



MATURITNÍ PRÁCE

INFORMATIKA

2021/2022

TŘÍDIČKA SBĚRATELSKÝCH KARET MAGIC

Matyáš Vančák R8.A

Vedoucí práce: Emil Miler

Obsah

Obsah	1
Prohlášení o autorství projektu	2
Úvod	3
Použité technologie	4
Hardware	4
Software	5
Použití stroje	5
Jak třídění funguje	6
Použité knihovny	6
Testování	7
Teoretické testování	7
Testování jednotlivých funkcionalit	7
Obtíže a jejich řešení	8
Závěr	10

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 22.8.2022

Úvod

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. Cílem práce tedy bylo vytvořit stroj, který po vložení balíčku karet bude schopen je automaticky rozpoznat a podle uživatelem navolených kritérií roztrždit. Stroj je vytvořený na třídění karet Magic: the Gathering, avšak s jeho program je uzpůsobený tak, aby byl schopen pracovat i s jinými sběratelskými kartami stejného fyzického formátu, například kartami Pokémon. Jedinou nutnou změnou by byla záměna databáze karet, s níž je stroj připravený pracovat.



V aktuálním stavu je třídička schopná balíček karet (cca 60) za zhruba 10 minut, čas však záleží na zvolených kritériích pro třídění.

Bohužel má třídička jisté limity z různých důvodů. Pro funkční rozpoznání karet musí být karty bez obalu, poněvadž jinak se příliš lesknou a stroj není schopen je spolehlivě naskenovat. V aktuálním stavu také není schopen poznat velmi staré karty, kvůli změně v designu v roce 2003, což ovšem příliš nevádí, poněvadž tyto karty mají obvykle příliš vysokou hodnotu na to, aby byly uchovávány bez obalu.



Použité technologie

Hardware

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. Tato deska byla vybrána kvůli snadnosti ovládání v kombinaci s vhodnou velikostí.

Obraz karty je získáván pomocí kamery s rozlišením 3280×2464 pixelů pro co nejostřejší obraz a tudíž vysokou šanci na poznání, s nižší kvalitou byly potíže, poněvadž během zpracování došlo k narušení obrazu natolik, že OCR nebyla schopna kartu poznat.

O stabilní osvětlení karty při focení se starají dvě LED, poněvadž použití jedné by mohlo způsobit odrazový flek kvůli lesklosti karet a bez osvětlení by karty nebyly vidět

Pohyb je zajištěn krokovými motory 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.

Samotný třídící modul se pak pohybuje po trapézové tyči s 3mm závitem - jedna celá otáčka znamená posun po závitu o 3mm. Kvůli tomu je microstepping spíše nevhodný, protože přesnost 0.015mm je bohatě dostačující.

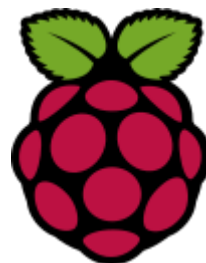
Mezi budičem a Raspberry Pi bylo nutno použít tzv. Logic Level Converter (v češtině konvertor úrovní), zařízení převádějící signál z Raspberry, jenž má 3,3V, na signál 5V, který je budič schopen rozpoznat a převést na signál pro motor.

Pro volbu kritérií podle kterých mají být karty tříděny je na stroji sada kolébkových vypínačů.

Celý stroj je pak uložený ve dřevěné konstrukci, jež byla na míru vyřezána laserem pro vysokou přesnost.

Software

Třídíčka 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.

Již v minulé sekci bylo zmíněno použití OCR na rozpoznání karet. Tou je Tesseract, jedna z nejpreciznějších open-source programů na rozpoznávání znaků v obrazu. Nemajíc grafické uživatelské rozhraní, je méně náročná na výkon, což ji dělá vhodnou pro toto použití. Díky Apache License, pod níž je Tesseract distribuován, není žádný právní problém s využitím právě Tesseract OCR.

Použití stroje

1. Stroj je umístěn na vodorovnou plochu a 12V zdroj (černý) zapojen do elektrické sítě 230V. Pro určování stran se předpokládá pohled kolmo na stranu s kolébkovými vypínači se směrem dolů odpovídajícím směru k podlaze.
2. Do vrchní části uživatel umístí karty bez obalů či jiných obstrukcí lícem dolů a vrchem karty směřujícím doprava.
3. Poté zvolí kategorie pro třídění pomocí kolébkových vypínačů na čelní straně stroje - přepínač v poloze nahoru znamená "třídít podle tohoto faktoru", dolů "ignorovat tento faktor".
4. Přichází čas zapojit druhý zdroj (bílý). Chvilu je stroj v klidu, než nastartuje.
5. Poté stroj resetuje svou polohu a začíná třídít, uživatel již nemusí nijak se strojem interagovat
6. Po vyřídění všech karet je bílý zdroj vypojen ze sítě a stroj je připraven na opakování tohoto procesu od kroku 2

Jak třídění funguje

1. Při zapnutí si stroj nejprve připraví potřebná data pro fungování - databázi karet a nutné proměnné
2. Stroj vyfotí kartu a detekuje text v poli pro její název pomocí Tesseract OCR.
 - a. Pokud není detekován žádný non-whitespace text, je karta vyhodnocena jako vadná či špatně otočená a je umístěna na speciální "leftover" hromádku
3. Tento text je porovnáván s názvy karet v databázi, čehož je dosahováno pomocí Levenshteinovy vzdálenosti, tj. nejnižšího počtu úprav (přidání, záměna či vypuštění znaku), v poměru k délce textu.
4. Kartě jsou přiřazeny hodnoty karty s názvem nejbližše detekovanému textu.
5. Tyto hodnoty jsou porovnány s hodnotami karet na jednotlivých hromádkách
 - a. Pokud některé z hromádek odpovídají, je tato hromádka nastavena jako cílová
 - b. Pokud hodnoty neodpovídají žádné z hromádek a ještě nebyl dosažen limit počtu hromádek, je založena nová hromádka a jsou jí přiřazeny hodnoty této karty
 - c. Pokud hodnoty neodpovídají žádné hromádce a byl dosažen maximální počet hromádek, je karta určena na "leftover" hromádku
6. Po zpracování se třídící modul přesune pomocí trapézové tyče k určenému páru hromádek a na správnou vyhodí kartu
 - a. Stroj si po každém přesunu zapamatuje svou polohu, díky čemuž může hned pokračovat s další kartou a není nutné se vracet do výchozí polohy.
7. Tento postup se opakuje od bodu 2 dokud nedojdou karty

Použité knihovny

- pigpio - knihovna pro komunikaci s GPIO piny na Raspberry Pi
- os - konkrétně její metoda `system()` je použita pro spouštění povelů v systémovém shellu
- PIL - knihovna pro úpravu obrazu pro vyčištění před detekcí textu
- difflib - její třída `SequenceMatcher` je použita pro nalezení nejbližší shody detekovaného textu s názvem karty
- picamera - ovládání kamery, vyfocení názvu karty
- pytesseract - knihovna umožňující komunikaci s Tesseract OCR v rámci Pythonu
- time - doslovný překlad "čas" napovídá, že tato knihovna je používána k vytváření časových prodlev, konkrétně mezi signály pro budiče motorů

Testování

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 maximální kvalita a spolehlivost.

Teoretické testování

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 (vizte Obtíže a jejich řešení část B).

Testování jednotlivých funkcionalit

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.

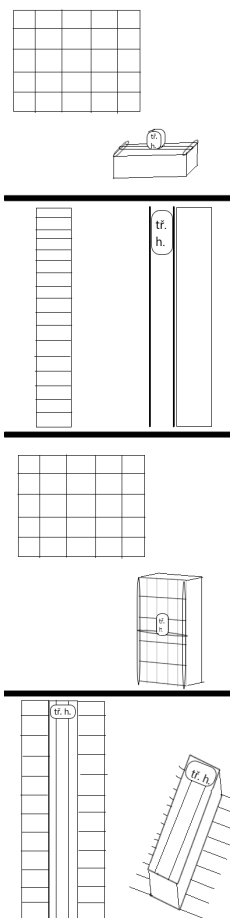
Ve druhé verzi došlo k celkové změně software, kvůli zjištěným nedostatkům v efektivitě. Bylo tedy třeba testovat jak software, tak hardware zcela nanovo. Po otestování třídění karet podle názvu přišlo testování rozpoznání názvu ve fotce karty, poté kombinace těchto dvou. Ta z počátku probíhala manuálně, později za využití dalšího programu na sehnání testovacích obrázků běžela dokonce automaticky na tisících karet 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)

Obtíže a jejich řešení

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

A. 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



B. 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 kruciólní pro funkci stroje

- c. 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 by bylo možné realizovat také pomocí tranzistorů, avšak takto byla aplikace snazší a má menší riziko na selhání než vlastnoručně sestavené obvody. Tato varianta také znamenala nutnost separátního zdroje pro Raspberry, poněvadž Arduino nebylo použitelné jako regulátor napětí.
- C. Pohyb třídicí hlavy v rámci konstrukce:
- a. Kolečka - pokud by se hlava pohybovala po kolejkách, bylo by nutné ji pohánět zevnitř, což by přidávalo hmotnost a tím zhoršovalo stabilitu celého stroje
 - b. 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
 - c. 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
 - d. Závitová tyč - závitová tyč je vhodným řešením jak pro přesnost, tak pro přesunutí motoru mimo samotnou třídicí 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, avšak tyto tyče nejsou běžně dostupné v potřebné délce a tloušťce.
- D. 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.
- a. 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

Závěr

V závěru práce bych rád zmínil, že výsledná verze není dokonalá, mezi její mouchy patří: potřeba dvou zdrojů napětí, nekompatibilita se starými kartami a kartami s alternativním rozložením a rychlost. 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 nějž 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 a náročnostně odpovídající, poněvadž čas strávený nad získáváním znalostí a jejich uplatňováním byl v řádu vyšších desítek hodin, možná více.