



MATURITNÍ PRÁCE

INFORMATIKA

2025

TŘÍDIČKA SBĚRATELSKÝCH KARET MAGIC

Matyáš Vančák R8.A

Vedoucí práce: Emil Miler

Prohlášení o autorství projektu

Já, Matyáš Vančák, tímto prohlašuji, že jsem jediným autorem tohoto projektu i všech jeho částí.

Dílo vzniklo výhradně mou tvůrčí činností a nevzniklo jako dílo spoluautorů.

V Úholičkách dne 3. 3. 2025

Obsah

1 Úvod	3
1.1 Co je MTG a proč stavět třídičku?	3
1.2 Cíl projektu	3
1.3 Existující řešení	3
2 Analýza a specifikace požadavků	3
2.1 Typy karet, které bude zařízení třídit	3
2.2 Vstupy a výstupy systému	3
2.3 Požadavky na hardware a software	3
2.4 Bezpečnostní a ergonomické požadavky	4
3 Návrh hardwarové části	4
3.1 Výběr orientace a tvaru	4
3.1.1 Jednosměrné lineární rozložení	4
3.1.2 Dvousměrné lineární rozložení	4
3.1.3 Mřížka	4
3.1.4 Moje rozhodnutí	5
3.2 Výběr materiálu	5
3.2.1 Dřevo	5
3.2.2 Kov	5
3.2.3 Plast	5
3.3 Volba mechanických součástí	5
3.3.1 Pohyb třídící hlavy	5
3.3.2 Ozubená kolejnice	5
3.3.3 Trapézová tyč	5
3.3.4 Řemen	6
3.3.5 Zvolená varianta	6
3.4 Řídicí elektronika (mikrokontrolér, motor driver, zdroj napájení)	6
3.5 Použité senzory (např. kamera, čtečka kódů, IR senzory)	6
4 Návrh softwarové části	6
4.1 Výběr programovacího jazyka a prostředí	6
4.2 Použité knihovny	7
4.3 Algoritmus třídění	7
4.4 Uživatelské rozhraní	7
5 Implementace	7
5.1 Postup výroby zařízení	7
5.2 Testování jednotlivých komponent	7
5.3 Sestavení celého systému	8
5.4 Použití stroje	8
6 Testování a ladění	8
6.1 Jak se ověřovalo, že třídění funguje správně?	8
6.2 Výsledky testů	8
6.3 Rychlost a přesnost třídění	9
7 Závěr	9
7.1 Shrnutí výsledků	9

7.2 Co se povedlo, co by se dalo vylepšit	9
7.2.1 Prostorová náročnost	9
7.2.2 Poškozená deska Arduino Nano	10
7.2.3 Pohyb třídící hlavy v rámci konstrukce:	10
7.2.4 Pull Up/Down rezistory v Raspberry Pi	10
7.3 Možnosti rozšíření do budoucna	10
8 Zdroje	10

1 Úvod

1.1 Co je MTG a proč stavět třídičku?

Magic: the Gathering je sběratelská karetní hra. Existují hráči, kteří mají pouze karty, s nimiž aktivně hrají, existují však i sběratelé, kteří mají stovky, tisíce či dokonce řádově vyšší počty karet. V takové sbírce je ale potřeba být schopen najít konkrétní kartu, případně zjistit, jaké karty člověku chybí. Proto je důležité mít sbírku uspořádanou podle nějakých pravidel.

Třídička je tedy strojem, který zjednoduší tuto údržbu, případně může pomoci v hromadě karet najít potřebné karty pro stavění balíčků



1.2 Cíl projektu

Cílem práce tedy bylo vytvořit stroj, který po vložení balíčku karet bude schopen je automaticky rozpoznat a podle uživatelem navolených parametrů roztrždit, a to do alespoň 25 kategorií.

1.3 Existující řešení

Vzhledem k popularitě MTG je pochopitelné, že již existují třídičky na tyto karty, ovšem jejich ceny se pohybují v řádech vyšších desetitisíců až statisíců korun, nejsou tedy dostupná pro běžného hráče.

2 Analýza a specifikace požadavků

2.1 Typy karet, které bude zařízení třídit

Stroj je určený ke třídění karet Magic: The Gathering od osmé edice, kdy byl design karet upraven pro lepší čitelnost. Taktéž z důvodu čitelnosti je nutné, aby byly karty čisté, nepoškozené a bez obalů, které by mohly způsobovat odlesky.

2.2 Vstupy a výstupy systému

Vstupem stroje je srovnaný balíček karet, tak, aby jejich názvy byly ve stejné pozici, a parametry pro třídění, které uživatel zvolí během startu stroje.

Výstupem pak je několik balíčků roztržděných karet. Zajímavým výstupem by také mohl být seznam karet na jednotlivých výstupních balíčcích.

2.3 Požadavky na hardware a software

Hardwarově je nutné zajistit mechanismus na přesun karet, jejich scanování a samozřejmě počítač pro zpracovávání dat a ovládání stroje.

Softwarové požadavky jsou: UI pro zadání parametrů pro třídění, algoritmus pro rozpoznání karty, získání parametrů karty a jejich porovnání se vstupy

2.4 Bezpečnostní a ergonomické požadavky

Stroj se musí vejít na desku stolu a být přístupný pro vložení a vyjmutí karet a ovládání včetně úvodního nastavení parametrů.

3 Návrh hardwarové části

3.1 Výběr orientace a tvaru

Orientace a tvar zařízení mají vliv na zabíraný prostor i na mechanismy třídění. Cílem tohoto kroku je nalézt prostorově efektivní, ale zároveň rozumně mechanicky zpracovatelný způsob třídění karet.

Vzhledem k obdélníkovému tvaru karet je pro prostorovou efektivitu relevantní preferovat taktéž obdélníkové rozložení zařízení. V případě, že bychom toto pominuli, bylo by možné také uvažovat kruh či válec, v němž by byly skupiny uspořádány po obvodu pro možnost třídění se vstupním balíčkem uprostřed zařízení, toto by znamenalo nižší úroveň mechanické složitosti, ovšem zvýšilo by to nároky na prostor. Mechanická složitost je také důležitým faktorem, částečně kvůli rentabilitě zařízení, ale především kvůli možnosti opravy – čím méně částí bude zařízení mít, tím snazší bude jeho údržba a případné opravy.

3.1.1 Jednosměrné lineární rozložení

Toto rozložení si můžeme představit jako přímku dělenou na úsečky odpovídající velikosti karet. Paralelně s těmito pak je pohyblivá třídící hlava, která dodává karty. Jeho výhodou je bezesporu snadnost zpracování, jak materiální, tak mechanická, a nízká prostorová náročnost ve dvou dimenzích. Nevýhodou je vyšší prostorová náročnost v jedné dimenzi, přesun mezi jednotlivými výstupními balíčky by tedy byl delší, jak pro stroj, tak pro uživatele při vyšších počtech balíčků.

Vertikální varianta tohoto uspořádání by byla velmi nestabilní, byla tedy zavržena z tohoto důvodu. Horizontální varianta je funkční možností pro nižší počty výstupních balíčků, je také v praxi realizována v některých komerčních variantách tohoto zařízení, například PhyzBatch-9000

3.1.2 Dvousměrné lineární rozložení

Jedná se o rozšíření předchozího tvaru, kde jsou dvě paralelní linky výstupních balíčků, mezi kterými třídící hlava jezdí a je schopná přidávat na obě strany. Výhodou tohoto rozložení oproti jednosměrnému je poloviční prostorová náročnost v nejdelším rozměru, nevýhodou zůstává složitější sběr výstupních balíčků a přibývá mechanická náročnost na vícesměrné třídění. I přes vyšší stabilitu nebyl tento tvar zvolen ve vertikální podobě kvůli nedostatečné stabilitě, v horizontálním směru nebyl implementován z důvodu nedostatečné efektivity.

3.1.3 Mřížka

Mřížka rozkládá prostorovou náročnost do dvou rozměrů, čímž spíše vyhovuje umístění na běžné pracovním stole. Nevýhodou však je složitější mechanismus přesunu karet, jelikož je nutné s nimi pohybovat minimálně ve dvou, spíše ve třech rozměrech, což přidává na mechanické komplexitě zařízení. Vertikální varianta tohoto zařízení je prostorově nejúspornější možností, jelikož nejdelší rozměr karty může být použit jako nejkratší rozměr stroje (tzn. když bude více karet vedle sebe či nad sebou, bude se násobit kratší rozměry). Horizontální varianta mřížky je pohodlnější pro uživatele, jelikož výstupní balíčky mohou být rovnou lícem k němu, je tedy možné je od sebe snáze rozlišit. Tento přístup je zvolen i v komerčním stroji Magic Sorter.

3.1.4 Moje rozhodnutí

Po analýze možných rozložení jsem se rozhodl pro horizontální lineární rozložení, jelikož pro toto použití je dostačujících 9 celkových balíčků – jeden vstupní a osm výstupních, pro každý card type jeden. Odhad nejdelší strany stroje by byl tedy kolem 60–70 cm. Nevýhodou tohoto rozhodnutí je počet kombinací paralelně tříděných parametrů, stroj by však byl mnohem komplikovanější, pokud by byly zahrnuty i tyto, pokud by člověk po stroji požadoval například třídění dle roku vydání karty, bylo by aktuálně nutné mít 32 výstupních balíčků, pokud by bylo cílem třídít podle jednotlivých setů, pohybovali bychom se kolem 170 balíčků.

3.2 Výběr materiálu

Materiál konstrukce hraje velkou roli na estetiku, stabilitu, rozložitelnost a opravitelnost i cenu zařízení. Hlavní materiály, které připadaly v úvahu byly dřevo, hliníkové profily či 3D tištěný plast. Další materiály z finančního i praktického hlediska nepřípadaly v úvahu, pravděpodobně by bylo nutné je vyrábět na míru, což při jednom kuse nebývá výhodné, u komerčního produktu by to však stálo za zvážení.

3.2.1 Dřevo

Výhodou dřeva je snadnost zpracování, široká dostupnost v různých velikostech a typech, vcelku nízké pořizovací náklady. Důvodem, proč jsem jej nezvolil jako součást finálního designu, je fakt, že s jinými materiály se pracuje výrazně snáze co se týče vzájemného propojování a jsou pevnější pro toto použití.

3.2.2 Kov

Kovová konstrukce je nejpevnější variantou, což je pro stroj s pohyblivými částmi relevantní. Jelikož se jedná o projekt, kde se počítá s úpravami, jako ideální volba se jeví hliníkové profily, které je snadno možné vzájemně spojovat a rozdělovat po celé délce. Svářená konstrukce by byla sice pevnější, ale neposkytovala by tuto modularitu, která je pro projekt podstatná. Taktéž esteticky je pro stroj preferován kov oproti jiným materiálům, ideálně lesklý.

3.2.3 Plast

Plast sám o sobě nebývá optimální pro celkovou konstrukci stroje, jelikož se typicky snáze deformuje. V čem ale spočívá jeho výhoda, je možnost výroby 3D tištěných součástí, například třídící hlavy tvarované na míru.

3.3 Volba mechanických součástí

Pohyb je možné řešit různými způsoby. V tomto stroji jej rozdělíme na dvě části – pohyb celé třídící hlavy a přesun karty.

3.3.1 Pohyb třídící hlavy

Třídící hlava se musí pohybovat po lince circa 55 cm s devíti body, kde se zastaví. Pro tento pohyb budeme zvažovat ozubenou kolejnici, trapézovou tyč nebo řemen jakožto nejužívanější způsoby převodu rotačního pohybu motoru na lineární.

3.3.2 Ozubená kolejnice

Lineární pohyb pomocí ozubené kolejnice by znamenal připevnění motoru k třídící hlavě, což není preferovanou možností, je jistější mít co nejvíce součástí a váhy mimo třídící hlavu. Výhodou by bylo zmenšení prostoru zabraného pohybujícími se součástmi zařízení - jejich centralizace v modulu třídící hlavy

3.3.3 Trapézová tyč

Trapézová tyč nabízí nejvyšší přesnost, ale zároveň nejnižší rychlost. V tomto využití si můžeme dovolit zvýšit přesnost ze setin milimetrů teoreticky i na nižší jednotky, rychlost je přednější.

3.3.4 Řemen

Pohyb pomocí řemene kombinuje rozumnou přesnost s vyšší možnou rychlostí. Jeho největší slabinou je největší míra opotřebování a nutnost občasného napínání řemene.

Tyto jsou ale v mírách třídičky v podstatě irelevantní.

3.3.5 Zvolená varianta

Pohyb je zajištěn krokovým motorem Nema 17 pro přesnost při cestě ke správné hromádce karet, motor sám má přesnost 200 kroků na otáčku, pomocí budiče A4988 (v angličtině driver) je však možné dosáhnout dvoj-, čtyř-, osmi- až šetnáctinásobného počtu kroků na otáčku (tzv. microstepping), tj. 3200. Pro tento pohyb však microstepping není nutný kvůli preferenci rychlosti před přesností.

3.4 Řídicí elektronika (mikrokontrolér, motor driver, zdroj napájení)

Mozkem celého projektu je Raspberry Pi 4B, jednodeskový počítač (SBC) o velikosti platební, nebo ve spojení s tímto projektem ještě lépe hrací, karty, který řídí veškeré další elektronické součástky. Teoreticky by bylo možné zařízení ovládat i ze stolního počítače či notebooku, ovšem ty nemají možnost přímo ovládat motory či jinak obsluhovat HW stroje. Raspberry Pi je pro takováto využití přímo stavěné se svými 40 GPIO piny.

Jelikož však Raspberry není stavěné tak, aby skrze něj tekly vyšší proudy potřebné pro operaci krokovacích motorů, je nutné jej doplnit o budiče (drivery) těchto motorů. Ve stroji využívám budiče založené na čipu A4988 kvůli jejich kompatibilitě s využitými motory, snadnému užívání a dobré dostupnosti na trhu.

Motory je nutné napájet vyšším napětím než samotnou logiku (Raspberry Pi a budiče) a zároveň operují na vcelku vysokých proudech (až .9A na cívku), zvolil jsem tedy pro stroj zdroj s napětím 15V, který bude stačit motorům a zároveň mohou pomocí step-down modulu ze stejného zdroje získávat 5V pro logiku. Celkový proud nutný k napájení všeho je zhruba 7A, pro stabilitu v situacích s abnormálním odběrem byl použit zdroj s maximálním proudem 10A.

3.5 Použité senzory (např. kamera, čtečka kódů, IR senzory)

Pro snímání karet byla použita kamera dedikovaná pro Raspberry Pi, konkrétně modul Raspberry Pi Camera Module 2, s rozlišením 3280 × 2464 pixelů pro co nejostřejší obraz a tudíž vysokou šanci na poznání. S nižší kvalitou byly potíže, poněvadž OCR je trénovaná na dokumenty, ideálně čisté digitální či kvalitní skeny tištěného textu na kontrastních pozadích. O osvětlení karty při focení se starají LED, bez osvětlení by nebyly stabilní podmínky pro používání a bylo by potřeba při každém použití potřeba dopočítat masku pro dorovnání světelných rozdílů po délce textu.

4 Návrh softwarové části

4.1 Výběr programovacího jazyka a prostředí

Třídička běží na výchozím operačním systému pro Raspberry Pi - Raspberry Pi OS (dříve zvaný Raspbian). Jedná se o linuxový systém odvozený od Debianu, který byl upraven pro podporu těchto zařízení. Přestože existují i další kompatibilní systémy (například FreeBSD, RISC OS, Windows 10

ARM64 apod.), běží stroj na Raspberry Pi OS, poněvadž ten je přímo designovaný pro použití počítač a tudíž minimalizuje riziko problémů.

Samotný program je pak psaný v programovacím jazyce Python, kvůli jeho srozumitelnosti, jednoduchosti a z velké části také knihovnám nutným pro spojení všech potřebných technologií. Mezi další možnosti spadaly také C, C++ či například Java, u všech z nich byla hlavním pozitivem vyšší rychlost běhu. Ta ovšem v tomto případě není příliš relevantní, proto od nich bylo ustoupeno.

4.2 Použité knihovny

Jelikož je Python populárním programovacím jazykem, existuje pro něj již spousta kódu vytvořeného a sdíleného jinými lidmi. Pokud takový kód funguje a dělá, co by se hodilo v kódu měm, bylo by hloupé jej nevyužít. Tímto není myšlen hlavní kód třídičky, ale použití knihoven. Tyto velmi zesnadňují práci a umožňují vyšší efektivitu. Knihovny které jsem využil:

- **RPI.GPIO** – knihovna pro komunikaci s GPIO piny na Raspberry Pi pro vysílání a přijímání signálu do a z ostatních komponent stroje
- **os** - konkrétně její metoda `system()` je použita pro spouštění povelů v systémovém shellu, v kódu je volána na spuštění GUI pro nastavení parametrů třídění
- **PIL** – knihovna pro práci s obrazem, jak jeho vytažení ze souboru, tak přípravu před detekcí textu
- **difflib** – její třída `SequenceMatcher` je použita pro nalezení nejbližší shody detekovaného textu s názvem karty, jelikož OCR často rozezná prach, ohyby karty a podobné nedostatky jako další části textu
- **picamzero** – slouží k ovládání kamery, vyfocení názvu karty
- **pytesseract** – knihovna umožňující komunikaci s Tesseract OCR v rámci Pythonu, lze ji vnímat jako OCR, přestože technicky pouze komunikuje s externí aplikací a zprostředkovává toto spojení
- **time** – doslovný překlad “čas” napovídá, že tato knihovna je používána k vytváření časových prodlev, konkrétně mezi signály pro budiče motorů

4.3 Algoritmus třídění

Již několikrát bylo zmíněno použití OCR (Optical Character Recognition) na rozpoznání karet. Tou je Tesseract, jedna z nejpřesnějších open-source programů na rozpoznávání znaků v obrazu. Nemajíc grafické uživatelské rozhraní, je méně náročná na výkon, což ji dělá vhodnou pro toto použití. OCR detekuje text na fotografii a rozpozná o jaké znaky se jedná. Vzhledem k již zmiňovaným nepřesnostem ve výstupu OCR je tento předán do `SequenceMatcher`, který pomocí počítání Levensteinovy vzdálenosti (počtu nutných úprav textu pro jeho změnu do jiného textu) najde kartu s nejpodobnějším názvem. Poté dojde k porovnání parametrů detekované karty s parametry zvolenými pro jednotlivé balíčky a karta je přiřazena na balíček shodných parametrů či na koncový balíček s kartami neodpovídajícími jinému z balíčků.

4.4 Uživatelské rozhraní

Spuštění programu je nutné manuálně z plochy počítače. Hlavní část programu nemá grafické rozhraní, má jej jen část určená pro volbu parametrů balíčků. Po zbytek běhu programu ovšem není grafické rozhraní potřebné

5 Implementace

5.1 Postup výroby zařízení

Pro výrobu stroje bylo nutné sehnat různé elektronické i konstrukční části, především získávané z českých eshopů (gme.cz, shraplayers.cz, dratek.cz, rpishop.cz). Během čekání na dodání dílů byl

prostor začít psát kód, testovat OCR na fotografiích získaných z internetu či navrhovat datové struktury pro snadnou a efektivní práci s informacemi.

Jakmile díly dorazily, bylo možné začít zkoušet jednotlivé komponenty i části kódu pro pohyb, vlastní focení apod.

5.2 Testování jednotlivých komponent

Testování softwarových komponent probíhalo pomocí automatizovaného stažení grafik karet, a porovnávání detekovaného textu s realitou karty. Úspěšnost se pohybovala kolem 85%, což je přijatelný výsledek.

Porovnávání parametrů a přidělování karet do virtuálních balíčků chtělo chvilku plánování kvůli různým strukturám, ve kterých jsou tyto údaje uloženy (CMC je float, legality karet jsou uloženy jako dictionary, barvy jako list).

Testování hardwaru bylo nejnáročnější, bylo nutné zjistit počet kroků motoru na přesun mezi balíčky, vyladit proud tečící do motorů, správně nastavit výšku endstopu, aby se karta chytila na přísavku, ale nebyla přitlačena příliš.

5.3 Sestavení celého systému

Stavba stroje byla náročná kvůli velkému množství komunikace mezi Raspberry Pi a ostatní elektronikou, limitům na proud z GPIO pinů a podobně. První kompletní sestavení trvalo necelé čtyři hodiny. Číslovka implikuje, že stroj byl rozebírán a stavěn znovu. Mimo jiné kvůli převozu na konzultaci či útoku kočky, které se líbil kabel, pravděpodobně připomínající myší ocas.

5.4 Použití stroje

Stroj je nutné postavit na vodorovnou plochu, zkontrolovat veškerá propojení a poté připojit k elektrické síti. Po nabootování systému je potřeba z plochy spustit hlavní třídící program. Po krátké přípravě se objeví grafické rozhraní pro nastavení parametrů pro jednotlivé balíčky, po jejich nastavení již třídička začne třídit.

6 Testování a ladění

6.1 Jak se ověřovalo, že třídění funguje správně?

Testování probíhalo v mnoha fázích různými způsoby, od teoretického, přes manuální k automatickému a zpět. Produkt byl testován jak po částech, tak jako celek, aby byla zaručena co nejvyšší kvalita a spolehlivost. Teoretické testování se skládalo ze sestavení diagramu funkcí jednotlivých částí finálního produktu a ověření, zda není potřeba ještě nějakého prvku, který by ovlivňoval funkčnost celku. Tento krok bylo potřeba zopakovat poté, co v první verzi přestalo fungovat Arduino, přičemž bylo odhaleno, že jeho funkce je možné nahradit jinými způsoby

6.2 Výsledky testů

Jednotlivé funkce byly testovány postupně, když byly vyvíjeny, první přišla detekce hodnot karty pomocí barev na kartě, přestože byla značně neefektivní, zdála se být funkční, protože byla v první verzi zachována. Stejně tak to bylo s ovládáním pohybu pomocí Arduina, které komunikovalo s Raspberry Pi pomocí I2C (Inter-Integrated Circuit), což znamenalo zdržení během posílání informací mezi deskami. Později přišlo rozřazování do jednotlivých skupin karet podle jejich vlastností, které bylo testováno jak textově, tak v kombinaci s rozpoznáním hodnot z karty. Všechny tyto testy probíhaly manuálně. Ve chvíli, kdy byly všechny části funkční, byl stroj zkompleťován a

puštěn jako celek, třídící karty automaticky. Bohužel krátce před termínem odevzdání došlo k vadě napájecího zdroje, která způsobila poškození Arduina, kvůli čemuž se stroj stal nepoužitelným.

Ve druhé verzi došlo k celkové změně software, kvůli zjištěným nedostatkům v efektivitě. Bylo tedy třeba testovat jak software, tak hardware zcela nanovo. Po otestování třídění karet podle názvu přišlo testování rozpoznání názvu ve fotce karty, poté kombinace těchto dvou. Ta z počátku probíhala manuálně, později za využití dalšího programu na sehnání testovacích obrázků běžela dokonce automaticky na tisících karech za sebou. S náhodnými kartami dosahoval program kolem 85% úspěšnosti při rozpoznávání karty, což vzhledem k faktu, že mezi náhodně vybranými byly mnohé se starým či alternativním designem, které nejsou podporovány kvůli značně jiné struktuře karty, je více než přijatelný výsledek. Jako celek byl stroj testován stejně jako první verze i uživatelem (autorem i člověkem, který dostal pouze pokyny k použití)

6.3 Rychlost a přesnost třídění

Rychlost třídění je zhruba 1 karta za 10 sekund, přesnost během testování se pohybovala mezi 85-90%. Lepší výsledky by vyžadovaly vyššího výpočetního výkonu, který použitý počítač nenabízí.

7 Závěr

7.1 Shrnutí výsledků

V závěru práce bych rád zmínil, že výsledná verze není dokonalá, mezi její mouchy patří: nekompatibilita se starými kartami a kartami s alternativním rozložením, rychlost třídění či počet možností paralelního filtrování. Přesto jde o uspokojivý produkt, který zvládá svůj primární účel pomoci s tříděním karet bez nutnosti lidské interakce za běhu.

Práce mi pomohla poznat mnoho nového, jak software, mezi něž patří Tesseract OCR, používání Raspberry Pi OS či skriptování, tak hardware, kde jsem se naučil pracovat s GPIO piny Arduina i Raspberry, zjistil, jak funguje budič krokových motorů a naučil se pracovat s Logic Level Converterem.



Práci tedy považuji za přínosnou, ovšem časově výrazně náročnější, než bylo očekáváno.

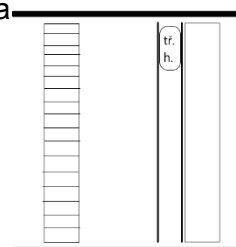
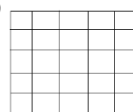
7.2 Co se povedlo, co by se dalo vylepšit

Tato část vysvětluje některé problémy, které bylo během tvorby nutno řešit a ukazuje různá potenciální řešení a řešení použité.

7.2.1 Prostorová náročnost

Díky tomu, že se jedná o fyzický stroj, zabírá tedy fyzický prostor. Je však více možností, jak by toto mohl dělat:

- **Horizontální mřížka**, po které by se třídící hlava přesouvala například po vzájemně kolmých kolejnicích – toto rozložení nebylo využito kvůli jeho prostorové náročnosti ve více rozměrech, za předpokladu rozměrů 5x5 karet by rozměry byly zhruba 50x40cm, což bylo vyhodnoceno jako příliš široké pro stůl, na němž byl stroj testován. Zároveň by to také znamenalo komplikovanější pohyb mezi hromádkami kvůli přidání druhého motoru a zvýšení nepřesnosti



- **Věž** – dostatečně vysoká kvádrová věž s pravidelně umístěnými horizontálními oddělovači pro vertikální stohování karet. Tato možnost má nevýhodu ve výrazně omezené výšce hromádky, po jejímž naplnění nebude možné dále přidávat. Druhým problémem je stabilita, poněvadž i s pevnou základnou bude těžká třídící hlava mít ve vysokých polohách větší riziko pádu, jak bylo během vývoje experimentálně ověřeno.
- **Vertikální mřížka** – technicky také popsatelná jako více spojených věží vedle sebe. Spojuje potíže předchozích dvou designů, od horizontální mřížky bere složitost a od věže její nestabilitu, přestože již jen v jednom směru, a omezení výšky hromádky karet
- **Horizontální linie hromádek** – tato možnost byla použita, přestože má velkou slabinu ve své délce, která je větší než u jiných možností. Pro zvětšení použitelného prostoru tedy stroj vyhazuje karty na obě strany podél své dráhy pro dvojnásobek hromádek

7.2.2 Poškozená deska Arduino Nano

první stroj přestal fungovat kvůli příliš vysokému napětí zdroje, které poškodilo desku Arduino Nano používanou jako regulátor napětí a především díky svým 5V outputům jako ovladač budičů. Možná řešení:

- Výměna zdroje - proběhla jistě, poněvadž nestabilní zdroj znamená riziko dalšího poškození. Výměna zdroje však neřeší chybějící součástku
- Koupě nového Arduina – z finančních důvodů byla tato možnost nedostupná, protože nebyla využita. Dalším důvodem pro toto rozhodnutí bylo, že Arduino není krucióální pro funkci stroje
- Použití Logic Level Converteru – tato jednoduchá součástka umožňuje komunikovat s budiči přímo z Raspberry Pi díky změně napětí signálu z 3.3V na 5V. Tuto změnu ale, jak jsem později zjistil, nebylo vůbec nutné provádět

7.2.3 Pohyb třídící hlavy v rámci konstrukce:

- **Koleje** – pokud by se hlava pohybovala po kolejkách, bylo by nutné ji pohánět zevnitř, což by přidávalo hmotnost a tím zhoršovalo stabilitu celého stroje
- **Lanko na cívce** – hlava by se mohla pohybovat pomocí cívky s lankem, která by se navíjela či odvíjela podle potřeby, to je však ve vodorovné poloze neefektivní
- **Okruh lanka** – by jistě také mohl fungovat, hlavními problémy jsou ale opotřebovávání lanka, a jeho pružnost, která by pravděpodobně způsobovala nepřesnosti při posunu po dráze
- **Závitová tyč** – závitová tyč je vhodným řešením jak pro přesnost, tak pro přesunutí motoru mimo samotnou třídící hlavu, díky čemuž byla vybrána jako finální řešení. Ideální by byla tyč s větším stoupáním závitu, poněvadž by zajistila vyšší rychlost, například trapézová tyč pro 3D tiskárnu

7.2.4 Pull Up/Down rezistory v Raspberry Pi

- mají za úkol nastavovat výchozí hodnotu jednotlivých pinů. Piny 0-8 začínají s hodnotou 1, tedy běžící napětí 3.3V, ostatní 0, tedy bez napětí.
- To může být problém, pokud do těchto pinů chceme připojit například výstup ze spínače, protože je důležité všem vstupním pinům nastavit jejich pull up/down rezistor na down, tedy výchozí stav 0

7.3 Možnosti rozšíření do budoucna

- Více pozic pro paralelní třídění
- podpora kombinování parametrů při třídění
- Detekce pomocí pHash

8 Zdroje

- <https://tcgmachines.com/shop/physbatch-9000>
- <https://www.magic-sorter.com/>
- <https://toon-beerten.medium.com/ocr-comparison-tesseract-versus-easyocr-vs-paddleocr-vs-mmocr-a362d9c79e66>
- <https://mtgison.com/>
- <https://pinout.xyz/pinout/pwm>