

Česká zemědělská univerzita v Praze

Provozně ekonomická fakulta

Katedra informačního inženýrství



Bakalářská práce

Generátor QR kódu podle grafického vzoru

Jan Jiroušek

© 2018 ČZU v Praze

ZADÁNÍ BAKALÁŘSKÉ PRÁCE

Jan Jiroušek

Informatika

Název práce

Generátor QR kódu podle grafického vzoru

Název anglicky

QR code generator by graphical pattern

Cíle práce

Cílem práce je vytvořit software, který bude schopen generovat QR kód podle zadané grafické předlohy. Logo, které bude použito jako šablona, bude viditelné a rozeznatelné v QR kódu.

Díličí cíle práce jsou:

- analyzovat vstupní grafiku pro potřeby programu
- aplikovat analyzovanou předlohu do QR kódu
- zkontrolovat dostupnost vygenerovaného QR kódu

Metodika

Použitá metodika řešené problematiky bakalářské práce je založena na analýze grafických předloh a vývoji software. Grafické předlohy budou brány jako stěžejí podoba generovaných QR kódů. Na podkladě poznatků a výsledků (vygenerovaných QR kódů) budou formulovány závěry této bakalářské práce a následně zobecněny pro další možná využití.

Doporučený rozsah práce

30-60 stran

Klíčová slova

analýza grafické předlohy, generování QR kódu, QR kód podle předlohy

Doporučené zdroje informací

PARKER, J. R. Algorithms for image processing and computer vision. New York: Wiley Computer Pub., c1997. ISBN 04-711-4056-2.

PECINOVSKÝ, Rudolf. Myslíme objektově v jazyku Java: kompletní učebnice pro začátečníky. 2., aktualiz. a rozš. vyd. Praha: Grada, 2009. Myslíme v—. ISBN 978-80-247-2653-3.

WATERS, Joe. QR codes for dummies. Portable ed. Hoboken, NJ: John Wiley, c2012. –For dummies. ISBN 978-1118337035.

Předběžný termín obhajoby

2017/18 LS – PEF

Vedoucí práce

doc. Ing. Vojtěch Merunka, Ph.D.

Garantující pracoviště

Katedra informačního inženýrství

Elektronicky schváleno dne 11. 1. 2018

Ing. Martin Pelikán, Ph.D.

Vedoucí katedry

Elektronicky schváleno dne 11. 1. 2018

Ing. Martin Pelikán, Ph.D.

Děkan

V Praze dne 07. 03. 2018

Čestné prohlášení

Prohlašuji, že svou bakalářskou práci "Generátor QR kódu podle grafického vzoru" jsem vypracoval samostatně pod vedením vedoucího bakalářské práce a s použitím odborné literatury a dalších informačních zdrojů, které jsou citovány v práci a uvedeny v seznamu použitých zdrojů na konci práce. Jako autor uvedené bakalářské práce dále prohlašuji, že jsem v souvislosti s jejím vytvořením neporušil autorská práva třetích osob.

V Praze dne _____

Poděkování

Rád bych touto cestou poděkoval svému vedoucímu práce panu docentu Merunkovi, který mi pomáhal a usměrňoval mě k dosažení cíle. Dále bych rád poděkoval své rodině, která mi dodávala motivaci při plnění této bakalářské práce.

Generátor QR kódu podle grafického vzoru

Abstrakt

Práce se zabývá QR kódy a jejich možným grafickým vylepšením. Současné QR kódy, které se používají pro reklamní účely, neobsahují žádnou grafiku. Vidíme pouze jakési černé a bílé čtverečky, které nám nic neříkají. Hovořím zde o grafickém vylepšení, které je na principu vložení zadaného vstupního obrázku do matice QR kódu. Tato akce poté nijak neomezí čitelnost, nezvýší časovou náročnost nebo nevytvoří chybovost překladu.

V práci je popsán stručný princip a způsob vytváření QR kódů. Dále je zde vysvětlen způsob práce se zadanou vstupní grafikou pro potřeby vytvořeného programu a jejich následnou aplikací do vytvořeného QR kódu.

Po vytvoření výsledného QR kódu je proveden test čitelnosti a překladu pro čtečky v porovnání s klasickým QR kódem bez vloženého loga. Ke konci práce je uvedeno vyhodnocení testů a určení závěru.

Klíčová slova: QR kódy, vložená grafika, tvorba QR kódu, vylepšení QR kódu, software, barevné kombinace matice, kontrast barev, marketing, java

QR code generator by graphical pattern

Abstract

The project consist of QR codes and their possible graphical improvement. Nowadays QR codes, which are used for advertising purpose do not contain graphic. It only seems like black and white little squares which tell us nothing. I mean an graphical enhancement that is based on the insertion of an image in a matrix of the QR code. This act does not limit a readability, does not increase a duration and does not make an error during a translation.

A brief principle and a way of making QR codes is described in this project. Secondly, is there explained a way of working with chossen input graphic for need of created software and their next application in created QR code.

After creating final QR codes a test of readibility and translation (for QR readers in comparation with classic QR code without input logo) is done.

Keywords:

A QR codes, embedded graphics, creating a QR code, improvement of a QR code, software, color variations of matrix, color contrast, marketing, java

Obsah

1	Úvod	12
2	Cíl práce.....	13
3	Metodika	14
3.1	Struktura práce	14
I.	LITERÁRNÍ REŠERŠE.....	15
4	O QR kódech	16
4.1	Online generátory a dekodéry.....	17
5	Tvorba QR kódu	18
5.1	Pixel vs modul.....	18
5.2	Matice bitů a její povinné části	18
5.2.1	Tichá zóna (Quiet zone).....	18
5.2.2	Orientační vzory (Finder patterns).....	19
5.2.3	Časovací vzory (Timing patterns).....	19
5.2.4	Seřizovací vzory (Alignment patterns)	19
5.2.5	Samostatný tmavý modul.....	19
5.2.6	Informace o verzi (Version and format information)	20
5.2.7	Datové a korekční bity	20
5.3	Výběr korekční úrovně	21
5.4	Zjištění velikosti (verze) QR kódu.....	21
5.5	Převod zprávy na bity	21
5.6	Výpočet korekčních bitů.....	22
5.7	Umístění bitů do matice	22
5.8	Maskování.....	23

5.8.1	Masky a jejich aplikace.....	23
5.8.2	Výběr nejvhodnější masky.....	24
5.9	Informační bity.....	25
6	Marketing	28
6.1	Současný stav.....	29
6.2	Zlepšení uplatnění QR kódů v reklamě	30
II.	PROJEKT.....	31
7	Struktura zdrojového kódu.....	32
7.1	Diagram aktivit	33
8	Uživatelské rozhraní aplikace.....	34
8.1	Ovládací prvky.....	34
9	Přidaná grafika do matice bitů	36
9.1	Použitá metoda a princip.....	36
9.1.1	Načtení obrázků	36
9.1.2	Převod obrázků	37
9.1.3	Vložení obrázků do vytvořeného QR kódu	38
9.2	Výsledný QR kód.....	38
9.3	Omezení čitelnosti	39
10	Testování vygenerovaných QR kódů	40
10.1	Testování pod různými úhly	41
10.2	Testování při různé světelné intenzitě.....	42
10.3	Testování při různé vzdálenosti	43
10.4	Testování určení kontrastu barevných kombinací loga	44
11	Vyhodnocení testování.....	46

11.1	Dosažené výsledky.....	48
12	Závěr	49
13	Použitá literatura	50
14	Internetové zdroje.....	51
III.	PŘÍLOHY	52

1 Úvod

QR kódy jsou všude kolem nás. Najdeme je na billboardech, obalech zboží, reklamních bannerech a na mnoho dalších snadno viditelných místech. Problémem je, že při samotném pohledu na dotyčný QR kód nevíme, respektive nevidíme, s jakou firmou máme tu čest. Vidíme pouze změť černých a bílých čtverečků, které nám nic nenapoví nebo neřekne. Proto musí být někde vedle na reklamní ploše zobrazeno logo firmy, které nás dále informuje o dotyčné firmě.

Na současném trhu již existují QR překladače, které umožňují vkládat loga do QR kódu. Nicméně loga jsou umístěna na QR kód a dochází k překrývání modulů. Musí se používat vysoká korekční úroveň a matice bitů je zbytečně velká a hustá.

Mým řešením je vložit logo firmy/společnosti do matice QR kódu a zanechat stejnou čitelnost za nulové (optimální) ztrátivosti při překladač. Tím se ušetří reklamní plocha pro další nutná reklamní sdělení, v této ploše by jinak bylo umístěno logo. Dosáhneme tím i grafické přívětivosti, která potencionální zákazníky upoutá a přiláká.

2 Cíl práce

Cílem práce je vytvořit software, jenž bude schopen vytvořit QR kód (přeložit zadanou zprávu) a dále do matice umístit logo, které uživatel zadá. Logo v QR kódu bude viditelné a rozeznatelné pohledem. Dále se zachová v rámci možností základní čitelnost a schopnost překladačů kódu pro čtečky.

Výsledné QR kódy se následně otestují, ověří se jejich čitelnost, schopnost překladačů v porovnání s klasickým QR kódem a určí se závěr.

3 Metodika

K dosažení zadaného cíle jsem naprogramoval samotný QR překladač. Ke správnému pochopení a fungování překladačů je nutné se naučit jejich princip vytváření a fungování. Největší potíž jsem zaznamenal při generování korekčních bitů, v nichž se používá několik matematických algoritmů.

Jakmile jsem měl hotový překladač, mohl jsem se pustit do načítání zadaných obrázků a následně je zpracovat. Tyto obrázky jsem si “přizpůsobil” pro potřeby mého překladače s co největší možnou kvalitou, kterou jsem byl schopen dosáhnout. Do vytvořeného QR kódu jsem poté umístil (sloučil) načtený obrázek, dále jsem určil optimální a kontrastní barvy bitů.

Poté jsem testoval vygenerované QR kódy pomocí různých testů. Testoval jsem dva vzorky QR kódu, které měli do sebe zakódovanou stejnou zprávu. První vzorek obsahoval vložené logo a druhý vzorek byl bez loga. Testoval jsem čitelnost a rychlost překladače při různých světelných podmínkách, při různé vzdálenosti čtečky nebo barevné kombinaci a kontrastu, při kterém bylo QR kód možno přečíst.

3.1 Struktura práce

1. V první kapitole popíšu stručně QR kódy
2. V druhé kapitole popíšu jak se QR kódy vytvářejí
3. Ve třetí kapitole popíšu můj princip načtení obrázků a jejich sloučení s QR kódem
4. Ve čtvrté kapitole popíšu jednotlivé testy a jejich výsledky
5. V páté kapitole vyhodnotím testování vytvořených QR kódů s logem a jejich čitelnost oproti QR kódu bez loga

I. Literární řešení

4 O QR kódech

QR kódy je poměrně nová technologie vymyšlená v roce 1994. Je to technologie, které by společnost měla dávat větší pozornost. Možnosti využití QR kódů jsou vsutku neomezené.

QR (Quick Response) kódy jsou jedny z mnoha dvojrozměrných kódů pro komerční, nebo nekomerční využití. Základní myšlenkou je přenos zprávy nebo sdělení, které se rychle zakóduje bez ztrátovosti a následným rychlým převodem zpět na původní zprávu při nulové (optimální) chybovosti.

Původní jednorozměrné BAR kódy mohli nést maximálně pouhých 20 znaků, což značně omezovalo jejich použití. Způsob čtení BAR kódů je na mechanickém principu pomocí laserového paprsku. Na to je potřeba speciálních čtecích zařízení, které jsou finančně náročnější.

Na druhou stranu QR kódy mohou nést nejen 100krát více informací, ale mohou být i digitálně čitelné (vyfocené) a následně přeloženy interním systémem nebo aplikací. Jednoduchá kamera na mobilním telefonu může číst tyto kódy, které následně v reálném (téměř okamžitém) čase přeloží a zobrazí výsledek. K tomu nám slouží spousta zdarma dostupných čteček. Některé systémy mají již zabudovaný QR dekodér, kde pak není potřeba externích aplikací, které nám zbytečně blokují prostor na disku zařízení.

QR kódy mohou nést nejen číslice, alfanumerické znaky, ale také znaky Kanji, Kana, symboly nebo různé kontrolní kódy. Maximální kapacita těchto kódů je 7,089 znaků [Waters. c1997].

Do matice QR kódů jsou vloženy korekční bity, které umožňují zpětný správný překlad pro například nějakým způsobem poškozené QR kódy. Díky těmto bitům získaných na základě korekčních algoritmů je možno obnovit až 30% kódových slov. Kódové slovo je jednotka, která představuje 8 po sobě následujících bitů. Samozřejmě možnost obnovení je plně závislé na poškození QR kódu [Thonky.com, 2015a].

Čtení QR kódů pomocí různých čteček není nijak orientačně závislé. Je možno QR kód skenovat v rozmezí 360 stupňů. Tomuto úkolu slouží tři velké čtverce v rozích, které jsou vidět pohledem oka. Tyto velké tři pozičně detekující čtverce informují čtečku o skutečnosti, že se jedná o QR kód a dále také slouží jako orientační a seřizovací body. Díky nim se čtečky seřídí a zorientují.

Díky skutečnosti, že je zde možno zakódovat jakékoli informace nebo sdělení, možnosti využití těchto kódů jsou nevyčerpatelné. Do QR kódů se může zakódovat čistý text (plain text), vizitka, telefon, URL odkaz nebo také odeslat sms či email s předdefinovaným textem.

4.1 Online generátory a dekodéry

Na internetu je již k dispozici celá řada všemožných generátorů nebo dekodérů QR kódů. Drtivá většina jich je zdarma a jejich ovládání je jednoduché a intuitivní.

Tyto generátory nám poskytují různé možnosti zakódování zprávy. Například vytvoření vizitky, URL odkazu, emailu, telefonu, polohy a mnoho dalšího. Tato práce se zabývá možností zakódováním čistého textu (plain textu).

Online webové dekodéry jsou jednoduché aplikace, které přeloží zadaný QR kód zpět na původní zprávu. Po přečtení a přeložení zadaného speciálního QR kódu (vizitka, URL, email, atd..) jsou tyto překladače schopny poskytnout výchozí akce pro dotyčný typ QR kódu.

5 Tvorba QR kódu

Tvorba QR kódu se skládá z několika na sebe navazujících kroků. Každý z těchto kroků je důležitý a jeho vynechání by znamenalo jak ztrátovost zpětného překladu, tak by nejspíše ani nešel vytvořit. Níže popíšu v jednoduchosti stručně všechny kroky tvorby QR kódu a jeho důležité části [Thonky.com, 2015a].

5.1 Pixel vs modul

Důležité je, si uvědomit rozdíl mezi pixelem a modulem. Většinu lidí napadne, že QR kódy jsou tvořeny z pixelů a myslí se tím tmavé nebo světlé čtverečky bez ohledu na to, jak jsou veliké. U QR kódů mluvíme o tzv. modulech.

Modul, jeden z čtverečků, ze kterých se QR kód skládá. Tento modul, může mít jakoukoli velikost. Čtečky, se pak na tuto velikost nastaví a přečtou celý QR kód. Jeho barva musí být jednoznačně kontrastní od druhé barvy. Pokud mezi těmito dvěma barvami není vysoký kontrast, čtečky QR kódů mají problém a vzniká ztrátovost nebo chybovost při překladu [Thonky.com, 2015b].

5.2 Matice bitů a její povinné části

Struktura matice má dále určené a vymezené zóny. V těchto zónách je určeno, jaké informace má obsahovat, o čem má informovat nebo jestli se jedná pouze o orientační moduly. Níže budou vypsány povinné zóny a oblasti.

Pro moji potřebu v kódu využívám jednoduché dvojrozměrné pole, kam ukládám jednotlivé bity (čísla). Neukládám pouze hodnoty 0 nebo 1, využívám hodnoty od -1 do 9. Díky tomu jsem schopen rozeznat o jakou část, respektive modul/šablonu se jedná. Liché čísla jsou použity pro tmavé bity a sudé pro světlé.

5.2.1 Tichá zóna (Quiet zone)

Tichá zóna, je okolí QR kódu, která nesmí nic obsahovat. Zde platí, že má mít velikost 4 modulů na všechny strany od QR kódu. Slouží především pro čtečky [esponce.com, 2018].

Dále nám tato zóna určuje, jaká bude světlá barva QR kódu. Na základě této barvy a dále nalezené tmavé barvy se vypočítá průměr, podle kterého se pak rozlišují jednotlivé bity.

5.2.2 Orientační vzory (Finder patterns)

Orientační vzory jsou tři hlavní čtverce, jejichž úkolem je čtečku informovat o skutečnosti, že se jedná o QR kód. Další jejich důležitou funkcí je orientace čtečky. Můžeme díky nim skenovat vzhůru nohama bez závislosti na ztrátovost překladu [esponce.com, 2018].

Jsou umístěny ve třech rozích. Tyto čtverce jsou umístěny ve třech rozích kromě pravého-dolního rohu. Celková velikost je 7×7 , kde kraj je tmavé barvy, uvnitř je o rozměrech 5×5 světlý obvod a nakonec nám zbývá 3×3 černý čtverec.

Orientační vzory se od zbytku QR kódu oddělují bílým pruhem o šířce/výšce jednoho modulu.

5.2.3 Časovací vzory (Timing patterns)

Pro čtečky QR kódu je důležité zjistit, jak jsou velké moduly a podle toho se seřídí a nastaví. K tomu slouží tzv. timing patterns. Lze je vidět a najít pohledem na QR kód. Najdeme je mezi finder patterns jako spojující sloupec nebo řádek, kde se střídají moduly o bitech 1 nebo 0 (tmavé moduly a světlé) [esponce.com, 2018].

5.2.4 Seřizovací vzory (Alignment patterns)

Pokud máme QR kód o větší velikosti (jedná se o velikosti 2 a výš), je nutné pro QR čtečky vložit do matice seřizovací bloky, které pomáhají čtečkám k seřízení a orientaci. Pokud bude čtený kód nějakým způsobem deformovaný nebo ohnutý, čtečky se díky nim seřídí a zorientují. QR kód o nejmenší velikosti (velikost 1) tyto bloky nevyžaduje [esponce.com, 2018].

Čím je hustota modulů větší, tím více je potřeba těchto bloků. Velikost seřizovacích bloků je 5×5 černého (tmavého) obvodu, uvnitř je 3×3 bílý (světlý) obvod a zbývá nám 1×1 modul černé (tmavé) barvy.

5.2.5 Samostatný tmavý modul

Jedná se o samostatný modul, který se nachází na souřadnici, kterou vypočítáme jako: $((4 * V) + 9), 8$. Kde V je verze QR kódu [Thonky.com, 2015b].

5.2.6 Informace o verzi (Version and format information)

Pro správný překlad čteček musíme zachovat informace o verzi a použitém formátu. Na to nám slouží další bloky, respektive jeden shluk informací, které se rozprostřou do více míst v matici.

Pro verze menší než 7 se používá řetězec bitů o délce 15 bitů. Tento řetězec je vyobrazen v matici dvakrát. První je umístěn u levého-horního orientačního vzoru, kde bity začínají od pravého horního bodu, hnedka vedle oddělovače a řetězec pokračuje dolů. Pokud narazí na časovací vzor, jednoduše ho přeskočí a pokračuje dál podél oddělovače od orientačního vzoru. Jakmile řetězec dosáhne hrotu oddělovače, pokračuje doleva až k levému kraji matice. Druhý se nachází pod pravým orientačním vzorem, kde začíná opět hned vedle oddělovače u pravé strany matice a postupuje doleva ke středu. Jakmile řetězec se dosáhne posledního horizontálního modulu oddělovače, pokračuje u levého dolního orientačního vzoru hned pod samostatným tmavým modulem.

Pro verze větší jak 7 (včetně) se používají pouze dvě oblasti pro informační bity. První se nachází nad levým dolním orientačním vzorem a má velikost 6x3 modulů. Druhý leží zleva u vedle levého horního orientačního vzoru o velikosti 3x6 modulů [Thonky.com, 2015c].

5.2.7 Datové a korekční bity

Ve zbytku matice máme samotná zakódovaná data, ale také korekční bity. Tyto bity jsou vypočítány pomocí polynomů. Počítání těchto bitů bude rozebráno v dalších kapitolách.

Datové a korekční bity se do matice vkládají z pravého-dolního rohu po dvojicích směrem nahoru. Jakmile dosáhneme situace, ve které již nemůžeme pokračovat nahoru, směr se otočí a opět bity pokračují po dvojicích dolů. Pokud se v cestě objeví seřizovací vzor, jednoduše ho přeskočíme a pokračuje se dál.

Do matice se nejdříve vkládají data, ty mají svou délku v závislosti na délce zprávy. Poté se hned pokračuje korekčními bity. V nejlepším případě, nám délka dat a korekčních bitů vyjdou přesně na počet volných pozic v matici. Často se stává, že nám došla jak data, tak korekční bity a v matici jsou stále prázdné bity. V tomto případě se do zbylých bitů vkládá terminátor, který čtečku informuje o konci zprávy [Thonky.com, 2015b].

5.3 Výběr korekční úrovně

Před zakódováním vstupní zprávy se musí určit korekční úroveň. QR kódy využívají Reed-Solomon korekční algoritmy, které mají další využití například v na diskách CD, DVD, Blu-ray nebo také v satelitní komunikaci.

Pomocí těchto algoritmů získáme na základě zakódovaných dat korekční bity, se kterými se dále pracuje a vkládají se do matice. Existují celkem čtyři korekční úrovně: L, M, Q, H. Každá úroveň obnovuje jiné procento dat. Čím větší máme toto procento, tím více je potřeba bitů a hustota matice roste, případně se mění i verze (velikost) QR kódu. V následující tabulce zobrazeny procentuální obnova dat.

Korekční úroveň	Korekční kapacita
L	Obnovuje 7% dat
M	Obnovuje 15% dat
Q	Obnovuje 25% dat
H	Obnovuje 30% dat

Tab. 1 Korekční kapacity [qrstuff.com, 2011]

5.4 Zjištění velikosti (verze) QR kódu

U QR kódů se velikost nazývá verzí. Nejmenší je verze 1 (21x21 modulů) a největší je verze 40 (177x177 modulů). Platí zde jednoduché pravidlo, kde každá další verze je o 4 moduly větší než předchozí [Esponce.com, 2018].

K určení velikosti se používá kapacitní tabulka velikostních kapacit, která je dostupná na odkazu, který je uveden v příloze. Pro správné určení musíme zjistit délku zadávaného řetězce k zakódování. Tuto délku pak porovnáváme v tabulce a hledáme nejnižší možnou kombinaci spolu s korekční úrovní. Např.: Řetězec má délku 11 bitů a vybraná korekční úroveň je Q. Použijeme tedy verzi 1.

5.5 Převod zprávy na bity

Následující postup budu psát pouze pro alfanumerický mód, jelikož pro ostatní módy se používá jiné kódování a můj naprogramovaný software umí „pouze“ alfanumerický mód.

Prvním krokem převodu zprávy na bity je určit mód konverze. Pro alfanumerický mód je indikátor řetězec 0010. K indikátoru dále připojíme řetězec, který nám udává délku kódované zprávy. Ten získáme na základě verze QR kódu a převedením do n-bitové soustavy (zobrazeno v tabulce).

Verze QR kódu	Počet bitů
1 – 9	9
10 – 26	11
27 – 40	13

Tab. 2 Počet bitů na verzi [Thonky.com, 2015d]

Dalším krokem je zakódování samotné zprávy. Algoritmus funguje na principu: Postupně bereme dvojice znaků. Hodnotu prvního znak vynásobíme 45 a přičteme k němu hodnotu znaku druhého. Hodnoty znaků jsou určeny v tabulce, která je dostupná na odkazu, který je uveden v přílohách. Výsledek pak převedeme do n-bitové soustavy, kde n je počet znaků kódovaného řetězce. Pokud nám na poslední znak nevyjde do dvojice, postupujeme stejně, ale výsledek převedeme do šesti bitového řetězce.

V minulém kroku jsme zjistili verzi QR kódu. V našem příkladu jsme získali 1-Q, kde je kapacita 13 znaků. Když to vynásobíme 8, získáme 104 bitů. Musíme tuto kapacitu vyplnit a využít. Pokud mód indikátor + počet znaků indikátor + zakódovaná data je menší než kapacita, přidáme tzv. terminátor, který má tvar 0000. Po přidání terminátoru musíme zaručit, že celková délka je násobku osmi. Pokud ne, na konec řetězce přidáme počet nul, který nám tento násobek splní. Jestli máme řetězec stále krátký na to, aby vyplnil kapacitu (např.: 104 bitů), přidáváme postupně následující řetězce: 11101100 nebo 00010001. Tyto řetězce mezi sebou střídáme a přidáváme, dokud není kapacita vyplněna [Thonky.com, 2015d].

5.6 Výpočet korekčních bitů

Pro výpočet korekčních bitů se využívá práce s polynomy a jejich kombinací mezi sebou v cyklu do určité podmínky, která umožní pokračovat do dalšího kroku. Dále se v tomto kroku využívá log – antilog tabulka, která nám udává jaké decimální hodnoty jsou pro dotyčnou alfu a opačně. Pro výpočet bitů se používá několik na sebe navazujících výpočtů. Jejich princip zde nebude popsán, jelikož se jedná o velmi rozsáhlou kapitolu. Odkaz, na kterém je vysvětlen postup a podrobný princip výpočtů bude uveden v přílohách.

5.7 Umístění bitů do matice

Jakmile máme zprávu zakódovanou a máme vypočítané korekční bity, následuje jejich umístění do matice. Bity se vkládají v tomto pořadí [Thonky.com, 2015b]:

1. Indikátor módu
2. Indikátor počtu znaků

3. Zakódovaná zpráva, včetně přidaného terminátoru a případných kapacitních bitů
4. Korekční bity

5.8 Maskování

Je část, při které aplikujeme na dosavadní matici bitů masky, které nám přetvářejí celkovou matici.

Dále vybíráme z osmi vytvořených masek tu, která bude mít co nejmenší odchylku a chybovost. S vybranou maskou se již dále nebude nijak pracovat (co se týče datových a korekčních bitů). Maskováním zajistíme, že čtečky budou mít co nejméně práce s přepočítáváním a překladem, jelikož jsme vybrali tu „nejjednodušší“ na překlad.

Maskování využívá pouze tu část matice, kde jsou obsaženy datové bity spolu s korekčními bity. Orientační, seřizovací, časové vzory se všechny vynechávají a přeskakují, maska se na ně neaplikuje. Jak jsem již psal dříve, pro moji potřebu si ukládám jednotlivé hodnoty pro určité části a šablony. Díky tomu, jsem schopen maskování zrychlit a zjednodušit. Maskování je operace, která je poměrně výkonově náročná a má vysokou složitost. Je důležité zde složitosti nějakým způsobem pomoci a tím výpočet zrychlit [plcviewer.com, 2011].

5.8.1 Masky a jejich aplikace

Masky určují, které bity zůstanou bílé (světlé) a tmavé (černé). Podle toho jaká maska se použila, matice se mění. Pro správný překlad, musíme zachovat informaci o tom, kterou masku jsme použili. Tato informace se potom zakóduje a vloží do informačních bitů.

Každá maska má svůj vzorec, pomocí kterého vypočítáváme hodnoty bitů. Vždycky se jedná o nějaké pravidlo, které když platí, mění se hodnota bitu a tudíž barva. Do vzorce se dosazuje pozice v řádku a sloupci. Ve výsledku dostaneme matici, která vypadá jinak než původní (matice před maskováním). V následující tabulce jsou zobrazeny vzorce pro jednotlivé masky.

Číslo masky	Pravdivostní vzorec pro masku
0	$(\text{row}^1 + \text{column}^2) \bmod^3 2 == 0$
1	$(\text{row}) \bmod 2 == 0$
2	$(\text{column}) \bmod 3 == 0$
3	$(\text{row} + \text{column}) \bmod 3 == 0$
4	$(\text{floor}(\text{row} / 2) + \text{floor}(\text{column} / 3)) \bmod 2 == 0$
5	$((\text{row} * \text{column}) \bmod 2) + ((\text{row} * \text{column}) \bmod 3) == 0$
6	$((\text{row} * \text{column}) \bmod 2) + ((\text{row} * \text{column}) \bmod 3) \bmod 2 == 0$
7	$((\text{row} + \text{column}) \bmod 2) + ((\text{row} * \text{column}) \bmod 3) \bmod 2 == 0$

Tab. 3 Vzory pro masky [Thonky.com, 2015e]

5.8.2 Výběr nejvhodnější masky

Jakmile získáme maskované matice je na čase vypočítat pro každou masku její chybovost a tu dále porovnat se zbylými maskami. Chybovost je hodnota, jakési skóre, které nám udává množství podobnostních chyb s ostatními povinnými částmi matice. Masky s nejnižším chybovým skóre se vybere pro finální výstup [Thonky.com, 2015e].

Výpočet skóre pro jednu masku je součtem čtyř pravidel. Každé pravidlo kontroluje a hledá jinou sekvenci bitů a každé pravidlo tyto nalezené sekvence ohodnocuje jiným měřítkem. Jakmile algoritmus pro kontrolu najde stejnou sekvenci, jak určuje pravidlo, přičte se k celkovému skóre odpovídající postihové „body“. Do výpočtu tohoto skóre se zahrnuje celková matice včetně funkčních šablon, které se například nepoužívali do maskování.

První pravidlo hledá sekvenci pěti po sobě následujících bitů se stejnou hodnotou. Pět stejných bitů za sebou jsou ohodnoceny třemi body. Jakmile po pěti bitech následuje další stejné bity, je za každý stejný bit přičten jeden bod. Tímto způsobem projedeme horizontálně všechny řádky. Poté provedeme totéž, nýbrž vertikálně všechny sloupce. Ohodnocení je stejné jako u horizontálního vyhledávání.

Druhé pravidlo hledá čtverce o velikosti 2x2 stejné barvy. Každý takto nalezený čtverec je ohodnocen třemi body. Algoritmus vyhledává i překrývající se čtverce navzájem. To znamená, pokud máme celkem 3x3 čtverec stejné barvy,

¹ Row: číslo řádku

² Col: číslo sloupce

³ Mod: zbytek po dělení

nalezneme v něm ve výsledku čtyři čtverce 2x2. Přičemž ohodnocení zůstává pro každý čtverec stejný.

Třetí pravidlo hledá dvě sekvence bitů v celé matici, jak horizontálně, tak vertikálně. První horizontální sekvence má tvar: 1-0-1-1-1-0-1-0-0-0-0 kde (1 je tmavý bit a 0 světlý bit). Druhá, vertikální sekvence hledá podobnou sekvenci, jen je zrcadlově otočená. Tudíž se jedná o tvar: 0-0-0-0-1-0-1-1-1-0-1. Každá nalezená podobnost je vysoce ohodnocena 80 body.

Čtvrté pravidlo je početní pravidlo a obsahuje několik po sobě následujících vzorců. Postup obsahuje následující kroky:

1. Součet všech modulů v matici. Dá se určit jako velikost matice na druhou.
2. Součet všech tmavých bitů v matici.
3. Spočítat procentuální podíl tmavých bitů ku celkovému počtu bitů.
4. Z vypočítaného procentuálního podílů vybereme nejbližší dvě hodnoty násobku pěti. Např.: pokud máme průměr tmavých bitů 62%, nejbližšími násobky jsou 60 a 65.
5. Od dvou získaných hodnot z minulého kroku odečteme v absolutní hodnotě hodnotu 50. Např.: z minulého kroku jsme získali 60 a 65 a po aplikaci aktuální kroku získáme 10 a 15.
6. Získané hodnoty z předešlého kroku dáme do čitatele zlomku. Ve jmenovateli bude hodnota 5. A vypočítáme pro každou hodnotu.
7. Pro výsledek použijeme menší hodnotu z kroku č. 6. Tu vynásobíme hodnotou 10. Tím získáme výsledek, který nám vyjadřuje body za čtvrté pravidlo.

Jakmile spočítáme všechny čtyři pravidla, sečteme jejich výsledky a získáme chybové skóre, které porovnááme s ostatními maskami. Masku s nejnižším skórem vybereme pro další postupy a finální podobu matice bitů.

5.9 Informační bity

Informační bity nám uchovávají informace o použité masce a korekční úrovni. Celkem máme čtyři korekční úrovně, k tomu dále je možno použít sedm mask, ve výsledku můžeme získat až 28 informačních řetězců.

Pro verze QR kódu menší než 7 je informační řetězec vždy 15 bitů dlouhý. Prvních pět bitů představuje korekční úroveň a použitá maska. Pomocí prvních pěti bitů vygenerujeme dalších deset bitů (podrobněji níže). Pro správný výsledek

je potřeba na všech 15 bitů aplikovat masku 101010000010010 pomocí XOR [Thonky.com, 2015c].

Prvním krokem je vytvoření prvních pěti bitů. Každá korekční úroveň představuje číselný ekvivalent. První dvojici bitů tedy získáme z tabulky:

Vybraná korekční úroveň	Bity	Číselný ekvivalent
L	01	1
M	00	0
Q	11	3
H	10	2

Tab. 4 Číselné ekvivalenty [Thonky.com, 2015c]

Za tuto dvojici bitů se dále připojí použitá maska (její číslo), převedená do tří bitového řetězce. Např.: Použili jsme korekční úroveň H (10) a maska číslo 4 (100) – výsledný řetězec odpovídá zápisu 10100.

Dalších deset bitů získáme použitím prvních pěti bitů (formátu) a následujícího řetězce (polynomu) 10100110111. Algoritmus spočívá na principu [Thonky.com, 2015c]:

1. Vezmeme formát (vstup), odebereme nuly zleva a přidáme tolik nul zprava, abychom získali řetězec délky 15.
2. Na získaný formát aplikujeme polynom pomocí XOR a odebereme nuly zleva.
3. Pokud výsledný formát je delší než 11 bitů opakujeme všechny předešlé kroky (vstup je výsledný formát).
4. Jakmile formát je kratší 11 bitů na jeho začátek přidáme původní formát (o délce pěti bitů).
5. Pro výsledek aplikujeme na formát masku 101010000010010 pomocí XOR a získáváme informační řetězec, který se může umístit do matice.

Pro QR kódy verze větších než 7 (včetně) se používá řetězec o délce 18 bitů. Algoritmus pro výpočet je stejný jako pro menší verze. Prvních šest bitů obsahují číselné vyjádření verze QR kódu převedené do šestibitového řetězce následovaných dvanácti bity korekčních bitů.

Algoritmus je skoro stejný jako pro menší verze. Změna je taková, že opakujeme cyklus do doby, kdy máme řetězec o délce menší než dvanáct. Polynom je taky jiný, jeho tvar je 111100100101.

K výslednému řetězci z algoritmu opět přidáme zleva prvních šest bitů, které představují verzi QR kódu a můžeme vložit do matice [Thonky.com, 2015c].

6 Marketing

V dnešní době jsou dvojdimenzionální čárové kódy vesměs standardem pro marketing a to převážně ve světě, zejména v Japonsku. Na rozdíl od České Republiky, kde se tyto kódy využívají málo a lidé je vnímají spíše jako kuriozitu než samozřejmost.

Nicméně drtivá většina lidí již vlastní chytrý telefon, který má zabudovanou vnitřní aplikaci fotoaparát, jenž zároveň umožňuje číst čárové kódy. To je ale pro většinu Čechů velká neznámá. Pokud jdeme po ulici, vidíme spoustu lidí, kteří jdou a koukají do svého mobilního telefonu. Není nic jednoduššího, než lidem postavit „do cesty“ připravené QR kódy. Po načtení a přečtení QR kódů je vyhráno a reklama se začne šířit.

Problémem u nás v ČR je nejspíše fakt, že některé firmy se neumí s využitím QR kódů jako zdrojem reklamy vypořádat. S tím v zásadě souhlasím, dle mého názoru, podloženého četbou internetových zdrojů, (týkajících se marketingu firem) jsem toho mínění, že České firmy zatím nevědí, jak QR kódy využít a používají to zatím spíš jako důkaz, že jsou moderní.

Firmy se zatím jen lehce snaží inspirovat v zahraničí. To je v zásadě správné, ale zatím nevědí, jak s informacemi pracovat.

Četnost využití QR kódů se s vývojem společnosti zvětšuje. Je ale časté, že se po načtení stránky z onoho čárového kódu zákazníkovi zobrazí pouze odkaz na samotnou úvodní stránku produktu. Což není pro zákazníka moc přínosné a z jeho strany zatím uživatelsky nekomfortní.

Vypadá to ale, že tento stav bude do několika let minulostí.

6.1 Současný stav

QR kódy se používají vzhledem k jejich potenciálu stále minimálně. Tato skutečnost se každým rokem mění k lepšímu, jak jsem již výše zmínil. Dvojdímenzionální čárové kódy stále nenachází u potencionálních zákazníků větší potřebu tyto kódy „číst“. Většina lidí totiž nechápe, k čemu ony kódy jsou [Rostlá, 2009].

Musím zmínit, že se jedná i o chybu zákazníků, kteří se nesnaží dosáhnout nového zjištění, nebo je pro ně použití QR kódu zatím složitý úkon, zatímco firmy už se začali v používání zlepšovat.

I demografické složení ČR může ovlivňovat četnost využití těchto čárových kódů, a to jak z důvodu starší populace, pro niž je QR kód neznámá, tak i kvůli firmám, které si v současné době vybírají pouze mladé lidi jako svou cílovou skupinu [Kotler, 2007].

Mladší generace se ale snaží kódy začít využívat. Zde bych rád uvedl jeden z povedených využití QR kódu (v roce 2011). Tím se může pyšnit firma MALL.CZ. Sice byl její nápad převzat z jihokorejského Soulu, ale v ČR byl také úspěšný. Firma se pokusila o realizaci projektu následovně [Čermák, 2013]:

- 1. Nalepila nabídku produktu (pracích prášků) do pražského metra
 - Velmi dobrý tah, velká koncentrace lidí
- 2. Po načtení QR kódu se zákazníkovi automaticky vložil produkt do košíku a kód ho odkázal na internetovou stránku obchodu
- 3. Na další kliknutí měl zákazník objednáno
 - Tento projekt byl hodnocen jako jednoduchý a efektivní

Firma Raiffeisenbank se také pokusila o realizaci projektu s QR kódy a zavedla tak provádění plateb skrze jejich načtení. O využití QR plateb se již začaly snažit i obchody a e-shopy.

Až společnost zjistí, že tyto kódy jim značně ulehčují práci, bude i pro ně samotné spousta věcí jednodušší.

6.2 Zlepšení uplatnění QR kódů v reklamě

Aby došlo ke zlepšení uplatnění QR kódů je důležité, aby se firmy nebály a začaly se více zapojovat v jejich využití.

Dobrým krokem může být i výzkum, který provedla firma insightlab, specialista na zákaznické komunity a inovativní výzkum. Tato firma se pokusila o sociologický výzkum, jenž realizovala na podzim roku. Ve výzkumu se lidí dotazovala na otázky ohledně využití QR kódů v běžném životě. Začala pozvolna s otázkou, zda lidé vědí, o co se jedná a poté přišla otázka, na jakých místech by QR kódy lidé uvítali. Jako nejlepší místo se ukázala restaurace. Firma vidí využití těchto čárových kódů jako možnost pro zákazníky, kteří se tímto úkonem (tedy načtení kódu) dostali ke zjištění kvality poskytovaných služeb, například v restauracích [Insightlab.cz, 2013].

Například firma futuristica, expert na mobilní technologie a mobilní marketing, se pokusila o vysvětlení, jak začít efektivně QR kódy využívat. Snažila se tak firmám ukázat, jak správně směřovat kroky k zefektivnění jejich propagace prostřednictvím QR kódů, což mi přijde jako velmi důležitý tah.

Dle mého názoru by toto vysvětlení potřebovala i veřejnost, buďto v rámci televizní reklamy, která by upozornila, že něco takového existuje či ve školách během výuky Informačních a komunikačních technologií, nebo i v rámci fyziky jako ukázky moderních inovací.

II. Projekt

7 Struktura zdrojového kódu

Struktura zdrojového kódu aplikace je dělená do několika tříd. Zajistil jsem tím základní přehlednost projektu pro snadnou orientaci v kódu. Dále jsem tím splnil jednu ze základních vlastností tříd a funkcí. Myslím tím myšlenku, kde každá třída a funkce mají mít vlastní důvod použití. Každá třída by měla obsahovat instrukce, funkce, které se vztahují pro úkoly dané třídy [Čada, 2009].

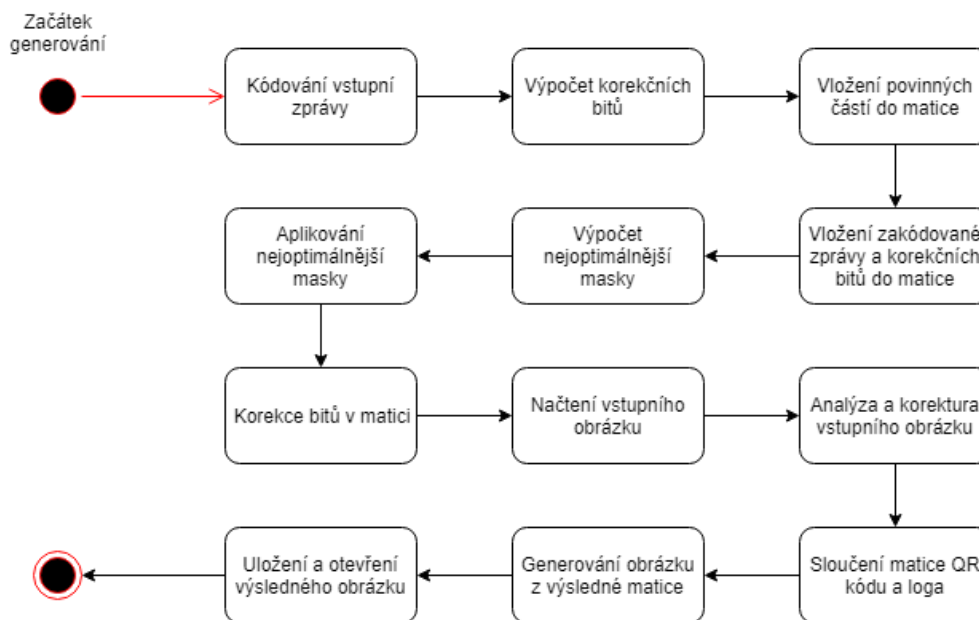
Každá třída obsahuje svoje metody a funkce. Metody využívám pro minimální opakování stejných částí kódu a tím i lepší upravitelnost a editaci kódu.

Všechny třídy obsahují tzv. „klienta“. Tento klient funguje jako mezičlánek mezi kořenovou třídou a ostatními třídami. Tím jsem zajistil možnost, kdy v nějaké metodě jakékoli třídy potřebuji data z úplně jiné třídy. Metoda vyžádá pomocí klienta tyto data, a klient vyžádaná data poskytne.

Při jakémkoli zakládání instance třídy se současně zadá parametr kořenové třídy. Tím pádem pak každá třída vlastní svého, avšak stejného klienta. Klient je druh třídy, která se nazývá interface. Právě toto rozhraní umožňuje aplikacím přenášet data mezi třídami. Všechny metody deklarované v tomto interface musí být striktně definované pod stejným názvem i v kořenové třídě [Pecinovský, 2009a].

7.1 Diagram aktivit

Následující diagram aktivit zobrazuje stručně princip postupu programu při vytváření QR kódu s logem. Tento diagram zobrazuje pouze základní aktivity, do kterých nejsou zahrnuty podaktivity v rámci aktivit, jelikož by tento diagram byl velice obsáhlý a rozsáhlý.



Obr. 1 Diagram aktivit

8 Uživatelské rozhraní aplikace

Pro co největší možnou kompatibilitu a možné víceplatformní používání aplikace jsem se rozhodl pro aplikaci v jazyku java. Aplikace bude fungovat jak na windows, linux nebo mac.

Uživatelské rozhraní jsem udělal v rámci možností jednoduché. Existuje zde dost parametrů, které uživatel může zadat nebo nastavit (text zprávy, korekční úroveň, vstupní grafika, výstupní cesta uložení QR kódu, barvy tmavých/světlých bitů). Už z tohoto důvodu aplikace musí obsahovat řadu ovládacích prvků. Nicméně počítám se širokými využitím aplikace, co se týče rozmanitosti uživatelů, od pokročilých uživatelů po běžné uživatele.

Ovládání aplikace a ovládacích prvků jsem se snažil udělat intuitivní a jednoduché. Kládl jsem snahu na co nejmenší potřebnou interakci od uživatele, aby uživatel nemusel nastavovat všechny parametry nebo prvky. Mluvím zde o interakci například, když uživatel vybere obrázek, aplikace automaticky vyplní cílovou cestu uložení výsledného QR kódu do stejné složky, jako je vstupní grafika. Dále jsem předdefinoval korekční úroveň, kontrastní barvy, které vypadají hezky.

Všechny tyto komponenty a ovládací prvky mají nastavené popisky (tooltipy), které uživateli vysvětlí a informují o příslušných nastaveních. Tím nedochází k neznalosti nebo případným nesrovnalostem v porozumění ovládání a nastavení. Tyto popisky se uživatelům zobrazují po najetí myši na příslušné ovládací prvky.

8.1 Ovládací prvky

V této části jsou zde stručně popsány jednotlivé ovládací prvky, u kterých má uživatel možnost aktivovat nebo nastavit.

Textové pole pro zprávu

Jedná se o jednoduché jednořádkové textové pole, do kterého uživatel zadává text zprávy. Není zde nastavena žádná korektura nebo jiné omezující pravidla. Text zprávy je čistě na uživateli. Je umístěno v levném horním rohu aplikace

Combobox pro výběr korekční úroveň

Rozbalovací menu, ve kterém je možno nastavit korekční úroveň. V základu je přednastavená možnost úrovně L. Tato komponenta je umístěna pod textovým polem pro zprávu.

Prvky pro výběr barev

Tyto ovládací prvky mají přednastavené barvy, které jsou odzkoušené a otestované. Samozřejmě má uživatel možnost tyto barvy ručně nastavit nebo změnit. Pokud uživatel nastaví tmavou barvu, automaticky se určí kontrastní světlá barva. Pro opačnou barvu funguje princip stejný. Vypočítaná barva má dostatečný kontrast, který by měl být pro čtečky bezproblémový.

Uživatel barvu vybírá pomocí ColorPickeru, který nabídne celou škálu barev. Tato komponenta je interní komponentou javy frameworku swing. Barvy lze vybrat klikem na příslušnou barvu z palety barev, nebo lze nastavit parametry RGB nebo HSB. Vše se nastavuje a vybírá v zobrazeném dialogu této komponenty.

Obrázky jsou pro lepší přehlednost a orientaci popsány jak nápisem, titulkem, tak i popiskem (tooltipem). Pro uživatele je v tomto popisku vysvětleno jakého nastavení (jaké barvy) se tlačítko týká.

Komponenta pro výběr obrázku a cílové cesty

Pro vybrání obrázku aplikace využívá interních knihoven a komponenty JFileChooser. Při výběru obrázku je zde nastaven filtr pouze pro obrázkové typy souborů. Tím se uživateli usnadňuje výběr a ulehčuje hledání. Po vybrání obrázku se do textového pole vloží absolutní cesta k obrázku. Uživatel má možnost ji upravit nebo zadat ručně.

Zadávání cesty pro cílový obrázek jsem se snažil uživatelům usnadnit. Po výběru zdrojového obrázku se do textového pole pro cílovou cestu vloží stejná absolutní cesta, jako je pro zdrojový obrázek. Nicméně zdrojový obrázek je odebrán z této cesty a nahrazen přednastaveným názvem obrázku. Uživatel má možnost zadat cílovou cestu pomocí stejného okna jako při výběru obrázku nebo ji zadat ručně. V případě, že nebude zadán název obrázku, program použije pro vytvoření cíle přednastavený název.

9 Přidaná grafika do matice bitů

Dostáváme se k samotnému vylepšení QR kódů. Myšlenkou je vložení obrázku do matice QR kódu, který bude korespondovat s celkovým stylem těchto kódů a zachová stejnou čitelnost s nulovou (optimální) ztrátovostí nebo chybovostí překladu. Obrázek ve výsledku je složený ze čtverců (modulů), stejně jako jsou moduly v matici. Je možno nastavit i hustotu těchto modulů obrázku, které zvýší čitelnost obrázku pro lidské oko. Nicméně po konzultaci s vedoucím práce jsem tuto možnost zakázal a ponechal pouze nastavení pro nejvyšší možnou hustotu obrázku. Obrázek je pak v QR kódu pro oko lépe čitelný.

Vkládaný obrázek je čistě na uživateli. Platí zde doporučení čtvercového vstupního tvaru a kontrastních barev. Nejlepší je použití černobílých obrázků. Nicméně algoritmus na převod pro potřeby programu zvládá celou škálu barev. Jde zde pouze o málo kontrastní barvy, které by mohli být špatně převedeny a ve výsledku špatně nebo vůbec nevykresleny kontury nebo tvar obrázku. Co se týče velikosti obrázku, doporučuji použít spíše větší obrázek než menší. Pokud se zadá obrázek menší, než bude samotný vygenerovaný QR kód, hustota obrázku nebude optimální a získáme zvětšený, rozostřený obrázek.

Veškeré práce a operace s obrázky se provádí až po vygenerování QR kódu, respektive po získání celkové matice bitů. Tím získáme i velikost, která bude potřeba při změně velikosti obrázku.

9.1 Použitá metoda a princip

Pro načtení obrázků jsem použil velice jednoduchý, můžeme říct až školácký postup. Nicméně pro potřeby programu je efektivní a snadno pochopitelný. Možná je tento postup trochu výpočetně náročný, nicméně je kvalitní a nevynechá žádné detaily obrázku, které by znamenaly jinak ztrátu kontur nebo tvaru výsledného obrázku.

Pro obrázky používám třídu `BufferedImage`, který má všechny funkce, které splňují můj záměr a princip algoritmů.

9.1.1 Načtení obrázků

Vstupní obrázek může mít pozadí, ale i nemusí. Jedná se o alfa vrstvu, kterou mají například obrázky typu PNG. Software je na detekci alfa vrstvy uzpůsobený a díky interních knihoven javy jsem schopen jednoduše rozeznat, jestli obrázek má nebo nemá alfa vrstvu.

Při načítání obrázků využívám interních knihoven javy, které mi poskytují všechny funkce, které jsou potřeba pro načtení a převod. Jsou spolehlivé, roky ověřené a hlavně rychlé [Pecinovský, 2009b].

V první řadě, si vstupní obrázek převedu na stejnou velikost, jako je vygenerovaný QR kód. Jak jsem již psal výše, je zde doporučený čtvercový tvar obrázků, aby nevznikla změna tvaru a kontur (například ze čtverce se stal obdélník nebo z kruhu by se stal ovál). Do změny velikosti zahrnu i nastavenou požadovanou hustotu, aby obrázek byl ve výsledku hladký a odpovídal vzoru.

Dále si obrázek převedu na pole bitů. Pokud obrázek nemá alfa vrstvu, jeden pixel představují tři bity po sobě. V opačném případě jeden pixel představují čtyři po sobě následující bity. Dalším krokem, si zjišťuji jakou má pixel hodnotu, respektive barvu. Podle toho dále posuzuji, jestli se bude jednat o tmavý nebo světlý bit. [Parker, c1997]

Číselné označení barev je na škále od 0 (černá) do 16 777 215 (bílá). Z toho vycházím při mém zpracování jednotlivých pixelů a jejich barev. Musím dále rozpoznat, jestli se jedná o tmavý nebo světlý bit. Můj software funguje tak, že si získám číselnou hodnotu/součet všech bitů, které se vztahují pro daný pixel. Dále pak tuto získanou hodnotu odečítám od bílé hodnoty. Vezmu polovinu z bílé hodnoty a porovnávám se získaným součtem bitových hodnot. Pokud je hodnota větší než polovina z bílé, jedná se o světlý bit. Všimněme si, že logicky by to mělo být opačně. Je to z důsledku bitových posunů. Ty používám při získávání číselných hodnot bitů. Pokud má obrázek formát RGB, pak posouváme bity v tomto pořadí: R neposouváme; G posouváme o 8 bitů; B posouváme o 16 bitů [Roubalová, 2015].

9.1.2 Převod obrázků

Po získání, rozpoznání odstínu bitu (světlý/tmavý bit) zachovávám tyto informace stejným způsobem, jakým si udržuji informace o matici QR kódu. Tudíž mám vytvoření dvojrozměrné číselné pole, kde jednotlivé souřadnice představují stejně jako v matici QR kódu odstíny bitů.

Jediný rozdíl je zde takový, že matice obrázkových bitů je zvětšena o hustotu v poměru 10:1, abychom zachovali detaily vstupního obrázku. To se později bude zohledňovat při slučování matice obrázku a matice QR kódu.

9.1.3 Vložení obrázků do vytvořeného QR kódu

V případě, že máme vytvořený QR kód a zpracovaný vstupní obrázek, je na řadě jejich sloučení a vložení obrázku do matice QR kódu. V první řadě, musíme nastavit a určit barvy, které chceme, aby byly zobrazeny. Nastavujeme tmavou barvu klasických bitů (černých) a tmavých bitů obrázku, které mají na pozicích stejné hodnoty. Dále můžeme nastavit ručně světlou barvu obrázkových bitů, nebo můžeme zvolit možnost v uživatelském rozhraní automatické světlé barvy. Ta bude vypočítána na optimální kontrast.

Barvy máme nastavené, můžeme slučovat obě matice do jedné. Pomocí jednoduchých cyklů prochází algoritmus obě matice a porovnává hodnoty. Algoritmus také zohledňuje velikost obrázkové matice ku matici QR kódu. Pokud se hodnota (1 – tmavý bit) na “stejných” souřadnicích (schválně slovo stejných dávám do uvozovek, jelikož se přihlíží na větší velikost obrázkové matice) shoduje v obou maticích, poté zvolíme nastavenou barvu pro tmavé moduly obrázku. Pokud naopak máme v matici QR kódu hodnotu 0 (světlý bit) a v matici obrázku hodnotu 1 (tmavý bit) aplikuje se nastavená hodnota pro světlý bit obrázku.

Tímto způsobem algoritmus projede obě matice a v průběhu skládá finální obrázek. Při každém průchodu každé souřadnice se současně určená barva aplikuje do vytvářeného obrázku. Podle aktuálních souřadnic průchodu matic se vkládá barva na aktuální souřadnici výsledného obrázku – QR kódu.

9.2 Výsledný QR kód

Sloučením obou matic (matice QR kódu a matice obrázku) získáme výsledný QR kód, který je již jako vygenerovaný obrázek. Vzniklý obrázek nijak neuchovávám v matici nebo jinak podobným způsobem. Nenašel jsem důvod, abych ho takto zachovával.

Po celkovém vygenerování obrázku QR kódu se obrázek automaticky otevře výchozím programem operačního systému. Tuto funkci umožňuje vnitřní funkce a knihovny javy, které otevírá dotyčný soubor (v tomto případě obrázek) pomocí výchozí aplikací nastavenou pro určitý typ souboru. Samozřejmě ho lze nalézt již dříve na nastavené cestě, kterou si uživatel zadal.

9.3 Omezení čitelnosti

Vzniklý QR kód s vloženým logem nemá nijak závratně omezenou čitelnost nebo překlad. Již dříve v kapitole o načítání obrázků jsem psal o kontrastních barvách. Podle mne je to nejdůležitější volba uživatele, na které je závislý zpětný překlad a rychlost překladu.

Čtečky fungují na principu detekování všech barev na QR kódu a následným zprůměrováním barev. Dále prochází matici a každou barvu porovnává s průměrnou barvou. Pokud je tato barva světlejší než průměrná, čtečka rozhodne o světlém bitu. V případě, že barva je tmavší než barva průměrná, jedná se o tmavý bit.

Jakmile uživatel zadá málo kontrastní barvy, vzniká chybovost a víceméně nemožný překlad. U čteček to můžeme zaznamenat jako opakované rozostřování a zaostřování kamery. Kde po chvilce tohoto opakovaného rozostřování ničeho nedosáhneme, nebo dosáhneme špatného výsledku.

Výsledný QR kód by tedy měl obsahovat vzájemně kontrastní barvy. Většinou jde tyto barvy určit jednoduchým pohledem na ně. Pokud uživatel ponechá přednastavené barvy, následný překlad vygenerovaných QR kódů by měl být bezproblémový.

10 Testování vygenerovaných QR kódů

Abych mohl rozhodnout o funkčnosti mého generátoru a výsledku náročnosti, rychlosti nebo ztrátovosti překladu, rozhodl jsem se vzniklé kódy s logem otestovat a tím vynést závěr.

Testovat vygenerované QR kódy budu testovat při mnoha úhlech, v různých světelných podmínkách nebo z různé dálky. Výsledky budu porovnávat se stejným QR kódem, který bude bez přidaného loga. Pro správnost a objektivní výsledek testování jsem vygeneroval QR kód bez loga pomocí mého generátoru. Díky tomu získáme QR kódy, které jsou stejné a mají stejně nastavené možnosti, co se týče například korekční úrovně nebo použité masky.

Ve stažitelných aplikacích se mi, i přes mou snahu, bohužel nepodařilo najít QR čtečku, která by umožňovala zobrazit rychlost překladu. Tudíž dále uvedené rychlostní výsledky překladu jsou hodnoceny subjektivně, z pohledu uživatele a jeho dojmům z rychlosti.

Testovat budu následující dva QR kódy v této barevné kombinaci. Oba mají v sobě zakódovanou alfanumerickou zprávu: „<http://www.pef.czu.cz>“. Čtečky automaticky tuto zprávu detekují jako URL odkaz a nabízejí otevření v již nainstalovaných prohlížečích případně výchozím prohlížečem. Vypočtený barevný kontrast pro zvolenou barevnou kombinaci je hodnota 4 (jedná se o stejnou hodnotu, jaká se používá pro pravidla přístupnosti webu. Optimální hodnota je 4,5).



Obr. 2 QR kód s logem



Obr. 3 QR kód bez loga

V následujících dvou tabulkách jsou vysvětleny použité termíny při testování. Tyto termíny jsou používány pro tabulky výsledků testování. Tím je zajištěna alespoň minimální objektivnost hodnocení.

Termín	Vysvětlení
S logem	Jedná se o QR kód, kde je vloženo logo
Bez loga	Jedná se o QR kód, kde není vloženo logo
Přečteno	Rozhodnutí, zda čtečka správně přeložila zadaný QR kód
Rychlost	Subjektivní pojem pozorování rychlosti překladu
Jas monitoru	Nastavený jas monitoru, na kterém bylo prováděno testování

Tab. 5 Vysvětlení použitých termínů

Termín	Vysvětlení
Okamžitě	Nebylo pozorováno žádné zpoždění překladu
Menší prodleva	Zpoždění do půl vteřiny
Střední prodleva	Zpoždění do jedné vteřiny
Velká prodleva	Zpoždění delší než vteřinu

Tab. 6 Vysvětlení časových termínů

10.1 Testování pod různými úhly

Testování pod různými úhly se rozumí souhrn testů, při kterých byla čtečka od čteného QR kódu pod různým úhlem nebo z různého směru. Dále jsem do těchto testů zahrnul situace, kdy čtený QR kód byl nějakým způsobem deformovaný (prohnutý nebo vypouklý).

Úhel	Bez loga		S logem	
	Přečteno	Rychlost	Přečteno	Rychlost
Kolmo ⁴	Ano	Okamžitě	Ano	Okamžitě
Kolmo 90°	Ano	Okamžitě	Ano	Okamžitě
Kolmo 180°	Ano	Okamžitě	Ano	Velká prodleva
Šikmo ⁵ 0°	Ano	Okamžitě	Ano	Okamžitě
Šikmo 90°	Ano	Velká prodleva	Ne	
Šikmo 180°	Ne		Ne	

Tab. 7 Test překladu pod různými úhly čtečky od loga

⁴ Kolmo – znamená, že čtečka byla směřovaná na QR kód pod kolmým úhlem

⁵ Šikmo – znamená, že čtečka byla směřovaná na QR kód pod úhlem 45°

Deformace	Bez loga		S logem	
	Přečteno	Rychlost	Přečteno	Rychlost
Prohnutý kód	Ano	Okamžitě	Ano	Okamžitě
Vypouklý kód 1 ⁶	Ano	Střední prodleva	Ano	Střední prodleva
Vypouklý kód 2 ⁷	Ne		Ne	

Tab. 8 Test překladu s různou deformací loga

10.2 Testování při různé světelné intenzitě

Pro testování při různých světelných podmínkách jsem uznal jako nejvhodnější testovat pomocí měřičů intenzity osvětlení. K tomu využívám aplikaci Lux Light (pro android), která je volně stažitelná v Google Play. Odkaz na používanou aplikaci je umístěn v přílohách.

Intenzitu osvětlení určuje jednotka Lux (lx). Co se týče obecných hodnot, tak platí, že pro denní intenzitu jsou hodnoty na rozmezí od 100 do 10 000lx. Pro osvětlení v interiéru domácnosti jsou hodnoty od 100 do 500lx. Pokud máme zataženou oblohu, hodnoty se pohybují okolo 1000lx.

QR kódy pro tento typ testu nebyly nijak deformované nebo znečištěné. Jedná se o optimální podmínky čtení, kde je měněna pouze intenzita osvětlení.

Následující test je prováděn v denních světelných podmínkách. Hovořím zde o zataženém počasí, kde je jasná viditelnost. QR loga jsou čteny na IPS, antireflexním monitoru počítače. Tudíž nedochází ke zkreslování barev nebo dalších nečitelností. Intenzita osvětlení při tomto testu byla 275lx a měnil se, nastavoval se jas monitoru. Vzdálenost čtečky od monitoru byla jeden metr.

275lx	Bez loga		S logem	
Jas monitoru	Přečteno	Rychlost	Přečteno	Rychlost
100%	Ano	Okamžitě	Ano	Okamžitě
75%	Ano	Okamžitě	Ano	Okamžitě
25%	Ano	Okamžitě	Ano	Okamžitě
1%	Ano	Okamžitě	Ano	Menší prodleva

Tab. 9 Test překladu (intenzita osvětlení) QR kódů z monitoru

⁶ Vypouklý kód 1 – situace, kdy byla vidět tichá zóna kódu

⁷ Vypouklý kód 2 – situace, kdy nebyla vidět tichá zóna kódu

Další test je prováděn na dvou papírech, kde jsou obě loga vytisknutá. Světelné podmínky v době testu byly zatažená denní obloha. Pro širší škálu testu jsem měnil světelnou intenzitu. Vzdálenost kamery od QR kódy byla jeden metr.

Světelná intenzita (lx)	Bez loga		S logem	
	Přečteno	Rychlost	Přečteno	Rychlost
2500	Ano	Okamžitě	Ano	Okamžitě
1800	Ano	Okamžitě	Ano	Okamžitě
900	Ano	Okamžitě	Ano	Okamžitě
500	Ano	Okamžitě	Ano	Okamžitě
250	Ano	Okamžitě	Ano	Okamžitě
100	Ano	Okamžitě	Ano	Okamžitě
50	Ano	Okamžitě	Ano	Okamžitě
25	Ano	Menší prodleva	Ano	Menší prodleva
10	Ano	Menší prodleva	Ano	Střední prodleva
5	Ano	Menší prodleva	Ano	Velká prodleva
1	Ne		Ne	

Tab. 10 Test překladu (intenzita osvětlení) QR kódů z papíru

10.3 Testování při různé vzdálenosti

Pro testování čtení QR kódů pro různou vzdálenost jsem se rozhodl pro dva typy testů. První je test za optimálních světelných podmínek. Kde QR kód nenarušuje nedostatečné světlo nebo další stíny. Druhý test je opačný a za horších světelných podmínek. Intenzita světla v tomto testu bude minimální. V tomto druhém testu jsem testoval spíše skutečnost, kdy QR čtečka zadaný kód přečte.

První test je tedy za optimálních světelných podmínek a mění se pouze vzdálenost čtecí kamery od QR kódu. QR kódy jsou vytisknuté na bílém papíře a mají rozměr 20x20cm včetně tiché zóny. Intenzita osvětlení při testování byla 850lx. Testování začíná na vzdálenosti 30cm, jelikož je to minimální vzdálenost, kdy je v čtecím prostoru překladové aplikace umístěn celý QR kód (bez tiché zóny).

Vzdálenost (cm)	Bez loga		S logem	
	Přečteno	Rychlost	Přečteno	Rychlost
30	Ano	Okamžitě	Ano	Okamžitě
75	Ano	Okamžitě	Ano	Okamžitě
100	Ano	Okamžitě	Ano	Okamžitě
150	Ano	Okamžitě	Ano	Menší prodleva
200	Ano	Menší prodleva	Ano	Menší prodleva
250	Ano	Menší prodleva	Ano	Menší prodleva
375	Ano	Menší prodleva	Ano	Střední prodleva

450	Ano	Střední prodleva	Ne	
575 ⁸	Ano	Větší prodleva	Ne	

Tab. 11 Test překladu (vzdálenosti) QR kódů (20x20cm) při optimálních světelných podmínkách

Dalším testem je test vzdálenosti při špatných světelných podmínkách. QR kódy byly umístěny do místa, kde je nízká světelná intenzita cca 15lx a dále jsou umístěny ve stínu. Celkově se jedná o tmavé místo.

Vzdálenost (cm)	Bez loga		S logem	
	Přečteno	Rychlost	Přečteno	Rychlost
30	Ano	Okamžitě	Ano	Okamžitě
75	Ano	Okamžitě	Ano	Menší prodleva
100	Ano	Menší prodleva	Ano	Menší prodleva
150	Ano	Menší prodleva	Ano	Menší prodleva
200	Ano	Menší prodleva	Ano	Střední prodleva
250	Ano	Menší prodleva	Ne	
360 ⁹	Ano	Střední prodleva	Ne	

Tab. 12 Test překladu (vzdálenosti) QR kódů při špatných světelných podmínkách

10.4 Testování určení kontrastu barevných kombinací loga

Při vytváření QR kódů s logy existuje vysoká závislost na volbě barev matice. Jedná se o kontrast mezi zadanými barvami. Při špatné volbě, nedostatečně kontrastní může docházet ke špatně čitelným kódům. Následující test je zaměřen na volbu barev a jejich vzájemným kontrastem. Kontrast je vypočítáván na webové stránce, na kterou je odkaz uveden v přílohách.

Zadávané barvy budou ve formátu RGB. V pravém sloupci od napsané barvy je umístěn náhled zadané barvy. Ve sloupci vypočítaný kontrast je hodnota, která byla vypočtena na webové stránce (odkaz na stránku je v přílohách). Text zprávy, která je uložena do QR kódů zůstává stejná a to: „<http://www.pef.czu.cz>“. QR kódy byly čteny a testovány na IPS, antireflexním monitoru při denních světelných podmínkách. Vzdálenost kamery od kódu byla 40cm.

⁸ Největší možná vzdálenost, kdy čtečka přečetla zadaný QR kód.

⁹ Největší možná vzdálenost, kdy čtečka přečetla zadaný QR kód.

Tmavá barva	Světlá barva	Vypočítaný kontrast	Přečteno
rgb(172,19,13)	rgb(255,189,220)	4.8	Ano
rgb(172,19,13)	rgb(255,182,214)	4.5	Ano
rgb(172,19,13)	rgb(255,176,211)	4.3	Ano
rgb(172,19,13)	rgb(255,167,172)	4	Ano
rgb(172,19,13)	rgb(255,152,193)	3.7	Ano
rgb(172,19,13)	rgb(255,141,193)	3.4	Ano
rgb(172,19,13)	rgb(255,127,188)	3.2	Ano
rgb(172,19,13)	rgb(255,123,172)	3	Ano

Tab. 13 Test překlady (určení kontrastu) QR kódů pro rudé odstíny

Tmavá barva	Světlá barva	Vypočítaný kontrast	Přečteno
rgb(0,102,51)	rgb(14,245,93)	4.8	Ano
rgb(0,102,51)	rgb(14,237,93)	4.5	Ano
rgb(0,102,51)	rgb(14,232,89)	4.3	Ano
rgb(0,102,51)	rgb(14,225,89)	4	Ano
rgb(0,102,51)	rgb(14,216,78)	3.7	Ne
rgb(0,102,51)	rgb(14,208,78)	3.4	Ne
rgb(0,102,51)	rgb(0,202,73)	3.2	Ne
rgb(0,102,51)	rgb(0,193,70)	3	Ne

Tab. 14 Test překlady (určení kontrastu) QR kódů pro zelené odstíny

Tmavá barva	Světlá barva	Vypočítaný kontrast	Přečteno
rgb(30,38,122)	rgb(0,137,232)	4.8	Ano
rgb(30,38,122)	rgb(0,131,232)	4.5	Ano
rgb(30,38,122)	rgb(0,127,230)	4.3	Ano
rgb(30,38,122)	rgb(0,119,230)	4	Ano
rgb(30,38,122)	rgb(0,110,230)	3.7	Ne
rgb(30,38,122)	rgb(0,103,230)	3.4	Ne
rgb(30,38,122)	rgb(0,95,230)	3.2	Ne
rgb(30,38,122)	rgb(0,95,216)	3	Ne

Tab. 15 Test překlady (určení kontrastu) QR kódů pro modré odstíny

Výsledek testu rudého odstínu a volba světlých barev se ukázala jako optimální pro čtečku. Logo je v matici pohledem rozeznatelné, jasné viditelné a čtečka si poradila se všemi odstíny.

11 Vyhodnocení testování

Myslím, že provedené testování je dostačující k tomu, abych byl schopen vynést závěr, který má podložené výsledky testování.

Testované QR kódy s vloženým logem se ukázaly jako velice úspěšné, co se týče rychlosti překladu nebo možnosti překladu vůbec. V žádném testovaném případě nedošlo k chybnému překladu. Čtečka QR kódů buď přeložila zadávaný QR kód správně, nebo vůbec. Rozdíly výsledků jsou minimální, díky tomu splňují svůj cíl práce.

Z testování deformovaných QR kódů je zřejmé, že čtečka musí vidět tichou zónu kódu. V testech, kdy byl kód deformovaný (vypouklý) vyplynulo, že čtečka nemá problém číst QR kód s logem, avšak pokud je vidět tichá zóna čteného kódu. V případě, kdy nebyla vidět tato zóna, čtečka nebyla schopna přečíst kód. Stejně výsledky vyplynuly i u testu QR kódu bez loga. Testy ostatních deformací se ukázaly jako nezávadné, výsledky obou testovaných QR kódů byly stejné, včetně rychlosti překladu.

Čtečka nicméně zaznamenala potíže čtení QR kódů při nestandardních úhlech kamery od kódu. V případě kolmého umístění čtečky od skenovaného QR kódu se potíže nevyskytovaly. Jediný rozdíl jsem zaznamenal, kdy čtečce trvalo déle přečíst kód s logem, který byl skenovaný pod úhlem 180 stupňů (vzhůru nohama). QR kód bez loga při stejném úhlu čtečka přečetla okamžitě. Potíže nastaly, kdy se kód skenoval pod šikmým úhlem 45° a dále natočený na určitý úhel. V testu šikmého přímého čtení výsledky byly totožné a to okamžitě přeložený QR kód. V šikmém testu a natočeného QR kódu o 90° už měla čtečka delší dobu překladu kódu bez loga. Kód s logem vůbec nepřečetla. Extrémní případ šikmého testu a dále otočeného kódu o 180° (vzhůru nohama) čtečka nebyla schopna přečíst ani jeden z testovaných kódů.

Na základě těchto výsledků testu deformačních situací jsem usoudil. Pro situace, které se běžně vyskytují v praxi (bez extrémních případů) není nijak omezena schopnost překladu čteček ani nijak omezena rychlost překladu.

Výsledky testů při různých světelných intenzitách jsou v obou případech většinou totožné. Liší se pouze doba překladu v extrémních pokusech testování, v podmínkách téměř totožné s tmou. Nezahrnul jsem do testu možnost přisvětlení, test by tím ztratil smysl, jelikož by vynikly lépe kontury a barevné rozdíly. Pro čtečku by to bylo jednoduché takto nasvícený kód přečíst. Test při různých

světelných podmínkách vyhodnocuji jako úspěšný, QR kód s logem nemá značný vliv na čtení kódu čtečkou.

Testování čtení QR kódů při různých vzdálenostech bych ohodnotil jako zdařilé. Výsledky, kde čtečka měla problém přečíst, nepřečetla QR kód, jsou spíše na extrémních vzdálenostech. Tam kde čtečka začínala číst QR kód bez loga s menším zpožděním, verze QR kódu s logem byla se středním zpožděním. V optimálních světelných podmínkách QR čtečka byla schopna číst QR kódy na ctihodnou vzdálenost. Kód bez loga o rozměrech 20x20cm čtečka přečetla na necelých šest metrů. Kód s logem byla čtečka přečíst na necelé čtyři metry. Jedná se o celkem malý QR kód, který je v praxi umisťován na místa, kde se zákazníci potýkají s QR kódy na menší vzdálenosti. Tudiž verze kódů s logem jsou pro praxi použitelné.

Testy ohledně výběru barev ukázaly, že se dají pro výběr barev použít pravidla přístupnosti webu. Kde je pravidlo, které tvrdí, že text by měl mít minimální kontrast s pozadím o hodnotě 4,5. Čtečka, která četla zadávané QR kódy s logem, které měly barevný kontrast 4.5 se přeložily okamžitě bez potíží. Čím menší byl kontrast barev, tím se prodlužovala doba překladu.

Celkový počet testů	32
Průměrný vypočítaný kontrast	3.86
Průměrný kontrast čitelnosti	3.56

Tab. 16 Tabulka průměru čitelnosti kontrastu

Pro správné barevné kombinace lze použít jedno z pravidel přístupnosti webu, které hovoří o doporučeném kontrastu textu a pozadí. Dostatečný doporučený kontrast je s hodnotou 4,5 nebo 765 (záleží na výpočetním algoritmu nebo kalkulačce). Testy ukázaly, že víceméně totožný barevný kontrast pro zvolené barvy QR kódu s logem se jeví jako začátek čitelnosti QR kódů pro čtečku.

Pokud pro slovní ohodnocení rychlosti překladu ohodnotíme známkou, můžeme z měřených testů vypočítat průměrnou dobu překladu.

Termín	Známka
Okamžitě	1
Menší prodleva	2
Střední prodleva	3
Větší prodleva	4
Nepřečteno	5

Tab. 17 Známkové ohodnocení pro rychlosti překladu

Následující tabulka obsahuje výsledky ze všech testů, které byly provedeny a vypočítány průměry pro jednotlivé výsledky. Celkově bylo provedeno 40 měření. Do následujícího výpočtu jsem nezapočítal testování barevného kontrastu.

	Bez loga	S logem
Přečteno	37	32
Nepřečteno	3	8
Okamžitě	25	18
Menší prodleva	10	8
Střední prodleva	3	4
Větší prodleva	2	2
Průměr	1.92	2.35

Tab. 18 Rychlostní výsledky ze všech testů

Na základě tabulky celkových výsledků jsem určil průměrnou dobu překladu. Pro testování bez loga byla průměrná doba překladu: 1.92. Pokud to převedeme na slovní hodnocení, jedná se o dobu mezi „menší prodlevou“ a „střední prodlevou“. Pro případ s logem byl vypočítán průměr doby překladu: 2.35. Ve slovním ohodnocení to můžeme určit jako dobu překladu lehce nad jednu vteřinu.

Doba překladu tudíž není nijak závratně omezena, do testů byly započítány i v praxi nemožné situace, které mohou zkreslovat výsledek průměrné doby překladu. Pro případy, které se běžně v praxi vyskytují se značné rozdíly překladu kódu s logem nebo bez loga nevyskytují.

11.1 Dosažené výsledky

Výsledky testování vygenerovaných QR kódů s logem ukázaly, že myšlenka snadného, správného čtení a překladu těchto kódů je reálná. Výsledky porovnávaných dvou QR kódů byly v drtivé většině totožné. Rozdíl výsledků testování se vyskytoval pouze u extrémních testů. Jako například příliš velká vzdálenost čtecí kamery od QR kódu nebo nedostatečné světelné podmínky. Testy čtení deformovaných, otočených nebo abnormálních úhlů se ukázaly jako totožné. Pro objektivitu čtení kódů jsem využíval pouze jednu čtecí aplikaci. Odkaz na tuto aplikaci je uveden v příloze.

12 Závěr

Cílem práce bylo vytvořit software, který bude schopen vytvářet QR kódy s vnořenou grafikou. Aplikaci na vytváření těchto kódů jsem vytvořil.

V uživatelském rozhraní lze nastavit základní parametry nastavení QR kódů, mluvím zde o samotné zprávě kódu, nastavení korekční úrovně, barevných kombinací nebo cílové cestě vygenerovaného obrázku (QR kódu). Aplikace je napsána v jazyce java, tudíž ji lze spouštět na více platformách kromě MS Windows.

Vstupní grafika se analyzovala pro potřeby programu tak, aby software byl schopen tuto grafiku vložit do vygenerované matice QR kódu. Byl kladen důraz na co největší možnou kvalitu načtení a zpracování zadávané vstupní grafiky pro co nejlepší koncový výsledek. Logo, které je umístěno ve výsledném QR kódu je jasně viditelné a pohledem rozeznatelné, to bylo jedním z cílů práce.

Vygenerované obrázky jsou ve formátu JPG s co největší možnou dostupnou kvalitou, kterou jsem byl schopen zajistit. Dále tyto obrázky mají přednastavenou velikost.

Navržené a vytvořené QR kódy s vloženou grafikou mohou nalézt uplatnění v reklamě. Jsou graficky přívětivější a více člověka upoutají. Jelikož testování prokázalo, že čitelnost těchto kódů není nějak razantně omezena, můžeme říci, že použití těchto kódů na veřejnosti a v praxi není nijak omezeno.

Rozšíření této práce by mohlo být například přenesení aplikace do webového rozhraní, možnosti nastavení vlastní velikosti vygenerovaného QR kódu, možnosti zakódovat do zprávy nejenom plaintext, ale také vizitky, emaily, sms a mnoho dalšího.

13 Použitá literatura

- ČADA, Ondřej, 2009. *Objektové programování: naučte se pravidla objektového myšlení*. Praha: Grada. Průvodce (Grada). ISBN 978-80-247-2745-5.
- KOTLER, Philip, 2007. *Moderní marketing: 4. evropské vydání*. Praha: Grada. ISBN 978-80-247-1545-2.
- PARKER, J. R. c1997. *Algorithms for image processing and computer vision*. New York: Wiley Computer Pub. ISBN 04-711-4056-2.
- PECINOVSKÝ, Rudolf, 2009b. *Myslíme objektově v jazyku Java: kompletní učebnice pro začátečníky*. 2., aktualiz. a rozš. vyd. Praha: Grada. *Myslíme v--*. ISBN 978-80-247-2653-3.
- ROUBALOVÁ, Eliška, 2015. *Java bez předchozích znalostí*. Brno: Computer Press. ISBN 978-80-251-4572-2.
- WATERS, Joe, c2012. *QR codes for dummies*. Portable ed. Hoboken, NJ: John Wiley. --For dummies. ISBN 978-1118337035

14 Internetové zdroje

- Čermák Tomáš, 2013. 9 tipů jak efektivně využívat QR kódy (nejen) v marketingu. [online]. © 2013 [cit. 11.02.2018]. Dostupné z: <http://futuristica.cz/9-tipu-jak-efektivne-vyuzivat-qr-kody-nejen-v-marketingu/>
- Espace.com, 2018. About QR codes [online]. [cit. 01.01.2018]. Dostupné z: <http://www.espace.com/resources/about-qr-codes>
- Insightlab.cz, 2013. QR kódy. Jaký mají potenciál? [online]. © 2018 [cit. 13.02.2018]. Dostupné z: <https://www.insightlab.cz/qr-kody-jaky-maji-potencial/>
- Pecinovský, Rudolf. 2009a. Objektově orientované programování? Co to je? *Počítač ve škole*. [Online] 2009. http://www.pocitacveskole.cz/system/files/uzivatel/9/clanky/pecinovsky_pdf_16661.pdf.
- Plcviewer.com, 2011. QR CODE bit masking [online]. Dostupné z: <http://www.plcviewer.com/rs2/qrmasking.htm>
- Qrstuff.com, 2011. QR Code Error Correction. [online]. © 2018 [cit. 01.01.2018]. Dostupné z: <https://blog.qrstuff.com/2011/12/14/qr-code-error-correction>
- Rostlá Štěpánka, 2009. 2D čárové kódy : Marketing journal. [online]. © 2004 [cit. 10.02.2018]. Dostupné z: http://www.m-journal.cz/cs/marketing/marketingove-nastroje/2d-carove-kody__s418x5073.html
- Thonky.com, 2015a. QR Code Tutorial: Introduction [online]. [cit. 19.12.2017]. Dostupné z: <https://www.thonky.com/qr-code-tutorial/introduction>
- Thonky.com, 2015b. Module Placement in Matrix [online]. [cit. 19.12.2017]. Dostupné z: <https://www.thonky.com/qr-code-tutorial/module-placement-matrix>
- Thonky.com, 2015c. Format and Version Information [online]. [cit. 01.01.2018]. Dostupné z: <https://www.thonky.com/qr-code-tutorial/format-version-information>
- Thonky.com, 2015d. Data Encoding [online]. [cit. 03.01.2018]. Dostupné z: <https://www.thonky.com/qr-code-tutorial/data-encoding>
- Thonky.com, 2015e. Data Masking [online]. [cit. 6.01.2018]. Dostupné z: <https://www.thonky.com/qr-code-tutorial/data-masking>

III. Přílohy

Použité tabulky

Tab. 1	Korekční kapacity [qrstuff.com, 2011].....	21
Tab. 2	Počet bitů na verzi [Thonky.com, 2015d].....	22
Tab. 3	Vzory pro masky [Thonky.com, 2015e].....	24
Tab. 4	Číselné ekvivalenty [Thonky.com, 2015c].....	26
Tab. 5	Vysvětlení použitých termínů.....	41
Tab. 6	Vysvětlení časových termínů.....	41
Tab. 7	Test překladu pod různými úhly čtečky od loga.....	41
Tab. 8	Test překladu s různou deformací loga.....	42
Tab. 9	Test překladu (intenzita osvětlení) QR kódů z monitoru.....	42
Tab. 10	Test překladu (intenzita osvětlení) QR kódů z papíru.....	43
Tab. 11	Test překladu (vzdálenosti) QR kódů (20x20cm) při optimálních světelných podmínkách.....	44
Tab. 12	Test překladu (vzdálenosti) QR kódů při špatných světelných podmínkách.....	44
Tab. 13	Test překladu (určení kontrastu) QR kódů pro rudé odstíny.....	45
Tab. 14	Test překladu (určení kontrastu) QR kódů pro zelené odstíny.....	45
Tab. 15	Test překladu (určení kontrastu) QR kódů pro modré odstíny.....	45
Tab. 16	Tabulka průměru čitelnosti kontrastu.....	47
Tab. 17	Známkové ohodnocení pro rychlosti překladu.....	47
Tab. 18	Rychlostní výsledky ze všech testů.....	48

Použité obrázky

Obr. 1	Diagram aktivit	33
Obr. 3	QR kód bez loga.....	40
Obr. 2	QR kód s logem	40

Odkazy

Stránka na výpočet barevného kontrastu:

Contrast Ratio: Easily calculate color contrast ratios. Passing WCAG was never this easy!.
GitHub Pages [online]. Dostupné z: <http://leaverou.github.io/contrast-ratio/>

Stránka, kde je popsán postup výpočtu korekčních bitů:

Error Correction Coding - QR Code Tutorial. - Thonky.com [online]. Copyright © [cit. 02.01.2018]. Dostupné z: <https://www.thonky.com/qr-code-tutorial/error-correction-coding>

Stránka, hodnot znaků pro kódování:

Character Capacities by Version, Mode, and Error Correction - QR Code Tutorial. Thonky.com - Thonky.com [online]. Copyright © [cit. 02.01.2018]. Dostupné z: <https://www.thonky.com/qr-code-tutorial/character-capacities>

Odkaz na stažení používané aplikace pro čtení QR kódů

Lightning QRcode Scanner – Aplikace pro Android ve službě Google Play. [online]. Copyright © 2018 Google [cit. 25.12.2017]. Dostupné z: https://play.google.com/store/apps/details?id=com.application_4u.qrcode.barcode.scanner.reader.flashlight&hl=cs

Odkaz na stažení používané aplikace pro měření intenzity osvětlení:

Lux Light Meter – Aplikace pro Android ve službě Google Play. [online]. Copyright © 2018 Google [cit. 01.01.2018]. Dostupné z: <https://play.google.com/store/apps/details?id=com.doggoapps.luxlight&hl=cs>

Stránka, kde je popsán celkový postup tvorby QR kódů

QR Code Tutorial. - Thonky.com [online]. Copyright © [cit. 18.11.2017]. Dostupné z: <https://www.thonky.com/qr-code-tutorial/introduction>

Stránka, hodnot znaků pro kódování:

Table of Alphanumeric Values - QR Code Tutorial: Introduction [online]. Copyright © [cit. 25.12.2017]. Dostupné z: <https://www.thonky.com/qr-code-tutorial/alphanumeric-table>