

# ZÁVĚREČNÁ STUDIJNÍ PRÁCE

## dokumentace

### Arduino gamepad



**Autor:** Matěj Vilášek  
**Obor:** 18-20-M/01 INFORMAČNÍ TECHNOLOGIE  
se zaměřením na počítačové sítě a programování  
**Třída:** IT4  
**Školní rok:** 2024/25



## **Poděkování**

Děkuji panu učiteli Mgr. Marcelu Godovskému za cenné rady a za pomoc s plošnou deskou, panu Mgr. Markovi Lučnému za rady a panu Ing. Petru Grussmanovi za rady a nápady.

## **Prohlášení**

Prohlašuji, že jsem závěrečnou práci vypracoval samostatně a uvedl veškeré použité informační zdroje.

Souhlasím, aby tato studijní práce byla použita k výukovým a prezentačním účelům na Střední průmyslové a umělecké škole v Opavě, Praskova 399/8.

V Opavě 1. 1. 2025

.....  
Podpis autora



## **Abstrakt**

Tento projekt se zaměřuje na návrh a výrobu vlastního gamepadu využívajícího platformu Arduino. Cílem bylo vytvořit zařízení kombinující tlačítka a dotykový displej, které by sloužilo pro jednoduché herní aplikace nebo ovládání jiných zařízení. K realizaci byl použit vývojový kit Arduino UNO R3, dotykový displej 2,4" SPI TFT ILI9341 a vlastní plošná deska s pěti tlačítky. Projekt zahrnoval jak návrh hardwaru, tak vývoj softwaru v prostředí PlatformIO, které využívá jazyk C/C++. Software obsahoval ovládání tlačítek, zobrazení grafických prvků na displeji a detekci dotykového vstupu. Pro zajištění správné funkčnosti byly použity knihovny Adafruit-GFX.h, Adafruit-ILI9341.h, URTouch.h a Arduino.h. Výsledkem je plně funkční gamepad, který lze využít pro různé herní nebo ovládací aplikace. Tento projekt poskytl cenné zkušenosti v oblasti elektroniky, pájení, návrhu desek a programování mikrokontrolérů.

## **Klíčová slova**

Arduino, gamepad, PlatformIO, C/C++, Adafruit-GFX, Adafruit-ILI9341, URTouch ...

# Obsah

<b>Úvod</b>	<b>7</b>
<b>1 Využité technologie</b>	<b>9</b>
1.1 Hardware . . . . .	9
1.1.1 Seznam součástek . . . . .	9
1.1.2 Arduino UNO R3 (ATmega328P) . . . . .	10
1.1.3 Displej ILI9341 (2,4" TFT SPI) . . . . .	11
1.1.4 Plošná deska a její zapojení . . . . .	12
1.2 Software . . . . .	13
1.2.1 PlatformIO . . . . .	13
1.2.2 Knihovny pro Arduino . . . . .	13
1.2.2.1 Adafruit-GFX.h . . . . .	13
1.2.2.2 Adafruit-ILI9341.h . . . . .	13
1.2.2.3 URTouch.h . . . . .	13
1.2.3 Programování v jazyce Arduino . . . . .	14
<b>2 PROGRAMOVÁNÍ ARDUINO GAMEPADU</b>	<b>15</b>
2.1 Zobrazení grafiky na displeji . . . . .	15
2.2 Ovládání pomocí tlačítek . . . . .	16
2.3 Dotykové funkce displeje . . . . .	16
2.4 Programování hry Pong . . . . .	17
2.4.1 Klíčové části programu . . . . .	17
<b>3 VÝSLEDKY A ZPŮSOBY ŘEŠENÍ</b>	<b>19</b>
3.1 Podoba hardwarového zařízení . . . . .	19
3.2 Programování hry . . . . .	20
3.3 Problémy a jejich řešení . . . . .	21
3.3.1 Problémy s velikostí děr na plošném spoji . . . . .	21
3.3.2 Omezení paměti Arduina . . . . .	21
<b>Závěr</b>	<b>22</b>
<b>Seznam použitých zdrojů</b>	<b>23</b>

# ÚVOD

Cílem tohoto projektu bylo navrhnout a vyrobit vlastní gamepad založený na platformě Arduino. K tomu jsem použil Arduino UNO R3, dotykový displej 2,4" SPI TFT ILI9341 a vlastní plošnou desku s pěti tlačítky. Tento projekt mi umožnil propojit práci s hardwarem a softwarem, což přispělo k rozšíření mých dovedností v oblasti elektroniky a programování. Práci jsem realizoval ve vývojovém prostředí PlatformIO, které využívá programovací jazyk C/C++, což je standardní jazyk pro vývoj na platformě Arduino.

Začal jsem návrhem plošné desky, na kterou byly umístěny tlačítka a konektory pro připojení displeje. Desku jsem vyrobil pomocí pájení a leptání, což mi umožnilo snadno připojit všechny součástky k Arduinu. Pro připojení a ovládání dotykového displeje jsem použil návod dostupný na internetu, který mi ukázal správné zapojení a knihovny pro práci s displejem.

Programování zahrnovalo využití knihoven jako Adafruit-GFX.h, Adafruit-ILI9341.h, UR-Touch.h a Arduino.h. Vytvořil jsem kód, který umožňuje ovládat tlačítka a zobrazovat grafiku na displeji. PlatformIO, s jeho podporou pro C/C++, mi umožnilo efektivně psát, testovat a ladit kód. Po úspěšném testování všech komponentů zařízení fungovalo dle očekávání a je připraveno k dalšímu použití nebo vylepšení.



# 1 VYUŽITÉ TECHNOLOGIE

## 1.1 HARDWARE

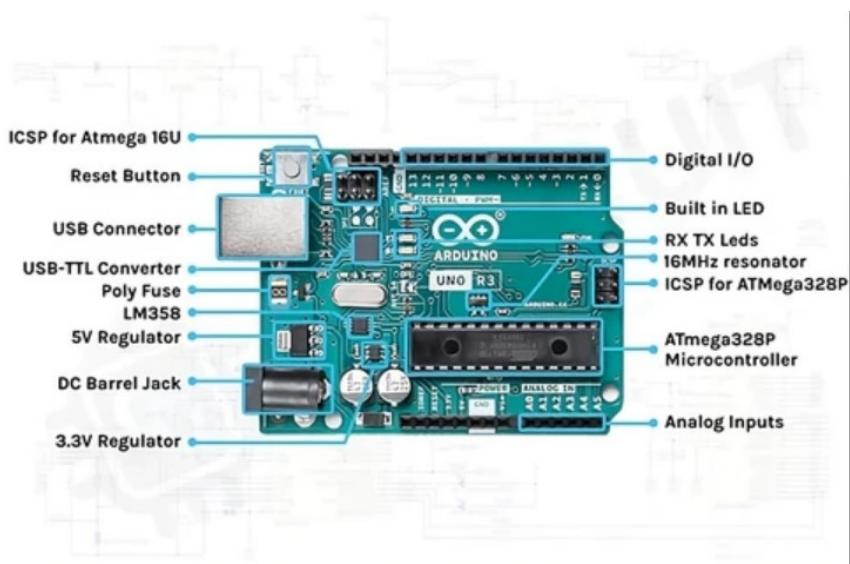
### 1.1.1 Seznam součástek

- Arduino UNO R3 (ATmega328P)
- 2,4" TFT SPI displej (ILI9341)
- Dvou vrstvý plošný spoj (PCB)
- 10k rezistor
- Mikrospínač - tlačítko 6 × 6 × 6 mm
- Dutinková lišta - Samice
- Kolíková lišta - Samce

### 1.1.2 Arduino UNO R3 (ATmega328P)

Základ projektu tvoří vývojová deska Arduino UNO R3 (obrázek č. 1.1). Deska je založena na mikrokontroléru ATmega328P od společnosti Atmel a nabízí 14 digitálních GPIO pinů, z nichž 6 lze využít jako PWM výstupy, a 6 analogových vstupů (obrázek č. 1.1). Paměť pro program má velikost 32 kB, z čehož 2 kB jsou vyhrazeny pro bootloader.

Arduino UNO pracuje s napětím 5 V a je možné ho napájet buď přes USB konektor, nebo pomocí externího napájení přes DC jack. Deska také obsahuje integrovaný převodník USB-TTL, což umožňuje snadné nahrávání programu a komunikaci s počítačem.



Obrázek 1.1: Vývojová deska Arduino UNO R3

### 1.1.3 Displej ILI9341 (2,4" TFT SPI)

Jako výstupní zařízení jsem použil TFT displej 2,4" s řadičem ILI9341, který dokáže zobrazit až 240x320 pixelů v 16bitových barvách. Pro připojení jsem využil SPI sériové rozhraní, což umožnilo připojení displeje pomocí několika vodičů. Funkčnost displeje je zajištěna knihovnami Adafruit-ILI9341 a Adafruit-GFX. Tento displej umožňuje jednoduché vykreslování textu, grafiky a dalších vizuálních prvků.



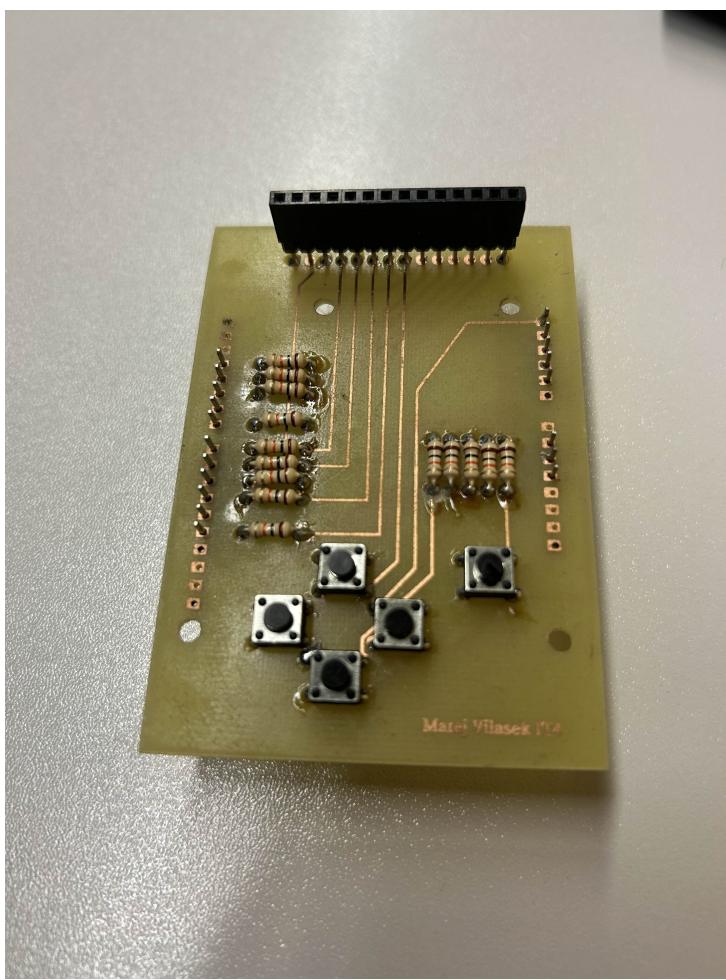
Obrázek 1.2: Displej ILI9341

#### 1.1.4 Plošná deska a její zapojení

Plošnou desku jsem navrhl jako dvouvrstvou v návrhovém softwaru EasyEDA. Po dokončení návrhu jsem si nechal desku vyrobit prostřednictvím služby na webu plosnaky.cz. Návrh byl vytvořen tak, aby deska přesně seděla na vývojovou desku Arduino, čímž se zajistilo kompaktní a efektivní propojení všech komponentů.

Při realizaci jsem však narazil na problém s příliš malými otvory pro některé součástky. Tyto otvory jsem musel manuálně zvětšit pomocí mikrovrtáčky. Následně jsem všechny součástky, jako jsou rezistory, tlačítka a konektory, ručně napájel na desku.

Celý proces výroby desky byl náročný, ale výsledek stojí za to. Plošná deska nyní slouží jako pevný a spolehlivý základ pro celý projekt, přičemž zajišťuje bezpečné a funkční propojení všech částí.



Obrázek 1.3: Plošná deska

## **1.2 SOFTWARE**

### **1.2.1 PlatformIO**

Jako vývojové prostředí jsem použil PlatformIO, které je moderní alternativou k tradičnímu Arduino IDE. PlatformIO podporuje širokou škálu mikrokontrolérů a umožňuje snadné spravování knihoven a projektů. Pro připojení externích knihoven a správu závislostí jsem využil soubor platformio.ini, který obsahuje všechny potřebné konfigurace pro komplikaci a nahrání kódu na zařízení. PlatformIO nabízí také integrovaný debugger a pokročilé možnosti testování, což značně zjednoduší vývoj a ladění projektů.

### **1.2.2 Knihovny pro Arduino**

#### **1.2.2.1 Adafruit-GFX.h**

Jako knihovnu pro grafiku jsem zvolil Adafruit-GFX, protože poskytuje širokou škálu funkcí pro vykreslování základních grafických prvků, jako jsou čáry, obdélníky, kruhy, text a další. Tato knihovna umožňuje snadnou manipulaci s textem a grafikou na displeji, což je ideální pro projekty, které vyžadují vizualizaci informací. Umožňuje měnit barvy textu a pozadí, měnit velikost písma, a dokonce i vykreslovat vlastní pixely, což ji činí velmi flexibilní pro různé grafické aplikace. Kromě toho je kompatibilní s mnoha různými typy displejů, což ji činí univerzálním nástrojem pro různé projekty.

#### **1.2.2.2 Adafruit-ILI9341.h**

Další knihovnu pro TFT displej jsem zvolil Adafruit-ILI9341, protože je navržena pro snadnou komunikaci s displeji, které používají řadič ILI9341, a umožňuje rychlé a efektivní ovládání displeje prostřednictvím SPI rozhraní. Tato knihovna zjednoduší proces inicializace displeje, nastavení orientace, vykreslování textu, obrázků a dalších grafických prvků na obrazovce. Díky vysokému rozlišení displeje (240x320 pixelů) a možnosti ovládání jasu a kontrastu je tato knihovna ideální pro projekty, které vyžadují kvalitní vizualizace a animace.

#### **1.2.2.3 URTouch.h**

Jako poslední tu máme knihovnu URTouch, která je určena pro práci s dotykovými panely. Zvolil jsem ji, protože umožňuje snadno detektovat dotykové události na displeji a získat přesnou pozici dotyku. Tato knihovna podporuje různé typy dotykových panelů a je velmi efektivní pro implementaci interaktivních funkcí. Umožňuje čtení souřadnic dotyku, což je klíčové pro

vytváření tlačítek, posuvníků a dalších interaktivních prvků na displeji. Díky této knihovně lze snadno přidat do projektu dotykové ovládání a vytvořit uživatelsky přívětivé rozhraní, které reaguje na vstupy uživatele.

### **1.2.3 Programování v jazyce Arduino**

Program sloužící k ovládání hardwarových součástek jsem napsal v jazyce C/C++ v prostředí PlatformIO, které je plně kompatibilní s Arduino knihovnami. Jazyk C/C++ je standardní volbou pro programování mikrokontrolérů a umožňuje efektivní komunikaci s hardwarem. PlatformIO nabízí rozšířené možnosti správy projektů a ladění, což zjednodušuje práci na složitějších aplikacích.

## 2 PROGRAMOVÁNÍ ARDUINO GAMEPADU

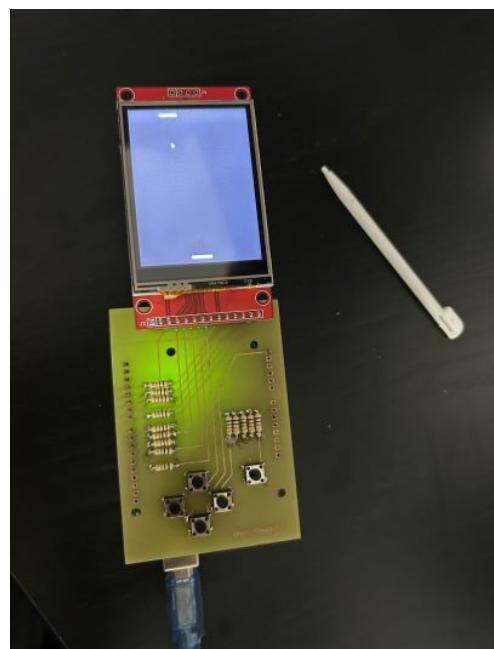
### 2.1 ZOBRAZENÍ GRAFIKY NA DISPLEJI

Zobrazení grafiky na displeji je jednou z klíčových funkcí projektu. Na displeji jsou zobrazeny všechny herní prvky, jako jsou pálky, míček a skóre hráčů, a to v reálném čase. Grafika je navržena tak, aby zajišťovala plynulý průběh hry a byla snadno čitelná.

Pro zobrazení pálky a míčku jsou využity jednoduché geometrické tvary, jako jsou obdélníky a čtverce. Tyto prvky jsou pravidelně aktualizovány, aby odpovídaly jejich aktuální poloze ve hře. Pro zajištění plynulého pohybu se současně provádí mazání předchozích pozic objektů, aby na displeji nevznikaly nechtěné artefakty.

Kromě samotné hry je na displeji zobrazeno také skóre, které se dynamicky mění podle průběhu hry. Text je umístěn v horní části obrazovky a je jasně viditelný díky kontrastnímu pozadí.

Celý systém je optimalizován tak, aby běžel plynule i na jednoduchém hardwaru Arduina.



Obrázek 2.1: Ukázka grafiky

## 2.2 OVLÁDÁNÍ POMOCÍ TLAČÍTEK

Hra je ovládána pomocí čtyř tlačítek, která jsou připojena k vývojové desce Arduino. Tlačítka jsou zapojena na analogové vstupy A2, A3, A4 a A5. Ačkoli na desce Arduino zbyly dva volné digitální piny, rozhodl jsem se využít analogové vstupy, aby nedocházelo k případným prodlevám při detekci stisků tlačítek.

Tlačítka jsou zapojena s využitím vnitřních pull-up rezistorů Arduina. Tento způsob zapojení minimalizuje počet potřebných externích součástek a zajišťuje spolehlivé ovládání hry.

Tento přístup umožňuje plynulé a bezproblémové ovládání celé herní logiky, což je klíčové pro dynamiku hry.

## 2.3 DOTYKOVÉ FUNKCE displeje

Displej ILI9341 je vybaven dotykovou vrstvou, která umožňuje kapacitní dotykové ovládání pomocí speciální tužky určené k tomuto účelu. Tento displej používá SPI (Serial Peripheral Interface) pro komunikaci s mikrokontrolérem, zatímco dotykové vstupy jsou detekovány pomocí knihovny URTouch.

Dotyková vrstva pracuje na principu detekce změn elektrického pole při dotyku povrchu displeje. Po dotyku displeje se souřadnice X a Y přenesou do mikrokontroléru, který je dále zpracovává. V případě této implementace jsou dotyková data využívána k ovládání pásky hráče ve hře.

```
1 if (ts.dataAvailable()) {  
2     ts.read();  
3     int touchY = ts.getY();  
4     if (touchY != -1) {  
5         paddleY = screenHeight - touchY - paddleHeight / 2;  
6     }  
7 }
```

Kód 2.1: Ukázka kódu pro dotykové ovládání hry

## 2.4 PROGRAMOVÁNÍ HRY PONG

Hra Pong byla navržena s cílem poskytnout jednoduchou, ale zábavnou hratelnost. Hlavními herními prvky jsou dvě pálky a míček. Hráč ovládá spodní pálku buď pomocí tlačítka, nebo dotykového ovládání displeje, zatímco horní pálka je ovládána jednoduchou umělou inteligencí (AI), která sleduje pozici míčku. Cílem je zabránit míčku opustit obrazovku na straně hráče.

### 2.4.1 Klíčové části programu

- **Inicializace hardwaru**

Nejprve se nastavují piny pro ovládání displeje, dotykové vrstvy a tlačítek. Displej i dotykový senzor jsou inicializovány pro zajištění správné funkčnosti herního rozhraní.

- **Vykreslování herních prvků**

Pálky a míček jsou na displeji vykreslovány pomocí jednoduchých grafických funkcí, které umožňují rychlé zobrazení. To je klíčové pro dosažení plynulého průběhu hry. Funkce jako `fillRect()` se používají k vykreslení a mazání prvků, což zajišťuje aktualizaci obrazu při každém pohybu herních objektů.

```
1 void drawPaddle(int x, int y) {  
2     tft.fillRect(x, y, paddleWidth, paddleHeight,  
3                   ILI9341_WHITE);  
4 }
```

Kód 2.2: Kod pro vykreslení pálky

- **Pohyb pálky a míčku**

Pohyb levé pálky je řízen buď tlačítky, nebo dotykovým ovládáním. Dotykové rozhraní umožňuje plynulý pohyb pálky podle polohy prstu na displeji. Míček se pohybuje konstantní rychlostí a při každém odrazu od pálky dochází k jeho zrychlení, což postupně zvyšuje obtížnost hry.

- **Kolizní detekce**

Kolize míčku s pálkami a stěnami je jedním z hlavních herních mechanizmů. Při nárazu na pálku se mění směr pohybu míčku a jeho rychlosť se zvýší. Kolize je detekována pomocí podmínek, které kontrolují vzájemnou pozici pálky a míčku.

Ukázka kódu pro detekci kolize s levou pálkou:

```
1 if (ballX <= paddleX + paddleWidth && ballX >= paddleX) {  
2     if (ballY + ballSize >= paddleY && ballY <= paddleY +  
3         paddleHeight) {  
4             ballSpeedX = -ballSpeedX;  
5             ballSpeedX *= ballSpeedMultiplier;  
6             ballSpeedY *= ballSpeedMultiplier;  
7         }  
8     }
```

Kód 2.3: Ukázka kódu pro detekci kolize s levou pálkou

- **AI pro pravou pálku**

Pravá pálka je ovládána jednoduchou AI, která sleduje pozici míčku a snaží se jej odrazit. AI se přizpůsobuje pohybu míčku a plynule upravuje pozici pálky, aby hra zůstala vyvážená.

- **Plynulost herního cyklu**

Pro zajištění plynulého průběhu hry bez použití zpožďovacích funkcí (`delay()`) je využívána metoda založená na časových intervalech s funkcí `millis()`. Ta umožňuje aktualizovat herní prvky v pravidelných intervalech, což zajišťuje konzistentní herní zážitek.

# **3 VÝSLEDKY A ZPŮSOBY ŘEŠENÍ**

Tato sekce popisuje dosažené výsledky projektu a přístup k řešení jednotlivých částí systému.

## **3.1 PODOBA HARDWAROVÉHO ZAŘÍZENÍ**

Zařízení pro hru Pong je v současné době plně funkční a splňuje má počáteční očekávání. Veškeré komponenty jsou osazeny na vlastnoručně navrženém a vyrobeném plošném spoji, který je přesně přizpůsoben rozměrům použitého Arduina. Toto řešení zajišťuje dostatečnou pevnost a spolehlivost hardwarového propojení.

Napájení zařízení je v současné době řešeno pomocí připojení přes USB kabel, což znamená, že uživatel musí mít k dispozici buď powerbanku, nebo dostatečně dlouhý kabel k elektrické síti. Do budoucna by bylo vhodné integrovat možnost napájení pomocí baterie, která by zvýšila mobilitu zařízení a odstranila závislost na externích zdrojích energie.

Dalším možným vylepšením by mohl být přechod na výkonnější mikrokontrolér, který by umožnil využití pokročilejších knihoven pro vykreslování na displeji. Tím by se eliminovaly současně drobné problémy, jako jsou například občasné "stíny" za objekty při rychlém pohybu.

## 3.2 PROGRAMOVÁNÍ HRY

Hra Pong splňuje svůj účel a nabízí jednoduchý, ale zábavný herní zážitek. Herní logika je navržena tak, aby běžela plynule i na hardwarově omezeném Arduinu. Veškerý kód je optimalizovaný pro maximální výkon a efektivitu, což zahrnuje mimo jiné i vykreslování herních objektů.

Z důvodu omezené paměti mikrokontroléru je využito překreslování celé obrazovky místo separátního vykreslování jednotlivých objektů. Tento přístup, ačkoliv méně efektivní, byl nutný vzhledem k hardwarovým limitům Arduina.

Pohyb pásky i míčku je řízen tak, aby hráči měli zajištěnou rychlou a přesnou odezvu při hře. Herní parametry, jako jsou rychlosť objektů a jejich zrychlování, byly pečlivě laděny tak, aby hra poskytovala adekvátní obtížnost.

Kromě základních funkcí byl kód navržen tak, aby byl co nejvíce přehledný a umožňoval případné budoucí rozšiřování nebo úpravy. Do budoucna bych rád implementoval systém pro lepší správu vykreslování objektů na displeji a případně i nové herní režimy nebo vylepšení.



Obrázek 3.1: Finální verze projektu

## 3.3 PROBLÉMY A JEJICH ŘEŠENÍ

### 3.3.1 Problémy s velikostí děr na plošném spoji

Jedním z prvních problémů, na které jsem narazil, byla nevhodná velikost vrtaných děr na plošném spoji. Některé součástky, například piny a tlačítka, se do otvorů nevešly. Tento problém jsem vyřešil manuálně – použil jsem mikrovrtáčku, abych otvory zvětšil na požadovanou velikost. Přestože to bylo časově náročné, umožnilo mi to zachovat funkčnost plošného spoje a pokračovat v projektu bez nutnosti výroby nového návrhu.

### 3.3.2 Omezení paměti Arduina

Druhým problémem byla omezená velikost paměti mého Arduina. Chtěl jsem původně použít jinou grafickou knihovnu, která by podporovala separátní překreslování jednotlivých objektů na displeji (tzv. "dirty rectangle" metoda), což by zlepšilo výkon systému. Bohužel tato knihovna byla příliš náročná na paměť, a Arduino ji nebylo schopno spustit.

Namísto toho jsem musel využít přístup, kdy se celá obrazovka překresluje znova při každé změně. Tento způsob vykreslování je méně efektivní a vedl k občasným zásekům hry nebo k tomu, že objekty (míček či pásky) za sebou zanechávaly stíny na obrazovce.

Abych tento problém minimalizoval, provedl jsem optimalizaci kódu:

- **Snížení rychlosti objektů:** Zpomalením pohybu míčku a pálek jsem snížil četnost aktualizace grafiky, což přispělo k plynulejšímu chodu hry.
- **Efektivní práce s pamětí:** Vyhnal jsem se zbytečnému vytváření proměnných a optimalizoval logiku pohybu i kolizí míčku, čímž jsem snížil zatížení procesoru.

I přes tyto optimalizace jsem si vědom, že kompletní řešení tohoto problému by vyžadovalo výkonnější hardware nebo použití efektivnější grafické knihovny, což však přesahovalo možnosti tohoto projektu. Přesto se mi podařilo dosáhnout funkčního řešení, které umožňuje hraní hry bez větších problémů.

# ZÁVĚR

Cílem projektu bylo vytvořit funkční herní zařízení, které dokáže spustit jednoduchou hru Pong na hardwaru s omezenými prostředky, jako je Arduino. Součástí projektu bylo také zprovoznění dotykového ovládání přes displej, což se mi podařilo, a hráči tak mohou ovládat hru nejen pomocí tlačítek, ale i dotykem. Všechny vytyčené cíle byly splněny – hra je plně funkční a poskytuje hráčům plynulý herní zážitek.

Během vývoje jsem se však setkal s několika problémy, jako byla omezená paměť Arduina nebo nevhodná velikost otvorů na plošném spoji. Tyto problémy jsem postupně vyřešil, at' už optimalizací kódu, nebo úpravou hardwaru.

Při realizaci mě napadlo několik možných vylepšení, která by zařízení posunula na vyšší úroveň. Jedním z nich je například napájení prostřednictvím baterie, což by zvýšilo přenosnost celého zařízení. Dále bych rád implementoval pokročilejší systém vykreslování herních objektů na displeji, což by mohlo eliminovat případné vizuální chyby, jako jsou stíny za pohybujícími se objekty.

Kromě optimalizace samotné hry plánuji přidat i další herní tituly, jako například Tetris nebo jiné klasické arkádové hry, čímž by zařízení získalo širší možnosti využití. Přestože na tato vylepšení už nezbýl čas, hodnotím projekt jako úspěšný a v budoucnu se těším na jeho další zdokonalování.

## SEZNAM POUŽITÝCH ZDROJŮ

- [1] *Arduino TFT LCD Touch Screen* [Online]. [cit. 2024-12-30]. Dostupné z:  
<https://www.youtube.com/watch?v=beyDkTBhpgs&t=235s>
- [2] *TFT LCD dotykový displej* [online]. [cit. 2024-12-30]. Dostupné z:  
<https://dratek.cz/docs/produkty/0/482/1420987998.pdf>
- [3] *Dotyková část kodu* [online]. [cit. 2024-12-30]. Dostupné z:  
<https://github.com/educ8s/Arduino-ILI9341-Example-Programs/blob/main/ArduinoILI9341Touch.ino>
- [4] *adafruit/Adafruit IO Arduino* [online]. [cit. 2024-12-30]. Dostupné z:  
<https://registry.platformio.org/libraries/adafruit/Adafruit%20IO%20Arduino/examples>
- [5] *URTouch na platformio* [online]. [cit. 2024-12-30]. Dostupné z:  
<https://github.com/f1rmb/URTouch>