

Pedagogická fakulta
Faculty of Education

Jihočeská univerzita v Českých Budějovicích
University of South Bohemia in České Budějovice

Jihočeská univerzita v Českých Budějovicích

Pedagogická fakulta

Katedra informatiky

Bakalářská práce

Programování robotů Lego Mindstorms v mobilních aplikacích

Vypracoval: Lukáš Vališ

Vedoucí práce: doc. PaedDr. Jiří Vaníček, Ph.D.

České Budějovice 2017

Prohlášení

Prohlašuji, že svoji bakalářskou práci jsem vypracoval samostatně pouze s použitím pramenů a literatury uvedených v seznamu citované literatury.

Prohlašuji, že v souladu s § 47b zákona č. 111/1998 Sb. v platném znění souhlasím se zveřejněním své bakalářské práce, a to v nezkrácené podobě elektronickou cestou ve veřejně přístupné části databáze STAG provozované Jihočeskou univerzitou v Českých Budějovicích na jejích internetových stránkách, a to se zachováním mého autorského práva k odevzdanému textu této kvalifikační práce. Souhlasím dále s tím, aby toutéž elektronickou cestou byly v souladu s uvedeným ustanovením zákona č. 111/1998 Sb. zveřejněny posudky školitele a oponentů práce i záznam o průběhu a výsledku obhajoby kvalifikační práce. Rovněž souhlasím s porovnáním textu mé kvalifikační práce s databází kvalifikačních prací Theses.cz provozovanou Národním registrem vysokoškolských kvalifikačních prací a systémem na odhalování plagiátů

V Českých Budějovicích dne 2017

Lukáš Vališ

Abstrakt

Bakalářská práce se zabývá využitím robotů pro výuku. Díky současným moderním technologiím, které jsou využívány v oblasti mobilních telefonů, je možné vytvářet programy pro jednoduché ovládání robotů Lego Mindstorms. Základem je analýza dostupných aplikací pro komunikaci s těmito roboty. V rámci této analýzy budou jednotlivé aplikace posuzovány z hlediska jejich možností a jednoduchosti vytváření programů. Zjištěné informace poslouží k návrhu a sestavení sady úloh testujících základních funkce robota.

V rámci praktické části práce bude využit Lego robot, který bude schopen vykonávat rozličné úkony dle zadaného programu a zajistí tak bezproblémové provádění všech úloh z vytvořené sady. Realizované úlohy pro finální podobu robota budou popsané ve vytvořeném manuálu, který seznámí čtenáře s jeho funkcemi a umožní programování z prostředí smartphonů či tabletu. Testování úloh bude provedeno formou sezení se žáky dle vytvořeného manualu. Výsledky z této aktivity budou následně vyhodnoceny.

Abstract

The bachelor thesis deals with the use of robots for teaching. With the modern mobile phone technologies, it is possible to create programs for easy control of Lego Mindstorms robots. The basis is an analysis of available applications for communication with these robots. Within this analysis, individual applications will be judged from the point of view of their possibilities and the simplicity of programming. The discovered information will be used to design and build a set of tasks testing basic robot functions.

In the practical part of the work will be used Lego robot, which will be able to perform different tasks according to the specified program and thus ensure smooth performance of all the tasks from the created set. Tasks for the robot will be described in a manual that will introduce the reader with its features and enable programming from a smartphone or tablet environment. Testing of the tasks will be in lessons with pupils according to the manual prepared. The results from this activity will then be evaluated.

Klíčová slova

- Lego Mindstorms NXT 2.0
- Programování
- Mobilní aplikace
- Programování v mobilních aplikací
- MIT App Inventor
- Catroid
- Android

Keywords

- Lego Mindstorms NXT 2.0
- Programing
- Smartphone app
- Programing in mobile app
- MIT App Inventor
- Catroid
- Android

JIHOČESKÁ UNIVERZITA V ČESKÝCH BUDĚJOVICÍCH
Fakulta pedagogická
Akademický rok: **2014/2015**

ZADÁNÍ BAKALÁŘSKÉ PRÁCE

(PROJEKTU, UMĚLECKÉHO DÍLA, UMĚLECKÉHO VÝKONU)

Jméno a příjmení:	**Lukáš VALIŠ**
Osobní číslo:	**P13114**
Studijní program:	**B7507 Specializace v pedagogice**
Studijní obor:	**Informační technologie a e-learning**
Název tématu:	**Programování robotů LEGO Mindstorms z mobilních aplikací**
Zadávající katedra:	**Katedra informatiky**

Zásady pro vypracování:

Student zpracuje přehled možností vytváření programů pro roboty ve stavebnicích LEGO Mindstorms v různých programovacích prostředích včetně mobilních. Popíše způsob práce v těchto prostředích a porovná je z hlediska, jak jsou vhodná pro použití na základní škole. Zpracuje a prakticky vyzkouší návody, jak vytvářet, implementovat a ladit programy v některém programovacím prostředí v mobilních zařízeních (Catroid, MIT App Inventor apod.), které rozšiřuje možnosti programovat roboty LEGO. Student vytvoří sadu programů pro programování zvoleného robota ze základní sady nabízené producentem stavebnice pomocí programovacího prostředí v mobilním zařízení. Student současně s danými programy vytvoří i sadu praktických úloh, které budou realizovatelné v daném mobilním vývojovém prostředí, pro některé úlohy navrhne své vlastní modely robotů. Prakticky ověří vytvořené úlohy a modely robotů se žáky základní školy.

Rozsah grafických prací:	**CD ROM**
Rozsah pracovní zprávy:	**40**
Forma zpracování bakalářské práce:	**tištěná**

Seznam odborné literatury:

1. **Gujberová, Monika, Mayerová, Karolina, & Veselovská, Michaela. Edukačná robotika na 2. stupni ZŠ a zručnosti pre 21. storočie. In Zborník didinfo 2014 [online]. Banská Bystrica: Univerzita Mateja Bela, 2014, s. 45-52. Dostupné z: http://didinfo.umb.sk/public/filestore/documents/richtext/161/zbornik_didinfo_2 978-80-557-0698-6**
2. **Slany, Wolfgang. Catroid: A Mobile Visual Programming System for Children, Austria 2012 [online]. Austria: Graz University of Technology, 2012. Dostupné z: http://arxiv.org/ftp/arxiv/papers/1204/1204.6411.pdf. ISBN 978-80-557-0698-6**
3. **Walter, Derek, & Sherman, Mark. Learning MIT App Inventor. [online] Addison Wesley, 2014 Dostupné z: http://ptgmedia.pearsoncmg.com/images/ 9780133798630/samplepages/9780133798630.pdf. ISBN 978-0-133-79863-0**

Vedoucí bakalářské práce:	**doc. PaedDr. Jiří Vaníček, Ph.D.** Katedra informatiky
Datum zadání bakalářské práce:	**28. dubna 2015**
Termín odevzdání bakalářské práce:	**29. dubna 2016**

Michal Vančura

Mgr. Michal Vančura, Ph.D.
děkan

L.S.

doc. PaedDr. Jiří Vaníček, Ph.D.
vedoucí katedry

V Českých Budějovicích dne 28. dubna 2015

Poděkování

Rád bych tímto poděkoval svému vedoucímu práce doc. PaedDr. Jiřímu Vaníčkovi, Ph.D za vstřícnost, rady a věcné připomínky při vedení mé bakalářské práce. Poděkovat bych také chtěl své úžasné rodině a přátelům, kteří mě podporovali a měli se mnou trpělivost. Další poděkování patří rodině Pulců, kteří mají velký podíl na mém vysokoškolském vzdělání a to díky podpoře ve velké části celého studia.

Velký dík patří i dobrovolníkům ze ZŠ a MŠ Lomnice nad Lužnící, kteří utvořili skupinu, kterým byl vývoj aplikací předveden. Byli to žáci Nikola Nedvědová, Eliška Koutková a Tomáš Řehoř.

Obsah

1 Úvod . . . 10
1.1 Cíle práce . . . 10
1.2 Východiska práce . . . 11
1.3 Metody práce . . . 11
2 Vymezení pojmů . . . 13
2.1 Android . . . 13
2.2 Vývojové prostředí . . . 13
2.3 Emulátor . . . 13
2.4 Drag & drop . . . 13
2.5 Catroid . . . 13
2.6 MIT App Inventor . . . 14
3 Přehled vývojových prostředí . . . 15
4 Porovnání již vytvořených aplikací . . . 16
4.1 NXT Simple Remote . . . 17
4.2 NXT Remote Control . . . 18
4.3 Aplikace pro WP a iOS . . . 19
4.3.1 App RoboLiterate: LEGO NXT Remote APK for Windows Phone . . . 19
4.3.2 iNXT Remote . . . 20
4.4 Přehled aplikací dle dostupnosti . . . 20
5 Srovnání vlastností porovnaných aplikací . . . 21
5.1 Základní funkce . . . 21
5.2 Vlastnosti a rozšiřující funkce . . . 22
6 MIT App Inventor . . . 23
6.1 Přihlášení . . . 24
6.1.1 Možné problémy . . . 24
6.2 Nejvyužívanější záložky menu . . . 24
6.2.1 Levá část menu . . . 24
6.2.1.1 Záložky - Projetcs . . . 24
6.2.2 Pravá část menu . . . 25
6.3 GUI vývojového prostředí . . . 25
6.3.1 Bloková část . . . 29
6.3.2 Blocks - pravá část GUI Blocks . . . 30
6.3.2.1 Built-in . . . 30
6.3.3 Screen - komponenty používané v GUI aplikace . . . 33
6.3.4 Viewer - návrhová část GUI Blocks . . . 34
6.4 Základ každé aplikace v MIT App Inventoru . . . 36
6.4.1 První krok - výběr komponentů . . . 36

6.4.2 Druhý krok - Nastavení vlastností komponentů . . . 37
6.4.3 Třetí krok - vytvoření akcí (funkčních bloků) . . . 38
6.4.4 Třetí krok - testování . . . 43
7 Pocket Code - Learn programming . . . 44
7.1 GUI aplikace . . . 44
7.1.1 Menu & Možnosti prostředí . . . 44
7.1.2 Potřebná nastavení pro vývoj programu na NXT . . . 45
7.1.3 Přehled objektů programu & detail objektů . . . 45
7.1.4 Rozdělení scriptů a jejich skládání . . . 46
7.1.5 Nejčastěji používané akce jednotlivých seznamů . . . 46
7.1.6 Nastavení jednotlivých akcí . . . 47
7.1.7 Spuštění vyvinuté aplikace . . . 49
8 Porovnání aplikací . . . 49
8.1 Využití NXT senzorů . . . 49
8.2 Využití komponent smartphonu - světlo, kamera . . . 50
8.2.1 Pocket Code . . . 50
8.2.2 MIT App Inventor . . . 50
8.3 Práce s vytvořenými aplikacemi . . . 50
8.3.1 MIT App Inventor . . . 50
8.3.2 Pocket Code . . . 50
8.4 Rozdíly vývoje aplikací . . . 51
8.5 Rozdíly mezi škálami senzorů NXT . . . 51
8.5.1 MIT App Inventor . . . 52
8.5.2 Pocket Code . . . 52
8.6 Nutné opatření ve vývoji aplikací . . . 52
8.6.1 Pocket Code . . . 52
8.6.2 MIT App Inventor . . . 52
8.7 Využití logických, matematických funkcí . . . 53
8.8 Využití souborů NXT nebo smartphonu . . . 53
8.8.1 Pocket Code . . . 53
8.8.2 MIT Appp Inventor . . . 53
8.9 Práce s txt. řetezci . . . 54
8.9.1 Pocket Code . . . 54
8.9.2 MIT App Inventor . . . 54
9 Praktická část . . . 55
9.1 Konstrukce robota . . . 55
9.2 Robot Controler . . . 57
9.2.1 Návod pro MIT App Inventor . . . 57
9.2.2 Návod Pocket Code . . . 59

9.2.3 Rozdíly vývoje . . . 59
9.3 Robot Control from List . . . 60
9.3.1 Návod vývoje v MIT App Inventoru . . . 60
9.3.2 Návod vývoje v Pocket Code . . . 61
9.3.3 Úskalí aplikace . . . 64
9.4 Detekce barev pomocí Color Senzoru . . . 64
9.4.1 Úskalí aplikace . . . 66
9.5 Pohyb robota pomocí úrovně a počtu zaznamenaného zvuku . . . 66
9.5.1 Úskalí aplikace . . . 70
9.6 Striker . . . 70
9.6.1 Návod vývoje v MIT App Inventoru . . . 71
9.6.2 Návod vývoje v Pocket Code . . . 72
9.6.3 Úskalí aplikace . . . 74
10 Předání informací žákům základních škol . . . 74
10.1 Náplň sezení . . . 75
10.2 Jednoduchý dotazník . . . 76
11 Vytvořená videa . . . 78
12 Vytvořená webová prezentace . . . 78
13 Problémy s BP . . . 79
13.1 Vybité baterie . . . 79
13.2 Neúspěšný návrat z SOŠ Velešín . . . 79
13.3 Problémy s vývojem v MIT APP Inventoru a Pocket Code . . . 80
13.4 Problém s nepřesností Color senzoru . . . 80
13.5 Nefunkčnost Light senzoru . . . 80
14 Závěr . . . 81
15 Literatura a použité zdroje . . . 82
16 Seznam obrázků . . . 83
17 Seznam tabulek . . . 87
18 Přílohy . . . 88
18.1 Přiložené CD . . . 88

1 Úvod

Programování robotů ve školách nabývá v současnosti vzrůstajícího trendu. Vznikají zájmové kroužky a předměty věnující se této problematice. Na základních školách především proto, aby žákům předali povědomí o programování a jeho principech. Na středních školách s nimi žáci vytvářejí složitější konstrukce a programy. Každoroční soutěž v tomto oboru pořádá například ČVUT v Praze.

Programování robotů v mobilních aplikacích je možné s využitím smartphonů a potřebných aplikací. Robota Lego Mindstorms lze programovat bez znalostí programovacích jazyků. Vhodným prostředím pro začátečníky může být možnost programovat pomocí Drag&drop[1] funkčních bloků. Dále představuje další způsob, jak efektně využít smartphony dnešních dětí při výuce. Při dobrém rozvržení náhledové plochy a správném programu lze mobil použít jako ovladač robota.[11]

Práce představuje přehled aplikací a návodů, jak realizovat nové programy ve zmíněné stavebnici. Především se zabývá programováním na platformě Android. Tato platforma nabízí podporu pro aplikace Pocket Code a MIT App Inventor. Druhá zmiňovaná aplikace není plně uzpůsoběna pro programování v případném smartphonu. Důležitým aspektem v této aplikaci je její desktopové prostředí, ve kterém se program vyvíjí. Následně se pak přes aplikaci MIT App Inventor a Ai2 Companion spustí ve smartphonu.

Školy, které se vybavily stavebnicemi Lega NXT 2.0 v době jejího vzniku dnes programují na těchto robotech, nebo je dále poskytují partnerským základním či středním školám a gymnáziím. Pro školy, které stavebnice pořizují v současné době je na trhu k dispozici novější typ stavebnice, který nese název EV3.

1.1 Cíle práce

Jedním z cílů bakalářské práce je zmapování možností programování v jednotlivých prostředí pro roboty Lego. K tomu budou sloužit důkladné rozbory těchto prostředí, kde bude zjišťována celková funkcionalita vývojového prostředí a implementace. Následně bude provedeno jejich srovnání a zjištění do jaké míry a zda-li vůbec umí pracovat se stavebnicí Lego Mindstorms.

Takto zjištěných informací bude dále využito při návrhu jednotlivých úloh, což je dalším cílem práce. Bude vytvářena úloha pro danou aplikaci. Následně bude úloha vyvíjena v ostatních aplikacích. Pokud úloha nepůjde v ostatních aplikacích naprogramovat, jelikož pro to nemá vhodnou funkci a daná funkce nepůjde ani dovytvořit, úloha dostane přiřazení, že v dané aplikaci nejde zrealizovat. Složitost vytváření programů v těchto prostředí bude odzkoušena při sezení se žáky vybrané školy. Dalším cílem je vytvořit webovou prezentaci a videa s ukázkami.

Cílem je vytvořit několik úloh, které ukáží plné využití daného prostředí. Pro tyto úlohy bude dále zpracován manuál pro jejich programování. Budou rozvrženy následovně:

- 2 úlohy pro MIT App Inventor
- 2 úlohy pro Pocket Code

[1] Drag and drop je v informatice operace používaná v GUI, kdy uživatel v počítači „uchopí“ pomocí ukazovacího zařízení (např. myši) virtuální objekt (např. soubor nebo ikonu) a přesune ho „přetažením“ na jiné místo (nebo na jiný objekt).[7]

1.2 Východiska práce

K rozboru složitějších funkčností, které daný robot poskytuje, poslouží informace na stránkách výrobce, ale také další zdroje. Dalším východiskem budou instruktážní videa od autorů, zabývajících se Lego roboty a jejich programováním. Neméně podstatné jsou mé zkušenosti v této problematice.

Informace pro programování v MIT App Inventorem jsou dostupné v odborných článcích a návodech, které popisují, jak programovat v tomto prostředí. Mgr. Petr Němec píše v jednom článku na spomocnik.rvp.cz o vytvoření programu jednoduchého programu právě v prostředím MIT App Inventor. [1] Programování v MIT App Inventorem vychází z vytvoření programu v prostředí webové stránky. Toto prostředí bychom mohli považovat za cloudové prostředí. Pro zobrazení průběhu programu a jeho kompilace je nutné si do smartphonu nainstalovat potřebnou aplikaci, nebo se spokojit s prací v emulátoru[2], která bude emulovat prostředí Androidu. Vývojové prostředí využívá blokové skládání programu, stejné jako se používá v originálním prostředí NXT 2.0.

Studování instrukcí, jak programovat v programu Catroid, budou předcházet informace z odborných článků v anglickém jazyce. Jedna z odborných prací nese název *Catroid: A Mobile Visual Programming System for Children*, kde autor Wolfgang Slaný popisuje široké možnosti využití.[2] Tento způsob programování se realizuje pouze na smartphonu uživatele (programátora). Tato aplikace je dostupná v obchodě Google Play. Aktuálně se nedá dohledat pod názvem Catroid, nyní je aplikace dostupná v balíku programu nesoucí název Pocket Code: Learn programing. Aplikace je od stejného vývojáře tj. Catrobat.

V České republice jsou základní a střední školy, které nabízejí zájmové kroužky právě s tímto druhem stavebnic od firmy Lego. Od roku 2014 se díky projektu nesoucí název *Podpora přírodovědného a technického vzdělávání v Karlovarském kraji* mohou žáci seznamovat na Základní škole Ostrov. Jednou ze škol je i Základní škola Hustopeče, kde ke vzniku kroužku napomohl projekt *Modernizace vzdělávání*. Bakalářská práce bude přínosem ve využití moderních technologií při práci s roboty v rámci výuky či kroužku.

1.3 Metody práce

Prvním krokem této práce bude zmapování oblasti aplikací vhodných pro programování Lego Mindstorms a to i aplikací pro platformu Apple. Následně provedení analýzy pro nashromáždění potřebných informací. Stěžejní je způsob, jakým má daná aplikace implementované prostředí, jak umožňuje pracovat s řídícími strukturami jako jsou podmínky, cykly, ale i samotné proměnné či konstanty. Tento rozbor se bude týkat především aplikací pro platformu Android z důvodu velké rozšířenosti tohoto operačního systému.

Tyto rozbory budou sloužit ke vzájemnému porovnání jednotlivých prostředí a jejich parametrů. Při realizaci teoretické části bude popsána tvorba programů, čehož bude dále využito v praktické části konkrétně při tvorbě sady úloh. Poslouží také ke zmapování úskalí vývoje v daném prostředí a jeho funkcionalitám. Jednotlivé úlohy jsou rozvrženy v části “Cíle práce”.

[2]Emulátor je druh softwaru, který umožňuje běh programu na jiné platfromě, než na kterou byl program vytvořen.Emulace je speciální případem virtualizace. [5]

V praktické části je potřeba, aby všechny vytvořené programy byly nasaditelné na zvolenou konstrukci robota bez nutnosti vytvoření manuálu na sestavení robota. Pokud se využije jedné ze základních konstrukcí, postačí návod dodávaný ke každé stavebnici.

2 Vymezení pojmů

2.1 Android

Android je mobilní operační systém založený na jádře Linux. Systém je dostupný jako open source. Je používán především na smartphonech, tabletech a dalších tzv. chytrých zařízeních. Systém je vyvíjen pod záštitou firmy Google.

2.2 Vývojové prostředí

Vývojové prostředí je software, který usnadňuje práci programátorů vetšinou zaměřených na jeden programovací jazyk. Prostředí je navržené tak, aby zvýšilo produktivitu programátora pomocí provázaných komponent s GUI.

2.3 Emulátor

Emulátor je druh softwaru, který umožňuje běh programu na jiné platfromě, než na kterou byl program vytvořen. Emulace je speciální případem virtualizace.[5]

2.4 Drag & drop

Drag & drop je v informatice operace používaná v GUI, kdy uživatel v počítači „uchopí“ pomocí ukazovacího zařízení (např. myši) virtuální objekt (např. soubor nebo ikonu) a přesune ho „přetažením“ na jiné místo (nebo na jiný objekt).

Operace drag & drop může být využita k vyvolání různých druhů akcí nebo vytvářet různé typy asociací mezi dvěma virtuálními objekty.

Podpora operace drag & drop není ve všech programech, i když je poměrně rychlá a její technika se dá snadno osvojit. Nevýhodou je, že uživateli nemusí být jasné, že objekt lze přetáhnout nebo že se některé příkazy provádějí operací drag-and-drop. [7]

Na nových dotykových mobilních telefonech čili smartphonech je technologie drag & drop nezbytná. Na základě této technologie je postavena např. funkce odinstalování aplikace nebo přesouvání souborů či aplikaci do složek.

2.5 Catroid

Pocket Code je mobilní aplikace od vývojářské firmy Catrobat dostupná na platformě Android, která umožnuje ovládat či programovat roboty Lego Mindstorms přimo z mobilních telefonů nebo-li smartphonu. [4]

Pocket Code disponuje i mnohem rozsáhlejšími funkcemi. Dokáže fungovat jako samotná aplikace pro vytáření programů s grafickými prvky. Při vytváření se zde skládají za sebe či různě do sebe jednotlivé funkční bloky. Díky drag & drop technologii je utváření programu snažší a přehlednější. [4]

2.6 MIT App Inventor

MIT App Inventor je inovativní úvod do programování a aplikací stvoření pro začátečníka, které transformuje složitý textový jazyk kódování do vizuální podoby.

Už z názvu je patrné, že MIT App Inventor není plnohodnotná mobilní aplikace. Jedná se o webové rozhraní, které se spojuje pomocí dostupné Wifi sítě, vygenerovaného kódu nebo QR kódu, potřebnou aplikací od vývojáře s mobilním telefonem.

Využíváno je zde drag-and-drop stavebních bloků. Díky jednoduchému grafickému rozhraní dokáže i nezkušený nováček vytvořit základní, plně funkční aplikace během jedné hodiny nebo méně. Jelikož, se jedná o desktopové řešení, má MIT App Inventor mnohem více funkcí. [3]

3 Přehled vývojových prostředí

Z úvodu práce je vytvořen přehled všech vývojových prostředí pro vývoj programů či aplikací pro Lego Mindstorms. Z tohoto přehledu bude následně vybrané prostředí, které bude splňovat hlediska programování na základní škole. Porovnání prostředích bude nastíněno v příslušné tabulce níže.

Graficky orientované prostředí

- Lego Mindstorms NXT - je vývojové prostředí, které je dodávané v sadě s robotem. Toto prostředí umí využívat i senzory od jiných výrobců.
- MIT App Inventor - je desktopové vývojové prostředí pro tvorbu aplikací pro platformu Android.
- Pocket Code - je mobilní vývojové prostředí pro tvorbu grafických aplikací, které umí pracovat s několika stavebnicemi.
- Robolab grafické prostředí pro práci s Lego roboty, které bylo vyvinuto pro LabVIEW [10]
- LabVIEW - je grafické prostředí pro návrh systémů vyžíváné spíše v SW inženírství [10]
- Scratch X beta - aktualizované grafické prostředí ve vývoji. Aktuálně umí pracovat se stvabenicí EV3
- Microsoft's Robotics Developer Studio - prostředí pro robotické ovládání nebo simulování pro akademické, koníčkové nebo komerční použití [10]

Textově orientované prostředí

- Robot C - programovací jazyk založený na programovacím jazyku C [10]
- NBC - jazyk symbolických znaků [10]
- NXC - výšší programovací jazyk založený na NBC [10]
- pbLua - univerzální scriptovací jazyk
- Robotary Review - programovací jazyk obdobný programovacím jazýkům Java nebo C# [10]

Název aplikace	Podobnost se Scratch	Přehlednost prostředí	Jednoduché blokové skládání	Barevné oddělení bloků
Lego Mindstorms NXT	✗	✓	✓	✓
MIT App Inventor	✓	✓	✓	✓
Pocket Code	✓	✓	✓	✓
Robolab	✗	✗	✗	✗
LabVIEW	✗	✗	✗	✗
Scratch X beta	✓	✓	✓	✓
Microsoft's Robotics Developer Studio	✗	✗	✗	✓

Tabulka 1: *Porovnání z hlediska programování na základní škole - 1. část*

Název aplikace	Snadná práce s txt. řetězci a proměnými	Základní matematické operace	Jazykové mutace	Vývoj mobilních aplikací
Lego Mindstorms NXT	✓	✓	✗	✗
MIT App Inventor	✓	✓	✓	✓
Pocket Code	✓	✓	✓	✓
Robolab	✗	✓	✗	✗
LabVIEW	✗	✓	✗	✗
Scratch X beta	✓	✓	✓	✗
Microsoft's Robotics Developer Studio	✗	✓	✗	✗

Tabulka 2: *Porovnání z hlediska programování na základní škole - 2. část*

4 Porovnání již vytvořených aplikací

Práce se zabývá porovnáním a popsáním aplikací MIT App Inventor a Catroid. Aplikace jsou porovnávány v několika aspektech jako je např. uživatelský dojem z prostředí, nutnost instalace potřebné aplikace nebo možnost využití externích zdrojů, jež mohou být přínosem. Nicméně práce popisuje vytváření programu v každém prostředí a zhodnocuje klady a zápory dané aplikace. Dále zde naleznete seznam mobilních aplikací napříč mobilními platformami, které dokáží pracovat se stavebnicemi Lego Mindstorms. Dalším kritériem je dostupnost aplikace - v práci je tento aspekt klíčový.

Seznam aplikací je vybírán z výčtu aplikací[3] z roku 2014 roboticky zaměřeného maďarského portálu

[3]Přístupné na adrese https://www.smashingrobotics.com/mobile-apps-for-interacting-with-your-mindstorms-nxt-robot/

Smashing robots (smashingrobotics.com) a serveru Microsoft (microsoft.com) a Windows phone apks (windowsphoneapks.com). Přičemž jsou ze seznamu vybrány jen ty nejznámější. [4]

Název aplikace	Android	iOS	Windows phone
NXT Remote Control	✓	✗	✗
NXT Simple Remote	✓	✗	✓
Lego Drive	✗	✓	✗
iNxt	✗	✓	✗
Robo Remote	✗	✓	✗
NXT pilot	✗	✗	✓
Mindstorms Remote	✗	✗	✓
Lego Nxt Remote	✗	✗	✓
NXT Robot Controler	✗	✗	✓

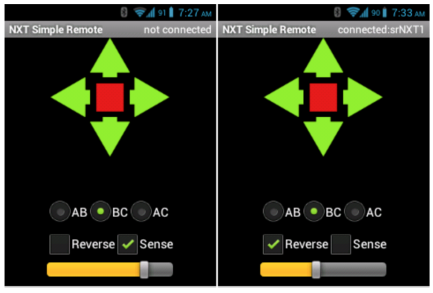
Tabulka 3: *Seznam dostupných aplikací na mobilní platformy [**?**]*

4.1 NXT Simple Remote

NXT Simple Remote patří do sady aplikací od Ferdinanda Stuecklera, které pracují různými způsoby právě s Lego roboty. Pro připojení smartphonu je nutné mít obě zařízení zpárovaná.

NXT Simple Remote umožňuje ovládat robota skrze šipky na ovládací ploše. Pokud máme zakškrtnutou volbu ”Sense”, je robot uveden do pohybu jen po dobu stlačeného příslušného tlačítka. Avšak, pokud tato volba není zaškrtnuta, pohybuje se robot příšlušným směrem do doby, dokud je stisknuté jiné tlačítko. Plusem je, že tyto volby lze provádět během chodu přímo na pracovní ploše. Aplikace disponuje i možností reverzovat chod motoru. Mínusem je však nemožnost zvolit si, který směr chceme reverzovat, jak je to např. v aplikaci NXT Remote Control. Signál je vysílán po dobu stlačení tlačítka.

Aplikace je dostupná na platformě Windows phone a Android. Pro spojení s robotem je využíváno technologie Bluetooth.



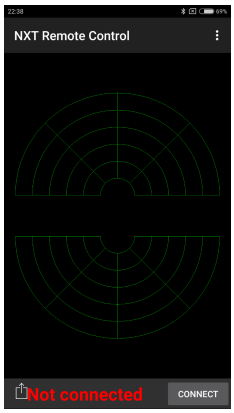
Obrázek 1: Ovládací plocha NXT Simple Remote

[4]Celý seznam naleznete na https://www.smashingrobotics.com/mobile-apps-for-interacting-with-your-mindstorms-nxt-robot/, https://www.microsoft.com/en-us/store/search/apps?q=nxt,

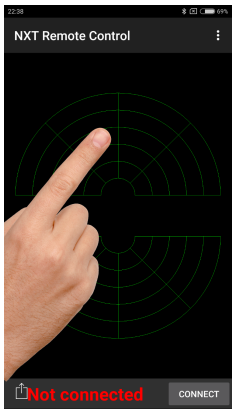
4.2 NXT Remote Control

Aplikace je obdobou NXT Simple Remote a taktéž je od vývojáře Ferdinanda Stuecklera. Uživatelský dojem z této aplikace je o poznání lepší a to díky větší dotykové ploše směrových tlačítek. S touto aplikací se připojuje k robotu stejným způsobem, jako tomu je u aplikace NXT Simple Remote a taktéž je vyžadováno zpárování s příslušným telefonem.

Pracovní plocha v defaultním nastavení je složena ze šipkových tlačítek pro udání směru viz Obrázek 6: Ovládací plocha NXT Remote Control. Avšak, nastavení nám umožňuje ovládat robota i přes jiné ovládácí prvky. Jedním z nich je tzv. Touch bar controler, kde jsou dva půlkruhy - jeden pro každý směr. Půlkruh je rozdělen výseky na 4 sektory vždy po 22,5° pro udání směru robota. Dále je půlkruh napříč sektory rozdělen cylindry na 4 stejně velké bloky. Toto rozvržení dotykové plochy umožňuje jedním přítlakem prstu nastavit směr a rychlost robota.



Obrázek 2: Ovládací plocha Touch Bar

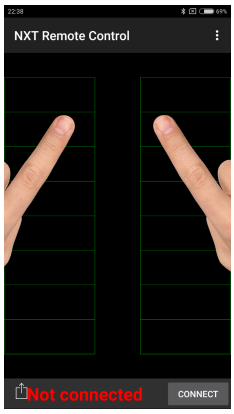


Obrázek 3: Ukázka ovládání

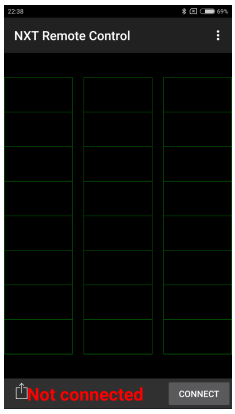
Další rozvržení plochy je nazýváno Tank Controls, kde jsou 2 sloupce rozděleny na 8 buněk. Čtyří horní buňky slouží pro určení rychlostí vpřed a čtyří dolní pro určení rychlosti vzad. Pokud jsou stisknuta jednotlivě pole v jednom sloupci nezávazně na tom druhém, začne robot zatáčet na místě dle toho, v jakém sloupci se nachází buňka. Ovládání vpřed a vzad je zde stejné jako v tanku.

Pokud jsou stisknuty stejné buňky v obou sloupích, tak robot jede vzad nebo v před, dle toho v jaké úrovni sloupce se buňky nachází (viz. Obrázek 4 - Ovládací plocha Tank). Když stiskneme naráz libovolné buňky v každém sloupci, robot zatáčí za jízdy a to buď dopředu, nebo dozadu dle urovně jednotlivých buněk.

Následujíci rozvržení nazývající se 3 motors Control ovládá robota na stejném principu jako Tank Control. Rozvžení se liší jen tím, že je zde o sloupec navíc pro ovládání třetího motoru.

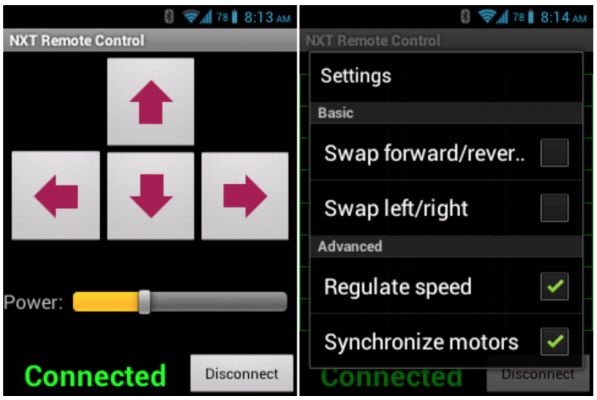


Obrázek 4: Ovládací plocha Tank



Obrázek 5: Ovládací plocha 3 Motors

Plusem této free aplikace je možnost nastavení reverze motoru, kde jsou na výběr v případě opačného zapojení motorů možnosti pro reverzi zatáčení nebo zvlášť pro chod vpřed a vzad. Dále aplikace disponuje jezdcem pro zvolení rychlosti viz. Obrázek 5.



Obrázek 6: Ovládací plocha NXT Remote Control

4.2

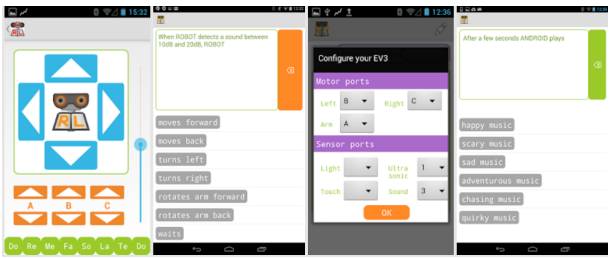
4.3 Aplikace pro WP a iOS

4.3.1 App RoboLiterate: LEGO NXT Remote APK for Windows Phone

V době psaní teoretické části nebyl k dispozici smartphone se systémem Windows Phone, proto nemohla být funkcionalita těchto aplikací otestována. Tato aplikace pracuje stejně jako jiné dostupné na platformě

Adnroid a to na základě bluetooth připojení. Pracovní plocha je složena ze šipkových tlačítek pro udání směru. Dále je vybavena tlačítky pro výběr připojeného motoru.

Na obrázku dole jsou evidentní prvky ze systému Android, ale všechny dostupné screeny naznačují, že takto vypadá aplikace i na WP.

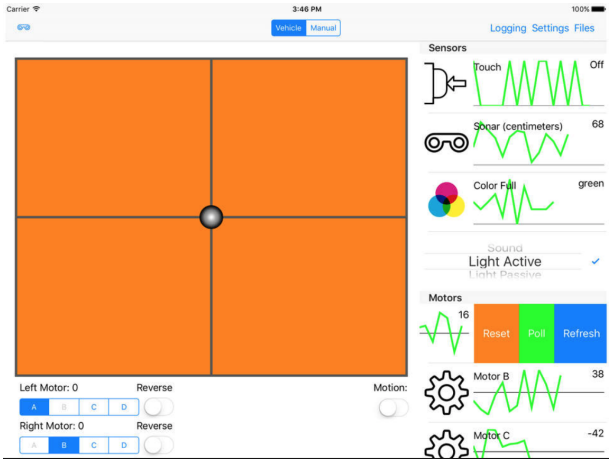


Obrázek 7: App RoboLiterate: LEGO NXT Remote AP for WP

4.3.2 iNXT Remote

iNXT Remote je kompletní aplikace, která umožňuje dálkové ovládání NXT robota buď dotykem, nebo pomocí akcelerometru v zařízení. Můžete také snímat a logovat hodnoty ze snímačů připojených k NXT kostce, kontrolovat provozní režim senzorů, nahrávat, stahovat nebo mazat soubory z paměti NXT. Aplikace přenáší informace do robota skrze Bluetooth. Lze zakoupit v obchodě iTunes nebo Apple store za $ 4,99. [8]

I přes možnost otestování této aplikace na dostupném zařízení s iOS bylo rozhodnuto žádnou aplikaci nekupovat. Jediná bezplatná aplikace je dostupná na systém iOS 6 a nižší, proto v současné době neexistuje žádná volně dostupná aplikace ke stažení.



Obrázek 8: Ovládací plocha iNXT Remote

4.4 Přehled aplikací dle dostupnosti

V celém seznamu dostupných aplikací jsou k nalezení i aplikace, které nejdříve musíme koupit, než by se začali používat. Tyto aplikace jsou vytvářeny především na platformu Apple a jejich systémy. Tyto aplikace

by tedy měly přinášet uživateli odlehčení od řešení tzv. bugů. [5]

Tato práce je především zaměřena na aplikace, které jsou volně ke stažení na určených místech jednotlivých platforem, aby nikterak nezatěžovala školní nebo soukromý rozpočet. Údaje o dostupnosti aplikací jsou zaznamenány v příslušné tabulce.

Název aplikace	Volné	Placené
NXT Remote Control	✓	✗
NXT Simple Remote	✓	✗
Lego Drive	✓	✗
iNxt	✗	✓
Robo Remote	✗	✓
NXT pilot	✗	✓
Mindstorms Remote	✓	✗
Lego Nxt Remote	✓	✗
NXT Robot Controler	✓	✗

Tabulka 4: *Přehled dostupnosti aplikací [**?**]*

5 Srovnání vlastností porovnaných aplikací

Systémy se liší ve funkcích, které poskytují. Aby bylo možné je porovnávat, jsou stanovena kritéria, která odpovídají hlavním funkcím těchto systémů. Tato kritéria jsou pro názornost zaneseny do příslušných tabulek.

5.1 Základní funkce

V tabulce 5 je sestaven přehled základních zjišťovaných možností. Kromě ovládání pohybu robota jsou zkoumány možnosti nastavení rychlosti a změna chodu motoru. Všechny aplikace kromě aplikace App RoboLiterate: LEGO NXT Remote APK for Windows Phone těmito možnostmi disponují.

[5] Bug je druh softwarové chyby, kterou udělal programátor při vytváření počítačového programu.

Název aplikace	pohyb robota	nastavení rychlosti	reverze motoru
NXT Simple Remote	✓	✓	✓
NXT Remote Control	✓	✓	✓
LEGO NXT Remote APK for WP	✓	✓	✗
iNXT	✓	✓	✓

Tabulka 5: *Základní funkce aplikací*

5.2 Vlastnosti a rozšiřující funkce

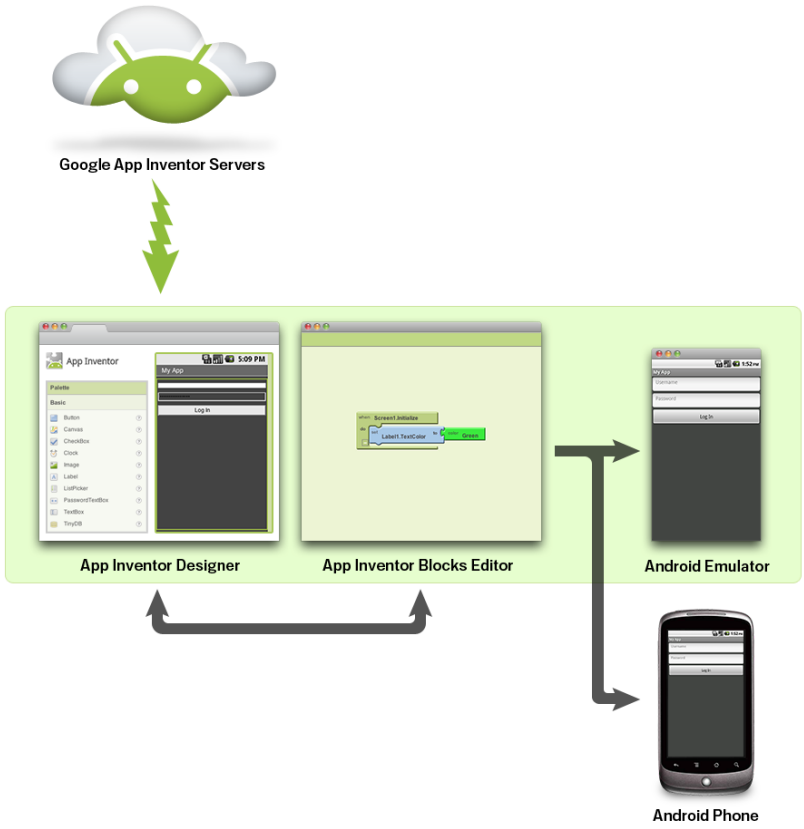
Každý systém poskytuje určitou paletu rozšiřujících funkcí. Mezi zkoumané patří možnosti výběru zapojení motorů, práci se senzory a změna pracovní plochy. Všechny aplikace disponují anglickým jazykem bez možností jeho přepnutí. Žádná z aplikací se k robotu nepřipojí automaticky. Vždy je vyžadováno párování s příslušným telefonem. Po zpárování se NXT zobrazí jako položka pro přípojení, avšak nejprve je potřeba začít scanovat okolní prostředí.

Název aplikace	zapojení motorů	práce se senzory	změna ovládací plochy
NXT Simple Remote	✓	✗	✗
NXT Remote Control	✓	✗	✓
LEGO NXT Remote APK for WP	✓	✓	✗
iNXT	✓	✓	✓

Tabulka 6: *Rozšiřující funkce analyzovaných systémů*

6 MIT App Inventor

MIT App inventor umožňuje dle oficiálních stránek vyvíjet aplikace pro systém Android pomocí webového prohlížeče. Tyto aplikace lze spustit skrze potřebnou aplikaci MIT Ai2 companion, nebo lze využít přímo možnost emulátoru v prohlížeči. K vývoji je využíváno webové prostředí. K uchování a přehledu veškerých vytvořených projektů a programů slouží aktuálně Google App Inventor severy. Názorně zobrazeno na obrázku oficiální distribuce MIT.



Obrázek 9: Princip fungování MIT App Inventor

Při sestavování aplikace se pracuje s dvěmi části prostředí. První prostředí se nazývá App Inventor Designer, kde si vybíráme komponenty pro svou aplikaci. Druhá část App Inventor Blok Editor, je používána pro nastavení programových bloků, které určují, jak se budou jednotlivé komponenty chovat. Jednotlivé funkcionality komponent se sestavují dohromady jako kousky skládačky.

Pokud není k dispozici telefon s Androidem, můžete si vytvořit své aplikace pomocí Android emulátoru, který je spuštěn na počítači a chová se stejně jako telefon. Vývojové prostředí MIT App Inventor je podporované na platformách Mac OS X, GNU / Linux, operačních systémech Windows a několik populárních modelů se systémem Android.

Aplikace vytvořené pomocí App Inventor mohou být instalována na libovolném telefonu se systémem Android. Dříve než budete moci používat App Inventor na PC, je třeba nainstalovat App Inventor Setup balíček.[9]

6.1 Přihlášení

Přihlášení je prvotní akcí, kterou celý proces vytváření vlastní aplikace začít. K přihlášení je zapotřebí vlastnit účet od Googlu, který by měl být součástí vlastnictví smartphonu se systémem Android. Tento systém vyžaduje při prvním spuštění přihlášení do příslušného účtu právě od firmy Google. Pokud účet ještě není založený, je nutností si jej vytvořit registrací.

6.1.1 Možné problémy

Problém může nastat v momentě, kdy jsou zapomenuty přihlašovací údaje. Zejména se jedná spíše o zapomenuté heslo, jelikož uživatelské jméno je k nalezení v "Nastavení"pod záložkou Účty. Poté si na příslušné stránce lze vygenerovat nové heslo a problém je vyřešen.

V případě nenalezení uživatelského jména nebo jiných problémů, které mohou nastat při generování nového hesla, je možné vytvořit účet úplně nový. Zde mohou nastat zase jiné problémy při zadání již použitých iniciálů v předchozí registraci. Zde by bylo vhodné řešení registrovat účet na jiného příslušníka rodiny nebo zadávané údaje změnit.

6.2 Nejvyužívanější záložky menu

Horní menu je rozděleno na dvě části - levou a pravou. Levá část slouží pro práci s projektem, kde jsou nástroje pro sdílení programu, propojení aplikace s příslušným telefonem a případnou rychlou pomoc. Pravá část je určena pro změnu a informace o přihašeném účtě a pro případné řešení problému v průvodci či na oficiálním fóru.

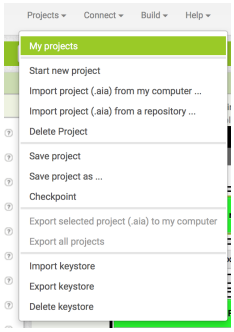
Nejvyužívanější záložky jsou však spíše v levé části menu, kde jsou k nalazení funkce pro uložení, spuštění programu a pomoc s vývojem aplikace.

6.2.1 Levá část menu

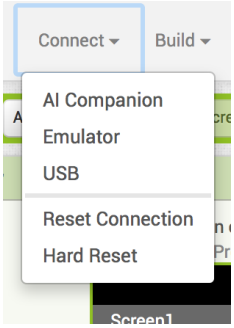
Tato část obsahuje celkem 4 záložky s drop-down menu, kde je k nalezení několik funkčncích podzáložek. Důležité jsou zejména první dvě záložky, které jsou využity po celou dobu vývoje aplikace. Zbylé 2 záložky slouží jako doplňkové služby a rychlou nápovědu při vývoji. Jednotlivé záložky a jejich funkčnosti jsou rozvedeny a následně popsány v položkách seznamu.

6.2.1.1 Záložky - Projetcs

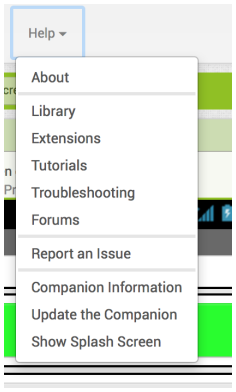
- Projects - obsahujes funkce převážně pro práci s vyvíjenou aplikací.
- Connect - již z názvu záložky je patrné, že jsou zde k nalezení funkce pro spuštění aplikace a to hned několika způsoby.
- Help - ze záložky help je nevyužívanější Report an Issue, kde jsou k nalezení veškeré informace ohledně každé komponenty. Nejvíce bylo využíváno pro komponenty NXT.



Obrázek 10: Drop-down Project



Obrázek 11: Drop down Connect



Obrázek 12: Dropdown Help [**?**]

6.2.2 Pravá část menu

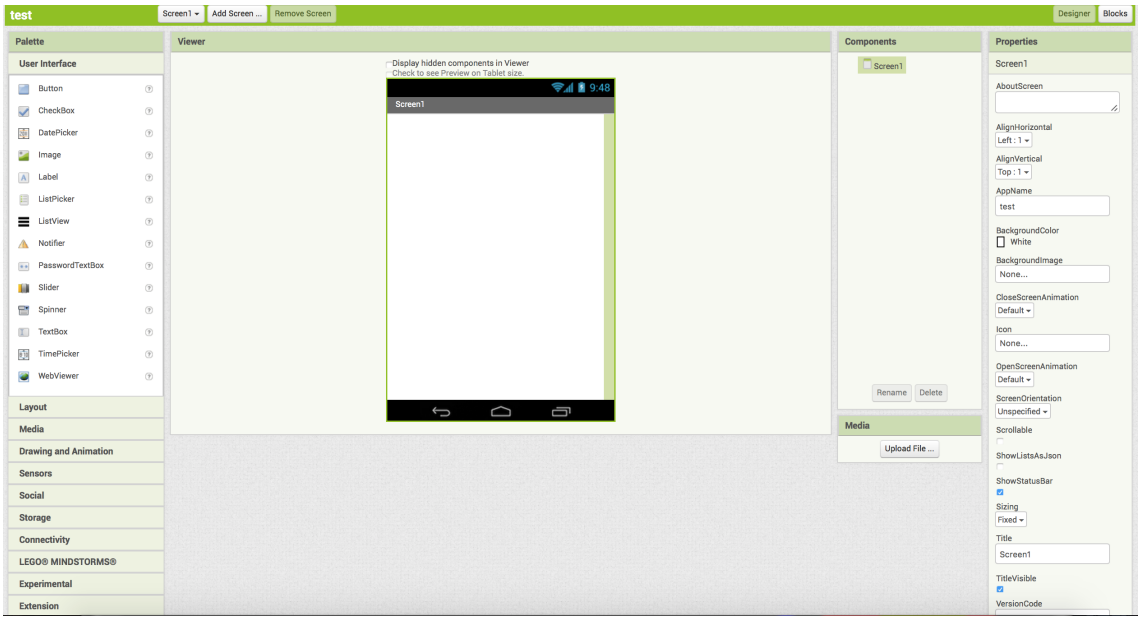
Pravá část menu prakticky žádné drop-down menu neobsahuje. Vyjma drop-down pro práci s účtem a mutace, která je popsaná v sekci Mutace. Funkce jednotlivých záložek je nastíněna v následujícím seznamu. Nejvyužívanější prvky jsou zde převažně stejné záložky jako v levé části menu.

- My projects - odkazuje na stránku, kde se nachází výčet vytvořených projektů příslušného účtu. Jedná se také o uvodní stránku po přihlášení.
- Report an Issue - jsou zde k nalezení veškeré informace ohledně každé komponenty. Nejvíce bylo využíváno pro komponenty NXT.

6.3 GUI vývojového prostředí

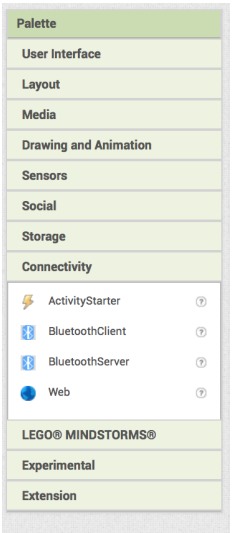
GUI je rozděleno na 5 částí. Každá část má svou jedinečnou funkci ve vývoji, ale zároveň se navzájem doplňují. Rozvržení GUI je navrženo tak, aby korespondovalo s postupem vývoje aplikace. Vývoj aplikace se skládá z několika kroků, které musí být splněny. Vývoj aplikace je zdokumentován v následujících částech baklářské práce.

Rozvržení začíná horním panelem, kde je k dispozici název příslušné aplikace a tlačítka nastavení pro výběr, přidání nebo smazání vývojářské plochy. V pravé části tohoto panelu jsou tlačítka pro přepnutí mezi designerskou částí, kde je hlavním úkolem sestavit GUI aplikace a blokovou částí, kde je jednotlivým částem navrženého GUI nastavována funkcionalita.



Obrázek 13: Rozvržení pracovní plochy [**?**]

Část která se nachází na levé straně vývojového prostředí, slouží k výběru jednotlivých komponent vyvíjené aplikace. Na výběr je z 11 paletek, které jsou nazvány tak, jaké komponenty jsou v dané paletce k nalezení. Po rozkliknutí paletky jsou vyobrazeny jednotlivé komponenty, které se musí v případě použití pomocí drag-and-drop přetáhnout do pracovní plochy vyvijené aplikace, která je další částí GUI vývojového prostředí.

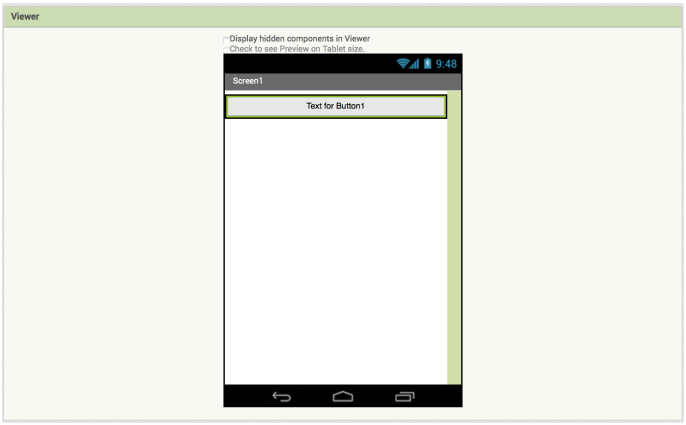


Obrázek 14: Levý panel GUI

Pracovní plocha, která je již zmíněna v předchozím odstavci, je tou kreativní částí, kde by se měla navrhovat GUI vyvíjené aplikace. Ta by se měla řídit alespoň částečně určitými UX pravidly. V pracovní ploše se skládají komponenty z paletek v levé části GUI a pro lepší rozvržení komponentů je doporučeno používat rozvržení z paletky Layout. Jednotlivé pracovní plochy lze přidávat v Horním panelu GUI. Lze tak vyřešit nedostatek místa při naplnění jednoho GUI pracovní plochy. Funkcionality pro otevření další

pracovní plochy je nastavována v blokové části vývojového prostředí. Tato funkcionalita v případě využití více ploch, musí být nastavena. V opačném případě se k ní uživatel nikdy nedostane.

Z hlediska náročnosti pochopení principu a chování aplikace je vhodnější vyvinout aplikaci s jednou ovládácí plochou. Je proto hned několik důvodů, ale ten hlavní spočívá v neznalosti druhé pracovních plochy naměřených hodnot, uložených proměných či použitých komponentů z předešlé či hlavní pracovní plochy.



Obrázek 15: Pracovní plocha - náhled

Další částí GUI vývojového prostředí je část s názvem Components. Zde je k dispozici přehled veškerých komponentů z levé části prostředí, které jsou použity ve Vieweru vytvářené aplikace. Specifické rozdělení zde utváří komponenty ze záložky Layout. Tyto komponenty rozdělují seznam komponent na podseznamy, dle toho v jakém použitém layoutu GUI se komponenta nachází.

Další částí této části GUI jsou dvě tlačítka s názvem Rename a Delete. Tlačítko Delete je velmi užitečné a to zhlediska, že je to jediné mazací tlačítko pro mazání komponentů. Pokud tedy je zvyklostí mazat komponenty rovnou z Vieweru (kde je rozvrženo GUI vývíjené aplikace) pomocí klávesy Delete popř. Backspace, tak zde zůstanou bez odezvy. Výhodou tohote řešení je nesmazání komponent mylným nebo chybným stisknutí klávesy.

Komponenty jsou při použití nazývány svým název a příslušnou hodnotou, dle počtu použité stejné komponenty. Pro lepší pochopení - pokud jsou použita 3 tlačítka, jsou názvány jako textButton1, textButton2, textButton3 dle pořadí, v kterém byla přidána do GUI aplikace. Při programování funkcí těchto tlačítek a použití těchto názvů se snadno vnese do aplikace chaos a nepřehlednost při řešení vzniklých problémů. Funkce Rename tedy slouží pro nastolení určíté přehlednosti a přináší řád do aplikace. Pokud se tedy vytvoří tlačítko, které slouží k připojení, je třeba ho nazvat správným a výstižným názvem např. connectButton nebo pripojovaciTlacitko. Je třeba myslet, že při nazývání komponent nebo funkcí se nepoužívá diakritika.

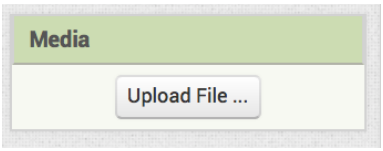
Více informací, jak si držet přehlednost aplikace je nastíněno v sekci Vývoj aplikace v MIT App Inventoru. V této sekci je nastíněn vývoje jednoduché aplikace, kde je postupováno tak, aby splňovala výše zmíněné nezbytné náležistosti.

V této části se vyskytuje ještě nástroj Media. Tento nástroj se dá využít pro náhrání příslušných obrázků,

zvuků, nebo videí. Nástroj Media slouží pro vylepšení vyvíjené aplikace buď lepšími grafickými prvky např. tlačítek nebo spouštění zvuků při stisku tlačítek, nebo zahájení určité fáze aplikace. Po kliknutí na tlačítko Uploud file se zobrazí Pop-up okno, kde je možnost pod tlačítkem vybrat soubor z počítače. Příslušný soubor se po potrvzení nahraje a zobrazí se nad tlačítkem Upload file. Menu s akcemi pro stažení nebo smazání je zobrazeno po kliknutí na název příslušného souboru.



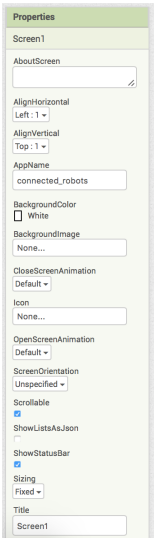
Obrázek 16: Komponenty



Obrázek 17: Media

Poslední částí GUI je pravá část nazývána Properties. Tato část slouží pro nastavení vlastností jednotlivých použitých komponent jako je zobrazovaný text, velikost textu, barva komponentů nebo jejich velikost. Obecně tedy slouží pro nastavení vlastností, což je také překlad slova properties. Defaultně je zobrazeno nastavení vlastností úvodní obrazovky vyvíjené aplikace.

Celé využití této části je nastíněno v sekci Vývoj aplikace v MIT App inventor a je hojne využíváno v aplikacích, které jsou popsány v praktické části bakalářské práce. Jedna z vlastností je např. volba pro nastavení obrázku na pozadí komponentu, které byly případně nahrány v části Media vývojového prostředí.



Obrázek 18: Část Properties

6.3.1 Bloková část

Jak už bylo zmíněno, vývojové prostředí MIT App Inventor je rozděleno na 2 části. Ta první, designerská část, kde se vybíraly a nastavovaly potřebné komponenty pro vývoj příslušné aplikace, je popsáná v několika odstavcích výše. V této blokové části se vybraným komponentům vštěpují funkce, aby aplikace pracovala, jak je žádáno. Jednotlivým komponentům se zde ”vdechuje život”.

GUI této části prostředí je rozděleno na méně částí, než část designerská. Prostředí může působit oproti předchozím trochu jednodušeji. Je rozděleno pouze na dvě části Blocs (bloky) a Viewer (náhled - prohlížeč).

První část Blocks je složena ze seznamu Built-in a vybraných komponentů, které byly vybrány v editorské části. Proto je stejný seznam k náhlednutí v části Components v GUI Designer MIT App Inventors. Built-in je složen z několika vyskakovacíh pod seznamů. Vyskakovacích z důvodu vyskočení postraního panelu v sekci Vieweru s jednotlivými funkčními bloky. Jednotlivé bloky se skládají pomocí drag-and-drop v příslušné pracovní ploše (Viewer). Pokud na některých z bloků je najeto kurzorem myši a posečkáno pár sekund, tak vyskočí informace ohledně funkce bloku. Bloky, které jednotlivé seznamy obsahují, budou popsány.

Druhá část viewer se chová podobně jako pracovní plocha v GUI designer. Pomocí drag-and-drop jsou na plochu přetahovány a sestavovány do funkčních bloků. Pracovní plocha dále disponuje užitečnými nástroji. Jedním z nich je Backpack, v překladu batoh. Nástroj slouží pro jednoduché uložení vytvořeného bloku. Druhým nástrojem je koš pro snadné vyhození bloků, které jsou přebytečné a neplní už žádnou funkci aplikace.

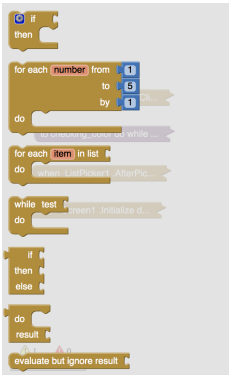
Pracovní plocha disponuje ještě indikátorem Chyb a varování, který je k nalezení v levém spodním rohu. V případě výskytu určitého upozornění je tento případ zaznamenán v příslušném poli. Nemá však vliv na funkčnost aplikace. Pakliže se vyskytne chyba, je taktéž zaznamenána v příslušném poli, tato indikace sděluje uživateli výskyt problémů, který je pro kompilaci aplikace nutné vyřešit.

6.3.2 Blocks - pravá část GUI Blocks

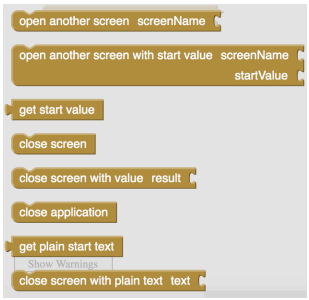
Jak je nastíněno výše, část Built-in slouží pro zvolení správné funkce. Tyto funkce jsou uloženy v několika seznamech, které jsou následně rozebrány a popsány. Jednotlivé funkce jsou vhodně roztříděné dle kategorie. Jedná se o kategorie Control, Logic, Math, Text, Lists, Colours, Variebles, Procedurec. Název každé kategorie sděluje, které nástroje pro funkčnost komponentů jsou v kategorii obsaženy. Část Screen, která je nazvána dle pojmenování, obsahuje funkční bloky pro přímé nastavování funkce nebo změnu jednotlivých komponentů.

6.3.2.1 Built-in

- Control - obsahuje funkční bloky pro vytvoření podmínek, cyklů (foreach, for a while). Dále Kategorie disponuje nástroji pro práci s obrazovkami (GUI vyvinuté aplikace) nebo pro ukončení aplikace.

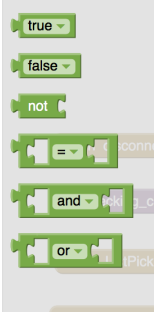


Obrázek 19: Ovládací bloky - první část



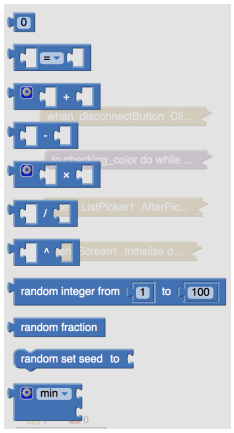
Obrázek 20: Ovládací bloky - druhá část

- Logic - v této sekci jsou k nalezení bloky pro logické operace. Zahrnuje operandy AND, OR, True, False.

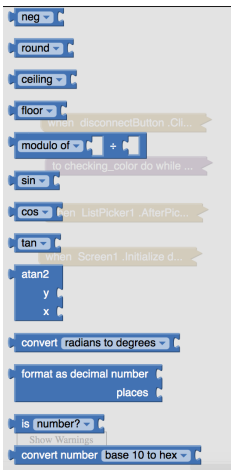


Obrázek 21: Logické bloky

- Math - zahrnuje operanty a nástroje pro většinu matematických funkcí. Obsahuje bloky pro náhodné vyplnění čísel i v určitém intervalu. Dále je možno využít bloky pro absolutní hodnotu, negaci, dělení se zbytkem, geometrické funkce nebo funkce isNumber(soustava Hex, dec...) sloužící pro zjištění nebo kontrolu soustavy.



Obrázek 22: Matematické bloky - první část

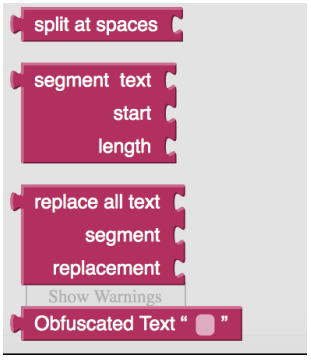


Obrázek 23: Matematické bloky - druhá část

- Text - obsahuje nástroje pro práci s textovým řetězcem. K dispozici jsou bloky, které vloží text do příslušné Stringové [6]proměnné, porovná dva různé texty, zjistí délku řetězce. Další funkce slouží pro rozdělení textového řetezece nebo pro vypuštění mezery či libovolného nechtěného znaku.



Obrázek 24: Bloky pro práci se String - první část



Obrázek 25: Bloky pro práci se String - druhá část

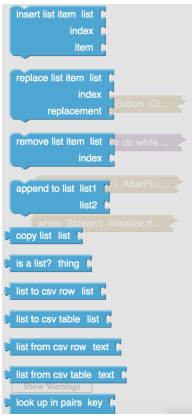
- Lists - funkční bloky pro vytvoření seznamu pro uložení. Vytvořit list nebo-li seznam je jediná

[6]Textový řetězec je v programování název datového typu sloužícího k uložení konečné posloupnosti znaků.

možnost v MIT, jak vytvořit datový objekt[7] obdobný poli. Další nástroje slouží pro práci s datovým objektem. K dispozici jsou funkce pro detekování seznamu, detekování prázdnosti seznamu, vložení položky na určený index[8] nebo výběr položky z určeného indexu. Další nástroje slouží pro import nebo export CSV[9] souborů.



Obrázek 26: Bloky pro práci se Seznamy - první část

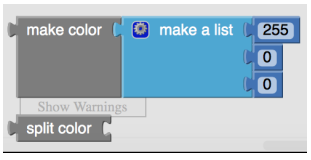


Obrázek 27: Bloky pro práci se Seznamy - druhá část

- Colors - obsahuje paletku 13 barev v blocích k okamžitému použití bez nutnosti zadávat RGBA[10]. Po použití a rozkliknutí libovolného bloku se zobrazí paletka o 63 barvách. Nástroj dále obsahuje funkci pro zobrazení vlastní barvy pomocí zadání RGB kódu. Posledním a užitečným nástrojem pro práci s NXT Color senzor je Split color pro rozdělení naměřené hodnoty a určení barvy.



Obrázek 28: Bloky pro práci se Seznamy - první část



Obrázek 29: Bloky pro práci se Seznamy - druhá část

[7] Datový objekt je vytvořený protor pro uložení hodnot, dostupné pod unikátním indexem

[8]

[9] CSV - Comma separated values - soubor obsahující hodnoty, které jsou oddělené čárkami

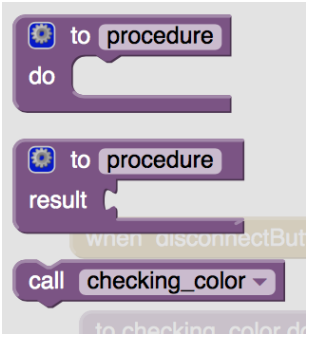
[10] RGBA - základní barevné schéma složené ze tří základnich barev. RedGreenBlue, A - značí alfa kanál (průhlednost od 0 do 1)

- Variables - obsahuje funkční bloky pro inicializaci (zavedení) globalních nebo locálních proměných do aplikace. Rozdíl mezi oběmi je nastíněn v sekci Globální a Lokální proměnné. Další bloky slouží pro nastavení nebo zjištění vytvořených proměných.



Obrázek 30: Bloky pro zavedení nových proměných

- Procedures - zahrnuje bloky pro vytvoření specifického funkčního bloku (procesu, který nemá MIT App inventor defaultně k dispozici. Lze vytvořit dva druhy. První proces provede zadanou funkci, ale nevrátí žádnou hodnotu. Druhý proces provede funkci a vrátí nastavenou hodnotu např. true nebo false. Dalším a zároveň posledním blokem je funkce pro zavolání vytvořeného procesu. Tím je vytvořená funkce zavedena do aplikace.



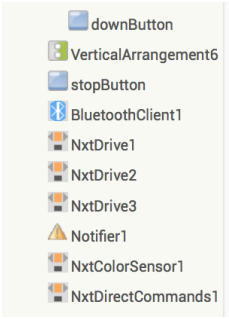
Obrázek 31: Bloky pro vytvoření procesu

6.3.3 Screen - komponenty používané v GUI aplikace

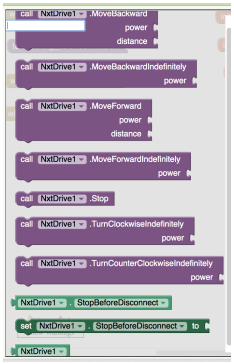
V této části se nacházejí všechny komponenty, které byly použity při návrhu GUI v části Designer. Zejména v této práci to jsou bloky z paletky Lego Mindstorms, kde jsou k nalezení všechny dostupné bloky pro práci s NXT. Dokonce i se stavebnicí EV3, to zde není nutné zmiňovat. K dispozici jsou zde bloky pro práci s dostupnými senzory, motory nebo blok pro přímé příkazy.

Další nezbytným prvkem je funkční blok BluetoothClient. Bez tohoto bloku je nemožné, aby se vytvořená aplikace připojila k NXT kostce. K funkčnímu připojení je nezbytný další funkční blok z paletky Uset Interface. Jedná se o blok ListPicker, který zobrazí seznam všech zpárovaných zařízení v dosahu a poté naváže spojení. Bez tohoto výběrového seznamu by nebyl znám bod připojení. Z toho důvodu by připojení neproběhlo.

Další zobrazené bloky jsou odvíjeny od návrhu GUI aplikace. Zcela jistě v této části budou i jiné funkční bloky, než výše zmíněné. Pro znázornění jsou k dispozici obrázky níže. Na první je náhled použitých bloků z GUI. Na tom druhém jsou funkční bloky pro práci s NXT drive.



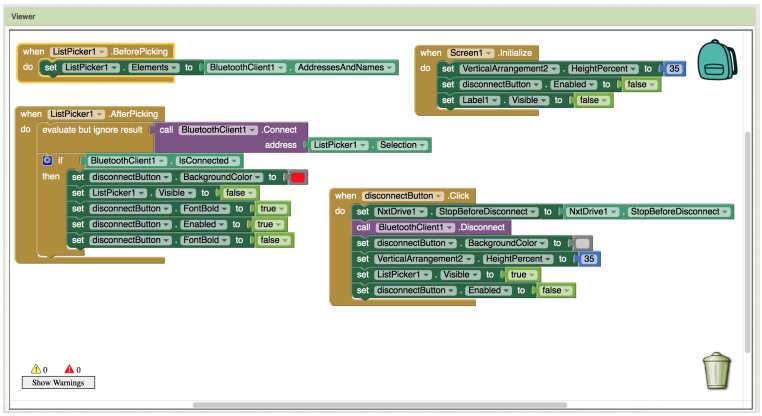
Obrázek 32: Nástroj pro bloky použité v GUI



Obrázek 33: Bloky pro práci s NXT Drive

6.3.4 Viewer - návrhová část GUI Blocks

Tato část GUI blocks je určena pro skládání funkcí jednotlivých komponent do akcí pomocí drag-and-drop technologie. Jednotlivé akce jsou složeny dle funkce vytvářené aplikace. Akce většinou začínají nabytím určitého stavu. Tomu může předcházet např. zavedení úvodní obrazovky nebo proměnné do programu, stisknutí příslušného tlačítka nebo získání hodnoty po splnění podmínky.



Obrázek 34: Pracovní plocha - blocks

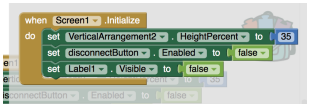
Návrhová část je dále vybavena indikátorem chyb a upozornění. Pokud je indikováno při vytváření jednotlivých akcí nějaké upozornění, tak výsledná aplikace bude zkompilována a následně spuštěna. O případném problému je informováno až v případě, že aplikace nefunguje správně. Je-li zaznamenána chyba u druhé indikace, tak nepůjde aplikace zkompilovat a následně spustit. Chyba popř. chyby musí být v první části odstraněny a vyřešeny.

Viewer v části block disponuje košem pro snadné odstranění jednotlivých funkčních bloků nebo celých akcí. Bloky stačí tahnutím přesunout na popelnici v pravém dolním rohu GUI Blocks.

Posledním prvkem, který Viewer obsahuje je Backpack (batoh). Slouží pro uložení vymazaných či často používaných bloků nebo celých akcí. Pokud batoh obsahuje některé bloky, tak se ikona změní viz. obrázky níže. Po rozkliknutí nabídne seznam všech obsahujících prvků.



Obrázek 35: Změna ikony - plný batoh

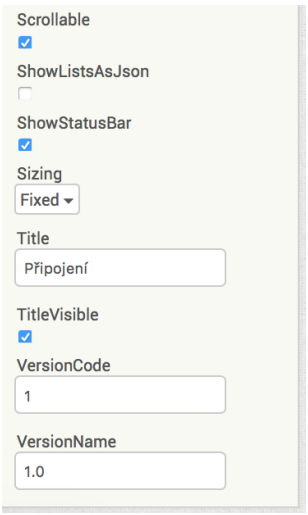


Obrázek 36: Seznam uložených akcí či bloků

6.4 Základ každé aplikace v MIT App Inventoru

Základem každé vytvářené aplikace pro práci se stavebnicí Lego Mindstorms je realizace správného připojení. Přípojení musí být vždy realizováno pomocí komponentů z ostatních paletek, kterými MIT App disponuje. Nejedná se tedy o určité bloky z paletky Lego Mindstorms. Jelikož je připojení důležité pro každý vývoj aplikace v tomto prostředí, tak je následně nastíněn postup jeho realizace.

Vývoj tedy začíná u prázdné pracovní plochy v GUI Designer, kde je potřeba navrhnout připojení s příslušnými komponenty. Popisek pracovní plochy je možno definovat v pravé části Properties (vlastnosti) viz. obrázek níže.



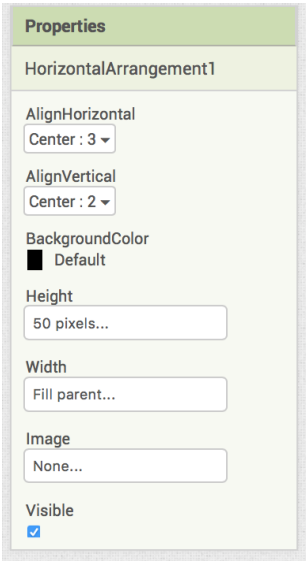
Obrázek 37: Nastavení popisku pracovní plochy

6.4.1 První krok - výběr komponentů

Prvním krokem při vývoji je závislý na výběru komponent. Následně je potřeba vybrat komponenty z levé části GUI. Výběr začíná v paletce Layout, kde je nutné přetáhnout do pracovní plochy komponentu HorizontalArranment. Ve vlastnostech této komponenty je nutné nastavit vlastnosti Width na hodnotu fill parent a Height na 50px viz. obrázek níže.

Další výběr komponent pokračuje v paletce User interface, kde je potřeba použít komponentu ListPicker, Button a Label. Tyto komponenty je nutno přetáhnout do již nastavené komponenty HorizontalArranments. Komponentu Button je povinností přejmenovat v části Components na disconnectButton, jelikož se bude jednat o odpojovací tlačítko. U komponenty Label1 nahradíme názvem ”pripojeno”. Komponentu ListPicker1 není nutné přejmenovávat.

Poslední komponentou připojení je BluetoothClient, který je k nalezení v paletce Conectivity. Tento komponent se po použití zařadí mezi Non-visible komponenty. BluetoothClient není třeba přejmenovávat. Následující kroky budou realizovány v části Properties, kde se nastaví hlavní vlastnosti.



Obrázek 38: Vlastnosti - HorizontalArranments

6.4.2 Druhý krok - Nastavení vlastností komponentů

ListPicker

Jedná se o připojovací tlačítko. Nastavení vlastností Width a Height je nutné nastavit na možnost Fill parent, BackgroundColor na Green. Checkbox u vlastnosti FontBold je možno zaškrtnout dle potřeby tučného textu. Nezbytným nastavením je změna zobrazeného textu v tlačítku. Tento text se přenastavuje pod vlastností Text, kde jej změníme na Připojit.

disconnectButton

Jedná se o odpojovací tlačítko. Nastavení vlastností Width a Height je nutné nastavit na možnost Fill parent. Defaultně zobrazovaný text, který je potřeba změnit nahodnotu Odpojit ve vlastnosti.

pripojeno

Jedná se o indikaci připojení. V případě úspěšného připojení bude text zobrazen na pracovní ploše. Je nutno nastavit vlastnosti popisku. Width na fill parent, fontsize na hodnotu 17.0, checkBox fontBold zaškrtnout, TextAlignment na hodnotu ”center”, Text - Připojeno a vlastnost Visible - odškrtnout (je potřeba, aby se text zobrazil až po úspešném připojení.

BluetoothClient

Není nutné jej nastavovat. Stačí, aby komponenta byla na pracovní ploše. Její použití na pracovní ploše lze vidět mezi Non-visible komponenty pod pracovní plochou.

Vybrané komponenty stačí pro zrealizování úspěšného připojení. Výběr komponentů pro připojení je kompletní a další vývoj základu aplikace pokračuje v GUI Blocks. Náledující kroky jsou tedy popsány v další části této práce.

6.4.3 Třetí krok - vytvoření akcí (funkčních bloků)

Vývoj aplikace se nyní přesouvá do části GUI blocks, kde je potřebné nastavit vybraným komponentům akci nebo-li funkci. K zprovoznění funkčního připojení je potřeba vytvořit 4 samostatné funkční bloky, kterým bude předcházet nabytí určitého stavu komponent.

whenScreen1.Initialize

Jako první je třeba vyvinout akci whenScreen1. Initialze (zavedení do programu). Tato akce je provedena ihned po připojení aplikace s telefonem. Akce by tedy měla obsahovat vše, co je potřebné zobrazit na úvodním obrazovce. V tomto případě je to dle UX pravidel aktivní a viditelné tlačítko Připojit (potřeba dominantního tlačítka, který vede uživatele k potřebné akci). Tento stav tlačítka je definován již díky nastavení vlastností v GUI Designer. Při vybírání komponentů a vytváření GUI aplikace byl použit i druhý akční prvek - tlačítko pro odpojení. Zde nastává situace, která musí uživatele přednostně vést k připojení. Jelikož žádné připojení neexistuje, nemůže být ani odpojeno, proto tlačítko Odpojit nemůže být ani aktivní. K tomuto případu slouží následně vytvořená akce.



Obrázek 39: Akce pro neaktivní tlačítko Odpojit

Akce je složena z bloku setdisconnectButton. Enabled, který je k dispozici v seznamu bloků disconnectButton v části Screen1 (nástroje vybraných komponentů v GUI Designer). Pro dosažení neaktivního tlačítka je potřeba tento blok nastavit logickou hodnotou false, která je k dispozici v seznamu nástrojů Logic v části Built-in.



Obrázek 40: Blok setdisconnectButton.Enabled



Obrázek 41: Logická hodnota false

whenListPicker1.BeforePicking

Akce sloužící k naplnění výběrového seznamu. Seznam je naplněn položkami získanými z komponenty bluetoothClient. V této fázi je nutné vybrat pro posunutí k dalšímu kroku vhodnou položku seznamu. Pokud žádná položka v seznamu neexistuje, není další krok vpřed. Neexistuje-li dostupné bluetooth zařízení, není ani seznam naplněn. V případě dostupných zařízení je seznam doplněn stejným počtem položek jako je dostupných zařízení v okolí.

Akce je složena ze seznamu, který je naplněn pomocí bloku setListPicker1.Elements, který je umístěn v seznamu bloků ListPicker1 v části Screen1. K tomuto bloku je připojen block, který definuje jednotlivé položky seznamu. V tomto případě je to block BluetoothClient1.AddressesAndNames, který naplní seznam



Obrázek 42: Akce pro naplnění seznamu

MAC adresou a názvem zařízení, který je v dosahu. Touto informací se naplní seznam zařízení, které jsou s NXT zpárované. Block je zařazen v seznamu nástrojů BluetoothClient1 v části Screen1. Nyní je vytvořen seznam zařízení, ke kterým se aplikace připojí po stisku tlačítka ”Připojit”. Nicméně ještě není ověřeno, zda se vůbec aplikace k NXT připojila. To řeší hned následně vytvořená akce.



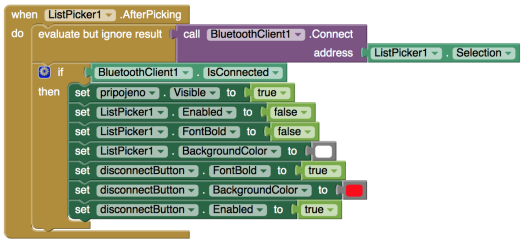
Obrázek 43: Blok setdisconnectButton.Enabled



Obrázek 44: Logická hodnota false

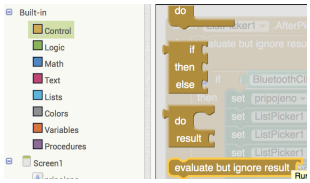
whenListPicker1.AfterPicking

Nyní je třeba definovat akci po vybrání, která byla naplněna pomocí předchozí akce *whenListPicker1.BeforePicking*. V této akci se nachází bloky pro navázání připojení. Celá akce dále nastavuje GUI aplikace tak, aby bylo patrné, že připojení proběhlo v pořádku a dle UX pravidel vedlo k možnému odpojení.

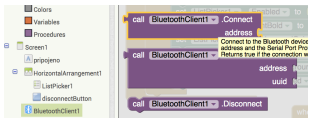


Obrázek 45: Akce pro naplnění seznamu

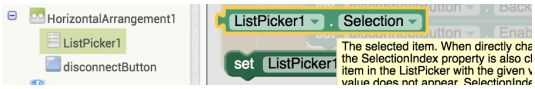
Akce se skládá z bloku *evaluate but ignore result*, který provede přiloženou akci, ale ignoruje výsledek. Block je umístěn seznamu Control. Dalším připojným blockem je *callBluetoothClient.Connect*, který je k nalezení v seznamu BluetoothClient1 v části Screen1. Tomuto bloku je nutné nastavit hodnotu, ke které adrese se realizuje připojení. Případnou MAC adresu pro přípojení zajistí block *ListPicker1.Selection*. Block je umístěn v seznamu ListPicker1 v části Screen1. Tento block vybere zajistí nastavení hodnoty dle vybrané položky seznamu, který byl naplněn položkami v předchozí akci.



Obrázek 46: evaluate but ignore result



Obrázek 47: Zavolání připojení Bluetooth



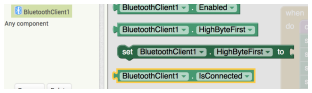
Obrázek 48: Výběr položky seznamu

Tímto je zrealizováno úspěšné přípojení aplikace k NXT. Nicméně akce tím nekončí, U akce je třeba ještě nastavení vhodné zobrazení indikace připojení a dostupnost tlačítek.

Akce dále obsahuje řídící struktury jako jsou podmíněné akce *IF-THEN*, které umožní, pokud je podmínka splněna, provádět další akce. Podmínka je umístěna v seznamu Control v části Built-in. Jelikož je potřeba ověřit správné připojení, tak musí být do podmínky umístěn block *BluetoothClient.IsConnected*, který vrátí hodnotu true nebo false. Block je umístěn v seznamu BluetoothClient1 v části Screen1. Pokud je podmínka splněna, budou provedeny následující kroky.



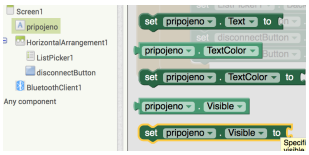
Obrázek 49: Podmínka IF



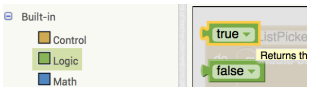
Obrázek 50: Metoda pro zjištění stavu připojení

Nastavovací kroky akce budou spadat do kategorie Setterů[11]. Jednotlivé vybrané komponenty je nutné nastavit do příslušného stavu. Pro indikaci připojení byl zvolen popisek, kde byli nastaveny vlastnosti Text na Připojeno a Visible na false. V tomto případě je na místě příslušný label nastaven na hodnotu true, aby indikovala úspešné připojení. Slouží k tomu block *setpripojeno.Visibleto*, který ve spojení s logickou hodnotou true, zobrazí indikační popisek. Logická hodnota je umístěna v seznamu Logic. Block setpripojeno.Visibleto je k dispozici v seznamu pripojeno v části Screen1.

[11] Setter - je v programování výraz pro metodu, která nastavuje proměnné příslušnou hodnotu

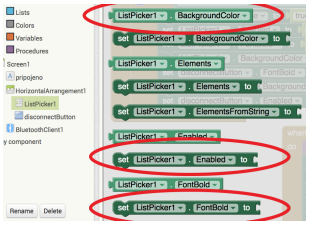


Obrázek 51: Nastavení indikace připojení



Obrázek 52: Logická hodnota true

Následující 3 kroky se týkají komponenty ListPicker1 nebo-li připojovací tlačítka. V tuto chvíli je aplikace připojená, a proto je na místě zneaktivnit tlačítko Připojit. Uživatel musí být veden, k možnosti odpojení jako k jediné volbě, jelikož připojení již proběhlo. Tohoto stavu přípojovací tlačítka je dosaženo díky blokům, které jsou umístěny v seznamu ListPicker1 v části Screen1. Jedná se o bloky *setListPicker1.Enabled a setListPicker1.FontBold*, které jsou pomocí logické hodnoty nastaveny na hodnotu false. Vzhled nefunkčního tlačítka dodá blok *setListPicker1.BackgroundColorTo*, který je záměrně nastaven na bílou barvu pomocí blocku White ze seznamu Color. Všechny použité settery dodají neaktivní vzhled tlačítka, který je potřebný.

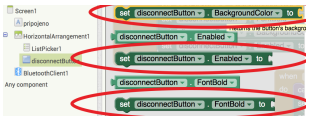


Obrázek 53: Práce s ListPicker bloky

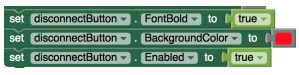


Obrázek 54: Nastavení na hodnotu False

Poslední částí akce je zviditelnění odpojovacího tlačítka, které je zde prioritní. DisconnectButton je od nastavování vlastností komponenty a inizializace programu nastaven na neaktivní. Důvod tohoto nastavení je nastíněn výše. K požadovanému nastavení jsou použity settery ze seznamu disconnectButton v části Screen1. Bloky *setdisconnectButton.Enabled a setdisconnectButton.FontBold* nastaveny na logickou hodnotu true umožní zaktivnění tlačítka a ztučněný textu (pro lepší rozpoznání textu v tlačítku). Barva tlačítka je nastavena blokem *setdisconnectButton.BackgroundColor* a červenou barvou ze seznamu Color v části Buil-in.



Obrázek 55: Práce s ListPicker bloky

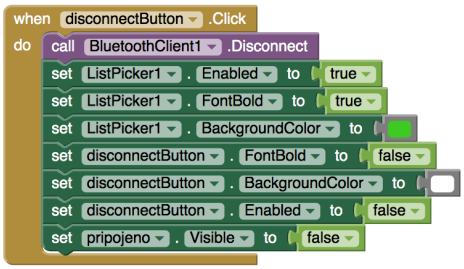


Obrázek 56: Nastavení na hodnotu False

whendisconnectButton.Click

V poslední části realizace připojení je třeba nastavit akci pro chování celé aplikace po stisku odpojovacího tlačítka - tedy odpojení aplikace od NXT. Celé odpojení je provedeno jedním blockem, ostatní

nastavení se týkají ostatních komponentů, které jsou použity na pracvoní ploše. GUI aplikace musí být po odpojení stejné jako úvodní obrazovka aplikace, která je zobrazena hned po zavedení programu.

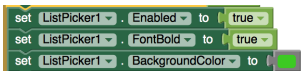


Obrázek 57: Chování aplikace po odpojení

Odpojení aplikace zajistí blok první blok akce, který zavoláním potřebné funkce ukončí připojení. Jedná se o blok *callBluetoothClient1.Disconnect*, který je umístěn v seznamu BluetoothClient1 v části Screen1. Nasledně jsou zařazeny nastavovací bloky (settery), které zaktivní tlačítko Připojit, jelikož se aplikace dostává do stavu, kdy je možné opětovného připojení. Zaktivnění tlačítka zajistí bloky *setListPicker1.Enabled a setListPicker1.FontBold* ze seznamu ListPicker1, které jsou nastavené na hodnotu true ze seznamu Logic. Defaultní barva tlačítka je nastavena blokem *setListPicker.BackgroundColor*, ke které připojen block Green se seznamu Color.



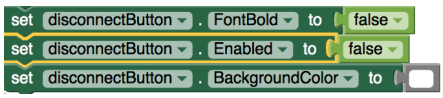
Obrázek 58: Zavolání metody pro odpojení



Obrázek 59: Nastavení na hodnotu True

Pro nastavení neaktivního tlačítka Odpojit jsou použity blocky (settery) ze seznamu disconnectButton. Blocky *setdisconnectButton.Enabled a setdisconnectButton.FontBold* nastaveny na logickou hodnotu false, zneaktivní tlačítko. Nastavení barvy na bílou pomocí složení blocků White ze seznamu Color *setdisconnectButton.BackgroundColorto* je dosaženo věrohodného vzhledu.

V poslední řade je skryt indikátor připojení. Indikace je skryta spojením bloku *setpripojeno.Visibleto*, který je nalezen v seznamu ”pripojeno”a logické hodnoty false.



Obrázek 60: Zneaktivnění tlačítka Odpojit



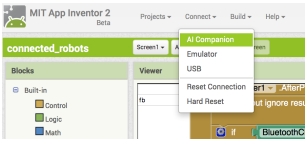
Obrázek 61: Skrytí indikace

V tuto chvíli je vývoj Připojení, který je nezbytnou součástí vývoje každé aplikace v MIT App u konce. V tuto chvíli je připojení potřeba řádně otestovat. Testování proběhne v další části.

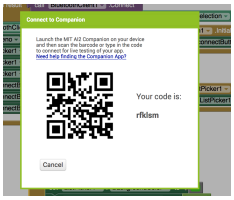
6.4.4 Třetí krok - testování

Součástí každého vývoje je otestování vytvořené aplikace byť jen části jako v tomto případě. Testování probíha stejně jakou dle možností připojení aplikace MIT App Inventoru.

V první řadě je nutnost zrealizování zpárování telefonu a NXT. Pokud jsou zařízení zpárovaná, tak je třeva propojit vytvořenou aplikaci v MIT app Inventoru s telefonem, který byl s NXT zpárován. Pro připojení slouží funkce Al Companion v záložce Connect horního menu. Po rozklinutí vyskočí v GUI Pop-up menu, kde je pro přípojení vygenerován QR nebo textový kód. Po přečtení nebo zadání těchto kódů v telefonní aplikaci MIT Ai2 Companion se vzájemně propojí. Propojením je vyvíjená aplikace přenesena do MIT Ai2 Companion až do jejího ukončení.

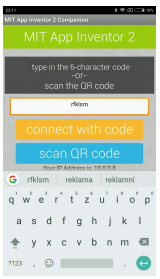


Obrázek 62: Navázání spojení Al Companion



Obrázek 63: Zobrazení připojovacích kódů

Dle přiložených obrázků je možno vidět, že se připojení chová přesně tak, jak bylo navrženo. Úvodní obrazovka aplikace k jedné možné volbě tj. k navázání připojení. Po rozkliknutí připojení se zobrazí naplněný seznam zpárovaných zařízení v okolí. Pro další připojení je povinností vybrání příslušné položky. V tomto případě je jediná. Po vybrání položky je aplikace připojena k NXT. Úspěšné připojení značí i indáktor na změněné ploše, která disponuje jedinou možností tj. odpojení aplikace od NXT. V případě odpojení je plocha změněna do stavu, ve kterém byla při připojení telefonu s MIT App Inventor.



Obrázek 64: Zadání kódů v MIT Ai2

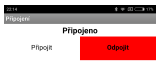


Obrázek 65: Zavedení aplikace po připojení

Testování proběhlo zcela v pořádku. Je tedy možné požadovat připojení za plně funkční.



Obrázek 66: Seznam pro výběr připojení



Obrázek 67: Aplikace po navázání připojení

7 Pocket Code - Learn programming

Pocket Code je vývojové prostředí pro platformu Android od vývojářů z firmy Catrobat. Catrobat vyvinul balík Pocket Code z předností předchozího vývojového prostředí Catroidu a desktopového prostředí Scratch.

Pocket Code funguje jako mobilní aplikace pro vytváření aplikací přímo z vašeho smartphonu. Tato aplikace je vyvinuta za účelem vývoje aplikací všeho druhu. Vytváření aplikací pro Lego Mindstorms je zde spíše okrajové. Aplikace dokáže ovládat ještě jiné různé aplikace, a proto jsou funkce pro ovládání komponentů stavebnice nekompletní.

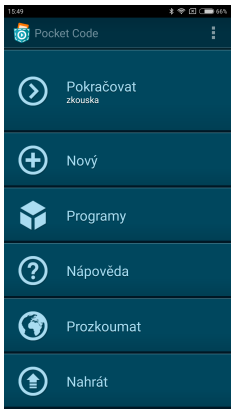
7.1 GUI aplikace

Celé vývojové prostředí je složeno z několika částí. Po natčení je vyobrazena první část, která je zaměřena pro nastavení a základní opercae s prostředím. Druhou částí GUI jsou části sloužící při vývoji aplikace.

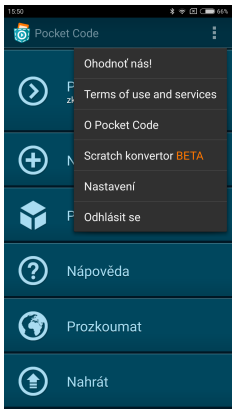
7.1.1 Menu & Možnosti prostředí

Základní GUI aplikace je složeno ze základního Menu aplikace, které umožňuje návrat k poslední vytvářené aplikaci, vytvořit nový program, zobrazit všechny programy uživatele, proklik k nápovědě s vývojem aplikací. Položka Prozkoumat a Nahrát slouží k prozkoumání příslušné galerie vytvořených aplikací od ostatních uživatelů nebo k nahrání vytvořené aplikace.

Záložka option slouží především k rozšířenému nastavení. Pocket Code umí vytvářet především pro smartphony a jejím rozšířením je právě možnost vytvářet aplikace pro různé stavebnice. Nejznámějšími stavebnicemi jsou Rapsbery PI, Arduino, EV 3 a NXT 2.0.



Obrázek 68: Menu - Pocket Code



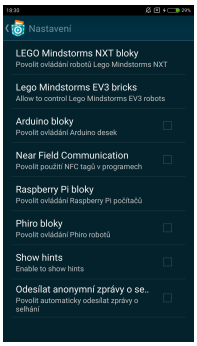
Obrázek 69: Pocket Code - Možnosti

7.1.2 Potřebná nastavení pro vývoj programu na NXT

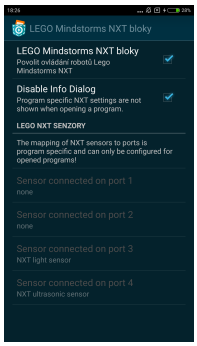
Díky všestranosti prostředí je potřeba jej správně nastavit. Nastavení se provede v záložce Lego Mindstorms bloky. Povolení bloků se provede pomocí zaškrtnutí první položky. V tomto nastavení je vidět obsazenost a zařazenost jednotlivých portů, které jsou určeny pro senzory robota.

Po tomto nastavení se objeví ve Scriptech objektů seznam obsahující scripty pro práci s NXT. Scripty umožňují ovládat pouze Servo motory NXT. Potřebné hodnoty z připojených senzorů se nacházejí v jiné části skládání scriptů.

Dalším potřebným nastavením je aktivní bluetooth připojení. Vhodným způsobem je tedy bluetooth připojení při vývoji aplikace nevypínat. Vhodné je smartphone před vývojem spárovat s robotem, tímto se předejde neúspěšnému připojení.



Obrázek 70: Nastavení - Bloky NXT



Obrázek 71: Zaškrtnutí nastavení

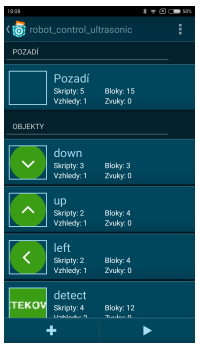
7.1.3 Přehled objektů programu & detail objektů

Celý vývoj aplikace se v tomto prostředí točí okolo grafický objektů, kterým jsou přiřazovány nejrůznější akce. Tyto objekty jsou součástí práce a ke stažení jsou na bp.valislukas.cz

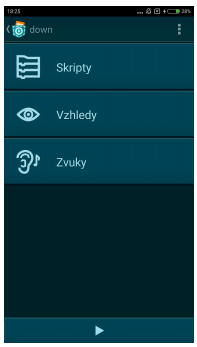
Dalším způsob, jak se dostat ke grafickým objektům pro aplikace je použití aplikace Pocket Paint. Již z názvu je patrné, že se jedná o rozšiřující aplikaci právě pro kreslení grafických objektů.

Vzhledem k tomu, že Pocket Code umí pracovat jen s grafickými objekty a texty vykreslují pomocí pixelů, tak je na místě vytvořit i pozadí aplikace s popisky, ke kterým jsou hodnoty proměnné napozicovány.

Ke grafickým objektům se ve vývoji přiřazují jejich funkce pomocí již zmíněných scriptů. Mimo to je možnost ke každému z objektů přiřadit další vzhledy nebo zvuky.



Obrázek 72: Přehled objektů při vývoji



Obrázek 73: Práce s objetky

7.1.4 Rozdělení scriptů a jejich skládání

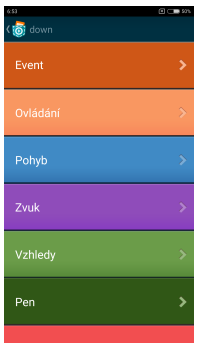
Jednotlivé scripty jsou umístěny v několika přehledných a barevně oddělených seznamech. Seznamy jsou pojmenovány dle toho, jaké scripty jsou v seznamu obsaženy.

Každý vytvořený script musí začít jedním z bloků, který obsahuje seznam Event - událostí. Poté se na tento blok skládají jiné akce. Je důležité, aby tyto akce byly skladané dle zvoleného pořadí.

Jednotlivé scripty lze libovolně duplikovat. Tato informace je důležitá především pro práci s událostmi v seznamu Event.



Obrázek 74: Přehled objektů při vývoji

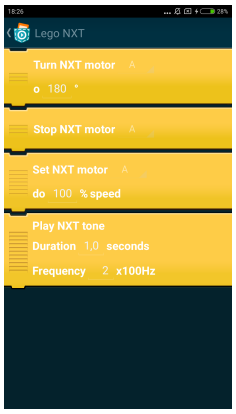


Obrázek 75: Práce s objetky

7.1.5 Nejčastěji používané akce jednotlivých seznamů

Díky všestranosti prostředí seznamy obsahují velkou škálu dostupných akcí. Při vývoji aplikací pro NXT byla použita jen menšina těchto akcí z každého seznamu. Seznam je vytvořen na základě použitých bloků v již vytvořených programech, které jsou součástí této práce.

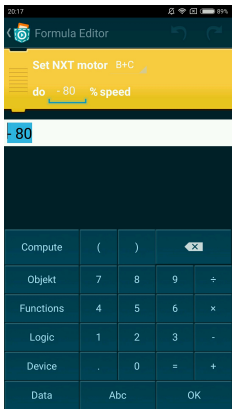
- Event - When programs starts, Na klepnutí, Při dotyku obrazovky, Když 1¡2 bude
- Ovládání - Čekat, Navždy, Pokud..., Pokud... Jinak..., Wait until, Opakovat, Opakovat dokud
- Pohyb - Umístěte na, Nastav x, Nastav x
- Zvuk - Spuštění zbvuku
- Vzhledy - Změnit vzhled, Předchozí vzhled, Nastavit velikost, Skrýt, Zobrazit, Říct, Nastavit průhlednost, Nastavit pozadí
- Pen - Seznam Pen nebyl nikterak využit
- Data - Nastavit proměnnou, Změnit proměnnou, Show variable, Přidat na seznam
- Lego NXT - Turn NXT motor, Stop NXT motor, Set NXT motor, Play NXT tone



Obrázek 76: Akce pro NXT

7.1.6 Nastavení jednotlivých akcí

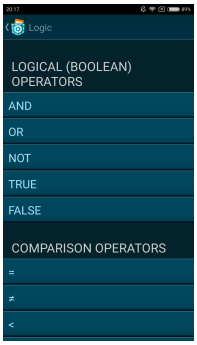
Jednotlivé použité akce při vytváření je potřeba dodatečně nastavit na správné jednotky nebo použít jednu z metod pro práci s datovými objekty či práci s hodnotami připojených senzorů. K použití jsou tu mimo jiné i logické hodnoty, porovnávací znaménka, logické operátory nebo matematické funkce a metody pro práci s listem nebo text. řetezci.



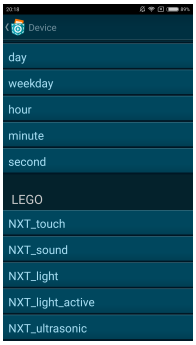
Obrázek 77: Nastavení hodnoty akce

Nejčastěji používané nastavení akcí je dosaženo pomocí číselné hodnoty, ale pro práci se senzory je možno používat metody nebo porty z části Device. Celý vývoj aplikace zahrnuje kombinaci prvků již zmíněných částí, ale také Funkcí z části Functions a dat z části Data. Data obsahují všechny vytvořené proměnné. Část Functions obsahuje matematické funkce, funkce pro práci s listem a textový řetezci. V části Logic jsou k dizpozici logické a porovnávací opearátory. Část Device obsahuje metody pro získání hodnot z robota nebo smartphonu. Nejčastěji používané prvky jsou shrnuty v následujícím seznamu.

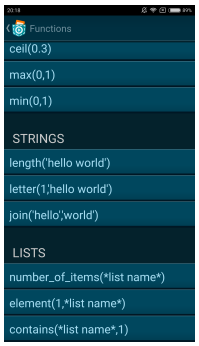
- Compute - Při vývoji není využíván
- Objekt - Při vývoji není využíván
- Functions - Strings(length(), letter(), join()), Lists(number_of_items(), element(), contains())
- Logic - všechny logické a porovnávací operátory
- Data - Zde jsou obsaženy vytvořené promněnné a listy potřebné pro vývoj aplikace



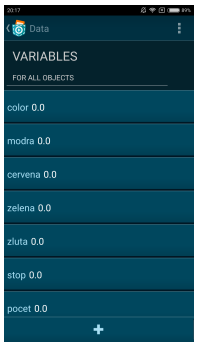
Obrázek 78: Prvky - Operace



Obrázek 79: Prvky - Device



Obrázek 80: Prvky - Funkce



Obrázek 81: Prvky - Variables

7.1.7 Spuštění vyvinuté aplikace

Spuštění aplikace v Pocket Code je provedeno tlačítkem v dolní části prostředí, které se nachází v přehledu objektů dané aplikace. Toto tlačítko je k dispozici i ve skládání scriptů jednotlivých objektů.

Při stisku tlačítka se objeví seznam dostupných zařízení v dosahu bluetooth. Z těchto zařízení je potřeba vybrat spravné NXT. Pokud je NXT zapnuté, proběhne automatické připojení a zavedení programu. Pokud NXT je vypnuté připojení neproběhne.

8 Porovnání aplikací

Napříč celou prací jsou obě vybraná vývojová prostředí porovnávána v několika aspektech. Obě prostředí jsou odlišná ve vývoji aplikací i ve využití prvků smartphonu nebo robota.

Vývojová prostředí jsou porovnávána především ve vývoji pro NXT, proto se zde nenalézá porovnání nevyužitých komponent či vlastností.

8.1 Využití NXT senzorů

Porovnání se zaměřuje na to, zda je vývojové prostředí schopno pracovat a využívat hodnoty získané ze senzorů robota.

Z tabulky je patrné, že prostředí Pocket Code neumí zpracovávat data z Color senzoru. Aplikace vytvořené v MIT App Inventoru, kde je využito dat z Color senzoru, nejsou zcela přesné. Data získaná z tohoto čidla jsou zavislá na intenzitě a teplotě denního světla, proto se mohou nasekonvaná data vždy lišit a rozpoznání barev je nepřesné.

Není tedy 100 % pro bezchybný vývoj aplikace pro detekování barev nebo jiného využití Color senzoru.

Název prostředí	Color	Touch	Sound	Light	Ultrasonic
Pocket Code	✗	✓	✓	✓	✓
MIT App Inventor	✓	✓	✓	✓	✓

Tabulka 7: *Porovnání z hlediska využití senzorů*

8.2 Využití komponent smartphonu - světlo, kamera

8.2.1 Pocket Code

Z tabulky je patrné, že toto vývojové prostředí umí využít obě komponenty smartphonu.

Použití kamery ve vývoji aplikace je zde specifické. Prostředí umí pracovat jak s přední, tak zadní kamerou, ale přes tyto kamery nelze pořídit fotografie. Využití těchto komponent v Pocket Code spočívá především v detekci lidského obličeje apod. Detekce obličeje funguje pouze při využití přední kamery smartphonu.

8.2.2 MIT App Inventor

Z tabulky je patrné, že toto vývojové prostředí umí využít pouze kameru.

Tato funkce je navíc značně omezena. Zavoláním funkce se otevře aplikace fotoaparát, která se poté chová jako defaultní fotoaparát smartphonu. V kombinaci s komponentou Sharing umí prostředí uložit ručně vytvořené foto do paměti telefonu.

Název prostředí	Světlo	Camera
Pocket Code	✓	✓
MIT App Inventor	✗	✓

Tabulka 8: *Porovnání dle využití prvků smartphonu*

8.3 Práce s vytvořenými aplikacemi

Porovnání, které je zaměřeno na sdílení vytvořených aplikací. Z tabulky je patrné, jak je možné dané aplikace z vývojových prostředí šířit nebo uchovávat v daných souborech.

8.3.1 MIT App Inventor

Tabulka vyobrazuje, že toto vývojové prostředí umožňuje aplikace nahrát do příslušné galerie MIT App Inventor komunity. Tato galerie slouží pro jednoduché publikování aplikace.

Export/import slouží ke snadnému přesunu vytvořených aplikací mezi účty. Možnost ukládání do .apk slouží ke spuštění aplikace bez potřebné aplikace pro simulaci na smartphonu.

8.3.2 Pocket Code

Tabulka vyobrazuje, že toto vývojové prostředí disponuje pouze možností publikování vytvořené aplikace do galerie komunity. Z této komunity je možnost stáhnout příslušné aplikace do Pocket Code. Tyto aplikace jsou po stažení k dispozici v záložce Programy.

Název prostředí	Uložení	Import/export	Publikování
Pocket Code	✗	✗	✓
MIT App Inventor	✓	✓	✓

Tabulka 9: *Porovnání práce vytvořených aplikací*

8.4 Rozdíly vývoje aplikací

Vývoj aplikací v obou prostředích je v několika aspektech stejný. Základem je výběr komponentům, kterým se poté přiřazují scripty nebo-li akce.

Rozdíl nastává v použití komponentů NXT, kde duplikace některých ze scriptů značně zjednodušuje vývoj.

MIT App Inventor tuto možnost u bloků začinajících na when nemá. Jsou to již předem připravené metody, které zajištují vykonání akcí v určitých stavech. Může to být detekce určité barvy, úrovně zvuku, stisk tlačítka nebo naopak jeho puštění.

Jak je zmíněno výše tyto bloky nelze duplikovat a prostředí po duplikaci zobrazuje Error upozornění. Nemožnost duplikace v mnohých případech zamezila i vývoj celé aplikace.

Pocket Code předem připravenými metodami nedisponuje. Proto je vývoj jasně daný a jednodušší.

Výhodou jsou nápovědy jednotlivých prostředí, které v nejasných případech osvětlí danou problematiku. Těmito nápovědami disponují obě prostředí. Nápověda se vždy nachází v menu prostředí.

Nastavení vybraných komponentů je dalším rozdílem vývoje mezi prostředími. V MIT App Inventoru vybrané komponenty lze vhodně nastavit. Nastavujícími parametry jsou např. zobrazený text, velikost textu, barva nebo visibilita [12].

Název prostředí	Duplikace bloků	Nápověda	Bloková nápověda
Pocket Code	✓	✓	✗
MIT App Inventor	✗	✓	✓

Tabulka 10: *Porovnání práce vytvořených aplikací*

8.5 Rozdíly mezi škálami senzorů NXT

Mezi vývojovými prostředími je rozdíl právě ve stupnicích senzorů. Ná základě této stupnice je po detekování přiřazena příslušná hodnota pomocí použití určitého scriptu nebo bloku.

[12] Visibilita je parametr, který nastavuje viditelnost komponenty

8.5.1 MIT App Inventor

V MIT App inventoru pracují NXT senzory s intervalem 0 - 1023. Pokud úroveň nemůže být změřena vrací metoda hodnotu -1. Prostředí disponuje bloky pro práci s touto stupnicí. Pro tyto bloky je stupnice rozdělena na třetiny. Poté bloky vykonávají svou funkci dle toho, do jakého intervalu zaznamenaná úroveň patří.

8.5.2 Pocket Code

Pocket Code disponuje stejnými stupnicemi jako originální prostředí Lego Mindstorms NXT tj. procentuální stupnice. Naměřená úroveň je vyjádřená procenty dle intenzity. Pomocí podmínkových scriptů a ručního nastavení na třetiny lze i v tomto vývojovém prostředí vyhodnocovat úroveň dle naměřené intenzity. Vývoj je v tomto případě obdobný vývoji v MIT App Inventoru, ale předchází tomu složitější příprava.

8.6 Nutné opatření ve vývoji aplikací

Při vývoji aplikace je potřeba obě aplikace řádně nastavit nebo přizpůsobit či opatřit aplikace pro hladký vývoj. Toto přizpůsobení se ve vývojových prostředí značně liší.

8.6.1 Pocket Code

Jediným potřebným nastavení v Pocket Code je spuštění práce s NXT bloky. Toto nastavení se nachází Možnostech vývojového prostředí. Po zapnutí bloků je do sciptů pro vývoj přidán seznamu scriptů pro ovládání NXT.

Na rozdíl od MIT App Inventoru je zde bluetooth zrealizováno bluetooth připojení automaticky. Toto připojení se automaticky provede po tom, co je v aplikaci použit jakýkoli prvek, který ovládá nebo přijímá data z NXT.

Zjednodušení vývoje může nastat po stažení vytvořených grafických objektů. Tyto objekty jsou k dispozici na webové adrese bp.valislukas.cz

8.6.2 MIT App Inventor

Na rozdíl od Pocket Code není nutno zde zapínat zobrazení bloků a komponent pro NXT. Vývojové prostředí je těmito komponenty vybaveno od prvního spuštení.

Vývoj v MIT App Inventoru je rozdílný v připojení bluetooth. Toto připojení je nejdříve potřeba vytvořit pomocí vhodných komponent a přiřazených scriptů. Postup vytvoření je popsán v části práce, kde je popsáno prostředí MIT App inventor.

Další nezbytnou částí vývoje je stažení aplikace MIT Ai2 Companion. Tato aplikace slouží k simulaci vytvořené aplikace do smartphonu. Tato aplikace slouží ke snadnému spuštění bez použití emulátoru nebo ukládání rozdělaných aplikací.

MIT Ai2 Companion je možno stáhnout přes odkaz, který je dostupný na webové adrese bp.valislukas.cz

Název prostředí	Realizace připojení	Povolení práce s NXT
Pocket Code	✗	✓
MIT App Inventor	✓	✗

Tabulka 11: *Porovnání nutných bodů při vývoji*

8.7 Využití logických, matematických funkcí

Obě prostředí jsou porovnána v oblasti využití matematických a logických funkcí. Porovnání zarhnuje i využití porovnávacích a logických operacích.

MIT App Inventor i Pocket Code těmito funkcemi disponují. Při vývoji aplikace jsou tyto funkce nezbytné. Především základní matematické operace, porovnávací a logické operace. Matematických funkcí při vývoji aplikacích v této práci nebylo využito.

Název prostředí	Matematické funkce	Logické funkce	Porovnávací operace
Pocket Code	✓	✓	✓
MIT App Inventor	✓	✓	✓

Tabulka 12: *Porovnání dle využití souborů*

8.8 Využití souborů NXT nebo smartphonu

Porovnání zabývající se prací se soubory umistěné v NXT kostce nebo příslušném smartphonu. Po provedení zkoumání jsou získané informace zaneseny do porovnávací tabulky.

8.8.1 Pocket Code

Ná základě získaných informací je dosaženo jasných výsledků. Pocket Code - mobilní vývojové prostředí má přístup a tedy umí využívat soubory, které jsou uloženy v paměti telefonu. Otestováno na zvukových a obrázkových souborech.

Naopak vzhledem k faktu, že se jedná o mobílní prostředí, tak zde není přístup k souborům, které jsou uloženy v NXT řídící kostce robota. Tyto soubory se mohou lišit dle stažených souborů přes desktopové prostředí Lego Mindstorms NXT.

8.8.2 MIT Appp Inventor

Opačným přístupem k souborům, než Pocket Code disponuje MIT App Inventor. Prostředí je vybaveno přístupem k souborům, které jsou uloženy v NXT řídící kostce. Tyto soubory jsou ovlivněny předchozím stažení skrze prostředí Lego Mindstorms NXT.

MIT App Inventor je pouze desktopové prostředí a vytvořený program je do smartphonu pouze simulován skrze příslušnou aplikaci. Tudíž prostředí nemá přístup k souborům, které jsou uloženy v paměti

telefonu.

Název prostředí	Přístup k souborům NXT	Přístup k souborům smartphonu
Pocket Code	✗	✓
MIT App Inventor	✓	✗

Tabulka 13: *Porovnání dle využití souborů*

8.9 Práce s txt. řetezci

Porovnání je zaměřeno především na zobrazování textových řetezců na ploše programu. V obou prostředích je základní funkcí vytvoření a uložení hodnoty do jakékoli proměnné. Základní rozdíl prostředí je ve vykreslení těchto proměných.

8.9.1 Pocket Code

Pocket Code narozdíl od MIT App Inventoru umí vyvíjet programy pouze s grafickými objekty a při vykreslení to není jinak. Pokud je požadováno vykrelení jakékoli proměnné, tak je tato proměnná vykreslená nečitelně pomocí několika pixelů.

Nejčitelněji vykreslené proměné jsou číselného druhu. Kdyby bylo zobrazení použito na textový řetězec byl by celý text nečitelný shluk pixelů.

Při vývoji aplikací je tedy vykreslení proměných použito jen u číselných hodnot, které jsou příslušně napozicovány. Případné popisky nebo potřebné text jsou zahrnuty v grafických objektech - pozadích.

8.9.2 MIT App Inventor

V MIT App Inventoru je práce s proměnými jednodušší. K zobrazení číselných i textových hodnot je zde využito textových komponentů. Při vývoji nejčasteji komponenta Label.

Tento label je vykreslen bez jakékoli deformace jako u Pocket Code. Je zde využito textového nástroje. Pokud je Label nastaven pomocí bloku do příslušné hodnoty, tak dojde k jejímu vykreslení na místě, kde je label umístěn.

Název prostředí	Práce s proměnými	Kvalitní vykreslení
Pocket Code	✓	✗
MIT App Inventor	✓	✓

Tabulka 14: *Porovnání dle využití souborů*

9 Praktická část

Vývoji aplikací předchází výběr správné konstrukce a sestavení robota. Konstrukce by měla být ze základní sady, jelikož potřebné návody na jejich sestavení jsou součástí stavebnice. Tyto návody jsou v tištěné nebo elektronické podobě na přiloženém CD.

Praktická část je také zaměřena na vytvořené aplikace v obou prostředích. Vývoj těchto aplikací je nastíněn ve vytvořených metodických návodech, jak danou aplikace vyvinout. Návod vždy obsahuje dvě části, ve kterých je rozdělen vývoj dle prostředí. U každé části je udána i časová náročnost vývoje a zjištěná uskalí, která mohou při vývoji nastat.

Součásti návodu je seznam použitých grafických objektů, které jsou ke stažení přes odkaz na webu bp.valislukas.cz. Pokud nejsou grafické objekty použity, tak je vytvořen seznam použitých komponent. Seznam použitých komponent se pojí s vývojem v MIT App Inventoru. Veškerým komponentů z kategorie NXT je potřeba nastavit v BluetoothClienta. Tento CLient se nastavuje v nastavení jednotlivých komponentů v pravé části GUI Designer prostředí MIT App Inventor. V Pocket Code není zapotřebí komponenty nastavovat.

Na přiloženém CD jsou k nalezení vytvořená videa, které vyobrazují funkčnost vytvořených aplikací. Tyto videa jsou pořízena na sezení se žáky základní školy. Na tomto sezení jsou žákům předány získané informace o vývoji aplikací přes smartphony. Vývoj jednoduchých aplikací si žáci samostatně osvojili. K jasnému zadání pro učitele či mentory kurzu předchází nastudování vytvořených návodů. Návody mohou být rovnou předány žákům jako podklad vývoje aplikací.

9.1 Konstrukce robota

Pro vývoj aplikací byla vybrána konstrukce pásového vozidla, který může disponovat střílecím zařízením. Návod je přiložen v originálním prostředí Lego Mindstorms, které je dostupné z přiloženého CD. Konstrukce byla vybrána na základě jednoduchého ovládání. Pásová konstrukce podvozku umožní pohyb všemi směry bez použití dalšího motoru pro natáčení nápravy.

Do konstrukce je zakomponované i střílecí zařízení. Návod na sestavení je také součástí návodu na sestavení. Toto střílecí zařízení je využito ve vývoji několika aplikací v obou prostředí. Konstrukce pamatuje na časté nabíjení baterií. Konstrukce je přizpůsobena k snadnému přístupu k bateriím.

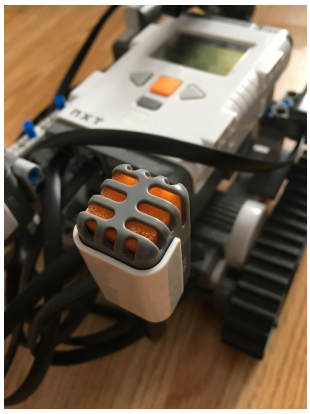
Pomocí Lego dílů stavebnice musí být všechny senzory jednotlivě umístěny na kostrukci robota. Toto umístění je provedeno individuálně, dle každého jednotlivce. Sound senzor by měl být v dostatečné vzdálenosti od robota tak, aby nebyla detekce ovlivněna běžným zvukem robota. RGB senzor by měl být namířen směrem k podlaze, kde budou umístěny jednotlivé barvy. Vzdálenost RGB a podlady by měla být cca 1 cm. Ultrasonického senzor by mělo být namířeno vždy buď dopředu nebo dopazdu, dle jeho využití. Vždy by mělo mít okolo sebe dostatečně velký prostor, aby nedetekoval nejprve konstrukci robota, která by ovlivňovala detekci.



Obrázek 82: Konstrukce robota



Obrázek 83: Střílecí zařízení



Obrázek 84: Umístění Sound senzoru



Obrázek 85: Umístění Touch senzoru



Obrázek 86: Umístění RGB senzoru



Obrázek 87: Umístění Ultrasonického senzoru

9.2 Robot Controler

V této aplikaci je vyvíjen program pro ovládání robora jako R/C auto. Vývoj spočívá ve vytvoření tlačítek, která ovládají pohyb robota 4 základními směry. Aplikace ja zařazena do kategorie lehkých úloh. Vývoj je obdobný v obou prostředí. Vždy se jedná o prvky, které reaguji na stlačení nebo kliknutí dle toho, jakými bloky prostředí disponuje.

Každému vývoji v tomto prostředí předchází potřebné zrealizování funkčního připojení. Toto připojení je popsáno v teoretické části v sekci MIT App Inventor.

9.2.1 Návod pro MIT App Inventor

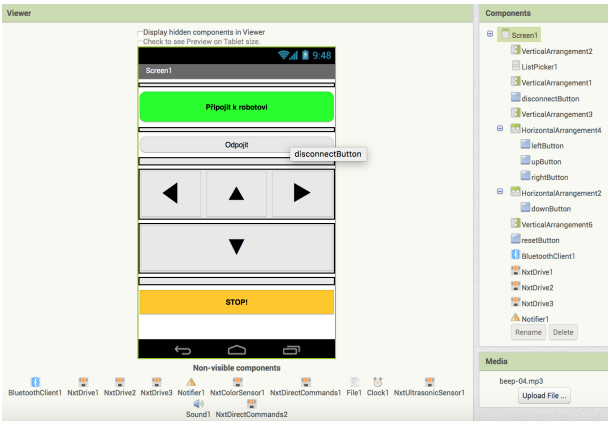
Jako první při vývoje je potřeba z příslušné paletky vybrat všechny komponenty. V seznamu jsou i komponenty pro uspořádání celé ovládácí plochy.

- VerticalArangments - 2x - tato komponenta v aplikaci slouží pro vertikální mezery mezi jednotlivými komponenty - vhodné nastavení je px nebo %
- HorizontalArangments - 2x - zde slouží pro použití více komponent vedle sebe. V tomto případě se jedná o vhodné rozložení tlačítek - vhodné nastavení je px nebo %

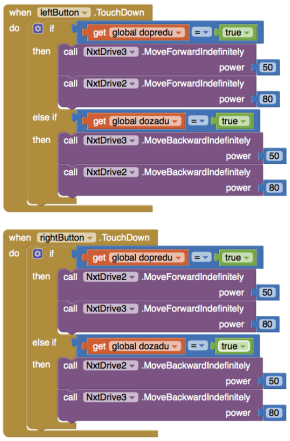
  Pokud je třeba využít plnou šíři obrazovky, je třeba tyto komponenty nastavit na hodnotu - fill parent
- Buttons - 4x - prvky pro ovládání robota. Prvky jsou pojmenovány dle směru, který mají udávat. Stejnou funkci mají i ASCII znaky vloženy jako text do příslušných tlačítek. Levé, horní a pravé tlačítka jsou vložena do HorizontalArangments a jsou nastavena po 33 % šířky.

  Dolní tlačítko je vybaveno taktéž vhodným ASCII znakem určující směr. Tlačítko je vloženo do dalšího HorizontalArrangments s nastavením šířky na hodnotu - fill parent.
- stopButton - slouží pro nouzové zastavení všech běžicích motorů. V aplikaci je vhodné pro nouzové vypnutí v případě, že nastane nečekaný problém.
- nxtDrive - 3x - Komponenta ovládající motory NXT. Komponenty je nutné nastavit na správně připojené porty. Pro přímý pohyb se nastavují vždy 2 porty - nejčastěji C,B. Pro zatáčení se motory vždy nastavují na jednotlivé porty B nebo C, dle zapojení do řídící kostky. NXT komponentům je vždy nuntné nastavit dostupné bluetooth připojení.

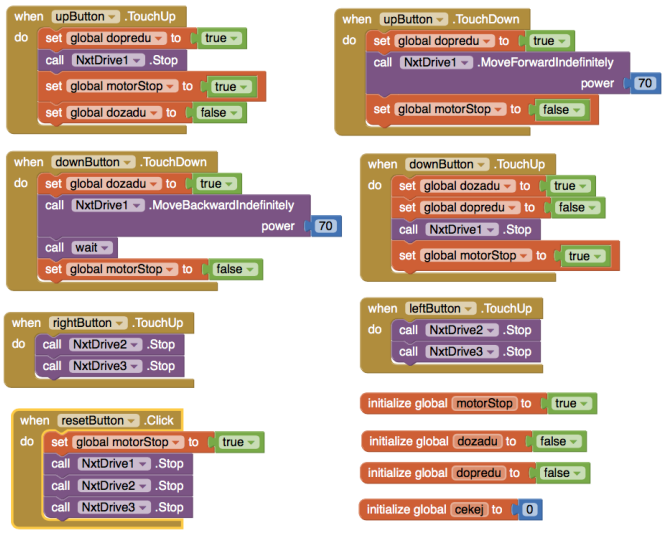
Všechny komponenty je vhodné vždy pojmenovat dle plnící funkce pro pozdější přehlednost programu a identifikaci. Dále je potřeba nastavit komponenty tak, aby byly vyobrazeny až po úspěšném připojení.



Obrázek 88: Použité - komponenty



Obrázek 89: Použité bloky část 1.



Obrázek 90: Použité bloky část 2.

Celý vývoj aplikace je shrnut v obrázcích výše. Na prvním je vyobrazeno použitích všech potřebných komponent. V dolní liště jsou neviditelné komponenty. Tyto všechny komponenty jsou následně zprovozněny v části - Blok.

Na příslušném obrázku jsou všechny funkční bloky ke správnému průběhu aplikace. Jsou zde využity metody, které se nacházejí v seznamu bloků jednotlivých tlačítek. Seznam je nazván dle názvu komponenty např. upButton. Do těchto metod jsou zabaleny komponenty pro spuštění jednotlivých motorů. Bloky se nacházejí v seznamech nxtDrive.

V metodách jsou dále zabaleny settery, které nastavují předem vytvořené proměnné do logických hodnot. Proměnné jsou vytvořeny dle bloků v seznamu Variables. Proměnné jsou při zavedení nastaveny na logickou hodnotu false. Logické hodnoty jsou k dispozici v seznamu Logic.

Metody *nazev.touchDown* jsou použity z důvodu sputění motoru pouze při stlačení příslušného tlačítka. Po jeho puštění by měl být motor zastaven. K tomu slouží metody tlačítek *nazev.touchUp*.

9.2.2 Návod Pocket Code

Při vývoji v Pocket Code je předpokládáno, že v příslušném smartphonu jsou staženy potřebné grafikcé objetky. Grafické objekty jsou pro vývoj základ.

Robot Control nevyžaduje, žádně grafické objekty ze složky *Pozadí*. Při vývoji je využito pouze ovládacích prvků ze složky *Buttons*.

Každému vývoji spočívá vytvoření nebo duplikace programu do kterého jsou vloženy grafické objekty. Tyto objekty jsou vkládány pomocí tlačítka s ÷v dolní části displeje. Tlačítko vedle se znakem play spouští vytvořenou aplikaci.

Po klepnutí na dané objekty je možno přidat vzhledy nebo zvuky, ale hlavním úkolem, který je nezbytný pro vývoj tj. přidání scriptů grafickým objektům.

Vývoj Robot Controleru tedy začíná u náhrání a napozicování jednotlivých objektů. Tyto objekty jsou napozicovány pomocí scriptů v seznamu *Pohyb*. Tlačítka musí být přizpůsobena na 50 % velikosti pomocí scriptu v seznamu *Vzhled*. Tyto scripty musí být složeny do hlavního bloku *When program starts*.

Po napozicování všech tlačítek, je na řadě po kliknutí na ně, uvést motory do pohybu daným směrem. Při udávání směru po kliknutí se veškeré srcipty skládají do scriptů *Na klepnutí* ze seznamu Events. Vývoj je u všech tlačítek obdobný. Vždy je použit script *setNxtMotor* nebo *stopNxtMotor*. U techto scriptů se nastavují porty připojených motorů. Pro pohyb vpřed nebo vzad jsou použity vždy oba porty. Pro pohyb do stran je používán vždy jen jeden port a to dle směru. Ve scriptu pro stopnutí je jistější, vždy zastavit motory pomocí obou portů. Tímto nastavením se předejde možným problémům.

Pro vývoj jsou tedy použity scripty ze seznamu Events, Pohyb, Lego NXT a Vzhled. Na obrázcích níže je nastíněn úplný vývoj Robot Controleru v Pocket Code.

9.2.3 Rozdíly vývoje

Rozdíl vývoje je především na reagování jednotlivých tlačítek. Na rozdíl od Pocket Code, který disponuje jen událostí *Na klepnutí* je u MIT App Inventoru k dispozici mnoho jiných událostí. Tyto události jsou použity právě při vývoji. Mimo ty použité jsou k dispozici metody Click, LongClick, GotFocus, LostFocus. V tomto ohledu se chování obou aplikací nepatrně liší - nemají však vliv na funkci.



Obrázek 91: Scripty buttonsUp



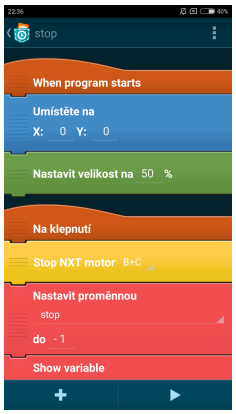
Obrázek 92: Scripty buttonsDown



Obrázek 93: Scripty buttonsLeft



Obrázek 94: Scripty buttonsRight



Obrázek 95: scripty buttonStop

9.3 Robot Control from List

Vývoj této aplikace vychází z vývoje aplikace *Robot Controler*. Princip aplikace je nepatrně pozměněn. Ovládací tlačítko zde neslouží pro okamžité rozpohybování robota. Směrová tlačítka zde slouží pro udání směru, který je zaznamenán do vytvořeného seznamu. Z tohoto seznamu směr robota přečten a zrealizován. Udáním sekvencí jednotlivých směrů je vytvořena cesta robota, na kterou se robot po stisku tlačítka vydá.

Jak je partné z předchozího odstavce, seznam použitých komponent je rozšířen. Seznam je rozšířen dle vývojového prostředí. Vždy se jedná maximálně o 4 nové komponenty.

Úloha spadá do kategorie těžkých úloh. Úloha vyžaduje povědomí o práci s Listem. List je datový objekt pro ukládání více hodnot, které jsou ke čtení pomocí indexů. Do Listu je možné mimo jiné také ukládat a zároveň číst hodnoty.

9.3.1 Návod vývoje v MIT App Inventoru

Při vývoji této aplikace je tedy použita celá aplikace *Robot Contoler*. Tuto aplikace tedy otevřeme a pomocí funkce *Save as*. Poté je potřeba přidat v části Designer další komponenty. Do horní části je vhodné umístit komponentu Label pro zobrazení směru. Do tohoto labelu jsou zařazeny jednotlivé směry, dle kterých je vidět celý průběh cesty.

Další komponentou je přidání startovacího tlačítka. Toto tlačítko je potřeba správně nazvat, např. startButton.

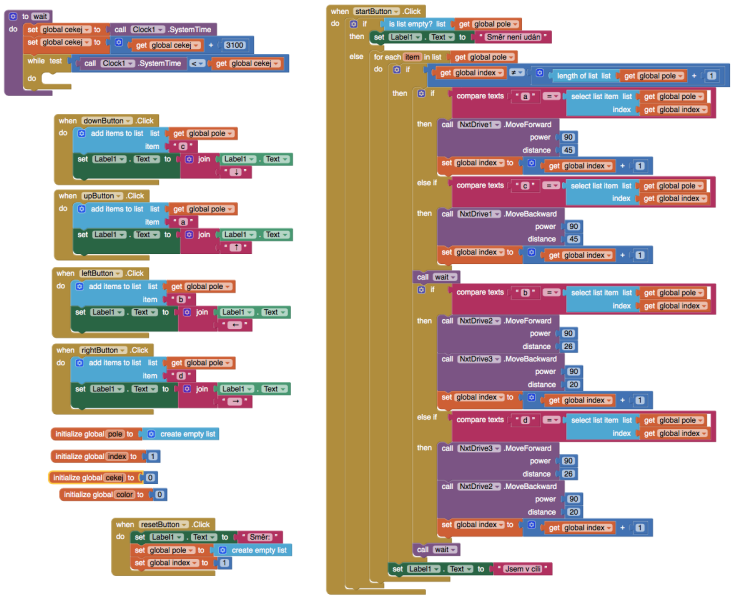
V části Blocks je dodatečně vytvořit potřebný List. Tento list je vytvořen pomocí bloku ze seznamu Variables a bloku *createEmptyList* v seznamu List. Do tohoto listu jsou uloženy hodnoty a,b,c,d dle toho, jaký směr je udán. Hodnota je do listu uložena po kliknutí na tlačítko, které slouží pro ovládání směru robota.

Hodnoty do listu jsou ukládány po kliknutí na tlačítko, proto je potřeba událost, na které tlačítko reaguje. Změna proběhne z události *TouchDown* na *Clic*. Bloky s událostí *TouchUp* musí být z aplikace vymazány.

Dalším krokem je neprodlená aktualizace udaného směru, která proběhne při ukládání do Listu. Vždy při kliknutí na příslušné tlačítko, je uložena hodnota do listu a pomocí ASCII znaků je aktualizována i komponenta *Label*. ASCII znaky reprezentují udaný směr.

V této aplikaci je vytořena metoda nazvaná *Wait*. Již z názvu je patrné, že se jedná o vytvořené prodlení, které slouží k potřebnému čekání mezi jednotlivými kroky cesty. Prodlení dopomáhá k pohybu v jednom směru a také k rozeznání právě prováděného kroku cesty.

V poslední řadě je nutné zprovoznit akce po kliknutí na tlačítko start. Celá akce je složena z podmínek, setterů, NXT bloků, porovnávacích bloků a metod pro práci s Listem. Bloky jsou použity ze seznamů Control, Text, NxtDrive, Label, Procedures, Math a List. Celý vývoj je zaznamenán v obrázcích níže.



Obrázek 96: Vývoj Robot Contoler from List

9.3.2 Návod vývoje v Pocket Code

Vzhledem k tomu, že Pocket Code nedisponuje jinými událostmi než je právě *Na klepnutí*, tak se vývoj od aplikace *Robot Controler* v tomto ohledu neliší.

V první řadě je nutné do aplikace přidat pozadí. Pozadí obsahuje tex, který zde slouží jako label ve vývojovém prostředí MIT App Inventor. Toto pozadi se přidává stejným způsobem jako grafické objekty

do aplikace. Přidáno je pozadí s názvem *smer*, který je součástí balíku vytvořených objektů. Do aplikace je potřeba přidat navíc grafické objekty ze složky Buttons. Jedná se o tlačítka Start. Tlačítko je jako každý objekt potřeba velikostně upravit a napozicovat.

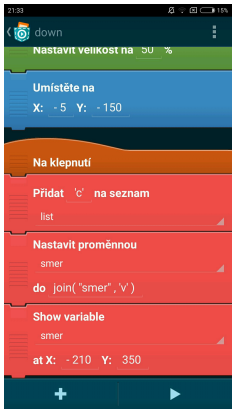
Dalším krokem vývoje je vytvořit potřebné proměnné, do kterých jsou ukládány textové znaky znázorňující směr. Proměnná by měla být napozicována k textu Směr, který je na pozadí.

Princip aplikace je stejný jako u MIT App Inventoru. Po klepnutí na pohybová tlačítka se robot nedá do pohybu, ale do vytvořeného seznamu se uloží hodnoty a, b, c, d. Příslušná hodnota je ukládaná dle zvoleného směru. Zadáním sekvencí jednotlivých směrů je vytvořena cesta, na kterou se robot vydá, až po stisku tlačítka Start.

Tlačítko Stop v této aplikaci slouží pro zastavení a zárveň pro reset předem udaného směru. Pokud je špatně zadáná cesta, tak je možné celý seznam vymazat. Seznam je vytvořen pomocí scriptů v seznamu *Data*. Dále jsou použity další scipty ze seznamů Vzhled a Ovládání. Celý vývoj je dostupný na obrázcích níže.



Obrázek 97: Scripty tlačítka UP



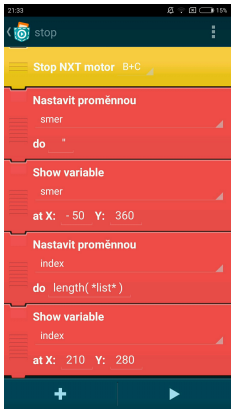
Obrázek 98: Scripty tlačítka Down



Obrázek 99: Scripty tlačítka Left



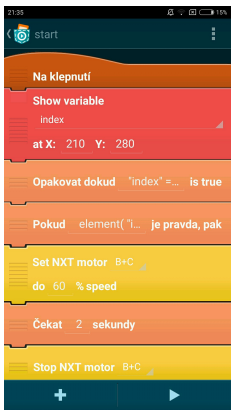
Obrázek 100: Scripty tlačítka Right



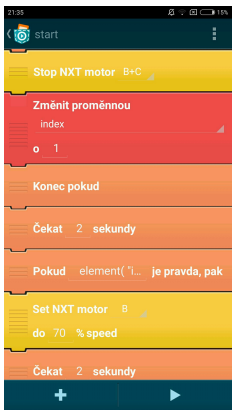
Obrázek 101: Scripty tlačítka Stop



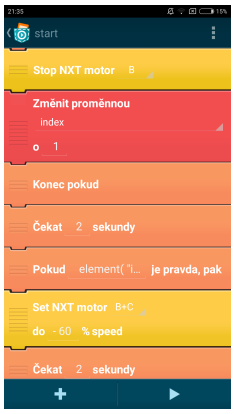
Obrázek 102: Scripty tlačítka Start 1.



Obrázek 103: Scripty tlačítka Start 2.



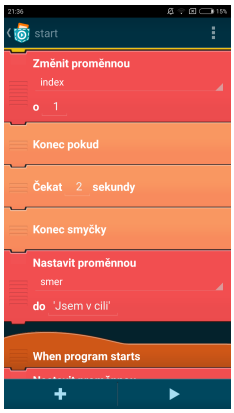
Obrázek 104: Scripty tlačítka Start 3.



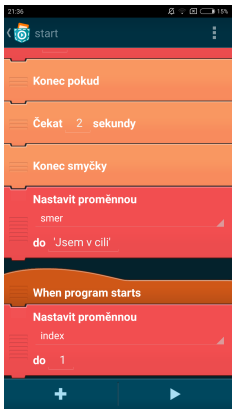
Obrázek 105: Scripty tlačítka Start 4.



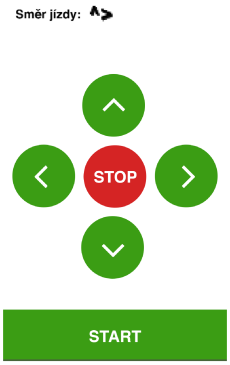
Obrázek 106: Scripty tlačítka Start 5.



Obrázek 107: Scripty tlačítka Start 6.



Obrázek 108: Scripty tlačítka Start 7.



Obrázek 109: GUI Robot Controler List

9.3.3 Úskalí aplikace

Úskalí této aplikace je skryto ve vnořených podmínkách, které jsou použity pro hladší průběh. Každá další vnořená podmínka znamena další větvení. Konec tohoto větvení je možno zaměnit s koncem celé podmínky. Pokud by se tak stalo, tak podmínky nebudou pracovat správně. Bohužel této nepřehlednosti nejde zabránit, pokud je při vývoji potřeba. Při vývoji je třeba dbát na zvýšenou pozornost a je nutné si hledat konce použitých scriptů.

9.4 Detekce barev pomocí Color Senzoru

Návod pro tuto aplikaci je vytvořen pouze pro prostředí MIT App Inventor. Je tak učiněno na základě, získaných informací ohledně nefunkčnosti Color senzoru v Pocket Code.

Časová náročnost vývoje se pohybuje od 1 do 3 vyučovacích hodin. Náročnost je proměnlivá dle toho, jak je porozuměno změnám oproti předchozím úlohám. Použité prvky již byly použity v předchozích úlohách.

Celá aplikace je pouze pro detekování barev nikoliv pro pohyb mezi příslušnými barvami pro detekování. Je to z důvodu špatné detekování naměřených hodnot. Tyto hodnoty se mohou lišit dle intenzity a teploty okolního světla. Téměř nikdy nebude aplikace fungovat správně, proto není její vývoj doporučen.

Aplikace navíc využívá plátno dodávané ke stavebnici NXT. Na tomto plátně se nachází barevná škála, která dobře posloužila při vývoji.

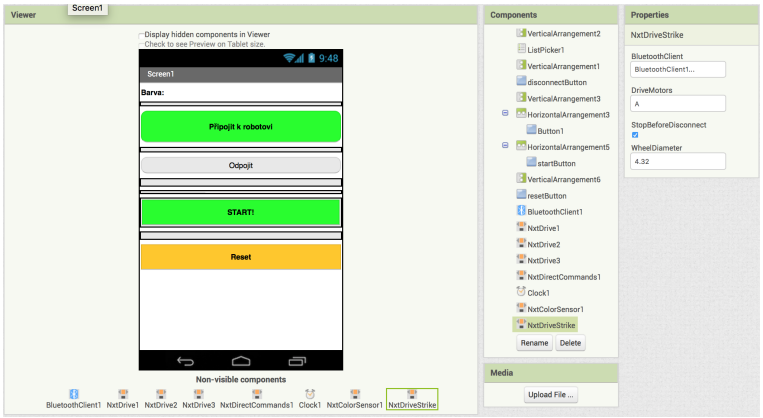
Aplikace pro detekování barev poprvé využívá jedno z mnoha senzorů. Color senzoru ten nejprve musí být připojen do jednoho ze čtyř portů. Tento zvolený port musí být i nastaven po výběru komponenty NxtColorSenzor.

Aplikace využívá další klasické komponenty. Label, který je použit již v Robot Control List a 2 tlačítka, ketré jsou rozloženy pomocí komponent ze seznamu Layout. Komponenta Label je zde defaultně nastavena na text Barva a slouží k oznámení detekované barvy. Tlačítka zde slouží pro start detekování a jeho případného resetování, dle toho by měla být i pojmenována a barevně odlišena.

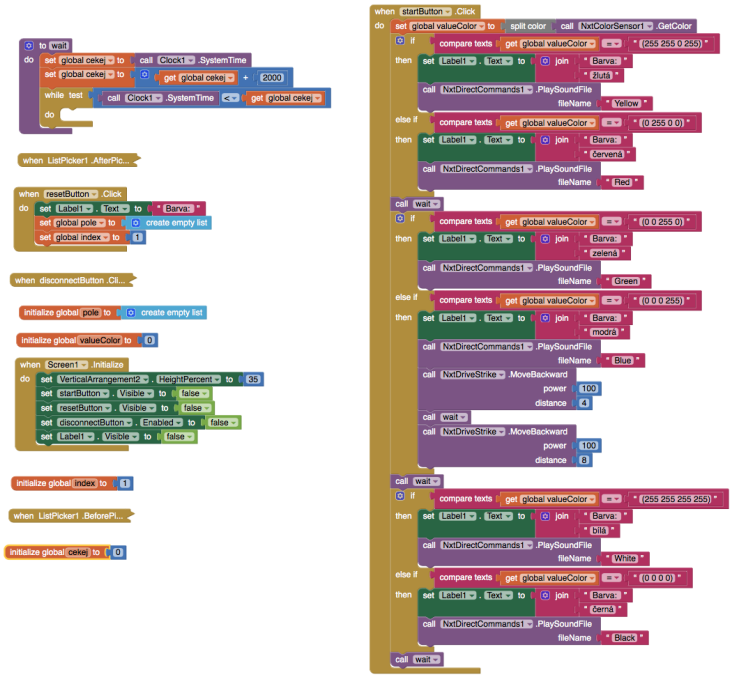
V této aplikaci je dále poprvé využito třetího motoru, který zde slouží pro vystřelení kuliček na detekování správné barvy. Dále je poprvé použitá komponenta NxtDirectCommands, která umožní pracovat se soubory uloženými v paměti NXT. V tomto případě aplikace bude spouštět soubor Red, Green, Blue, White, Black, Yellow dle detekované barvy. Pokud v NXT tyto soubory nejsou, je zapotřebí pomocí originálního prostředí Lego Mindstorms nahrát tyto soubory do kostky NXT. Tento motor je zpravidla zapojen v portu A. Tento port musí být předem nastaven.

V části blok jsou naskenované hodnoty uloženy do vytvořeného Listu do jednotlivých indexů. Po stisku tlačítka Start započne detekování barvy. Tato naskenované hodnota je porovnávaná s hodnotami, které jsou uloženy v Listu. Veškeré barevné hodnoty jsou pomocí bloku *SpliColor* v seznamu Colors převedeny do RGBA kódu. Tato metoda dá celé aplikaci přehlednost. Pokud je naskenovaná hodnota nalezena v Listu, tak se dle použitých vnořených podmínek aktualizuje příslušný Label a přehraje se zvukový soubor, dle toho o jakou barvu jde. Jestliže je naskenovaná modrá barva, tak robot vystřelí pomocí třetího motoru kuličku ze zásobníku.

Po stisku tlačítka reset je vymazaná hodnota s příslušné proměnné a aktualizován Label do defaultní hodnoty. V části Blocks jsou tedy použity bloky ze seznamu Text, List, Color, NxtDirectCommands, NxtDriveStrike, Label, Procesdures a Math. Celý vývoj je patrný z obrázků níže.



Obrázek 110: Detekce barev část Designer



Obrázek 111: Detekce barev část Blocks

9.4.1 Úskalí aplikace

Vývoji předchází naskenování barev na plátně dodávané ke stavebnici. Tyto hodnoty jsou uloženy do příslušné proměnné nebo jsou zaznamenány přesné číselné hodnoty, které pak slouží k porovnávaní právě naskenovaných hodnot.

Pokud tento scan proběhne za určité světelné intenzity a teploty, tak poté musí i celé aplikace probíhat za stejných hodnot. Jestliže není dosaženo potřebných aspektů naměřené hodnoty se nikde nebudou rovnat.

9.5 Pohyb robota pomocí úrovně a počtu zaznamenaného zvuku

Vývoj této aplikace je omezen pouze na prostředí Pocket Code, jelikož nebylo možné úspěšně dokončit 3 různé řešení v MIT App Inventoru. Ačkoliv umí MIT App Inventor umí vyvíjet aplikace se Sound senzorem, který je taktéž součástí stavebnice, tak se vývoj pokaždé zastavil na nemožnosti duplikace blokových akcí, které zaznamenávají hodnotu zvuku.

Jak je zmíněno výše, aplikace využívá pro pohyb robota Sound senzor, který dokáže rozeznat několik úrovní zvuku. Celá škála je udávaná v %. Princip aplikace ja založen na rozdělení stupnice na pomyslné třetiny. Stupnice je tedy rozdělena vždy po 33 %. Spodní úroveň 0 - 33 % - není využívána. Střední úroveň 33 - 66 % - využívána pro pohyb vpřed, vzad, popř. zastavení. Horní úroveň 66 - 100 % - využívána pro pohyb vlevo, vpravo, popř. zastavení.

Aplikace nevyžaduje žádné grafikcé objekty jako ovládací prvky, jak je použito v předchozích aplikací. Je zde využíváno pouze vytvořeného pozadí, které obsahuje popisky - Horní úroveň a Střední úroveň.
K těmto textům jsou napozicovány vytvořené proměné, které detekují úroveň a počet zaznamenaného zvuku.

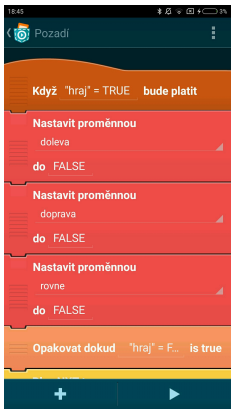
Princip aplikace je následující. Pokud je zaznamenána střední úroveň zvuku, je příslušná proměnná zvětšena o 1 a robot je uveden do pohybu vpřed. Pokud je robot uveden do pohybu, je nastavena logická hodnota příslušné proměnné do true. Jestliže robot stojí na místě, nebo je zastaven, je tato logická hodnota nastavena do false. V případě, že je zaznamenaná střední hodnota zvuku 2x za sebou, tak je robot uveden do pohybu vzad a stejně jako v prním případě je příslušná logická proměnná nastavena na true. Zároveň je ovlivněná proměnná, která ukazuje, kolikrát je konkrétní úroveň zvuku detekována. V případě detekování střední úrovně zvuku, kde je logická hodnota true (robot je v pohybu), tak je robota zastaven. Po zastavení se vrací logická hodnota do false.

Stejným principem robot reaguje na horní úroveň zvuku. Pokud je zaznamenána 1x, tak je robot uveden do pohybu vlevo. Zároveň se mění logická i číselná hodnota. Pokud je detekována tato úroveň 2x za sebou, tak je robot uveden do pohybu vpravo a zároveň jsou ovlivněny proměnné. Pokud je roboty v pohybu vlevo nebo vpravo, tak 2x detekovaná hodnota horní úrovně robota zastavuje. Číselná proměnná je nastavena na hodnotu 0 a logická na hodnotu false.

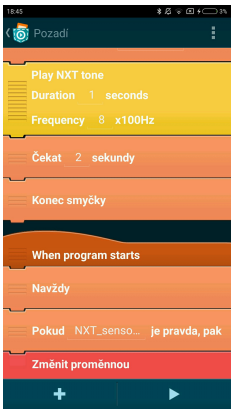
S vývojem aplikací bylo zapotřebí vytvořit příslušné zvuky, které usnadní práci s pohybem robota. Tyto zvuky jsou ke stažení na vytvořené webové stránce. Jedná se o zvuky, které umožní jedno nebo dvě po sobě jdoucí pípnutí. Úroveň hlasitosti musí být nastavena pomocí tlačítek Volume na smartphonu.

Dolní interval stupnice není použit. Zvuk, který vydávají motory robota patří do tohoto intervalu stupnice, proto by byla spuštěna akce, které je nastavená na detekci zvuku v této úrovni ihned, po uvedení robota do pohybu.

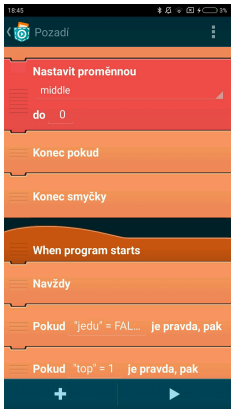
Aplikace spouští při pohyb robota vzad zvuk couvání (opakované pípání) podobně jako couvající auto. Jestliže je pípnutí moc intenzivní a zvukový senzor je umístěn moc blízko konstrukci, tak je toto pípuntí detekováno ve střední hodnotě. Tato detekce je tedy nastavena jiná hodnota a mění se podmínky a to je nežádoucí. V tomto případě je nutné senzor pomocí Lego kostek oddálit od konstrukce robota, nebo při vývoji nepoužívat scripty pro pípnutí při pohybu vzad. Při vývoji jsou využity již známé scripty. Jsou to scripty ze seznamů Events, Ovládání, Data a Lego NXT.



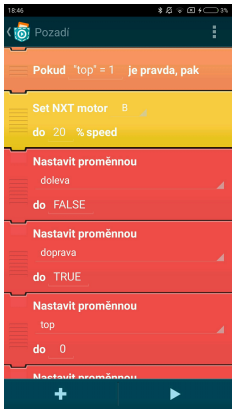
Obrázek 112: Vývoj část 1.



Obrázek 113: Vývoj část 2.



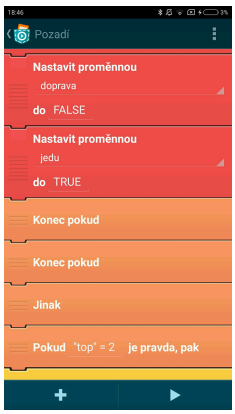
Obrázek 114: Vývoj část 3.



Obrázek 115: Vývoj část 4.



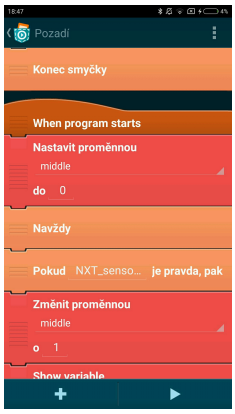
Obrázek 116: Vývoj část 5.



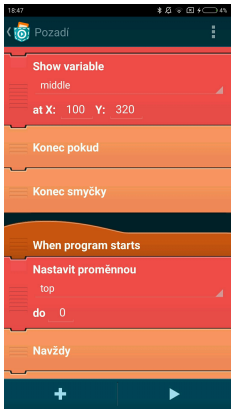
Obrázek 117: Vývoj část. 6



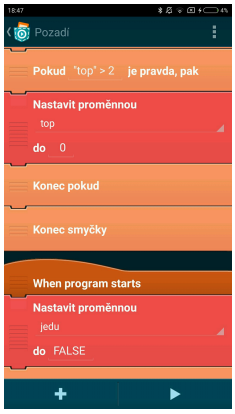
Obrázek 118: Vývoj část 7.



Obrázek 119: Vývoj část 8.



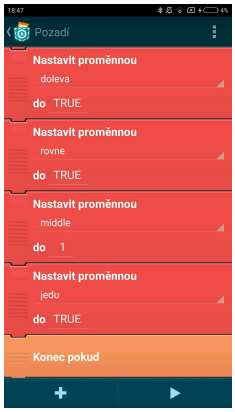
Obrázek 120: Vývoj část 9.



Obrázek 121: Vývoj část 10.



Obrázek 122: Vývoj část 11.



Obrázek 123: Vývoj část 12.



Obrázek 124: Vývoj část 13.



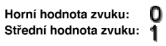
Obrázek 125: Vývoj část 14.



Obrázek 126: Vývoj část 15.



Obrázek 127: Vývoj část 16.



Obrázek 128: GUI Sounds Controler

9.5.1 Úskalí aplikace

Při spuštení připravených zvuků je těžké trefit správnou hladinu zvuku pro určený směr. Detekce úrovně se liší intenzitou. Intenzita stoupá a klesá dle vzdálenosti spuštení zvuku od daného senzoru. Pokud je tedy spuštěn zvuk střední úrovně pro pohyb vpřed, tak může být robot uveden do pohybu vlevo. Senzor v danou chvíli může vyhodnotit detekovaný zvuk jako zvuk horní úrovně a to v závislosti na příliš krátkou vzdálenost. Nicméně s trochou tréningu se dá robot obstojně ovládat.

9.6 Striker

Ze všech vytvořených aplikací je tato nejobsáhlejší. Aplikace striker využívá prvky z předešlých aplikací, a proto je zařazena na posledním místě. Vývoj aplikace by neměl být náročný, jelikož se zde opakují již vytvořené postupy.

Aplikace využívá střílecí zařízení, které je využito v aplikaci Detekování barev. Dále obsahuje ovládací prvky pro pohyb z aplikace Robot Controler a navíc je využito ultrasonického senzoru stavbenice pro detekování bezpečné vzdálenosti od robota. Pokud je tato vzdálenost překročena, tak robot upozorňuje hlasitým pípnutím různé frekvence.

Časová náročnost by měla být od 1-2 hodiny. Délka vývoje se může lišit dle pozornosti žáků na

předchozích aplikacích. Délka vývoje by však neměla přesáhnout 2 hodiny.

9.6.1 Návod vývoje v MIT App Inventoru

Aplikace vychází opět z aplikace Robot Controler proto, aby bylo možné dopravit robota na místo určení. Díky pohybovým prvků má Striker mnoho využití. Nyní jsou ovládací prvky lépe rozvrženy tak, aby bylo ovládání pohodlnější. Počet tlačítek se nemění, jen jsou jinak použity komponenty ze seznamu Layout. Jedná se o prvky *HorizontalArragments*, do kterých jsou vloženy tlačítka pro ovládání.

Dalším prvkem jsou komponenty Label, které byla využita u aplikace Robot Controler from List, kde tento Label sloužil jako ukazatel cesty. Label je vhodné zanechat v horní části aplikace. První label je nutné přejmenovat v defaultní nastavení na hodnotu ”Upozornění”. Tento label bude znázorňovat počet vydaných upozornění. Druhý Label, který je nastaven na hodnotu ”Zabito”znázorňuje počet zabitých delikventů. Tento label je aktulizován v momentě, kdy je z robota vystřelena kulička. Defaultní hodnota dalšího Labelu je ”Zabito”. Labely by měly být i vhodně pojmenovány, aby byly dobře rozpoznatelné.

Další novým prvkem je komponenta NxtUltRasonicSenzor, který je zařazen v dolní liště, kde jsou tzv. non-visible komponenty. U této komponenty je potřeba nastavit vlastnost *BottomOfRange* na hodnotu 40 a *TopOfRange* na hodnotu 50. Zadané hodnoty jsou v centimetrech. Poslední komponentou je NxtDriveStrike. Jedná se o použití třetího motoru, jen název je nastaven tak, aby byla komponenta snadno identifikována. Této komponentě je zapotřebí nastavit vhodný port přípojení. Motor je připojen na port A. Oboum NXT komponentům je nezbytné nastavit bluetooth klienta.

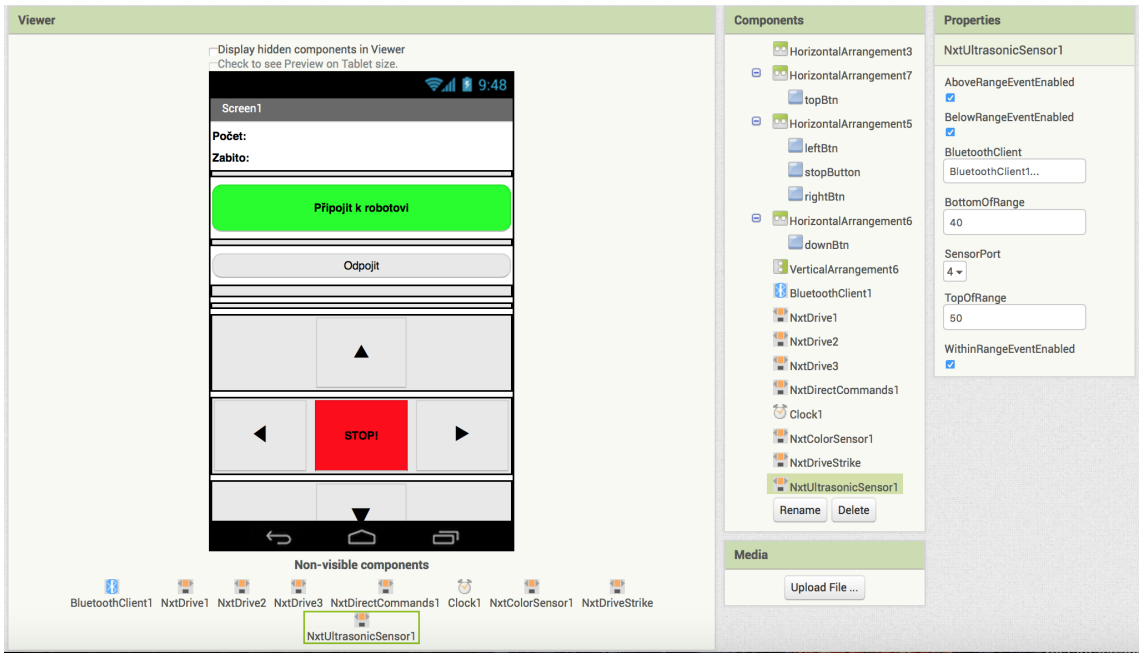
Princip střílení a upozornění je následující. Pokud je delikvent detekován pomocí Ultrasonicu v blízkosti 40 cm, je delikvent upozorněn pípnutím. Hodnota proměnné *Pocet* je zvětšena o 1. Hodnota této promenné je poté nastavena jako text příslušného Labelu. Po detekování delikventa je čekáno, než delikvent opustí svou oblast. Toho je dosaženo čekáním aplikace do té doby, než bude vzdálenost překážky větší, než 50 cm. Tímto je zamezeno nekonečnému střílení na stejnou překážku či delikventa.

Tento postup se opakuje do doby, než se proměnná *Pocet* nerovná hodnotě 3. V tomto případě je vystřeleno a zároveň zabito. Poté je proměnná *Zabito* zvýšená o 1 a proměnná *Pocet* je vynulována. Zároveň jsou vždy aktualizovány příslušné Labely.

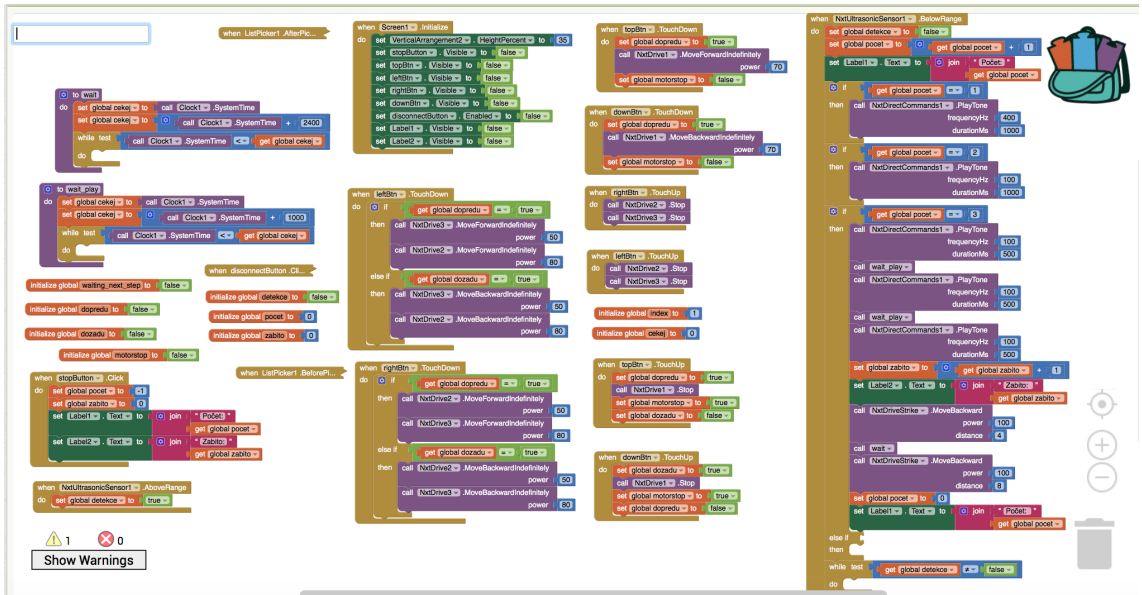
V části Block jsou použity již známe bloky. Novými bloky jsou metody pro práci s Ultrasonickým senzorem *whenUltrasonicSenzor.BelowRange* a *whenUltrasonicSenzor.AboveRange* do těchto metod jsou skládány jednotlivé bloky realizující upozorňují a výstřel ve správnou dobu.

Ostatní bloky jsou v aplikaci, pokud byla duplikovaná aplikace Robot Controler. Aplikace disponuje použitím klasických bloků ze seznamu Math, Variables, Procedures, Control, Logic, Lables, stopButton, leftBtn, rightBtn, upBtn, downBtn, nxtDriveStrike, nxtDrive1, nxtDrive2, nxtDrive3, nxtDirectCommands a Clock.

Následující obrázky zobrazují vývoj celé aplikace.



Obrázek 129: Vývoj Striker v části Designer



Obrázek 130: Vývoj Striker v části Blocks

9.6.2 Návod vývoje v Pocket Code

Princip aplikace se zde neliší, proto je zbytečné ho v této části popisovat znovu. Nastíněn je v úvodu této sekce a je detailněji popsán v části Vývoj pro MIT App Inventor.

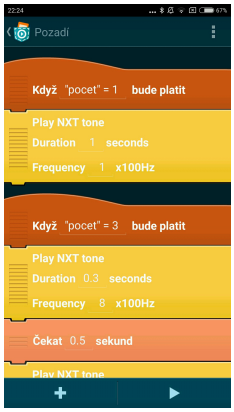
Vzhledem k tomu, že aplikace vychází z aplikace Robot Controler, kde je vyobrazen vývoj tlačítek, které pohybují robotem, tak je zde nastíněna jen část vývoje, která probíhá v části Pozadí. Tento vývoj je

paradoxně kratší, než vývoj aplikace Robot Controler from List.

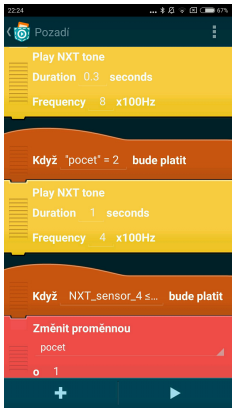
Při vývoji aplikace je zapotřebí změnit pozadí na strike.png, které je k dispozici ve složce Backgrounds ve vytvořených grafických prvcích. Pozadí obsahuje dva texty sloužící jako labely v MIT App Inevntoru. První text - Upozornění. Druhý text - Varovných výstřelů. K těmto textům jsou napozicovány předem vytvořené proměnné *Pocet a kill*.

Tyto proměnné jsou aktualizovány dle počtu upozornění. Celá aplikace se chová stejně jako aplikace vytvořená v MIT App Inventoru. Pokud je delikvent zaznamenán ve vzdálenosti 40 cm je upozorněn a příslušná proměnná je zvýšena o 1. Opět je čekáné do té doby, než bude vzdálenost vetší než 50 cm. Pokud je delikvent upozorěn 3x, tak je pomocí střílecího zařízení vystřela kulička. Poté je proměnná Pocet vynulována a proměnná Kill zvýšena o 1.

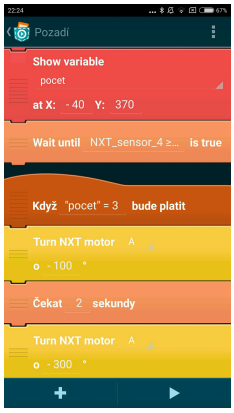
Další změna proběhne v grafickém objektu Stop, kde je zapotřebí vynulovat obě proměnné. To v minulých aplikací nebylo potřeba. Změna je vyobrazena níže, jako vývoj celé aplikace, která probíha zejména v části Pozadí.



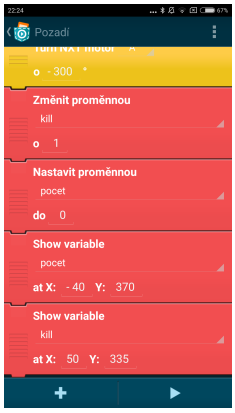
Obrázek 131: Vývoj Striker - část 1.



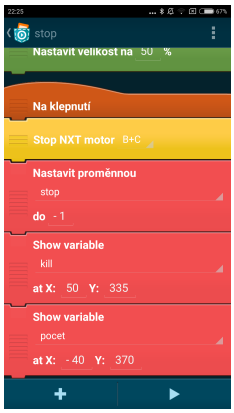
Obrázek 132: Vývoj Striker - část 2.



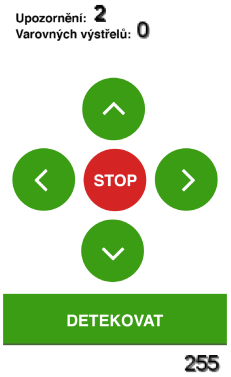
Obrázek 133: Vývoj Striker - část 3.



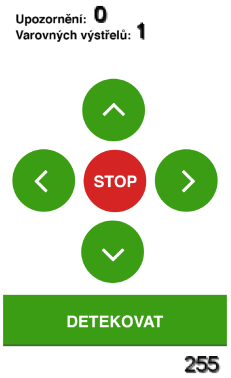
Obrázek 134: Vývoj Striker - část 4.



Obrázek 135: Vývoj Stiker - změna v tlačítku Stop



Obrázek 136: Náhled funkční aplikace - Striker



Obrázek 137: Druhý náhled funkční aplikace - Striker

9.6.3 Úskalí aplikace

Úskalí aplikace se může skrývat v použití několika podmínek za sebou. Myšleno v obou prostředí, ale jedná se jen o skládání jednotlivých bloků.

Novým prvkem je cyklus s podmínkou na konci, který se musí vždy nacházet na konci skládaných bloků (MIT App Inventor). Je v aplikaci kvůli čekání na příslušný stav. V tomto případě je zde pro čekání na změnu vzdálenosti nad 50cm. Stejný způsob je použit i v Pocket Code.

10 Předání informací žákům základních škol

Sezení se žáky základních škol proběhlo ve 3 termínech. Tyto termíny sloužili k předání informací ohledně NXT vývoje v Pocket Code. Skupina byla složena ze 3 žáků ve věku 13-15 let. Žáci docházejí nebo docházeli na ZŠ a MŠ Lomnice nad Lužnicí.

Tato sezení se konala mimo školní vyučovací dobu. Vývojové prostředí Pocket Code bylo vybráno na základě jednoduchého použití grafických objektů pro ovládání robota. Tyto prvky se nacházejí ve vytvořeném balíčku komponent. Dále je prostředí jednodušší stejně jako vývoj aplikací oproti prostředí MIT App inventor.

Pro vývoj nebo seznámení s vývojem stačí pouze vytvořený robot, smartphone a popř. potřebné návody. Těchto potřebných věcí je daleko méně, než při vývoji v MIT App Inventoru. V MIT App Inventoru je potřeba dostupnost WiFi připojení, PC pro každého člena skupiny (pokud bychom chtěli, aby každý pracoval sám), robota a popř. potřebné návody. Díky tomu, že pro vývoj v Pocket Code není zapotřebí WiFi připojení a PC, kterými disponuje každá IT učebna na školách, tak mohl být vybrán jakýkoliv prostor, který disponoval klidným prostředím pro soustředění a dostatkem prostoru pro žáky.

10.1 Náplň sezení

Každé sezení trvalo 90 minut. Bylo zapotřebí čas přizpůsobit vyučovacím hodinám. Při prvním sezení proběhlo seznámení s robotem, kterými senzory disponuje a do jakých portů jsou motory a senzory stavebnice zapojeny. Dále proběhlo stažení prostředí Pocket Code a jeho nainstalování. Po instalování proběhlo stažení balíku vytvořených prvků.

Stahování, instalování a následné seznamování trvalo většinu času. Poté zbyl čas (cca 30 min) pouze na první vytvoření jednodcuhé aplikace, která spustila zvuk 100 Mhz po dobu 1s na stisk jakéhokoliv tlačítka. Prvních 15 min měli žáci samostatně přijít na vytvořené jednoduché aplikace. Po neúspěšném vytvoření bylo v následujicích 15 - 20 min dosaženo úspěšného vytvoření. Tomu předcházely instrukce, které navedly žáky k vytvoření. Aplikace byla následně úspěšně odzkoušena.

Na druhém sezení již byli žáci seznámeni s prostředím a zároveň s vývojem aplikací, kde robot reagoval na stisk tlačítka. Tento princip je použit také v aplikaci Robot Controler. V této aplikaci je jen použito více ovládacích prvků. Tyto prvky reagují na klepnutí, ale spouštějí jiný script než v prvotně vytvořeném programu.

Žáci měli vytvořit ovládání pohybu robota pomocí již dosažených znalostí. Na vytvoření bylo k dispozici 45 min čili jedna vyučovací hodina. Žáci měli aplikovat již známý postup do této aplikace. To se bohužel nepodařilo. Ačkoliv každý z žáků skončil v jiné fázi vývoje, tak ani u jednoho nebyl vývoj dokončen. Vývoj byl posléze dokončen předáním informací, jak aplikaci dokončit. Aplikace byla následně úspěšně ověřena.

Ve zbývajícím čase byl započat vývoj aplikace Robot Control from List. Tomuto vývoji předcházelo nastínění principu aplikace. Tato aplikace využívá o tlačítko více, které spouští předem nastavenou cestu. Cesta je nastavena pomocí stisku jednotlivých tlačítek, které ukládají příslušné hodnoty do Listu. Žáci zvládli ve zbývajícím čase předělat reagování na stisk tlačítka. Po stisku tlačítka se ukládá textová hodnota do nově vytvořeného Listu. Touto částí bylo druhé sezení ukončeno.

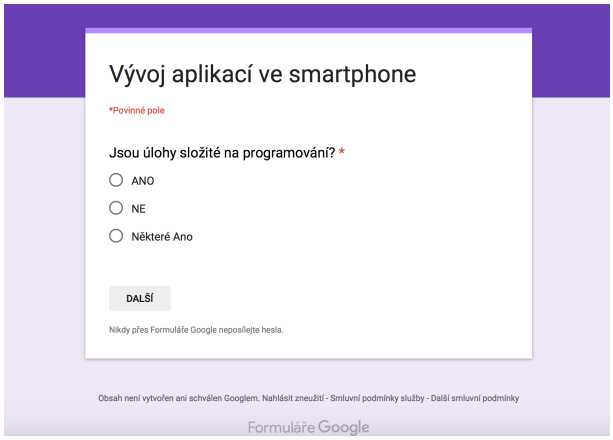
Třetího sezení bylo rozděleno na 2 části. První část se týkala dokončení Robot Controler from List, kde si žáci vyzkoušeli práci se smyčkami a podmínkami. Pro dodělání aplikace bylo vyhrazeno 45 min. Tento čas však nestačil k úplnému dokončení aplikace. Když bylo patrné, že žáci sami aplikaci nedokončí, tak byl veden zbytek vývoje jednotlivými instrukcemi, které vedly k úspěšné realizaci. Předáním informací proběhlo i odzkoušení vytvořených materiálů, které slouží k pochopení vývoje. Tento materiál by měl být k dispozici i žákům. Každému nedodělání aplikace z jejich strany předcházela neznalost programovacích struktur, základů programování a dalších prvků či metod spojených s problematikou programovaní. Programování se nyní v IT předmětech na základních školách nevyskytuje.

Druhá část posledního sezení sloužila k natáčení potřebných videí. Tyto videa ukazují funkčnost vytvořených aplikací. Videa jsou přiložena na CD, které je součástí bakalářské práce.

Ostatní aplikace nebyla se žáky zkoušena z důvodu jejich složitosti. Nicméně tyto aplikace byly předvedeny, aby žáci získali představu o možnostech robota a možném vývoji aplikací. S rostoucí složitostí aplikací, kde bylo využito více komponent robota, zajímaly žáky intenzivněji než jednoduché aplikace.

10.2 Jednoduchý dotazník

Pro žáky byl vytvořen jednoduchý dotazník, aby mohlo být zpracováno hodnocení žáků ze sezení. Dotazník je vytvořen pomocí funkce Forms, kterou disponuje Google.

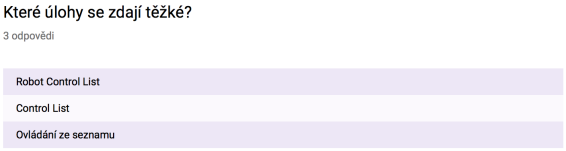


Obrázek 138: Náhled vytvořeného dotazníku

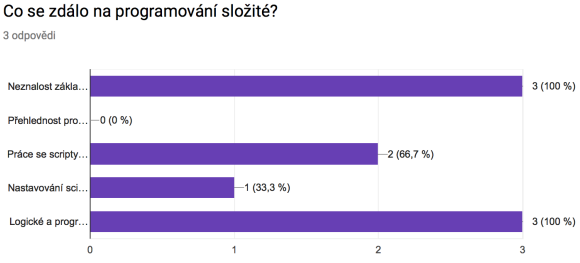
Dotazník je umístěn pod odkazem https://goo.gl/forms/EOd0Dw5VHZE8QlYz2. Tento odkaz byl poskytnut žákům, kteří jej měli povinnost vyplnit. Zaznamenané odpovědí zpracovává taktéž Google Forms. Zaznamenané údaje slouží ke zpracování následujících grafů.



Obrázek 139: Koláčový graf první otázky



Obrázek 140: Odpovědi na druhou otázku



Obrázek 141: Sloupcový graf třetí otázky



Obrázek 142: Koláčový graf čtvrté otázky

11 Vytvořená videa

Z jednotlivých videí by mělo být patrné fungování aplikací. Každá aplikace disponuje 2 videi, protože každé prostředí má své video, kde je ukázaná funkčnost v konkrétním prostředí. Tyto videa jsou řazena do jednotlivých adresářů na přiloženém CD. Při natáčení videí bylo využito skupiny žáků, kteří pomohli při natáčení pro aplikace, u kterých bylo zapotřebí více lidí.

Videa neslouží jako instruktážní video, jak jednotlivé aplikace vytvořit. Slouží pouze pro zhlédnutí správného fungování, které by mohlo sloužit k porovnání.

12 Vytvořená webová prezentace

Pro jednoduchou distribuci materiálů a důležitých odkazů pro začátek vývoje aplikací v obou prostředích byla vytvořena webová prezentace. Při vytváření bylo využito frameworku Bootstrap[13] pro jednoduché kódování layoutu. Layout celé stránky je plovoucí a responzivní pro smartphony a tablety uživatelů.

Na stránce jsou k nalezení odkazy, které vedou přímo na Google Play, kde je prostředí možno stáhnout nebo vede přímo do vývojového prostředí. Další odkazy vedou na užitečné aplikace pro dané prostředí nebo na officiální stránky vývojářů, kde jsou k nalezení i informace ohledně vývoje v daném prostředí.

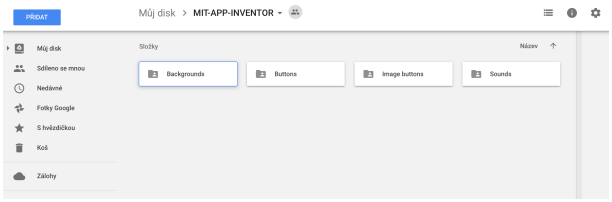
Poslední odkaz vede do soukromého disku, kde jsou veřejně sdíleny vytvořené grafické prvky. Jsou zde k nalezení pozadí, tlačítka, obrázková tlačítka a zvuky. Tyto prvky jsou převážně vytvořeny pro vývoj v Pocket Code, ale mohou být použity i v MIT App Inventoru.

V dolní části webové stránky jsou umístěny náhledy jednotlivých prvků a jejich možného využití. Tyto náhledy nejsou kompletní, jelikož grafické objekty byly v průběhu práce rozšiřovány.

[13]je jednoduchá a volně stažitelná sada nástrojů pro tvorbu webu a webových aplikací.[6]



Obrázek 143: Náhled webové stránky



Obrázek 144: Náhled na umístění grafických objektů

13 Problémy s BP

13.1 Vybité baterie

Pravidelným problémem, který nastával po celou dobu realizace praktické části bylo vybíjení baterií. Tužkové baterie určeny pro robota vydržely plně nabité cca 3 - 4 dny. Z toho vyplývá, že se baterie museli 2x do týdne dobíjet. Dostupná nabíječka na tyto baterie pojmula pouze 4 ze 6 potřebných baterií, proto nabíjení probíhalo na 2x.

Celý vývoj aplikací musel být po dobu nabíjení zcela přerušen. Většinou se jednalo o celý den. Pokud byl tedy čas nabíjení, tak vznikla nepříjemná prodleva. Další problém byl s bateriemi používáných telefonů. Zde nevznikala žádná prodleva při vývoji, jelikož se baterie nevybíjely, tak rychle jako u NXT.

13.2 Neúspěšný návrat z SOŠ Velešín

V počátcích bakalářské práce byla navštívena Střední odborná škola ve Velešíně. Na této škole probíhala v určitém datu návštěva žáků z partnerské školy z SK.

Na této škole byli zapůjčeny roboti Lego Mindstorms z KIN Jihočeské Univerzity. Žáci se s těmito stavebicemi a programováním v Lego Mindstorms NXT seznamovali v několika fázích.

Celé seznamování bylo pozorováno a dokumentováno. Bohužel při návratu z Velešína zkolabovala technika. Smartphone po obnovení funkčnosti neuchoval právě zdokumentovaný průběh seznámení s roboty.

13.3 Problémy s vývojem v MIT APP Inventoru a Pocket Code

Při vývoji v MIT App Inventoru byly zjištěny problémy. Tyto problémy jsou spojeny s některými komponenty, které by měly využívat prvky smartphonu např. komponenta Speech, která by měla předčítat daný text. Tuto komponentu se nepodařilo zprovoznit.

Další problém je v nemožnosti duplikování a tedy opakovaného použití bloků sloužících pro práci s NXT kostkou a senzory. Tento problém komplikuje a mnohdy zamezuje vývoj aplikace. Dle tohoto zjištění je potřeba s rozvahou vybírat smysl aplikace.

13.4 Problém s nepřesností Color senzoru

Při práci s Color senzorem byl zjištěn problém s nepřesností detekce barvy dle teploty a intenzity světla. Tento problém zamezuje správnému fungování vytvořené aplikace. Proto je doporučeno na základě techto informací nevyvíjet aplikace s Color senzorem. Pokud by i přes tyto informace vývoj začal, je potřeba vždy zajistit stejnorodé podmínky.

Práce s Color senzorem v Pocket Code není vůbec umožněna. Pocket Code neobsahuje scripty pro práci s tímto senzorem.

13.5 Nefunkčnost Light senzoru

Vývoj aplikace s Light senzorem není ověřený pro obě vývojová prostředí. Nejsou tedy vytvořené žádně úlohy pro aplikace s využitím s tímto senzorem. Light senzor by měl vracet hodnotu intenzity odraženého nebo dopadnutého světla na senzor. Dodané senzor bylo označeno za nevhodné či rozbité. senzor vracelo vždy hodnoty 100 % a to i po zakrytí čidla. Vzhledem k náročnosti všech úloh nebyl tento problém dále řešen např. výměnou čidla.

14 Závěr

Práce dosáhla všech cílů, které měla vytyčeny. Zpracovaný přehled, který porovnává dle vybraných parametrů pro programování na základní škole určuje dvě vhodná vývojová prostředí pro vývoj aplikací.

Z porovnání byla vybrána prostředí MIT App Inventor a Pocket Code. Jsou pospána a zároveň porovnána. Porovnání obsahuje rozdíly vývoje aplikací v jednotlivých vývojových prostředích.

Primárními cíly bylo vytvořit sadu úloh, vybrat vhodnou konstrukci a jedntolivé úlohy sepsat do návodů dle prostředí. Těchto cílů bylo úspěšně dosaženo. Byly vytvořeny celkem 3 úlohy a to pro každé prostředí. Navíc pro každé byla vytvořena ještě jedna další, která využívá specifik daného prostředí. Konstrukce je vybraná ze základní sady konstrukcí stavebnic. Tuto konstrukci je možné sestavit dle návodu dodávané ke stavebnici Lego Mindstorms. Popsání vývoje, který je doplněn obrázky celého vývoje slouží jako návod, jak vytvořit danou aplikaci ve zvoleném prostředí.

Tyto návody byly úspěšně ověřeny, při sezení s žáky základní školy. Dalším cílem bylo předání získaných znalostí a představení jednotlivých úloh skupině žáků. Žáci si při jednotlivých sezení vyzkoušeli vývoj aplikací v Pocket Code. Žákům byly předány jednotlivé návody vždy, kdy nevěděli, jak danou aplikaci úspěšně dokončit. Po pokračování dle dodaného návodu žáci aplikace vždy úspěšně dokončili. Každý ze skupiny žáků úspěšně ověřil funkci své aplikace pomocí vytvořeného robota a zároveň tím proběhlo ověření sady aplikací a jednotlivých návodů.

Dalšími cíly bylo vytvoření webové prezentace a natočení videí. Videa slouží pro zhlédnutí funkcionality jednotlivých aplikací a to z obou vývojových prostředí. Tyto videa jsou umístěna v adresářích na přiloženém CD. Webová prezentace byla vytvořená pro snadnou distribuci vytvořených grafických objektů, předání informací a užitečných odkazů všem, kteří se zajímají o vývoj aplikací pro Lego Mindstorms. Webová stránka velmi dobře posloužila při sezení se žáky. Nalezli zde potřebné zdroje a materiály.

Pro vývoj aplikací Lego Mindstorms v mobilních telefonech jsou vhodná obě vývojová prostředí, ale vhodnější je Pocket Code a to z hlediska jednouchosti vývoje, kterému nemusí předcházet vytvoření pracného bluetooth připojení, které musí obsahovat každá aplikace vyvíjená v MIT App Inventoru. Dále není zapotřebí PC, PC učebna nebo místnost, která disponuje WiFi připojením, které by mohlo být problémem zejména při realizaci výuky.

15 Literatura a použité zdroje

[1] *Zborník didinfo* 2014: 20. Ročník národnej konferencie , Bánská Bystrica 2014 [online]. Bánská Bystrica: Univerzita Mateja Bela, 2014. Dostupné z: http://didinfo.umb.sk/public/filestore/documents/richtext/161/zbornik_didinfo_2014.pdf. ISBN 978-80-557-0698-6

[2] Wolfgang Slany, Catroid: A Mobile Visual *Programming System for Children*, Austria 2012 [online]. Austria: Graz University of Technology, 2012. Dostupné z: http://arxiv.org/ftp/arxiv/papers/1204/1204.6411.pdf. ISB 978-80-557-0698-6

[3] About us - MIT App Inventor. *MIT App Inventor* [online]. Massachusetts: Massachusetts Institute of Technology, 2012 [cit. 2017-06-14]. Dostupné z: http://appinventor.mit.edu/explore/about-us.html

[4] Catrobat. *Catrobat* [online]. Graz: Graz University of Technology, 2010 [cit. 2017-06-14]. Dostupné z: https://www.catrobat.org

[5] Emulátor. In: *Wikipedia: the free encyclopedia* [online]. San Francisco (CA): Wikimedia Foundation, 2001- [cit. 2017-06-14]. Dostupné z: https://cs.wikipedia.org/wiki/Emulátor

[6] Bootstrap 3 Tutorial. *W3schools [online]*. 1999 [cit. 2017-07-03]. Dostupné z: https://www.w3schools.com/bootstrap/

[7] Drag and Drop. In: *The Tech Terms Computer Dictionary* [online]. 2011 [cit. 2016-03-08]. Dostupné z: http://techterms.com/definition/drag_and_drop

[8] MIHAI, Dan. Mobile Apps for Interacting with Your Mindstorms NXT Robot: iNXT Remote. In: *Smashingrobotics* [online]. 2014 [cit. 2017-02-04]. Dostupné z: https://www.smashingrobotics.com/mobile-apps-for-interacting-with-your-mindstorms-nxt-robot/

[9] *What is App Inventor?*[online]. Massachusetts: Massachusetts Institute of Technology, 2012 [cit. 2017-02-11]. Dostupné z: http://appinventor.mit.edu/explore/content/what-app-inventor.html

[10] ŽIDLÍK, Martin.*Využití stavebnice Lego Mindstorms NXT při výuce bezpečnostních technologií* [online]. Zlín, 2010 [cit. 2017-06-14]. Dostupné z: http://digilib.k.utb.cz/bitstream/handle/10563/12031/židl%C3%ADk_2010_bp.pdf?sequence=1. Bakalářské práce. Univerzita Tomáše Bati

[11] KRESA, Jan.*Výuka základů programování v prostředí Scratch* [online]. České Budějovice, 2014 [cit. 2017-06-18]. Dostupné z: http://theses.cz/id/b5f11x/DP_Krejsa_Scratch.pdf. Diplomová práce. Jihočeská Univerzita. Vedoucí práce Doc. PaedDr. Jiří Vaníček, Ph.D.

16 Seznam obrázků

1 Ovládací plocha NXT Simple Remote . . . 17
2 Ovládací plocha Touch Bar . . . 18
3 Ukázka ovládání . . . 18
4 Ovládací plocha Tank . . . 19
5 Ovládací plocha 3 Motors . . . 19
6 Ovládací plocha NXT Remote Control . . . 19
7 App RoboLiterate: LEGO NXT Remote AP for WP . . . 20
8 Ovládací plocha iNXT Remote . . . 20
9 Princip fungování MIT App Inventor . . . 23
10 Drop-down Project . . . 25
11 Drop down Connect . . . 25
12 Dropdown Help [?] . . . 25
13 Rozvržení pracovní plochy [?] . . . 26
14 Levý panel GUI . . . 26
15 Pracovní plocha - náhled . . . 27
16 Komponenty . . . 28
17 Media . . . 28
18 Část Properties . . . 29
19 Ovládací bloky - první část . . . 30
20 Ovládací bloky - druhá část . . . 30
21 Logické bloky . . . 30
22 Matematické bloky - první část . . . 31
23 Matematické bloky - druhá část . . . 31
24 Bloky pro práci se String - první část . . . 31
25 Bloky pro práci se String - druhá část . . . 31
26 Bloky pro práci se Seznamy - první část . . . 32
27 Bloky pro práci se Seznamy - druhá část . . . 32
28 Bloky pro práci se Seznamy - první část . . . 32
29 Bloky pro práci se Seznamy - druhá část . . . 32
30 Bloky pro zavedení nových proměných . . . 33
31 Bloky pro vytvoření procesu . . . 33
32 Nástroj pro bloky použité v GUI . . . 34
33 Bloky pro práci s NXT Drive . . . 34
34 Pracovní plocha - blocks . . . 34
35 Změna ikony - plný batoh . . . 35
36 Seznam uložených akcí či bloků . . . 35
37 Nastavení popisku pracovní plochy . . . 36

38 Vlastnosti - HorizontalArranments . . . 37
39 Akce pro neaktivní tlačítko Odpojit . . . 38
40 Blok setdisconnectButton.Enabled . . . 38
41 Logická hodnota false . . . 38
42 Akce pro naplnění seznamu . . . 39
43 Blok setdisconnectButton.Enabled . . . 39
44 Logická hodnota false . . . 39
45 Akce pro naplnění seznamu . . . 39
46 evaluate but ignore result . . . 40
47 Zavolání připojení Bluetooth . . . 40
48 Výběr položky seznamu . . . 40
49 Podmínka IF . . . 40
50 Metoda pro zjištění stavu připojení . . . 40
51 Nastavení indikace připojení . . . 41
52 Logická hodnota true . . . 41
53 Práce s ListPicker bloky . . . 41
54 Nastavení na hodnotu False . . . 41
55 Práce s ListPicker bloky . . . 41
56 Nastavení na hodnotu False . . . 41
57 Chování aplikace po odpojení . . . 42
58 Zavolání metody pro odpojení . . . 42
59 Nastavení na hodnotu True . . . 42
60 Zneaktivnění tlačítka Odpojit . . . 42
61 Skrytí indikace . . . 42
62 Navázání spojení Al Companion . . . 43
63 Zobrazení připojovacích kódů . . . 43
64 Zadání kódů v MIT Ai2 . . . 43
65 Zavedení aplikace po připojení . . . 43
66 Seznam pro výběr připojení . . . 44
67 Aplikace po navázání připojení . . . 44
68 Menu - Pocket Code . . . 45
69 Pocket Code - Možnosti . . . 45
70 Nastavení - Bloky NXT . . . 45
71 Zaškrtnutí nastavení . . . 45
72 Přehled objektů při vývoji . . . 46
73 Práce s objetky . . . 46
74 Přehled objektů při vývoji . . . 46
75 Práce s objetky . . . 46
76 Akce pro NXT . . . 47

77 Nastavení hodnoty akce . . . 48
78 Prvky - Operace . . . 48
79 Prvky - Device . . . 48
80 Prvky - Funkce . . . 49
81 Prvky - Variables . . . 49
82 Konstrukce robota . . . 56
83 Střílecí zařízení . . . 56
84 Umístění Sound senzoru . . . 56
85 Umístění Touch senzoru . . . 56
86 Umístění RGB senzoru . . . 56
87 Umístění Ultrasonického senzoru . . . 56
88 Použité - komponenty . . . 58
89 Použité bloky část 1. . . . 58
90 Použité bloky část 2. . . . 58
91 Scripty buttonsUp . . . 59
92 Scripty buttonsDown . . . 59
93 Scripty buttonsLeft . . . 60
94 Scripty buttonsRight . . . 60
95 scripty buttonStop . . . 60
96 Vývoj Robot Contoler from List . . . 61
97 Scripty tlačítka UP . . . 62
98 Scripty tlačítka Down . . . 62
99 Scripty tlačítka Left . . . 62
100 Scripty tlačítka Right . . . 62
101 Scripty tlačítka Stop . . . 63
102 Scripty tlačítka Start 1. . . . 63
103 Scripty tlačítka Start 2. . . . 63
104 Scripty tlačítka Start 3. . . . 63
105 Scripty tlačítka Start 4. . . . 63
106 Scripty tlačítka Start 5. . . . 63
107 Scripty tlačítka Start 6. . . . 64
108 Scripty tlačítka Start 7. . . . 64
109 GUI Robot Controler List . . . 64
110 Detekce barev část Designer . . . 65
111 Detekce barev část Blocks . . . 66
112 Vývoj část 1. . . . 67
113 Vývoj část 2. . . . 67
114 Vývoj část 3. . . . 68
115 Vývoj část 4. . . . 68

116 Vývoj část 5. 68
117 Vývoj část. 6 68
118 Vývoj část 7. 68
119 Vývoj část 8. 68
120 Vývoj část 9. 69
121 Vývoj část 10. 69
122 Vývoj část 11. 69
123 Vývoj část 12. 69
124 Vývoj část 13. 69
125 Vývoj část 14. 69
126 Vývoj část 15. 70
127 Vývoj část 16. 70
128 GUI Sounds Controler 70
129 Vývoj Striker v části Designer 72
130 Vývoj Striker v části Blocks 72
131 Vývoj Striker - část 1. 73
132 Vývoj Striker - část 2. 73
133 Vývoj Striker - část 3. 73
134 Vývoj Striker - část 4. 73
135 Vývoj Stiker - změna v tlačítku Stop 74
136 Náhled funkční aplikace - Striker 74
137 Druhý náhled funkční aplikace - Striker 74
138 Náhled vytvořeného dotazníku 76
139 Koláčový graf první otázky 76
140 Odpovědi na druhou otázku 77
141 Sloupcový graf třetí otázky 77
142 Koláčový graf čtvrté otázky 78
143 Náhled webové stránky 79
144 Náhled na umístění grafických objektů 79

17 Seznam tabulek

1 *Porovnání z hlediska programování na základní škole - 1. část* 16
2 *Porovnání z hlediska programování na základní škole - 2. část* 16
3 *Seznam dostupných aplikací na mobilní platformy [?]* 17
4 *Přehled dostupnosti aplikací [?]* 21
5 *Základní funkce aplikací* 22
6 *Rozšiřující funkce analyzovaných systémů* 22
7 *Porovnání z hlediska využití senzorů* 49
8 *Porovnání dle využití prvků smartphonu* 50
9 *Porovnání práce vytvořených aplikací* 51
10 *Porovnání práce vytvořených aplikací* 51
11 *Porovnání nutných bodů při vývoji* 53
12 *Porovnání dle využití souborů* 53
13 *Porovnání dle využití souborů* 54
14 *Porovnání dle využití souborů* 54

18 Přílohy

18.1 Přiložené CD

Na přiloženém CD se nachází plné znění bakalářské práce pod názvem ”valis_lukas_bp.pdf”, dále se na CD nachází kompletní videové soubory ke všem vytvořeným aplikacím. Dále jsou přiloženy veškeré vytvořené objetky tj. grafické a zvukové soubory.