomu, že rychlost počáteční je nižší než konečná, mohlo dojít vlivem nepřesnosti vyhodnocené pozice. V případě, že se v intervalu výpočtu konečné rychlosti nenacházela žádná rychlost nižší než počáteční, byl výpočet přerušen a hodnota zpomalení byla stanovena z předem naměřených hodnot. Pokud konečná rychlost byla nižší než počáteční, byla hodnota zpomalení spočtena z rozdílu velikostí a časů rychlostí. Hodnoty rychlosti a zpomalení byly vynulovány ve chvíli, kde se ve frontě délky n nenacházelo více než dvě různé pozice a byl vyhodnocen konec pohybu.

Úspěšnost vyhodnocení pohybu byla v případě druhého algoritmu podstatně lepší než v případě prvního. Úspěšnost vyhodnocení byla velmi ovlivněna parametry n a m . Aby

reakce aplikace byla dostatečně rychlá, byly zvoleny hodnoty $n=4$ a $m=3$. U vyhodnocení směru pohybu se pohybovala kolem 70%, což je ovšem stále velmi nízká hodnota, vezmeme-li v potaz to, že pokud by byl směr pohybu zvolen náhodně, teoreticky by se shodoval se směrem skutečným v 50% případů. V případě určování rychlosti byla úspěšnost rovněž velmi nízká. Skutečnosti odpovídala pouze asi v jedné třetině případů. Přesnost vyhodnocení rychlosti a zpomalení byla současně negativně ovlivněna malým množstvím štítků rozmístěných po obvodu předmětu, které ale bylo nutné kvůli zachování dostatečně rychlé vzorkovací frekvence.

U třetího vytvořeného algoritmu bylo upuštěno od snahy o převod pohybu do aplikace v co nejrychlejším čase a pohyb byl vyhodnocován ze sekvence pozic od začátku pohybu až do jeho ukončení.

Nejprve bylo nutné zjistit, kdy se předmět pohybuje. Způsob vyhodnocení toho, zda se předmět pohybuje se shodoval se způsobem použitým v druhém algoritmu, neboť se ukázal jako dostatečně spolehlivý.

Po vyhodnocení začátku a konce pohybu došlo ke kontrole, zda se předmět pohyboval po dostatečně dlouhou dobu. Pokud byl detekovaný pohyb příliš krátký, jednalo se často o hodnoty vzniklé difrakcí a jiným rušením a bylo je třeba odfiltrovat.

Následně bylo opět nutné vyhodnotit orientaci pohybu. I v tomto případě byl použit stejný princip, jako v druhém algoritmu a byly počítány vzdálenosti mezi jednotlivými pozicemi a směr s nejnižší hodnotou celkové vzdálenosti byl vyhodnocen jako směr pohybu reálného předmětu. Rozdíl mezi třetím a druhým algoritmem v této fázi vyhodnocování byl v délce sekvence, ze které byla orientace pohybu vyhodnocována. Zatímco v případě druhého algoritmu bylo použito pouze prvních $n+m$ hodnot, v případě algoritmu třetího byla orientace vyhodnocena z pozic získaných během celého trvání pohybu, díky čemuž se výrazně zvýšila úspěšnost.

V další fázi algoritmu bylo třeba získat hodnotu počáteční rychlosti a hodnotu zpomalení. Prvním pokusem a vyhodnocení těchto parametrů bylo umístění získaných pozic do patřičných vzdáleností a proložení hodnot vzdáleností kvadratickou funkcí s nulovým absolutním členem. Z rovnice kvadratické funkce bylo následně možné získat hodnotu zrychlení a z doby trvání pohybu poté počáteční rychlost. Tento způsob vyhodnocení rychlosti a zpomalení se ovšem ukázal velmi málo odolný vůči špatně vyhodnoceným pozicím a ve většině případů rychlost a zpomalení neodpovídaly realitě.

Dalším pokusem bylo místo výpočtu zpomalení ze získaných hodnot použít hodnotu získanou z provedených měření. V tomto případě odpovídaly vypočtené hodnoty skutečnosti podstatně více. Nevýhodou použití tohoto způsobu výpočtu je, že pokud hráč nenechá předmět zastavit vlivem tření a zastaví ho sám, bude sice teoreticky odpovídat doba pohybu předmětu v aplikaci se skutečným předmětem, ale nebude odpovídat počáteční rychlost pohybu. I se započtením této nevýhody byl nakonec použit tento způsob výpočtu, protože ve většině případů podával lepší výsledky než způsoby předešlé.

Třetí algoritmus vyhodnocení pohybu předmětu se ukázal jako dostatečně úspěšný a z tohoto důvodu byl nakonec použit ve finální verzi mobilní aplikace.

4.4 Testování vytvořených konstrukcí

Po vytvoření obou konstrukcí a algoritmu vyhodnocení pohybu bylo možné přistoupit k jejich testování. Cílem těchto testů bylo zjištění, u které z vytvořených konstrukcí bude vyhodnocení orientace spolehlivější a tedy která konstrukce bude pro účel hry vhodnější.

Úspěšnost vyhodnocení pohybu předmětu z provedených testů byla pro případy obou konstrukcí velmi podobná. U obou konstrukcí se při správném zvolení pozice předmětu vůči anténě čtečky úspěšnost vyhodnocení směru pohybu pohybovala kolem 90%, tedy v 10% případech, kdy bylo vyhodnoceno, že dochází k pohybu předmětu, byl vyhodnocený směr pohybu opačný než směr skutečný. V případě vyhodnocování začátku a konce pohybu byla úspěšnost vyhodnocení nižší. Při použití první konstrukce bylo v přibližně 75% případů z nasnímané sekvence pozic vyhodnoceno, že k pohybu dochází. Při použití konstrukce druhé byla úspěšnost podstatně vyšší a pohybovala se kolem 85%. Nejnižší úspěšnost vyhodnocení byla u obou testovaných konstrukcí u vyhodnocení času, po který se předmět pohyboval. Jelikož bude vyhodnocování pohybu použité ve hře, není nutné, aby byl čas, po který se předmět pohyboval, vyhodnocen zcela přesně, ale aby alespoň přibližně odpovídal realitě, protože sám hráč není schopný přesnou dobu pohybu určit. V případě první konstrukce byla doba pohybu vyhodnocena správně asi v 60% případů. V případě druhé konstrukce byla úspěšnost vyhodnocení opět vyšší a pohybovala se kolem 75%.

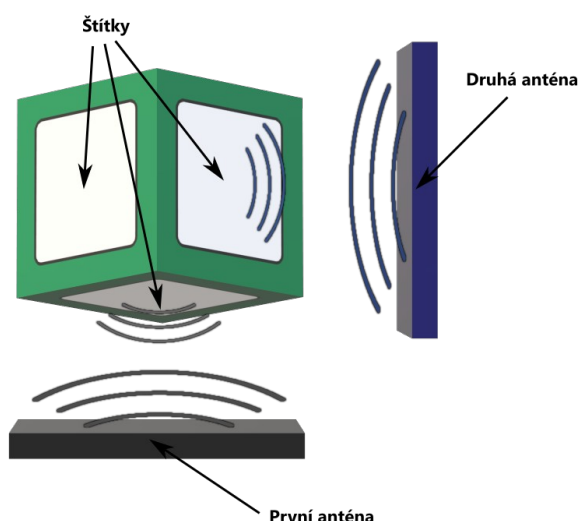
Z provedených testů se jako výrazně lepší ukázala druhá navržená konstrukce, u které byla úspěšnost vyhodnocování ve všech případech lepší. Z tohoto důvodu bude k ovládní hry použita druhá konstrukce sledovaného předmětu.

5 Rotační pohyb sférického předmětu

Posledním a nejsložitějším způsobem pohybu, o jehož snímání se bude tato práce snažit, bude rotační pohyb sférického předmětu. Stejně jako v předchozím případě snímání pohybu pomocí RFID bude pozice předmětu v prostoru pevná a povolené budou pouze rotace kolem os x , y a z . I při vyhodnocování rotačního pohybu tohoto předmětu bude uvažován velmi zjednodušený způsob pohybu, ve kterém předmět s konstantním zpomalením rotuje po celou dobu pohybu kolem jedné osy (tím není myšlena osa x , y a z ale libovolná osa rotace procházející středem předmětu). Cílem vyhodnocování tohoto typu pohybu bude pomocí výpočtu zjistit osu, kolem které předmět rotuje a následně odhadnout průběh úhlové dráhy, kterou předmět při rotaci kolem dané osy urazí.

5.1 Princip určování orientace sférického předmětu

Princip vyhodnocování orientace předmětu se třemi stupni volnosti bude téměř stejný, jako tomu bylo v předchozím případě. Hlavní rozdíl mezi vyhodnocováním orientace předmětu rotujícího kolem jedné pevně zvolené osy a orientací předmětu rotujícího kolem libovolné osy je ten, že zatímco v předchozím případě byly štítky uspořádány pouze ve dvou rozměrech, v tomto případě budou rozloženy do rozměrů tří. Dalším rozdílem bude nutnost použití k vyhodnocování orientace předmětu dvou antén, neboť polohu koule v prostoru je možné definovat pomocí nejméně tří bodů. Jeden z bodů, který bude polohu předmětu definovat, je střed koule, který, jak bylo zmíněno výše, bude pevně daný. Druhé dva body budou získány pomocí dvou antén. Ostatní parametry vyhodnocování orientace zůstávají nezměněny. Ve středu předmětu se bude nacházet kovový materiál, jehož úkolem bude stínit štítky, které se nacházejí na straně odvrácené od dané antény. Schematický náčrt principu určování orientace předmětu se třemi stupni volnosti je možné vidět na obrázku 18.



Obrázek 18: Princip určení orientace předmětu

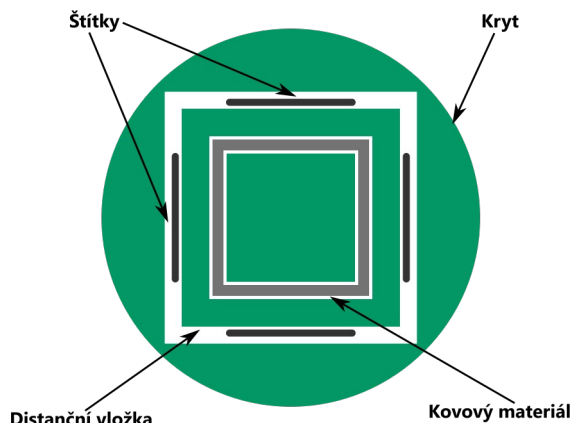
I v případě určování orientace tohoto předmětu bude mít velký vliv na stínění štítků difrakce. Jelikož velikost kovové části, která se nachází ve středu rotujícího předmětu, zůstala stejná jako v předchozím případě určování pohybu předmětu, bude i v tomto případě stín špatně definovaný a bude značně závislý na pozici předmětu vůči anténám.

Oproti předchozímu případu vyhodnocování bude vyhodnocování tohoto typu pohybu disponovat jednou zásadní nevýhodou, kterou je nutnost použití dvou antén. Vlivem použití dvou antén výrazně vzroste čas, který bude trvat získání všech hodnot štítků ze čtečky. Perioda, se kterou je čtečka schopná předat všechna data potřebná k vyhodnocení pozice do aplikace, se pohybuje kolem 200 ms a dle Nyquistova–Shannonova teorému nesmí frekvence, se kterou se předmět kolem osy otáčí, tedy přesáhnout 2,5 Hz. Dvě a půl otáčky za sekundu je velmi nízká frekvence otáčení a i když předmět bude uveden do pohybu hráčovou rukou, bude často vzorkovací teorém porušen.

5.2 Vytvoření konstrukce rotujícího předmětu

Ze stejného důvodu jako v předchozím případě snímání pohybu, budou i v tomto případě vytvořeny dvě konstrukce předmětu, jehož pohyb bude snahou vyhodnotit.

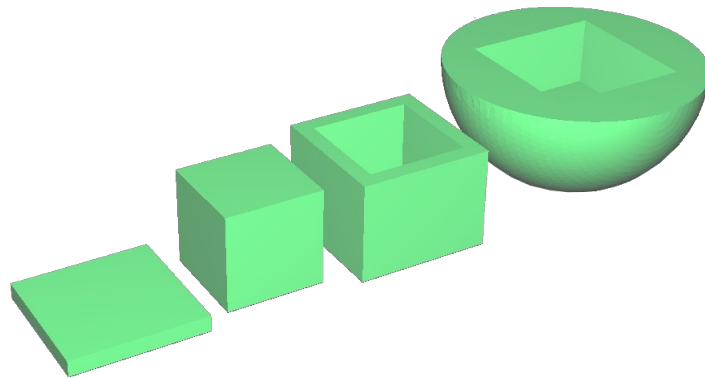
Tvar vnitřní části první konstrukce snímaného předmětu bude odpovídat tvaru předmětu na obrázku 18. Návrh první konstrukce je možné vidět na obrázku 19.



Obrázek 19: Návrh první konstrukce

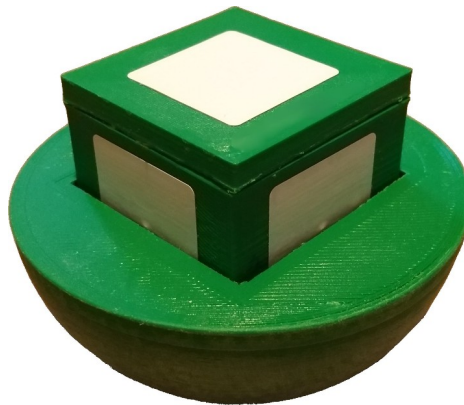
Návrh řezu první konstrukcí na obrázku 19 vypadá díky dvourozměrnosti obrázku téměř stejně jako řez konstrukcí na obrázku 12 použité v předchozím případě vyhodnocování pohybu. Zatímco ale v předchozím případě byl výsledný tvar snímaného předmětu válcový, v tomto případě je předmět i s krytem ve tvaru koule. Funkce jednotlivých dílů, ze kterých se předmět skládá, jsou stejné jako v předchozím případě. Kovová část stíní štítky nacházející se za ní, distanční vložka vytváří mezeru mezi kovem a štítky nutnou k tomu, aby štítky bylo možné přečíst a kryt chrání štítky před kontaktem s hráčovou rukou.

Modely částí první konstrukce, které byly tištěny na 3D tiskárně, je možné vidět na obrázku 20.



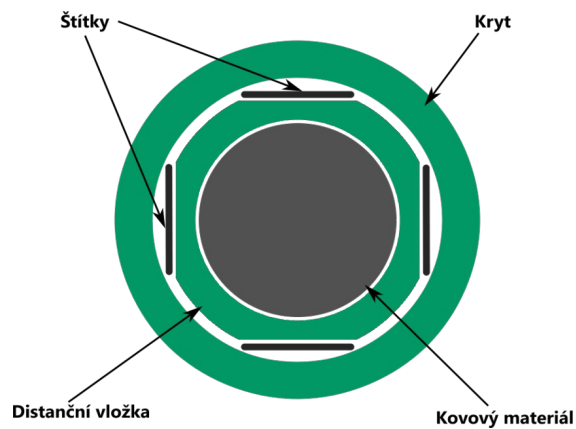
Obrázek 20: Modely tištěných dílů první konstrukce

Částečně složenou první vytvořenou konstrukci je možné vidět na obrázku 21.



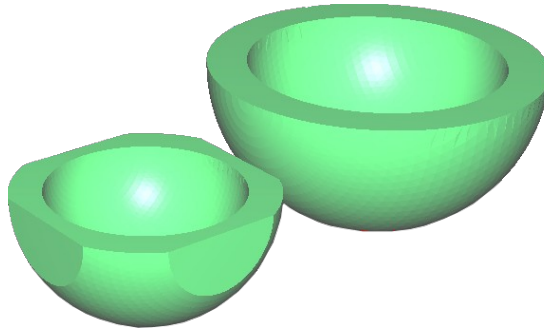
Obrázek 21: První vytvořená konstrukce

Jako u návrhu konstrukcí pro vyhodnocování předchozího typu pohybu, bude i v tomto případě vytvořena druhá konstrukce s rozdílným tvarem vnitřní kovové části stínící štítky. Návrh druhé vytvořené konstrukce je možné vidět na obrázku 22.



Obrázek 22: Návrh druhé konstrukce

Jak je z nákresu patrné, kovová část použitá ve druhé navržené konstrukci je namísto tvaru krychle, použitém v první konstrukci, ve tvaru koule. Kvůli zamezení ohybu a poškození štítků připevněných ke konstrukci byly na distanční vložce vytvořeny plošky, na které budou štítky nalepeny. Modely dílů určených pro tisk je možné vidět na obrázku 23 a vytištěnou vnitřní část opatřenou RFID štítky na obrázku 24.



Obrázek 23: Modely dílů druhé konstrukce



Obrázek 24: Vnitřní část druhé konstrukce

5.3 Algoritmus vyhodnocení rotačního pohybu

Cílem vyhodnocování rotačního pohybu předmětu se třemi stupni volnosti je získat osu, kolem které předmět rotuje a následně z celkové doby pohybu vypočítat pravděpodobný průběh úhlové dráhy. Jelikož se ani v případě předmětu rotujícího kolem pevně dané osy nepovedlo převádět rotaci předmětu do aplikace v reálném čase, bude v případě předmětu rotujícího kolem libovolné osy od začátku snahou pohyb předmětu vyhodnotit a posléze převést do aplikace z dat získaných během celé doby trvání pohybu. Vyhodnocování bude vycházet z předpokladu, že hráč pouze uvede předmět do pohybu a dále nebude předmět během doby pohybu nijak ovlivňovat. V aplikaci bude následně rozpoříván virtuální předmět, který se bude pohybovat ve stejném směru, jako by se pohyboval skutečný předmět, pokud by nebyla jeho poloha v prostoru pevně daná.

Aby bylo možné zjistit směr, ve kterém by se předmět bez fixace polohy pohyboval, bude nutné nejprve vyhodnotit orientaci předmětu. Tu bude možné získat pomocí dvou bodů, které budou odpovídat štítkům, které čte první a nebo druhá anténa. Každému štítku bude přidělen bod odpovídající poloze štítku na povrchu jednotkové koule. Následně budou všechny body přiřazené jednotlivým štítkům, které jsou čtené danou anténou, sečteny. Výsledný vektor bude poté normalizován a díky tomu bude vyhodnocen bod odpovídající dané množině čtených štítků. Stejný proces bude proveden i pro štítky, které jsou čtené druhou anténou. Tímto způsobem budou získány body definující orientaci předmětu.

Následně bude potřeba získat úhel a osu rotace, které transformují body získané z daných množin štítků do poloh odpovídajících jednotlivým anténám. Nejprve bude třeba získat úhel a osu rotace, které transformují první vyhodnocený bod do polohy první antény. Úhel rotace bude vypočten pomocí vzorce

$$\cos(\Theta) = \frac{P_1 \cdot A_1}{|P_1| |A_1|} , \quad (3)$$

kde P_1 je vektor odpovídající prvnímu bodu, A_1 je vektor odpovídající pozici první antény a operátor \cdot značí skalární součin vektorů. Z vzorce vyplývá, že nejprve jsou z vektorů P_1 a A_1 vytvořeny jednotkové vektory pomocí vydělení daných vektorů jejich velikostmi $|P_1|$ a $|A_1|$. Poté je pomocí skalárního součinu získána délka průmětu jednoho vektoru do směru vektoru druhého. Jelikož jsou vektory pomocí vydělení jejich velikostmi převedeny na jednotkové, odpovídá výsledná hodnota skalárního součinu hodnotě kosinu úhlu mezi vektory. Úhel je poté vypočítán jako

$$\Theta = \arccos\left(\frac{P_1 \cdot A_1}{|P_1| |A_1|}\right) . \quad (4)$$

Následně je vypočtena osa rotace, kolem které je nutné první bod získaný ze štítků rotovat o pomocí vzorce (4) vypočtený úhel, aby z něj byl bod odpovídající poloze první antény. Tato osa odpovídá normále plochy, která prochází body P_1 , A_1 a bodem $[0, 0]$, který je středem jednotkové koule. Normálu této roviny je možné získat pomocí vzorce

$$n = P_1 \times A_1 , \quad (5)$$

kde n je potřebná normála a operátor \times značí vektorový součin, pomocí něhož je získán vektor kolmý na vektory P_1 a A_1 . Pokud leží vektory P_1 a A_1 na jedné přímce, existuje nekonečné množství vektorů na ně kolmých. V tomto případě je možné jako vektor rotace, respektive normálu plochy, na které body P_1 a A_1 leží, vybrat libovolný z těchto vektorů (v aplikaci je vybrán vektor s nulovou hodnotou souřadnice z).

Po vypočtení osy a úhlu rotace pomocí nichž je možné transformovat první bod P_1 do pozice první antény A_1 , dojde pomocí těchto hodnot k transformaci druhého bodu P_2 . Tato situace bude odpovídat stavu, kdy by byl první bod P_1 přesunut do pozice antény A_1 a přiměřeně k tomu by byl přesunut i druhý bod P_2 . Rotace bodu P_2 kolem vypočtené osy o vypočtený úhel bude provedena pomocí jednotkového kvaternionu.

Informace ohledně kvaternionů a práce s nimi byly čerpány z [4], [5] a [6]. Kvaterniony byly poprvé popsány irským matematikem Williamem Rowanem Hamiltonem a jedná se o rozšíření komplexních čísel ve formě

$$q = a + b \cdot i + c \cdot j + d \cdot k \quad , \quad (6)$$

kde a , b , c a d jsou reálná čísla a i , j a k jsou parametry splňující vlastnosti

$$\begin{aligned} i^2 = j^2 = k^2 = i \cdot j \cdot k = -1 \quad , \\ i \cdot j = k \quad , \quad j \cdot i = -k \quad , \\ j \cdot k = i \quad , \quad k \cdot j = -i \quad , \\ k \cdot i = j \quad , \quad i \cdot k = -j \quad . \end{aligned} \quad (7)$$

Kvaterniony je možné mezi sebou sčítat a také násobit. Operace sčítání mezi kvaterniony je komutativní a kvaternion vzniklý ze součtu dvou kvaternionů $q_1 = (a_1 + b_1 \cdot i + c_1 \cdot j + d_1 \cdot k)$ a $q_2 = (a_2 + b_2 \cdot i + c_2 \cdot j + d_2 \cdot k)$ je možné vyjádřit jako

$$q_1 + q_2 = (a_1 + a_2) + (b_1 + b_2) \cdot i + (c_1 + c_2) \cdot j + (d_1 + d_2) \cdot k \quad . \quad (8)$$

Násobení mezi kvaterniony komutativní není a je jej možné zapsat jako

$$\begin{aligned} q_1 \cdot q_2 = & a_1 \cdot a_2 - b_1 \cdot b_2 - c_1 \cdot c_2 - d_1 \cdot d_2 \\ & + (a_1 \cdot b_2 + b_1 \cdot a_2 + c_1 \cdot d_2 - d_1 \cdot c_2) \cdot i \\ & + (a_1 \cdot c_2 + b_1 \cdot d_2 + c_1 \cdot a_2 - d_1 \cdot b_2) \cdot j \\ & + (a_1 \cdot d_2 + b_1 \cdot c_2 + c_1 \cdot b_2 - d_1 \cdot a_2) \cdot k \end{aligned} \quad (9)$$

Rotaci vektoru v kolem dané osy o daný úhel je možné pomocí kvaternionů převést na

$$v_{rot} = q \cdot v \cdot \bar{q} \quad , \quad (10)$$

kde q je jednotkový kvaternion vypočtený z osy a úhlu rotace a \bar{q} je k němu sdružený kvaternion definovaný jako

$$\bar{q} = q_a - q_b \cdot i - q_c \cdot j - q_d \cdot k \quad . \quad (11)$$

Má-li být tedy třírozměrný vektor pomocí kvaternionů transformován do pozice odpovídající rotaci vektoru kolem dané osy o daný úhel, je nejprve třeba z osy a úhlu vypočíst jednotkový kvaternion, pomocí něž bude transformace provedena. Tento kvaternion je možné vypočíst pomocí vzorce

$$q = (\cos(\Theta/2), x \cdot \sin(\Theta/2), y \cdot \sin(\Theta/2), z \cdot \sin(\Theta/2)) \quad , \quad (12)$$

kde Θ je úhel rotace a parametry x , y a z jsou souřadnice jednotkového vektoru jdoucího ve směru osy rotace.

Transformovaný bod P_{2rot} vzniklý rotací bodu P_2 kolem vypočtené osy o vypočtený úhel je možné získat jako

$$P_{2rot} = q \cdot P_2 \cdot \bar{q} \quad . \quad (13)$$

Následně bude třeba získat úhel a osu rotace, které transformují bod P_{2rot} do pozice druhé antény A_2 . Osa rotace bude zvolena ve směru vektoru P_1 , neboť pokud by byla zvolena jiná osa, došlo by k transformaci tohoto bodu, který se již nachází ve správné pozici. Bude tedy

nutné vypočítat úhel, který při rotaci kolem osy ve směru vektoru P_1 transformuje bod P_{2rot} na bod A_2 .

Za účelem výpočtu tohoto úhlu dojde nejprve k promítnutí vektorů P_{2rot} a A_2 do roviny S procházející bodem $[0,0]$ a s normálovým vektorem ve směru vektoru P_1 . Pro získání promítnutých vektorů P'_{2rot} a \acute{A}_2 je nejprve třeba získat vzdálenost bodů P_{2rot} a A_2 od roviny S , do které mají být promítnuty. Tu lze získat pomocí skalárního součinu daného vektoru s normálou roviny S . Následně dojde k přesunutí jednotlivých bodů o jejich vzdálenost od roviny S ve směru opačném ke směru normály této roviny.

Po vypočtení promítnutých bodů P'_{2rot} a \acute{A}_2 bude třeba zjistit úhel, který tyto vektory mezi sebou svírají. Ten bude možné získat stejným způsobem jako v předchozím případě a tedy jako

$$\arccos\left(\frac{P'_{2rot} \cdot \acute{A}_2}{|P'_{2rot}| | \acute{A}_2|}\right) . \quad (14)$$

Ze zjištěné osy a úhlu rotace bude poté opět vytvořen jednotkový kvaternion.

Velkou výhodou rotace pomocí kvaternionů je jejich snadné skládání. Výsledný kvaternion kombinující rotaci bodu P_1 do pozice první antény A_1 a rotaci bodu P_{2rot} do pozice druhé antény A_2 kolem osy ve směru vektoru P_1 je možné vyjádřit jako

$$q_3 = q_1 \cdot q_2 , \quad (15)$$

kde q_1 je kvaternion první rotace a q_2 je kvaternion druhé rotace. Jelikož násobení kvaternionů není komutativní, je nutné dát pozor na pořadí jejich násobení.

Tímto způsobem bude určena orientace sférického předmětu ve chvíli vyhodnocení bodů P_1 a P_2 . Pro vyhodnocení osy rotace předmětu bude třeba nejprve zjistit souřadnice bodů, které byly vyhodnoceny po bodech P_1 a P_2 . Tyto body P_{1n} a P_{2n} budou pomocí vypočteného kvaternionu q_3 transformovány do pozic P_{1nrot} a P_{2nrot} , ve kterých by se nacházely, pokud by souřadnice bodů P_1 a P_2 byly $P_1 = A_1$ a $P_2 = A_2$. Následně bude stejným způsobem jako v předchozím případě vypočten kvaternion q , který transformuje body odpovídající pozicím antén A_1 a A_2 do pozic P_{1nrot} a P_{2nrot} . Z tohoto kvaternionu q je následně možné získat velikost úhlu jako

$$\Theta = 2 \cdot \arccos(q_a) \quad (16)$$

a následně souřadnice osy rotace pomocí vzorce

$$n = \frac{(q_b, q_c, q_d)}{\sin\left(\frac{\Theta}{2}\right)} . \quad (17)$$

Aby bylo možné tímto způsobem pohyb předmětu vyhodnocovat, bude nejprve třeba vyhodnotit, kdy se snímaný předmět pohybuje. Algoritmus vyhodnocování toho, zda k pohybu předmětu dochází či ne, bude téměř stejný jako v předchozím případě.

Pro tento účel byly vytvořeny dvě fronty, jedna pro každou anténu, a do nich byly postupně ukládány body odpovídající čteným štítkům. Ve chvíli, kdy se ve frontách nacházelo n hodnot, došlo k vyhodnocení počtu unikátních bodů. Pokud se v alespoň jedné frontě nacházely tři a více unikátních bodů, bylo vyhodnoceno, že dochází k pohybu. Jelikož bylo snímání štítků v tomto případě pohybu podstatně náchylnější na rušení, byl algoritmus vyhodnocení začátku a konce pohybu rozšířen o kontrolu úhlů mezi po sobě následujícími body. Pokud se ve frontě nacházel dostatečný počet unikátních bodů k tomu, aby byl vyhodnocen pohyb, ale součet úhlů byl dostatečně malý, předmět byl označen jako nepohybující se.

Po vyhodnocení konce pohybu bylo možné přistoupit ke zjištění pravděpodobné osy, kolem které předmět rotuje. Pro každou pozici definovanou dvěma body získanými v daném okamžiku z přečtených štítků bude pomocí způsobu popsaném výše získána osa a úhel, který transformuje tuto pozici na pozici následující. Následně budou všechny takto získané osy rotace sečteny a výsledný vektor bude následně normalizován. Díky tomu byla získána průměrná osa rotace, která bude označena za pravděpodobnou osu, kolem které předmět rotuje.

Po získání osy rotace bude následně vyhodnocen průběh úhlové dráhy. Ten bude počítán jako v předchozím případě. Nejprve došlo k výpočtu průměrného zpomalení předmětu z naměřených dat. Díky takto vypočtené hodnotě zpomalení a z celkové doby trvání pohybu byla poté vypočtena počáteční rychlost.

5.4 Testování vytvořených konstrukcí

Po implementaci algoritmu vyhodnocení pohybu kulového předmětu popsaném v kapitole 5.3 a vytvoření obou konstrukcí bylo možné přistoupit k testům a zhodnocení úspěšnosti vyhodnocení pohybu.

V případě obou navržených konstrukcí se nepodařilo pohyb rotujícího předmětu vyhodnotit dostatečně přesně, aby bylo možné výsledky tohoto vyhodnocování použít pro ovládání hry na mobilní zařízení. Důvodů neúspěchu vyhodnocování délky trvání a osy rotace pohybu bylo několik.

Prvním důvodem byla velmi nízká vzorkovací frekvence orientace snímaného předmětu. Ta se v tomto případě pohybovala kolem 5 Hz a v naprosté většině případů tedy docházelo k porušení vzorkovacího teoremu. Jelikož je však pohyb předmětu vyhodnocen až po ukončení pohybu, bylo by teoreticky možné výsledný pohyb z dat získaných během pohybu předmětu zpětně zrekonstruovat.

Dalším důvodem neúspěchu při vyhodnocování tohoto typu pohybu byl kvůli difrakci špatně definovaný stín a díky tomu relativně malá úspěšnost při stínění potřebných štítků. Díky tomu vyhodnocené pozice často neodpovídaly skutečnosti a zpětná rekonstrukce pohybu by byla

možná pouze v případě, kdyby k pohybu docházelo dostatečně dlouho a bylo by získáno dostatečné množství dat, aby bylo možné špatně vyhodnocené pozice odfiltrovat.

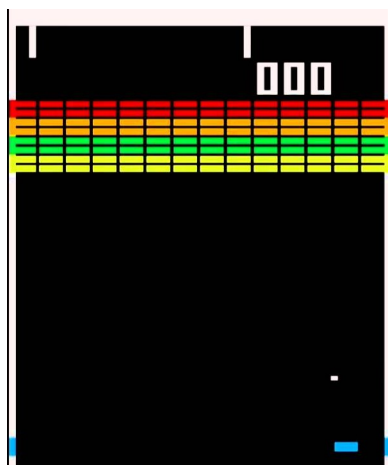
Posledním důvodem neúspěchu bylo velké tření mezi kulovým předmětem a stojanem, který zajišťoval správnou polohu snímaného předmětu vůči anténě čtečky. Díky tomuto tření, pohyb kulového předmětu trval mezi 0,5 až 1 sekundou a během této doby bylo možné vyhodnotit pouze 2 až 5 pozic, což při započítání chybně vyhodnocených pozic vlivem špatně definovaného stínu bylo nedostatečné k úspěšnému vyhodnocení potřebných parametrů. Toto tření bylo možné snížit opracováním ploch předmětu a stojanu, ale ani poté nebylo tření dostatečně malé, aby byla doba pohybu dostatečná k vyhodnocení pohybu. Použití maziva pro snížení tření by nebylo možné, neboť kulový předmět bude přicházet do přímého kontaktu s rukou hráče. Dalším možným způsobem snížení tření by bylo použití kuličkových jednotek. Jelikož se ovšem tento díl vyrábí pouze z kovových materiálů, znamenalo by jeho použití vznik dalšího rušení. Jelikož i při konstantním pohybování předmětu pomocí ruky docházelo vlivem odrazu vln od ruky k minimální přesnosti vyhodnocené pozice, bylo od použití tohoto dílu upuštěno.

6 Tvorbba her využívajících snímání pohybu pomocí RFID

V rámci této práce byly vytvořeny celkem tři různé hry využívající vyhodnocování pohybu popsané v předešlých kapitolách. Úkolem při tvorbě těchto her bylo jednak vymyslet vhodné hry, jejichž ovládání bude pomocí RFID technologie uskutečnitelné, následně hry naprogramovat a nakonec provázat jejich ovládání se systémem vyhodnocování polohy a pohybu předmětů.

6.1 Breakout

První vytvářenou hrou, která bude ovládána pomocí systému vyhodnocování polohy založeném na technologii RFID bude obdoba arkádové hry Breakout. Breakout je arkádová hra vydaná v roce 1976 společností Atari. Za návrhem této hry stáli Steve Wozniak, Nolan Bushnell a Steve Bristow [19]. Podobu originální hry je možné vidět na obrázku 25.



Obrázek 25: Původní hra Breakout [20]

Cílem v této hře je pomocí koule, která se pohybuje po herní ploše, zničit všechny obdélníkové útvary nacházející se na vrchu obrazovky a zároveň pomocí podložky zamezit propadnutí koule pod herní plochu.

Ve vytvářené době této hry bude princip hry zachován. Stejně jako v originálu bude cílem hráče zničit všechny objekty v herní ploše a zamezit propadnutí koule pod podložku. Rozdíl bude ve tvarech objektů, které bude hráčovým cílem zlikvidovat. Zatímco v původní hře jsou všechny objekty ve tvaru obdélníku, ve vytvářené podobě budou ve tvaru pravidelného n -úhelníků, kde $n=3...6$.

6.1.1. Vytvoření a vykreslení herní plochy

Prvním úkolem při vytváření hry, bylo vygenerovat herní plochu a tu následně vykreslit. V herní ploše se nacházejí celkem tři typy objektů. Prvním typem objektu je koule, která se po herní ploše pohybuje a odráží se od jejích hranic a hran dalších objektů. Dalším typem objektu je podložka, pomocí které hráč zamezuje propadnutí koule pod spodní hranici herní plochy a současně při správném nastavení pozice může změnit směr jejího pohybu. Posledním typem objektu, který se ve hře nachází, jsou n-úhelníky, které má hráč za cíl zničit. Počet hran a velikost n-úhelníku budou ve hře generovány náhodně a tudíž bylo třeba vytvořit metodu, která toto generování zajistí.

Při vytváření je nejprve vypočtena náhodná velikost generovaného tvaru. Následně je určen první bod, jehož souřadnice budou pro všechny n-úhelníky stejné a bude se jednat o souřadnice $[0, s]$, kde s je náhodná velikost tvaru. Dále bude vypočten sinus a kosinus úhlu, pomocí něhož bude první bod rotován a budou tímto způsobem vypočteny body ostatní. Nakonec bude každému takto vytvořenému tvaru přidělena náhodná barva, která bude použita při vykreslování.

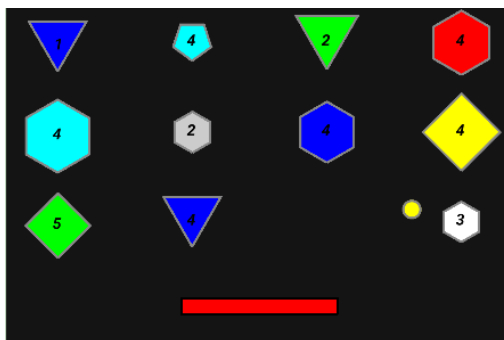
V případě podložky, kterou hráč ovládá, byly jednotlivé body n-úhelníku definovány takovým způsobem, aby podložka byla ve tvaru obdélníku.

Poslední objekt, který bylo třeba vytvořit, byla koule. Ta je definována svou pozicí v herní scéně a také svým poloměrem.

Před vykreslováním koule a podložky byly z důvodu optimalizace nejprve vytvořeny obrazy těchto tvarů a ty byly následně uloženy, aby je nebylo nutné generovat při každém překreslování herní plochy. V případě n-úhelníků byl nejprve vytvořen snímek celé herní plochy a do tohoto snímku byly následně obrazy jednotlivých n-úhelníků vykresleny. V případě, že byl n-úhelník zničen, došlo pouze k vyplnění obdélníku ohraničující daný tvar barvou pozadí, čímž byl zničený n-úhelník odstraněn ze snímku. Tento způsob vykreslování herních objektů byl zvolen z toho důvodu, protože po herní ploše se bude pohybovat pouze podložka a koule a vykreslování ostatních tvarů při každém překreslování by bylo zbytečné.

Vykreslování, současně s vyhodnocováním kolizí probíhá v samostatném vlákně. Aby bylo možné vykreslování provádět v jiném vlákně než hlavním, bylo využito komponenty *TextureView*, která je součástí vývojové sady Android Studio. Při jejím použití je před vykreslováním nutné nejprve zavolat metodu *lockCanvas*, která vrátí instanci třídy *Canvas*, pomocí které je následně možné potřebné objekty vykreslit. Zároveň při volání metody *lockCanvas* dojde k uzamčení plátna, aby nedocházelo k editaci bitmapy z většího množství vláken. Pro zobrazení výsledného snímku na obrazovce je nutné nakonec zavolat metodu *unlockCanvasAndPost*.

Výsledný vzhled herní plochy je možné vidět na obrázku 26.

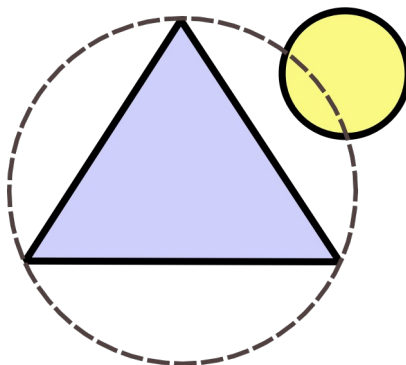


Obrázek 26: Vzhled hry Breakout

6.1.2. Vyhodnocování kolizí

Aby bylo možné hru hrát, bude nutné docílit toho, aby se koule odrážela od ostatních objektů v herní scéně. Z tohoto důvodu bude nutné vytvořit systém pro detekci kolizí. Pro tento účel je možné použít velké množství různých knihoven. Příkladem takovýchto knihoven je například dyn4j. Jelikož ale bude v této hře jediným typem kolize, kolize mezi n-úhelníkem a koulí respektive kruhem, který je relativně jednoduchý na implementaci, bude systém na vyhodnocování kolizí v rámci této práce celý naprogramován. Velkou výhodou bude pochopení základů vyhodnocování kolizí mezi dvourozměrnými objekty a také naprostá kontrola nad celým systémem a možnost jeho úprav. Informace v této kapitole byly čerpány z [7], [8] a [9].

Při vyhodnocování kolizí mezi n-úhelníkem a koulí dojde nejprve ke kontrole, zda se oba objekty nacházejí dostatečně blízko sebe, aby ke kolizi mezi nimi mohlo docházet. Za tímto účelem bude třeba vytvořit výpočet poloměru kružnice, která definuje nejmenší vzdálenost, ve které ještě mezi koulí a n-úhelníkem ke kolizi dojít nemůže. Jelikož je vzdálenost všech bodů od středu u vytvářených n-úhelníků i podložky stejná, je výpočet tohoto poloměru velmi jednoduchý. V tomto případě stačí vypočítat vzdálenost libovolného bodu od středu daného tvaru. Pokud by ale vzdálenost jednotlivých bodů od středu tvaru stejná nebyla, byl by výpočet buď podstatně náročnější a nebo by vypočtený poloměr byl větší, než by bylo nutné a v takovém případě by v aplikaci docházelo ke zbytečnému vyhodnocování kolizí u objektů, které by byly příliš daleko od sebe, než aby ke kolizi mezi nimi mohlo dojít.



Obrázek 27: Ohraničení n-úhelníku kružnicí

Jak je ale možné vidět na obrázku 27, to že je vzdálenost středu n -úhelníku a koule menší než součet poloměru kružnice ohraničující n -úhelník a poloměru koule, ještě nutně nemusí znamenat, že ke kolizi mezi tvary dochází. Aby bylo možné vyhodnotit, zda ke kolizi skutečně dochází, bude tedy nutné n -úhelníky, které se nacházejí dostatečně blízko koule, dále testovat.

V další fázi vyhodnocování kolizí dojde ke kontrole, zda se některý z vrcholů n -úhelníku nenachází uvnitř kružnice definující hranici koule. Pokud by bylo jediným cílem vyhodnotit, zda ke kolizi dochází či ne, bylo by možné, v případě, že je některý z vrcholů n -úhelníku uvnitř koule, po jeho identifikaci další vyhodnocování přerušit. Aby však bylo zaručeno, že před přemístěním koule do nové polohy v závislosti na její rychlosti ke kolizi mezi koulí a n -úhelníkem nedochází, dojde v případě detekce bodů uvnitř koule k výpočtu vzdálenosti mezi středem koule a bodem, který se nachází středu koule nejbližší a současně náleží některé z hran n -úhelníku. Pomocí tohoto bodu bude určena vzdálenost a směr, ve kterém je nutné kouli přemístit. Nejbližší bod přímkou ke středu kružnice je možné získat pomocí vzorců

$$\begin{aligned} dC &= C - P_1, \\ dP &= P_2 - P_1, \\ d &= \frac{dC \cdot dP}{dP \cdot dP}, \\ P &= P_1 + d \cdot dP. \end{aligned} \tag{18}$$

kde dC je vektor vzdálenosti středu kružnice C od prvního bodu přímky P_1 , dP je vektor vzdálenosti prvního bodu přímky P_1 od druhého bodu P_2 , d je poměr mezi velikostí vektoru vzniklého jako průmět vektoru dC do směru vektoru dP s velikostí vektoru dP a P je hledaný nejbližší bod na přímce definované body P_1 a P_2 ke středu kružnice C .

Po získání tohoto bodu je následně nutné zjistit, zda skutečně náleží dané hraně a nachází se mezi vrcholy P_1 a P_2 . Pokud ano je za nejbližší bod ke středu kružnice označen takto vypočtený bod P . V případě, že tomu takto není, je za nejbližší bod ke středu kružnice označen bližší z bodů P_1 a P_2 .

Tento proces bude proveden pro všechny hrany, jejichž alespoň jeden vrchol se nachází uvnitř kružnice definující hranici koule a tímto způsobem bude získán bod n -úhelníku, který je středu této kružnice nejbližší. Následně bude vypočtena vzdálenost, o kterou je nutné kouli posunout, aby s n -úhelníkem již nadále nekolidovala a směr tohoto posunu. Velikost posunu lze vypočítat jako $r - l$, kde r je poloměr kružnice a l je vzdálenost nejbližšího bodu n -úhelníku P od středu kružnice C a směr posunu lze získat jako $C - P$.

V případě, že se ve vnitřku kružnice udávající hranice koule nenachází žádný z vrcholů n -úhelníku, dojde stejným způsobem, jako tomu bylo v předešlém případě, k identifikaci nejbližších bodů jednotlivých hran n -úhelníku ke středu kružnice a následně bude provedena kontrola, zda se daný bod nachází mezi vrcholy definujícími danou hranu a zda je blíže ke středu kružnice než je její poloměr. Pokud nejbližší bod dané hrany splňuje tyto podmínky je možné opět vypočítat vzdálenost a směr posunu koule a vyhodnocování kolizí s dalšími hranami n -úhelníku přerušit. Přerušeni výpočtu kolizí pro další hrany daného n -úhelníku je možné z toho důvodu, že v okamžiku tohoto vyhodnocování může koule kolidovat už jen s jednou hranou n -úhelníku. To je zajištěno tím, že velikosti všech n -úhelníků jsou voleny a rychlost

koule limitována takovým způsobem, aby i při maximálním posunu koule nedocházelo k průchodu skrz n-úhelník. V případě, že by docházelo ke kolizi koule s více hranami n-úhelníku, došlo by tedy k zachycení těchto kolizí již v předešlém kroku výpočtu, neboť by se alespoň jeden z vrcholů nacházel uvnitř kružnice.

Po vyhodnocení kolizí a získání všech hran, se kterými koule koliduje, je nakonec z normál jednotlivých hran vypočten vektor definující směr odrazu koule.

6.1.3. Reakce na vyhodnocené kolize

Po dokončení systému pro detekci kolizí mezi koulí a ostatními objekty v herní scéně bylo třeba vytvořit program, který v závislosti na vyhodnocených kolizích patřičně změní rychlost a směr pohybu koule.

Před aktualizací hodnoty rychlosti koule bylo nutné postupně vyhodnotit kolize mezi ní a všemi objekty, od kterých se koule má odrážet. Z výstupu kolizního systému byl v případě, že ke kolizi mezi koulí a daným objektem došlo, získán seznam parametrů obsahující vektor udávající směr odrazu, vektor, pomocí kterého je možné kouli přesunout do pozice, kde již nebude s daným n-úhelníkem kolidovat, rychlost n-úhelníku a údaje o tom, zda má být rychlost koule ovlivněna třecí silou.

Po vyhodnocení kolizí a získání všech seznamů parametrů odpovídajících těmto kolizím bylo možné přistoupit k aktualizaci rychlosti koule. V případě, že koule kolidovala s pohybující se podložkou, byla nejprve vypočtena rychlost, která má být k rychlosti odražené koule přičtena. Tato rychlost byla počítána pomocí rychlosti podložky a vektoru odrazu. V případě, že koule kolidovala pouze s jednou hranou podložky, došlo k výpočtu změny rychlosti simulující tření mezi pohybující se podložkou a koulí. Velikost této změny rychlosti byla ovlivněna rychlostí podložky a absolutní hodnotou normály dané hrany.

Následně je vypočítán směr pohybu koule po odrazu od n-úhelníků. Ten je počítán podle vzorce

$$V_n = V - 2n(V \cdot n) \quad (19)$$

kde V_n je rychlost koule po odrazu, V je rychlost koule před odrazem a n je jednotkový vektor udávající směr odrazu, který je vypočítán z normál hran, se kterými koule koliduje. Pokud koule koliduje pouze s jednou hranou, je tento vektor roven normále dané hrany. Pokud je hran, se kterými koule koliduje, více, je tento vektor vypočítán jako normalizovaný součet normál jednotlivých hran. Ve své podstatě se jedná o osu, pomocí které je původní rychlost zrcadlena do nového směru.

Nakonec je možné přistoupit k výpočtu konečné rychlosti koule, kterou je možné získat jako

$$V_f = V_n + V_{add} \quad (20)$$

kde V_f je konečná rychlost koule, V_n je rychlost koule vzniklá odrazem a V_{add} je rychlost způsobená pohybující se podložkou.

Pokud dojde ke kolizi koule s hranicí herní plochy, dojde pouze k převrácení rychlosti v daném směru. Výjimkou je dolní hranice herní plochy, u které dojde k vyhodnocení propadnutí koule, na což hra adekvátně reaguje.

6.1.4. Provázání hry se systémem vyhodnocení polohy

Po vytvoření částí aplikace zajišťující vykreslování herní scény, detekci kolizí mezi koulí a ostatními herními objekty a reakci na vyhodnocené kolize, bylo třeba přistoupit k provázání těchto částí aplikace se systémem pro vyhodnocování polohy posuvného segmentu využívající technologii RFID, jehož návrh a tvorba byla popsána v kapitole 3 a dovolit hráči pomocí tohoto systému hru ovládat.

Jelikož je hra i systém pro vyhodnocení polohy rozdělen do jednotlivých objektů, jejichž kód je vykonáván v samostatných vláknech, bylo hlavním úkolem při provázání obou částí aplikace docílit bezpečného předávání informací mezi těmito objekty. Toho je docíleno pomocí návrhového vzoru pozorovatel a synchronizací příslušných metod pro předání dat.

Nejprve dojde k nastartování vlákna starajícího se o zpracování hodnot získaných ze čtečky a vlákna zajišťujícího vykreslování herní scény a detekci a reakci na kolize herních objektů. Vlákno pro zpracování hodnot ze čtečky následně čeká, dokud hodnoty ze čtečky neobdrží. Po jejich získání dochází pomocí algoritmu popsaném v kapitole 3.4 k vyhodnocení aktuální polohy posuvného segmentu. Tato poloha je následně pomocí definované metody předána do vlákna starajícího se o vykreslování herní scény a správu kolizí. Zde je v závislosti na vyhodnocené poloze aktualizována rychlost podložky. Pokud je vyhodnocená poloha stejná jako je poloha podložky v aplikaci, je rychlost nastavena na nulovou hodnotu. Pokud se vyhodnocená poloha od polohy podložky liší, je podložce nastavena taková rychlost, aby se postupně k vyhodnocené poloze přibližovala.

Aby při sdílení hodnot mezi vlákny nedocházelo k chybovým stavům vyplývajícím ze souběžného přístupu ke sdíleným prostředkům, bylo třeba metody pro předání dat mezi objekty synchronizovat. Synchronizace zajišťuje to, aby kód, ve kterém se pracuje se sdílenými prostředky, byl současně vykonáván pouze jedním vláknem. Díky tomu je zajištěno, že při předání nové polohy do vlákna spravujícího kolize dojde k správnému zpracování této hodnoty.

6.2 Spoj 4

Další hrou, která byla v rámci této práce naprogramována, byla elektronická verze známé deskové hry Spoj 4. Cílem hráče v této hře je vytvořit nepřerušenou linii (vertikální, horizontální nebo diagonální) čtyř svých žetonů dříve, než se to samé podaří protihráči. Žetony mohou hráči pokládat vždy do jednoho ze sedmi sloupců o šesti polích. Hráč nevybírá přímo pozici žetonu, ale vybírá pouze sloupec, do kterého má žeton být umístěn. Žeton při položení propadá až na dno vybraného sloupce. Hra končí ve chvíli, kdy jeden z hráčů vytvoří nepřerušenou linii čtyř svých žetonů, nebo ve chvíli, kdy je celá herní plocha zaplněna žetony obou hráčů. Deskovou verzi této hry je možné vidět na obrázku 28.



Obrázek 28: Desková verze hry Spoj 4 [21]

Cílem bude v rámci této práce vytvořit aplikaci, která bude graficky reprezentovat tuto hru, dokáže na základě implementovaného algoritmu vyhodnotit dostatečně přesně pozici položení žetonu, která vede k výhře a dovolí hráči pomocí systému vyhodnocování dotyku založeném na technologii RFID popsaném v kapitole 3 položit svůj žeton.

6.2.1. Vytvoření a vykreslení herní plochy

I v tomto případě bylo nejprve nutné vytvořit tu část aplikace starající se o vykreslování herní plochy. Zatímco v případě hry Breakout se v herní scéně nacházely tři typy objektů, které se vzájemně ovlivňovaly a dva z nich (koule a podložka) se po ploše pohybovaly a bylo tedy nutné herní scénu neustále ve smyčce překreslovat, situace v případě této hry bude jiná. Překreslení plochy bude nutné pouze v případě, když bude do jednoho ze sedmi sloupců umístěn nový žeton.

Prvním úkolem při vykreslování této hry bylo z hodnot proměnných reprezentujících herní plochu, které jsou vhodné pro vyhodnocení správné pozice k položení žetonu, vytvořit grafickou reprezentaci vhodnou pro hráče. Aktuální stav herní plochy je z důvodu optimalizace vyhodnocení správné pozice zakódován do proměnné typu long. Před položení nového žetonu

je třeba z proměnné reprezentující aktuální stav plochy vyhodnotit, zda je do zvoleného sloupce možné žeton položit a v případě že ano, vyhodnotit pozici ve sloupci, do které by žeton propadl, pokud by se jednalo o hru deskovou. Do této pozice bude následně obrázek žetonu nakreslen.

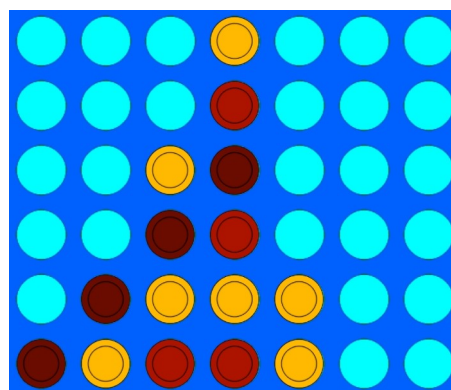
Aby byl princip hry zřejmý i hráči, který se s touto hrou dříve nesetkal, bylo snahou přiblížit vizuální podobu verze vznikající v rámci této práce co nejlépe její deskové podobě. Pokládání žetonů do sloupce je animované, díky čemuž bude pochopení principu hry podstatně jednodušší než v případě, kdy by se nově položený žeton objevil přímo ve své nové pozici.

Z tohoto důvodu bylo třeba herní plochu opět vykreslovat v samostatném vlákně, neboť pokud by vykreslování pádu žetonu do své pozice probíhalo v hlavním vlákně aplikace, došlo by během tohoto vykreslování k zastavení zpracovávání uživatelských vstupů a aplikace by se stala neresponzivní.

Kvůli animaci pádu žetonu bylo současně potřeba při každém posunutí padajícího žetonu překreslovat celou herní plochu. Aby nebylo nutné při každém překreslení herní plochy kreslit znovu všechny položené žetony, byl obrázek každého žetonu, který se nacházel ve své konečné pozici, stejně jako tomu bylo v případě n-úhelníků ve hře Breakout, nakreslen přímo do obrázku herní plochy.

Dále bylo při vykreslování této hry potřeba vykreslit do obrázku herní plochy rám, do kterého jsou v deskové verzi žetony umísťovány. Aby bylo možné současně při zobrazení rámu vykreslovat animaci pádu žetonu, bylo nutné vytvořit jednotlivá pole obrázku rámu průhledná.

Nakonec bylo vytvořeno zvýraznění výherní pozice. To je realizováno pomocí vykreslení poloprůhledných černých kruhů do pozic čtyř žetonů stejné barvy tvořící nepřerušovanou linii. Výslednou podobu hry s vyhodnocenou výhrou je možné vidět na obrázku 29.



Obrázek 29: Vzhled hry Spoj 4

6.2.2. Vyhodnocení optimální pozice pro položení žetonu

Po dokončení vykreslování hry bylo přistoupeno k vytvoření systému, který dokáže s dostatečnou přesností vyhodnotit pozici pro nově položený žeton, která povede k výhře. Toto vyhodnocování bude následně použito pro vytvoření virtuálního protihráče, což odstraní

nutnost hrát hru ve dvou. Cílem v rámci této práce nebylo vytvořit systém, který vždy zjistí optimální pozici nově položeného žetonu, ale vytvořit systém takový, který bude průměrnému hráči důstojným protivníkem. Informace ohledně použitých algoritmů, jejich optimalizaci a o vhodném zakódování herní pozice pro co nejrychlejší vyhodnocení vhodného tahu byly čerpány z práce [10].

Základním algoritmem použitým pro získání co nejlepšího tahu v dané pozici byl algoritmus minimax. Tento algoritmus je založen na rekurzivním procházení stromu vznikajících pozic a v závislosti na aktuálním hráči hledá takový tah, který vede k maximalizaci nebo minimalizaci hodnoty konfigurace položených žetonů. Velikost stromové struktury vznikajících pozic roste s hloubkou rekurze exponenciálně, neboť pro každou konfiguraci, kromě těch, kde se žetony nachází v nejvyšším řádku herní plochy, existuje sedm dalších konfigurací, které vzniknou učiněním nového tahu. Celkově existuje 4531985219092 možných koncových pozic [10] a díky tomu je prakticky nemožné v počátečních fázích hry prohledat všechny možné pozice a zjistit tímto způsobem optimální tah. Z tohoto důvodu bude nutné hloubku rekurzivního hledání ideálního tahu omezit. Pseudokód použitého hloubkou omezeného algoritmu minimax vypadá následovně

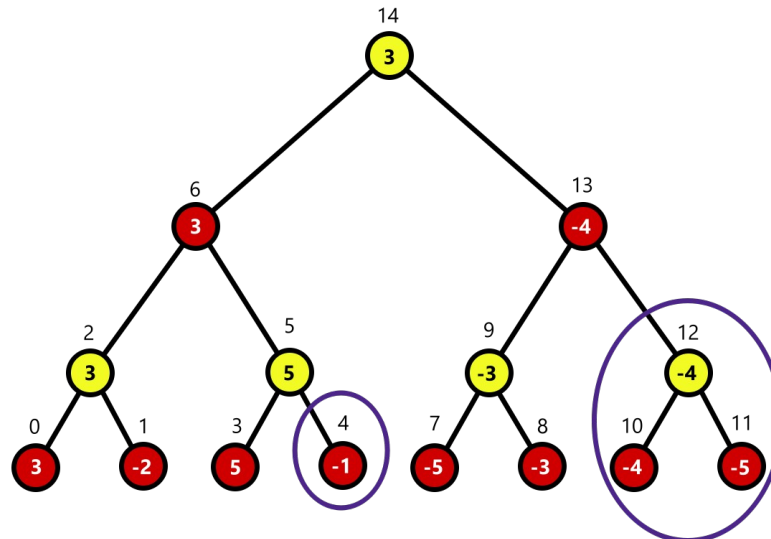
```

minimax(maximalizuj, pozice, hloubka)
  pokud hloubka == 0 nebo pozice je konečná potom
    vrať heuristickou hodnotu pozice
  pokud maximalizuj potom
    hodnota = nejnižší možná hodnota
    pro každý možný tah udělej
      nová pozice = pozice po provedení tahu
      nová hodnota = minimax(NE, nová pozice, hloubka - 1)
      pokud nová hodnota > hodnota potom
        hodnota = nová hodnota
    vrať hodnotu
  jinak
    hodnota = nejvyšší možná hodnota
    pro každý možný tah udělej
      nová pozice = pozice po provedení tahu
      nová hodnota = minimax(ANO, nová pozice, hloubka - 1)
      pokud nová hodnota < hodnota potom
        hodnota = nová hodnota
  vrať hodnotu

```

Kromě snížení doby trvání má omezení hloubky rekurze ovšem také negativní vliv na přesnost vyhodnocení optimálního tahu. Z tohoto důvodu bude cílem ponechat hloubku rekurze co největší a délku trvání vyhodnocování snížit optimalizací algoritmu.

Prvním možným způsobem optimalizace algoritmu minimax je takzvané alfa-beta ořezání. Při prohledávání stromu všech možných pozic dochází při použití neoptimalizovaného algoritmu minimax ke zbytečnému prohledávání některých větví. Tyto větve je možné při výpočtu vynechat a tím dojít ke stejnému výsledku v nižším čase. Tuto situaci je možné vidět na obrázku 30, ve kterém je zobrazen strom pro vyhodnocení tahu v případě, že je možné v každé pozici učinit tahy dva.



Obrázek 30: Princip algoritmu minimax a alfa-beta ořezání

Na tomto obrázku je žlutě označena maximalizující strana, červeně minimalizující strana, čísla uvnitř uzlů značí hodnotu daného uzlu, čísla nad jednotlivými uzly značí pořadí, ve kterém jsou určeny hodnoty daných uzlů a modře jsou zvýrazněny větve, jejichž hodnota byla určována zbytečně a bylo možné je ořezat.

Algoritmus nejprve dojde do konečného uzlu, v případě situace zobrazené na obrázku 30 se jedná o uzel s číslem 0 a pomocí heuristické funkce zjistí hodnotu aktuální pozice. Následně dojde k návratu o úroveň výše, v tomto případě do uzlu číslo 2 a je učiněn druhý ze dvou možných tahů. Pro vzniklou pozici je stejným způsobem jako v případě uzlu 0 zjištěna hodnota pozice. V uzlu číslo 2 následně dojde k porovnání hodnot z uzlů nacházejících se pod ním, tedy uzlů 0 a 1 a jelikož je na řadě maximalizující strana, je vybrána nejvyšší hodnota těchto uzlů. Následně dojde k návratu do uzlu číslo 6 a stejným způsobem jako u uzlu číslo 2 je určena hodnota uzlu 5. V případě uzlu 4 je však hodnota určována zbytečně a výpočet jeho hodnoty je možné vynechat. To je možné z toho důvodu, že uzel číslo 5, který se nachází nad ním, je uzel maximalizující strany a hodnota uzlu 4 by byla vybrána pouze v případě, že by byla větší než je hodnota dříve ohodnoceného uzlu 3. Minimální hodnota uzlu 5 před ohodnocením uzlu 4 je tedy v případě situace zobrazené na obrázku 30 větší, než je výsledná hodnota uzlu 2 a jelikož je uzel 6, pod kterým se uzly 2 a 5 nachází, minimalizující, bude vždy vybrána nižší hodnota z hodnot uzlů pod ním a i v případě, kdy by byla hodnota uzlu 4 vyšší než hodnota uzlu 3 a byla tedy vybrána uzlem 5, nebyla by vybrána uzlem 6.

Stejnou situaci je možné pozorovat v případě větve složené z uzlů číslo 10, 11 a 12. Ve chvíli, kdy je určována hodnota této větve, je již známá hodnota větve s vrcholem v uzlu 6 a také hodnota větve s vrcholem v uzlu 9. Hodnota uzlu 12 by byla uzlem 13 vybrána pouze v případě, pokud by byla nižší než je hodnota uzlu 9. Jelikož je ale už hodnota uzlu 9 nižší než je hodnota uzlu 6, nebyla by maximalizujícím uzlem 14 vybrána ani ještě nižší hodnota uzlu 12.

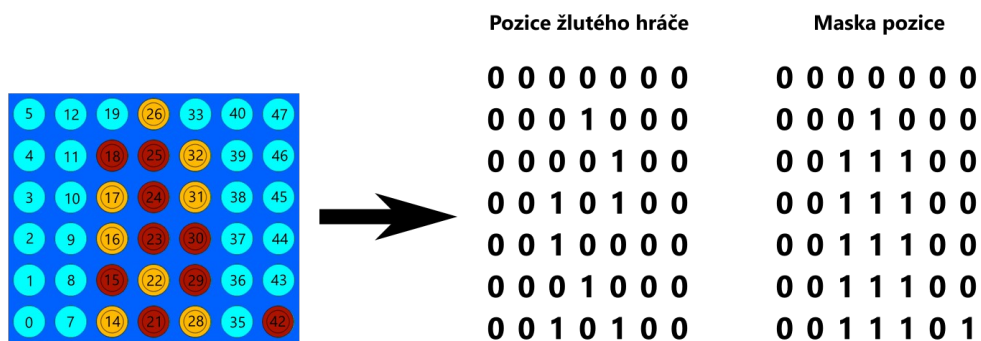
Pseudokód algoritmu minimax s alfa-beta ořezáním je

```

minimax(maximalizuj, pozice, hloubka, alfa, beta)
  pokud hloubka == 0 nebo pozice je konečná potom
    vrať heuristickou hodnotu pozice
  pokud maximalizuj potom
    hodnota = nejnižší možná hodnota
    pro každý možný tah udělej
      nová pozice = pozice po provedení tahu
      hodnota = minimax(NE, nová pozice, hloubka - 1, alfa, beta)
      pokud hodnota > alfa potom
        alfa = hodnota
      pokud alfa >= beta potom
        vrať hodnotu
    vrať hodnotu
  jinak
    hodnota = nejvyšší možná hodnota
    pro každý možný tah udělej
      nová pozice = pozice po provedení tahu
      hodnota = minimax(ANO, nová pozice, hloubka - 1, alfa, beta)
      pokud hodnota < beta potom
        beta = hodnota
      pokud alfa >= beta potom
        vrať hodnotu
  vrať hodnotu
  
```

Aby alfa-beta ořezání mělo co největší vliv a došlo k co největšímu zrychlení vyhodnocení tahu, je možné pomocí heuristických pravidel seřadit jednotlivé tahy od pravděpodobně nejlepšího po pravděpodobně nejhorší. Pokud tato pravidla budou vytvořena dobře, dojde k ořezání velkého množství větví, čímž se sníží množství iterací a čas potřebný k dokončení funkce.

Aby bylo řazení jednotlivých tahů dostatečně rychlé a přispělo k rychlosti dokončení funkce minimax, bude třeba pozice ukládat dostatečně efektivním způsobem. Z tohoto důvodu bude každá pozice jednoznačně zakódována do dvou 64 bitových proměnných typu long. V první proměnné se budou nacházet žetony aktuálně hrajícího hráče a bude pomocí ní zjišťováno, zda došlo k výhře. V druhé proměnné se bude nacházet maska dané pozice, tedy všechny žetony položené během hry. Zakódování pozice do proměnných je možné vidět na obrázku 31.



Obrázek 31: Zakódování herní pozice

Čísla uvnitř jednotlivých polí značí číslo bitu v proměnné, do kterého bude pole zakódováno. Pozici druhého hráče je možné získat pomocí provedení bitové operace xor mezi pozicí prvního hráče a masky. Aby bylo možné pomocí bitových operací zjistit, zda je daná pozice výherní, je v proměnných definujících aktuální stav hry každý sedmý bit nulový (v obrázku 31 se jedná o vrchní řádek nul) a díky tomu jsou jednotlivé sloupce uložené do proměnné od sebe odděleny. Při vyhodnocování výhry a určování hodnoty pozice dochází k bitovým posunům proměnné pozice aktuálního hráče a následnému vymaskování bitů ležících vedle sebe v daném směru (horizontálním, vertikálním a obou diagonálních) pomocí bitové operace and. Výhra je vyhodnocena v případě, že po provedených bitových operacích je výsledná hodnota nenulová. V závislosti na velikosti a počtu bitových posunů lze určit různé délky nepřerušovaných linií žetonů daného hráče a pomocí toho určit hodnotu pozice v případě, že pozice není konečná (nedošlo k výhře jednoho z hráčů ani k remíze).

Posledním v rámci této práce použitým způsobem optimalizace algoritmu bylo ukládání již vyhodnocených pozic do transpoziční tabulky. V případě, kdy byla pomocí jiného sledu tahů získána pozice, jejíž hodnota se již nacházela v tabulce, nebylo nutné její hodnotu získávat časově náročným iteračním způsobem.

Při použití všech výše zmíněných způsobů optimalizace bylo možné nastavit hloubku rekurze na 17. Výsledný algoritmus je tedy schopen vyhodnotit nejlepší možný tah, pokud je počet tahů obou hráčů vedoucích k výhře jednoho z nich menší nebo roven 17, respektive je algoritmus schopen vyhodnotit výhru v devíti tazích aktuálně hrajícího hráče.

Dalším možným způsobem optimalizace by bylo použití tabulky otevíracích tahů. Do té jsou uloženy nejlepší tahy pro dané počáteční pozice a v začátcích hry není díky tomu nutné vybírat tah pomocí rekursivního algoritmu. Jelikož je ale zamýšleným protivníkem v rámci této práce vytvářené umělé inteligence člověk, nebylo nutné tuto tabulku vytvářet.

6.2.3. Provázání hry se systémem vyhodnocení polohy

Po dokončení částí aplikace pro vykreslování a vyhodnocování vhodného tahu bylo možné opět přistoupit k provázání těchto částí se systémem pro vyhodnocení dotyku hráče popsaném v kapitole 3.

V případě vyhodnocení dotyku dojde nejprve k uložení hráčova tahu do proměnných pozice a maska. Následně jsou vytvořena dvě samostatná vlákna. První vlákno má na starost animaci pádu žetonu do své pozice, ve druhém vlákne probíhá vyhodnocení vhodného tahu. Během animace a vyhodnocování tahu je hráči zakázáno pokládat další žetony. Po vyhodnocení tahu dojde opět k jeho animovanému provedení a poté je hráči pokládání nových žetonů znovu povoleno.

Aby byly jednotlivé tahy provedeny správně, byl přístup ke sdíleným prostředkům (proměnným pozice a maska) prováděn v synchronizovaných funkcích.

6.3 Curling

Poslední vytvářenou hrou využívající systém pro vyhodnocování pohybu pomocí RFID technologie byla počítačová verze hry Curling. Původním záměrem při tvorbě této hry bylo ke směrování hráčova kamenu využít systém vyhodnocující pohyb sférického předmětu. Jelikož ale byl vytvořený systém při vyhodnocování neúspěšný, bylo nutné hru upravit. Místo systému vyhodnocujícího pohyb sférického předmětu, byl pro ovládání hry použit systém pro vyhodnocení pohybu předmětu válcového.

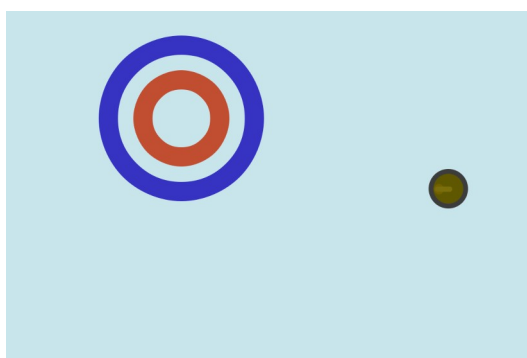
6.3.1. Vytvoření a vykreslení herní plochy

Stejně jako tomu bylo u předchozích her, byl i v případě této hry nejprve vytvořen program pro vykreslení herní plochy a jednotlivých herních objektů. Herní plocha této hry je ze všech her vzniklých v rámci této práce nejjednodušší a sestává se pouze z pozadí a hráčova kamenu.

Nejprve došlo k vytvoření obrázku pozadí. Do tohoto obrázku je při vytváření vykreslen cíl, ke kterému bude muset hráč dostatečně blízko přiblížit svůj kámen. Místo cíle je voleno náhodně v zadaných mezích. Tyto meze jsou definovány takovým způsobem, aby výsledná pozice cíle nebyla příliš blízko počáteční pozici hráčova kamenu a zároveň, aby se celý obrázek cíle nacházel v zobrazeném obrázku herní plochy.

Po dokončení metod pro vytvoření pozadí herní plochy bylo nutné vytvořit vykreslování pohybujícího se hráčova kamenu. Toto vykreslování funguje stejným způsobem jako vykreslování padajících žetonů v rámci této práce vytvořené hře Spoj 4. Po vyhodnocení pohybu pomocí systému popsaném v kapitole 4 dojde k vytvoření nového vlákna, které začne v závislosti na vyhodnoceném průběhu dráhy přemísťovat hráčův kámen a překreslovat herní plochu. Ve chvíli, kdy je rychlost kamenu nulová, dojde k zastavení vlákna a je vyhodnoceno, zda je kámen dostatečně blízko cíli. Pokud ano, dojde k vyhodnocení výhry, pokud ne, aplikace čeká na vyhodnocení nového průběhu pohybu, po němž dojde k opětovnému rozpohybování hráčova kamenu.

Podobu vytvořené hry je možné vidět na obrázku 32.



Obrázek 32: Vzhled hry Curling

6.3.2. Určování rychlosti a směru pohybu kamenu

Dalším úkolem při vytváření této hry bylo rozpohybovat hráčův kámen ve směru a s rychlostí určenou pomocí vyhodnocení pohybu válcového předmětu. Velikost posunu v daném čase je možné získat pomocí vzorce

$$dS = a \cdot t \cdot dt + v_0 \cdot dt, \quad (21)$$

kde dS je změna polohy, a je zpomalení, t je čas od začátku pohybu, dt je rozdíl mezi současným a předchozím časem výpočtu posunu a v_0 je získaná počáteční rychlost.

Hodnota zpomalení a byla zjištěna z provedených měření a byla stanovena na dvě otáčky za sekundu. Tuto hodnotu je pro použití v aplikaci nutné přepočítat na zpomalení v pixelech, ze kterých se herní plocha sestává. Hodnotu zpomalení v pixelech je možné vyjádřit jako $a \cdot \pi \cdot d$, kde d je průměr kruhu, respektive kamenu hráče.

Jak již bylo řečeno, jelikož bylo vyhodnocení pohybu sférického předmětu neúspěšné, bude směr pohybu hráčova kamenu určován pomocí vyhodnocení pohybu předmětu válcového. Jelikož je ale osa rotace tohoto předmětu pevná, bude pomocí vyhodnocování určeno pouze to, zda se bude kámen ve hře pohybovat směrem k cíli nebo od něj. Směrový vektor pohybu bude v aplikaci vyhodnocen jako $P_c - P_k$, kde P_c je pozice cíle a P_k je pozice kamenu.

6.3.3. Provázání hry se systémem vyhodnocení pohybu

Při převodu pohybu válcového předmětu do prostředí hry dojde nejprve k vyhodnocení směru a délky trvání pohybu. Z délky trvání pohybu a ze změřeného zpomalení bude následně vypočtena počáteční rychlost hráčova kamenu a následně dojde k nastartování vlákna, které začne pomocí vzorce 21 přesouvat kámen po herní ploše.

Během doby, po kterou se bude kámen v aplikaci pohybovat, bude přerušeno vyhodnocování pohybu válcového předmětu, čímž bude hráči znemožněno pohyb předmětu do jeho ukončení ovlivňovat.

V případě, že se bude kámen nacházet dostatečně blízko cíli, dojde k vyhodnocení konce hry. V opačném případě bude vyhodnocování pohybu opět odstartováno, čímž bude hráči umožněn nový pokus o přesun kamenu k cíli.

7 Závěr

Hlavním cílem této diplomové práce bylo vytvoření systémů pro snímání polohy a pohybu založených na technologii RFID a jejich následné použití k ovládání her. V rámci této práce byly vytvořeny tři různé systémy z nichž dva byly pro ovládání vytvořených her úspěšně použity.

Prvním vytvořeným systémem byl systém určující polohu předmětu pohybujícího se po ploše z RFID štítků. Při jeho návrhu a tvorbě byly navrženy a následně na 3D tiskárně vytisknuty jednotlivé části prvků, které byly ke snímání použity. Dále byl vytvořen systém pro softwarovou filtraci hodnot získaných ze čtečky, která byla nutná k odstranění vznikajícího šumu. Výsledný systém byl následně úspěšně použit pro ovládání v průběhu této práce vytvořených her Breakout a Spoj 4.

Druhým systémem, který byl během této práce vytvořen, byl systém pro snímání pohybu předmětu rotujícího kolem jedné pevně dané osy. Stejně jako v případě systému předchozího došlo nejprve k návrhu celého systému a následnému vytvoření modelů a vytištění potřebných dílů. Dále byly implementovány a otestovány algoritmy pro vyhodnocení průběhu úhlové dráhy. Prvotní snahou při snímání pohybu válcového předmětu bylo vyhodnotit tento pohyb a následně převést do aplikace v reálném čase. Jelikož byla úspěšnost systému, který dovoľoval přenést pohyb do aplikace s minimálním zpožděním, velmi nízká, byl nakonec použit systém, který pohyb předmětu do aplikace převádí až po jeho ukončení. U tohoto systému byla úspěšnost vyhodnocení výrazně vyšší a byl následně použit pro ovládání vytvořené hry Curling.

Posledním, při této práci vzniklým systémem byl systém pro vyhodnocení rotačního pohybu kulového předmětu. Z důvodu hledání tvaru vedoucího k co nejvyšší úspěšnosti vyhodnocení, byly navrženy a po sléze vyrobeny dvě odlišné konstrukce snímaného předmětu. Dále byl vytvořen algoritmus pro výpočet osy rotace a délky pohybu. Navzdory vynaložené snaze nebyl výsledný systém schopen určit potřebné hodnoty s dostatečnou přesností a z tohoto důvodu nebyl pro ovládání z žádných vytvořených her použit.

Možným zdokonalením systémů vzniklých v průběhu této práce by bylo použití neuronové sítě pro detekci a identifikaci snímaného pohybu. Tím by se teoreticky mohla zvýšit úspěšnost vyhodnocení a snížit čas, který je pro úspěšnou identifikaci pohybu potřebný.

Použitá literatura

- [1] VOTÝPKA, Martin. *RFID hlavolam*. Liberec, 2021. Semestrální práce. Technická Univerzita v Liberci. Vedoucí práce Ing. Jana Kolaja Ehlerová Ph.D.
- [2] DOBKIN, Daniel Mark. *The RF in RFID: UHF RFID in practice*. Second edition. Amsterdam: Newnes, Newnes is an imprint of Elsevier, 2013. ISBN 978-0-12-394583-9
- [3] CURTY, Jari-Pascal, Michel DECLERCQ, Catherine DEHOLLAIN a Norbert JOEHL. *Design and optimization of passive UHF RFID systems*. New York: Springer, [2010]. ISBN 978-1-4419-4199-2
- [4] Quaternion. In: *Wikipedia: the free encyclopedia* [online]. San Francisco (CA): Wikimedia Foundation, 2001- [cit. 2022-04-27]. Dostupné z: <https://en.wikipedia.org/wiki/Quaternion>
- [5] Quaternion. *Wolfram Mathworld* [online]. [cit. 2022-04-27]. Dostupné z: <https://mathworld.wolfram.com/Quaternion.html>
- [6] Visualizing quaternions: An explorable video series. *Ben Eater* [online]. [cit. 2022-04-27]. Dostupné z: <https://eater.net/quaternions>
- [7] Video Game Physics Tutorial - Part I: An Introduction to Rigid Body Dynamics. Toptal [online]. [cit. 2022-04-27]. Dostupné z: <https://www.toptal.com/game/video-game-physics-part-i-an-introduction-to-rigid-body-dynamics>
- [8] Video Game Physics Tutorial - Part II: Collision Detection for Solid Objects. Toptal [online]. [cit. 2022-04-27]. Dostupné z: <https://www.toptal.com/game/video-game-physics-part-ii-collision-detection-for-solid-objects>
- [9] Video Game Physics Tutorial - Part III: Constrained Rigid Body Simulation. Toptal [online]. [cit. 2022-04-27]. Dostupné z: <https://www.toptal.com/game/video-game-physics-part-iii-constrained-rigid-body-simulation>
- [10] Creating the (nearly) perfect connect-four bot with limited move time and file size. *Towards Data Science* [online]. 2017 [cit. 2022-04-27]. Dostupné z: <https://towardsdatascience.com/creating-the-perfect-connect-four-ai-bot-c165115557b0>
- [11] LIU, Ran, Goran HUSKIC a Andreas ZELL. *On Tracking Dynamic Objects with Long Range Passive UHF RFID Using a Mobile Robot* [online]. 2015 [cit. 2022-04-27]. Dostupné z: https://www.researchgate.net/publication/277897492_On_tracking_dynamic_objects_with_long_range_passive_UHF_RFID_using_a_mobile_robot. Research Article. University of Tubingen.
- [12] LEE, Young Ho a Ivan MARSIC. *Object motion detection based on passive UHF RFID tags using a hidden Markov model-based classifier* [online]. 2015 [cit. 2022-04-27]. Dostupné z: <https://www.sciencedirect.com/science/article/pii/S221418041730017X>. Rutgers University.

- [13]KRIGSLUND, R., S. DOSEN a P. POPOVSKI. A Novel Technology for Motion Capture Using Passive UHF RFID Tags [online]. 2015 [cit. 2022-04-27]. Dostupné z: <https://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.304.3868&rep=rep1&type=pdf>. Rutgers University.
- [14]CAEN RFID API Reference Manual [online]. 2016, 121 [cit. 2022-04-27]. Dostupné z: https://www.caenrfid.com/wp-content/uploads/2021/11/CAEN-RFID-API-Reference-Manual_rev_12.pdf
- [15]CAEN RFID API User Manual [online]. 2014, 22 [cit. 2022-04-27]. Dostupné z: https://www.caenrfid.com/wp-content/uploads/2021/11/CAEN-RFID-API_UserMan_rev_05.pdf
- [16]R4320C Hadron High Performance 4-port Embedded RAIN RFID Reader TECHNICAL INFORMATION MANUAL [online]. 2021, 33 [cit. 2022-04-27]. Dostupné z: https://www.caenrfid.com/wp-content/uploads/2021/11/R4320C_Hadron_Technical-Information-Manual_Rev_02-1.pdf
- [17]WANT020 Quad Circular Polarized Quadrifilar Antenna (ETSI) [online]. 2020, 2 [cit. 2022-04-27]. Dostupné z: https://www.caenrfid.com/wp-content/uploads/2021/11/WANT020_Quad_DS-2.pdf
- [18]USB host overview. Android Developers [online]. 2020 [cit. 2021-5-15]. Dostupné z: <https://developer.android.com/guide/topics/connectivity/usb/host>
- [19]Breakout (video_game). In: Wikipedia: the free encyclopedia [online]. San Francisco (CA): Wikimedia Foundation, 2001- [cit. 2022-05-05]. Dostupné z: [https://en.wikipedia.org/wiki/Breakout_\(video_game\)](https://en.wikipedia.org/wiki/Breakout_(video_game))
- [20]Breakout game screenshot. In: Wikipedia: the free encyclopedia [online]. San Francisco (CA): Wikimedia Foundation, 2001- [cit. 2022-05-05]. Dostupné z: [https://en.wikipedia.org/wiki/Breakout_\(video_game\)#/media/File:Breakout_game_screenshot.png](https://en.wikipedia.org/wiki/Breakout_(video_game)#/media/File:Breakout_game_screenshot.png)
- [21]HASBRO Connect 4 Game. In: Home Hardware [online]. [cit. 2022-05-05]. Dostupné z: <https://homehardware.sirv.com/products/7237/7237624.view?thumb>

Obsah přiloženého CD

- Fotky vytvořených systémů pro snímání pohybu
- Snímky obrazovek vytvořených aplikací
- Text diplomové práce ve formátu PDF
- Zdrojové kódy vytvořených her