



ZADÁNÍ BAKALÁŘSKÉ PRÁCE

Název: Zásuvné moduly aplikace Dráček III - výuka fyziky
Student: Jaroslav Štěpán
Vedoucí: Ing. Jiří Chludil
Studijní program: Informatika
Studijní obor: Softwarové inženýrství
Katedra: Katedra softwarového inženýrství
Platnost zadání: Do konce letního semestru 2017/18

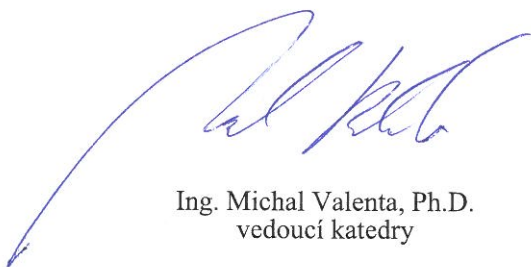
Pokyny pro vypracování

Dráček je dotyková vzdělávací aplikace pro OS Android pro žáky prvního stupně základní školy. Zásuvné moduly rozšiřují funkcionalitu klientské aplikace Dráček, která byla předmětem bakalářské práce z minulého roku.

1. Jeden z existujících modulů podrobte uživatelskému testování.
2. Analyzujte výsledky testování a navrhňte úpravy testovaného modulu.
3. Analyzujte učební osnovy pedagogů ze základních škol se zaměřením na výuku fyziky.
4. Navrhňte alespoň 5 nových modulů, které budou podporovat výuku fyziky.
5. Navrhňte editory pro tvorbu a úpravu zadání pro navržené moduly.
6. Implementujte alespoň 5 zásuvných modulů pro vybrané typy cvičení včetně editoru, vycházejte přitom z existujících modulů a v maximální míře využijte odladěný kód.
7. Hotové moduly podrobte vhodným testům

Seznam odborné literatury

Dodá vedoucí práce.



Ing. Michal Valenta, Ph.D.
vedoucí katedry



prof. Ing. Pavel Tvrdlík, CSc.
děkan

V Praze dne 5. února 2017

ČESKÉ VYSOKÉ UČENÍ TECHNICKÉ V PRAZE
FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
KATEDRA SOFTWAREVÉHO INŽENÝRSTVÍ



Bakalářská práce

Zásuvné moduly aplikace Dráček III – výuka fyziky

Jaroslav Štěpán

Vedoucí práce: Ing. Jiří Chludil

14. května 2017

Poděkování

Chtěl bych poděkovat svému vedoucímu bakalářské práce, Ing. Jiřímu Chludilovi za odborné vedení, za pomoc a rady při zpracování této práce. Dále bych rád poděkoval svým kolegům, Ondřeji Slabému a Jaroslavu Rybovi za velmi dobrou vzájemnou spolupráci.

Prohlášení

Prohlašuji, že jsem předloženou práci vypracoval(a) samostatně a že jsem uvedl(a) veškeré použité informační zdroje v souladu s Metodickým pokynem o etické přípravě vysokoškolských závěrečných prací.

Beru na vědomí, že se na moji práci vztahují práva a povinnosti vyplývající ze zákona č. 121/2000 Sb., autorského zákona, ve znění pozdějších předpisů. V souladu s ust. § 46 odst. 6 tohoto zákona tímto uděluji nevýhradní oprávnění (licenci) k užití této mojí práce, a to včetně všech počítačových programů, jež jsou její součástí či přílohou, a veškeré jejich dokumentace (dále souhrnně jen „Dílo“), a to všem osobám, které si přejí Dílo užít. Tyto osoby jsou oprávněny Dílo užít jakýmkoli způsobem, který nesnižuje hodnotu Díla, a za jakýmkoli účelem (včetně užití k výdělečným účelům). Toto oprávnění je časově, teritoriálně i množstevně neomezené. Každá osoba, která využije výše uvedenou licenci, se však zavazuje udělit ke každému dílu, které vznikne (byť jen zčásti) na základě Díla, úpravou Díla, spojením Díla s jiným dílem, zařazením Díla do díla souborného či zpracováním Díla (včetně překladu), licenci alespoň ve výše uvedeném rozsahu a zároveň zpřístupnit zdrojový kód takového díla alespoň srovnatelným způsobem a ve srovnatelném rozsahu, jako je zpřístupněn zdrojový kód Díla.

V Praze dne 14. května 2017

.....

České vysoké učení technické v Praze
Fakulta informačních technologií

© 2017 Jaroslav Štěpán. Všechna práva vyhrazena.

Tato práce vznikla jako školní dílo na Českém vysokém učení technickém v Praze, Fakultě informačních technologií. Práce je chráněna právními předpisy a mezinárodními úmluvami o právu autorském a právech souvisejících s právem autorským. K jejímu užití, s výjimkou bezúplatných zákonných licencí, je nezbytný souhlas autora.

Odkaz na tuto práci

Štěpán, Jaroslav. *Zásuvné moduly aplikace Dráček III – výuka fyziky*. Bakalářská práce. Praha: České vysoké učení technické v Praze, Fakulta informačních technologií, 2017.

Abstrakt

Práce navazuje na předchozí projekty Dráček, které mají za cíl podpořit výuku v základních školách pomocí moderních technologií. Cílem práce je upravit stávající aplikace pro systém Android a vytvořit pět nových modulů klientské aplikace. Práce zachycuje analýzu aplikací, které rozvíjejí výuku fyziky a důležitých aspektů, jež musí aplikace obsahovat. Na jejím základě jsou navrženy nové moduly klientské aplikace podporující výuku fyziky na prvním stupni základní školy. Výstupem práce je jejich implementace v jazyce Java, využívající knihovny pro fyzikální simulaci JBox2D. Součástí každého modulu je navíc editor pro tvorbu zadání. Funkčnost modulů byla otestována, ale před nasazením do produkce je třeba upravit a sjednotit vzhled aplikací.

Klíčová slova Zásuvný modul, výuka fyziky, Android, Java, tablet.

Abstract

The thesis is a continuation of previous projects Dráček, whose aim was to support education in schools using modern technologies. The goal of the thesis is to modify and fix existing applications for OS Android and make five new plugins for client application. The thesis describes the analysis of applications which support physics education and important aspects of them. On the basis of this analysis new plugins for client application were designed. These plugins help pupils of primary school to understand the basics of physics. The outcome of this thesis is an implementation of the plugins in Java language using the library for physics simulation JBox2D. Each plugin include an editor for creating new exercises. All modules were tested, all that is left is to make a unified appearance before deployment.

Keywords Plugins, education of physics, Android, Java, tablet.

Obsah

Úvod	1
1 Cíle práce	3
2 Analýza	5
2.1 Stávající řešení	5
2.2 Moduly Dráčka II	10
2.3 Uživatelské testování modulů Dráček II	10
2.4 Analýza učiva fyziky pro základní školy	12
2.5 Souhrn požadavků	15
2.6 Výběr technologií	17
3 Návrh	21
3.1 Společná knihovna	21
3.2 Moduly Dráčka III	23
3.3 Použité komponenty Android SDK	32
3.4 Třídní diagramy modulů	33
4 Implementace	39
4.1 Společná architektura modulů	39
4.2 Schéma nasazení	40
4.3 Omezení vyvinutého prototypu	40
4.4 Instalační příručka	41
4.5 Uživatelská příručka	42
4.6 Návod změny grafického obsahu	47
5 Testování	49
5.1 Unit testy	49
5.2 Integrované testy	49
5.3 Systémové testování	50

5.4	Uživatelské testování	50
5.5	Výsledky testování	50
	Závěr	51
	Literatura	53
	A Seznam použitých zkratk	55
	B Obsah příloženého média	57

Seznam obrázků

2.1	Aplikace Fyzika ve škole	6
2.2	Aplikace Jdu do školy	7
2.3	Aplikace PhyWiz - Physics Solver	8
2.4	Aplikace Bran it On! - Physics Puzzles	8
2.5	Modul Vybarvování před úpravou	11
2.6	Modul Vybarvování po úpravě	11
2.7	Popis experimentu <i>Rovnováha na páce</i>	14
2.8	Popis experimentu <i>Vlastnosti světla</i>	14
2.9	Popis experimentu <i>Pohyby těles</i>	15
2.10	Softwarová architektura OS Android, převzato z [1]	19
3.1	Třídní diagram aplikační vrstvy	22
3.2	Třídní diagram Prezentační vrstvy	22
3.3	Wireframe modulu Rovnováha	24
3.4	Diagram aktivit modulu Rovnováha	25
3.5	Wireframe modulu Odklon paprsku	26
3.6	Diagram aktivit modulů Odklon paprsku	28
3.7	Wireframe modulu Odklon paprsku 2	29
3.8	Wireframe modulu Odklon kuliček	29
3.9	Diagram aktivit modulů Odklon kuliček	31
3.10	Wireframe modulu Odklon kuliček 2	32
3.11	Společné třídy modulů	33
3.12	Třídní diagram modulu Rovnováha	35
3.13	Třídní diagram modulů Odklon paprsku	36
3.14	Třídní diagram modulů Odklon kuliček	37
4.1	Schéma nasazení aplikace Dráček	40
4.2	Obrazovka editoru modulu Rovnováha	43
4.3	Modul Rovnováha	43
4.4	Modul Odklon paprsku	44

4.5	Modul Odklon paprsku 2	45
4.6	Modul Odklon kuliček	46
4.7	Modul Odklon kuliček 2	47

Seznam tabulek

2.1	Hodnocení aplikací	9
-----	------------------------------	---

Úvod

Počítačová gramotnost dětí je s postupem času schopností, kterou ovládají od čím dál nižšího věku. Současně žáci základních škol trpí nedostatkem technického vzdělání a neochotou přemýšlet. Projekt Dráček III má za cíl tyto nedostatky zmírnit a pomoci s výukou školních předmětů, které tyto schopnosti rozvíjí.

Práce navazuje na projekty Dráček I a Dráček II. Dráček I měl za cíl vytvořit desktopovou aplikaci, která bude pomáhat dětem s poruchami učení. Úkolem Dráčka II bylo tuto aplikaci přesunout z PC na tablety, konkrétněji se systémem Android. Dráček III přidává do stávající aplikace novou funkcionalitu, a to prostřednictvím zásuvných modulů. Na projektu jsem spolupracoval s dalšími dvěma kolegy, přičemž každý z nás se zaměřil na jinou oblast výuky. Jmenovitě:

- Ondřej Slabý – výuka základů algoritmizace [2]
- Jaroslav Ryba – výuka matematiky [3]

Cíle práce

Cílem této práce bylo návázání na projekt Dráček a vytvoření pěti nových modulů, které budou podporovat výuku fyziky. Začátek práce je věnován analýze aplikace Dráček a stávajícímu řešení problematiky. Na jejím základě jsem navrhl moduly aplikace Dráček, které budou problematiku řešit. Navržené moduly implementuji pomocí technologií zvolených v analýze. Hotové moduly budou podrobeny uživatelskému testování cílovou skupinou v Usability laboratoři na Fakultě informačních technologií, ČVUT.

Cílovou skupinou budou žáci prvního stupně základní školy. Cílovými zařízeními budou tablety se systémem Android.

Analýza

V této kapitole popíši stávající řešení problematiky, shrnu dosavadní stav projektu Dráček a zhodnotím uživatelské testování modulů Dráčka II. V dalších podkapitolách popíšu nové moduly Dráčka III a zvolené technologie.

2.1 Stávající řešení

V současné době neexistuje mnoho kvalitních vzdělávacích her, které by se zaměřovaly na výuku fyziky a zároveň splňovaly cíl Dráčka. Při hledání výukových her na distribuční platformě Google Play lze nalézt mnoho aplikací, které jsou sice cizojazyčně lokalizované, ale samotné fyzikální zákonitosti se snaží podat hravou formou. Na poli česky lokalizovaných her lze nalézt aplikace, které buď stroze formulují teorii z učebnic, nebo jde o kvízové hry, pro které je dobré mít fyzikální základ. Dále existují aplikace, které jsou česky lokalizované, ale obsahují pouze fyzikální hádanky, které jsou pro děti příliš složité.

2.1.1 Výběrová metodika

Aplikace jsem vybíral pouze na distribuční platformě systému Android, Google Play, jelikož vybrané aplikace jsou dostupné i na konkurenční platformě pro iOS, App Store, či lze na ni nalézt odpovídající ekvivalent. Zvolil jsem aplikace buď na základě doporučení, či podle zařazení do kategorie „Vzdělávací“, či podle fulltextového vyhledávání klíčových slov physics, fyzika.

2.1.2 Testovací metodika

Následující aplikace jsem hodnotil podle několika kritérií. Byla to kritéria, která by měla odhalit, zda již existují vyhovující aplikace. Některá kritéria byla hodnocena podle mého subjektivního pohledu, zbytek informací byl převzat z internetových stránek výrobce, či stránek platformy Google Play. Veškeré

2. ANALÝZA

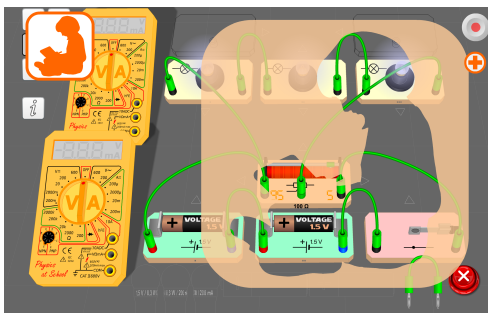
kategorie by měly obsahovat bodové ohodnocení, a to i ty (např. přítomnost české lokalizace), u kterých se takové hodnocení zdánlivě nehodí. Lze tak nastavovat významnost jednotlivých kategorií.

Hodnocení aplikací je tedy dáno následujícími kategoriemi:

- Lokalizace v českém jazyce: 0–5 bodů
- Zaměření na výuku fyziky: 0–5 bodů
- Úroveň gamifikace: 0-10 bodů
- Zpoplatnění: 0–4 body
- Úroveň podpory: 0–1 bod
- Přívětivost UI: 0–10 bodů
- Přítomnost nápovědy: 0–1 bod

2.1.3 Související aplikace

Fyzika ve škole Nejzajímavější nalezenou aplikací byl soubor animací *Fyzika ve škole* [4], který ilustruje experimenty z látky probírané na základních a středních školách. Aplikace je zpoplatněná a pořídit si ji mohou majitelé OS Android i iOS. Samotné animace sice jsou dobrou ilustrací vyučované teorie, každopádně nezasvěcenému žákovi (tedy cílové skupině Dráčka) většina experimentů nic neřekne. Aplikace je vhodná pro žáky, kteří se fyziku již učili.



Obrázek 2.1: Ukázka animace elektrického obvodu, obraz sejmut z aplikace [4]

Přítomnost české lokalizace hodnotím pěti body. Aplikace je zaměřená na výuku fyziky na základní i střední škole, tudíž taktéž hodnotím pěti body. Záporně musím ohodnotit gamifikaci a zpoplatnění. Aplikace neobsahuje žádný cíl, kterého by musel uživatel dosáhnout, každopádně v některých aplikacích lze animace ovlivňovat, či například měnit parametry rovnice. Gamifikaci tedy hodnotím čtyřmi body. Aplikaci lze na Google Play nalézt ve dvou verzích – zpoplatněné verzi a volné verzi, ve které obraz překrývá logo aplikace. Zpoplatnění hodnotím tedy dva body, kvůli volně dostupné verzi. Aplikace je často

aktualizována, dostává aktualizace i v podobě nových experimentů – hodnotím maximem bodů. UI bohužel není moc přívětivé a už vůbec ne pro dítě, protože u některých experimentů ani není znát, co který UI prvek dělá. UI tedy hodnotím pěti body. Za absenci nápovědy neuděluji žádné body.

Jdu do školy Další nalezenou aplikací je aplikace *Jdu do školy* [5], kterou jsme společně s kolegy testovali v usability laboratoři na Fakultě informačních technologií, ČVUT. Ta ovšem obsahuje pouze činnosti podobné modulům z *Dráčka II*, nejedná se tedy o aplikaci zaměřenou na výuku fyziky. Obsahuje cvičení pro děti, které se chystají do školy. Mezi takovéto cvičení patří např. na vybírání rozlišných tvarů, orientace v prostoru, či jednoduchých počtů. Výsledky testování dětmi rozeberu v podkapitole 2.3.

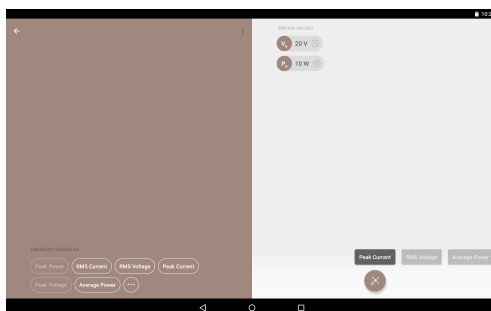


Obrázek 2.2: Ukázka výběru cvičení, obraz sejmут z aplikace [5]

Aplikace obsahuje nejen české texty, ale i český dabing, tudíž hodnotím plným počtem bodů. Aplikace neobsahuje žádné cvičení orientované na výuku fyziky, v této kategorii hodnotím nula body. Gamifikace je u této aplikace na velmi dobré úrovni. Veškeré úkoly jsou přizpůsobeny tomu, že je budou plnit malé děti. Navíc, za každé dílčí řešení dostane dítě od avatara buď pochvalu, nebo podnět pro další pokus. V této kategorii tedy aplikaci hodnotím deseti body. Aplikace není zpoplatněná, tudíž ji v tomto ohledu uděluji čtyři body. Autor sleduje zpětnou vazbu a vydává aktualizace, kategorii podpora hodnotím jedním bodem. UI aplikace je pro děti velmi přívětivé, hlavně díky avaturovi. UI uděluji devět bodů. Zadání úkolů je většinou hned jasné, každopádně vždy si lze po kliknutí na avatara zadání znovu poslechnout. Nápověda tedy přítomna je, hodnotím jedním bodem.

PhyWiz - Physics Solver Mezi další fyzikálně orientované aplikace patří i *PhyWiz - Physics Solver* [6]. Aplikace má několik částí. V první části si může uživatel nalézt problém, který momentálně řeší, zadat proměnné ze seznamu a aplikace pak vyřeší danou úlohu a popíše i postup řešení. V druhé části je několik stovek slovních úloh kategorizovaných podle tematiky - např. Kinematika, Energie, Optika. Ve třetí části je přítomno fulltextové vyhledávání, pomocí kterého lze nalézt např. řešení slovní úlohy, či znění rovnice.

2. ANALÝZA



Obrázek 2.3: Ukázka výpočtu fyzikálního problému, obraz sejmут z aplikace [6]

Aplikace je pouze v anglickém jazyce, lokalizaci hodnotím nula body. Obsahem jsou řešení fyzikálních úloh, tudíž v kategorii zaměřené na fyziku hodnotím maximem bodů. Úlohy nejsou nikterak gamifikované, slouží pouze jako postup pro vypočtení řešení, gamifikaci tedy hodnotím nula body. Aplikace je dostupná zadarmo, hodnotím ji tedy maximem bodů. Autor aktivně opravuje chyby a přidává nová cvičení, podporu hodnotím jedním bodem. Stejně jako většina aspektů i UI není orientováno na dětského uživatele, hodnotím jej pouze dvěma body. Aplikace sice obsahuje úvodní seznámení s vkládáním řešení, ale žádná další nápověda již přítomna není, nápovědu tedy hodnotím nulou.

Brain It On! - Physics Puzzles Jako poslední aplikaci jsem vybral fyzikální hříčku *Brain It On! - Physics Puzzles*. Ta již obsahuje gamifikované fyzikální jevy v takové míře, že budou pochopitelné i pro cílovou skupinu. Jediné negativum zde vidím v nepříliš atraktivním uživatelském rozhraní pro dětského hráče.



Obrázek 2.4: Ukázka úkolu, obraz sejmут z aplikace [7]

Aplikace obsahuje několik překladů včetně českého, v první kategorii ji tedy hodnotím plným počtem bodů. Aplikace obsahuje fyzikálně zaměřené hádanky, tudíž podporuje výuku fyziky a hodnotím ji maximem bodů. Podobně

jako u aplikace *Jdu do školy* je gamifikace na velmi dobré úrovni, uděluji deset bodů. Aplikace je volně dostupná, taktéž uděluji maximum bodů. Aplikace má aktivní podporu, jež zahrnuje přidávání a úpravu lokalizací, přidávání hádank, či propojení se sociálními sítěmi, hodnotím jedním bodem. UI je přívětivé pro dospělého člověka, každopádně pro dítě není ideální, hodnotím jej šesti body. Aplikace obsahuje rovněž videa s nápovědou k řešení každého zadání, tudíž hodnotím jedním bodem.

2.1.4 Zhodnocení aplikací

Nejlepší nalezenou aplikací je Brain It On!, která vyniká téměř ve všech testovaných kategoriích. Jedinou nevýhodou této aplikace je přílišná složitost pro dětského hráče. Při testování jsem strávil u několika úloh nemalou chvíli přemýšlením o vyhovujícím řešení. Každopádně, ze všech aplikací měla nejlepší systém hodnocení úkolů. Každý úkol se hodnotí podle tří kritérií – počet nakreslených tvarů, čas fyzikální simulace, splnění zadání. Pro dosažení nejvyššího hodnocení je nutnost splnit všechna tři kritéria.

Tabulka 2.1: Hodnocení aplikací

Metrika	Fyzika ve škole	Jdu do školy	PhyWiz	Brain It On!
Lokalizace	5	5	0	5
Výuka fyziky	5	0	5	5
Gamifikace	4	10	0	10
Zpoplatnění	2	4	4	4
Podpora	1	1	1	1
Přívětivost UI	5	9	2	6
Nápověda	0	1	0	1
Součet	22	30	12	32

Pro účely Dráčka III je tedy nutno vybrat některé fyzikální experimenty a gamifikovat je. Podobně, jako tomu je u aplikace *Brain it On!*. Tzn. vytvořit v aplikaci systém pravidel, cílů a odměn za plnění cílů. Aplikaci by měli být schopni hrát všichni žáci základní školy a již od útlého věku si uvědomovat vybrané zákonitosti. Na základě analýzy aplikací jsem si stanovil zajímavé vlastnosti, které by měly obsahovat nové moduly Dráčka:

- Poutavé UI
- Zobrazení výsledku po každém cvičení
- Avatar, který bude hráče doprovázet
- Hravá forma fyzikálního experimentu

2.2 Moduly Dráčka II

Výsledkem Dráčka II bylo přesunutí aplikace Dráček z PC na mobilní zařízení se systémem Android. Struktura aplikace zůstala zachována, celý systém se skládal ze serveru, aplikace s rozhraním pro učitele, klientské aplikace a zásuvných modulů pro klientskou aplikaci. Každý modul se zároveň ještě dále dělí na samotnou hru a editor, ve kterém se budou cvičení pro hru vytvářet a upravovat. Kolegům z Dráčka II se podařilo naimplementovat celkem 11 modulů. Jejich tématica byla následující:

- Michal Bureš – graficky orientované moduly [8]
- Miroslav Mazel – textově orientované moduly [9]

Graficky orientované moduly byly implementovány v jazyce Java pouze s použitím Android SDK. Textově orientované moduly byly implementovány v jazyce Kotlin, pro který má Android rovněž podporu. Na modulech oceňuji rozdělení na sdílenou knihovnu a zdrojové kódy specifické pro každý modul. Každý modul je navíc rozdělen do tří komponent podle návrhového vzoru MVC.

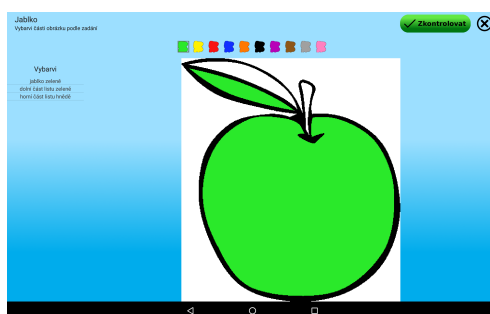
Podobného výsledku bych chtěl docílit při návrhu nových modulů. Navíc bych chtěl dosáhnout takové modularity, která by umožnila výměnu fyzikálního engine za jiný (v případě výkonostních problémů), či zobrazovací vrstvu (např. zobrazování pomocí Android SDK nahradit vlastním řešením pomocí OpenGL ES). Chtěl bych také co nejvíce zjednodušit práci v editoru.

2.3 Uživatelské testování modulů Dráček II

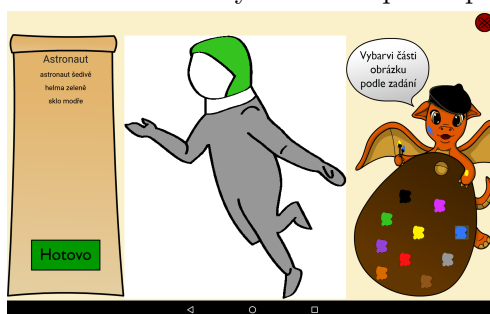
Jako část analýzy požadavků nám posloužilo uživatelské testování tří modulů Dráčka II a jedné veřejně dostupné aplikace z Google Play. Při pokusném testování jsme zjistili, že grafický návrh modulů je nezajímavý a nedokázal by udržet dětskou pozornost. Před finálním testováním bylo potřeba vytvořit nový, poutavější návrh UI a moduly předělat.

Grafický návrh a i samotné provedení za nás provedli studenti předmětu BI-MGA. Na obrázcích 2.5 a 2.6 lze vidět markantní rozdíl v grafickém zpracování. Ve všech modulech vystupuje avatar dráčka, který je v nesnázi a potřebuje pomoci. Podoba avatara je vyhovující dle Lorenzových kritérií roztočilosti [10].

S pomocí dospělého děti vyplnily před testováním a po testování vstupní, resp. výstupní dotazníky (dokument v příloze). Ze vstupních dotazníků jsme se dověděli, že děti dochází do třetí třídy a většina z nich má doma tablet či na něm již někdy nějakou hru hrála. Hlasy pro nejoblíbenější a nejméně oblíbený předmět jsou víceméně rovnoměrně rozdělené. Z vyplněných výstupních dotazníků lze vyčíst, že ačkoliv se moduly dětem líbily a nic by na nich neměnily, tak i přesto jako nejlepší aplikaci z testování jmenovaly aplikaci *Jdu do*



Obrázek 2.5: Modul Vybarvování před úpravou



Obrázek 2.6: Modul Vybarvování po úpravě

školy. Patrně kvůli animovanému avatarovi, který jednotlivé úkoly ve cvičení zadával mluveným slovem a ne textem jako tomu je u Dráčka.

Mimo dotazníky jsme měli možnost celé testování živě sledovat prostřednictvím kamer. Tento vstup nám odhalil více chyb, než děti napsaly do dotazníků. U každých z modulů jsme vypožorovali následující problémy.

2.3.1 Modul Vybarvování

Nedostatečná responzivita Dětem nebylo jasné, která barva z palety je vybrána. Někteří žáci nepočkali na načtení dalšího cvičení a klepnuli znovu na další cvičení, čímž načítané cvičení přeskočili.

Nejasné zadání V obrázcích zbývaly nevybarvené části, děti nevěděly, jak s nimi naložit.

Absence možnosti odbarvení Po vybarvení části, která nebyla v zadání není možnost tuto část zpět odbarvit. Což děti mátklo a myslely si, že celé cvičení pokazily.

Nejasné barvy Barvy nebyly jasně rozlišitelné. Ně kterým dětem dělalo problém rozeznat fialovou a růžovou barvu v paletě.

2.3.2 Modul Hodiny

Nedostatečná tolerance chyb Dětem se nedařilo uchopit hodinové ručičky a těžko se trefovaly na jednotlivé minuty.

Nejasné rozlišení ručiček Děti si pletly hodinovou a minutovou ručičku.

Nevýrazná čísla Děti nebraly v potaz čísla minut na ciferníku, počítaly minuty od nuly, nikoliv od nejbližší nižší části hodiny.

Matoucí posun ručiček Hodinová ručička se posouvá relativně k minutové. V mezních hodnotách se hodinová ručička posunula buď o hodinu méně, nebo více.

Nejasné ovládání Některé děti nevěděly, jak aplikaci ovládat.

2.3.3 Modul Otáčení

Při testování tohoto modulu jsme nenalezli žádné chyby. Zadání každý žák pochopil a cvičení hned vyplnil.

2.3.4 Aplikace Jdu do školy

Ačkoliv se jedná o aplikaci, která je již dostupná na distribuční platformě Google Play a měla by být tedy otestovaná, našli jsme i u této aplikace několik nedostatků.

Zdlouhavé vysvětlování Některá cvičení jsou na první pohled jasná a děti nebaví čekat, až avatar domluví.

Přehnané reakce Avatar reaguje na splnění každého dílčího úkolu a jelikož se stále opakují, stávají se po několika cvičeních otravnými.

2.3.5 Shrnutí testování

Jediným modulem, který uspěl v uživatelském testování byl modul Otáčení. Velikou nevýhodou Dráčka II oproti komerční aplikaci *Jdu do školy* je absence animovaného mluvícího avatara. Avatar srozumitelně vysvětlí celý průběh cvičení, případně ho i na požádání zopakuje. Pro děti byla tato aplikace poutavější, což je u vzdělávací aplikace jednoznačné plus.

2.4 Analýza učiva fyziky pro základní školy

Rámcový vzdělávací program je určen pro výuku žáků druhého stupně, ale do cílové skupiny Dráčka patří i žáci prvního stupně. Moduly tedy musí být přizpůsobeny tak, aby úkoly zvládli i mladší žáci. Dle aktuálního Rámcového vzdělávacího programu [11] je učební osnova následující:

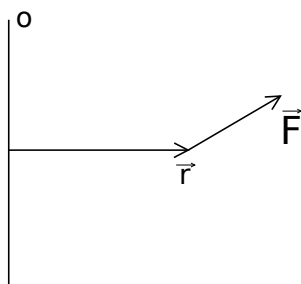
1. Látky a tělesa
 - Měřené veličiny
 - Skupenství látek
2. Pohyb těles; síly
 - Pohyby těles
 - Gravitační pole a gravitační síla
 - Tlaková síla a tlak
 - Třecí síla
 - Výslednice dvou sil stejných a opačných směrů
 - Newtonovy zákony
 - Rovnováha na páce a pevné kladce
3. Mechanické vlastnosti tekutin
 - Pascalův zákon
 - Hydrostatický a atmosferický tlak
 - Archimédův zákon
4. Energie
 - Formy energie
 - Přeměny skupenství
 - Obnovitelné a neobnovitelné zdroje energie
5. Zvukové děje
 - Vlastnosti zvuku
6. Elektromagnetické a světelné děje
 - Elektrický obvod
 - Elektrické a magnetické pole
 - Vlastnosti světla
7. Vesmír
 - Sluneční soustava
 - Hvězdy

Z učební osnovy jsem vybral tři oblasti, podle kterých vytvořím pět modulů do Dráčka III.

2.4.1 Rovnováha na páce

Pro rovnováhu na páce je klíčová rovnost momentů sil na obou stranách páky. Moment síly je podle vzorce $\tau = F \times r$, roven součinu síly a vzdálenosti bodu působení od osy otáčení. [12] Při experimentu *Rovnováha na páce* je snaha vyrovnat momenty sil působících na obou stranách páky. A to tak, že změnou velikosti síly působící na dané rameno, či změnou působíště síly (tedy vzdálenost od osy otáčení).

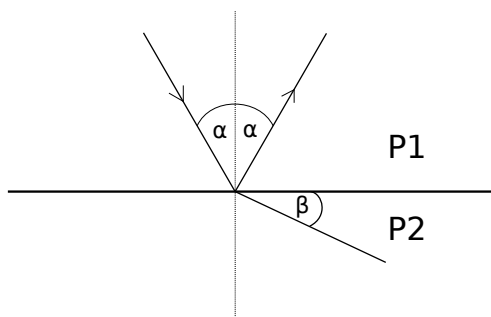
Pro tento experiment jsem navrhl aplikaci, ve které děti budou vyrovnávat páku pomocí rozmístování závaží. Bude možné měnit obě veličiny v rovnici.



Obrázek 2.7: Popis experimentu *Rovnováha na páce*

2.4.2 Vlastnosti světla

Z optických experimentů jsem vybral cvičení na odraz a lom světla. Aplikace by měly demonstrovat význam veličin jako úhel dopadu, úhel odrazu, či index lomu [13]. A to například tak, že by žák měl za úkol zalomit paprsky ze světelného zdroje tak, aby dorazily do světelného senzoru. Zalamovat by je mohl pomocí hranolů, které by umístil do hrací plochy. Hranoly by mohly být z různých materiálů, které by jinak zalamovaly paprsky, a tím by se ukázal význam indexu lomu.

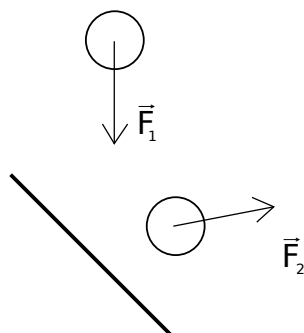


Obrázek 2.8: Popis experimentu *Vlastnosti světla*

Na obrázku 2.8 si lze všimnout úhlu dopadu paprsku α , který je roven úhlu odrazu. Dále je ilustrován úhel lomu β .

2.4.3 Pohyby těles a energie

Z kategorie kinematiky a dynamiky jsem vybral cvičení na rázy (srážky) těles, jelikož tyto experimenty by mohly být dětem blízké. Pomocí aplikací by se děti měly seznámit se zákony zachování hybnosti a mechanické energie [14]. Tyto zákony bych chtěl demonstrovat pomocí kuliček a odrazných ploch. Hráč bude muset postavit odrazné plochy tak, aby kuličky dorazily na požadované místo. Jak samotné kuličky, tak i odrazné plochy budou tuhými tělesy, tudíž se na nich mohou ukázat zmíněné zákony.



Obrázek 2.9: Popis experimentu *Pohyby těles*

2.5 Souhrn požadavků

Na základě analýzy stávajícího řešení, modulu Dráčka II [8], výsledků uživatelského testování a analýzy učebních osnov byly sepsány následující požadavky.

CFR1 Modul dokáže načítat zadání z jádra.

CFR2 Modul zobrazí výsledek po dílčím úkolu.

CFR3 Modul spočítá procentuální hodnocení ze všech zadání a změří čas jejich řešení.

CFR4 Žák může cvičení předčasně ukončit, výsledky rozpracovaného cvičení se neuloží.

CFR5 Editor umožní načíst zadání a uložit jej do jádra aplikace.

CFR6 Editor umožní vytvořit nové zadání.

CFR7 Editor umožní upravit stávající zadání.

CFR8 Editor umožní smazat stávající zadání.

CFR9 Modul bude obsahovat libovolný počet zadání.

CNR1 Modul bude v českém jazyce.

CNR2 Moduly budou implementovány pro platformu Android, verze API 21, Android 5.0 (Lollipop).

CNR3 Hru budou používat převážně žáci prvního stupně, a proto by prostředí mělo být přívětivé a působit nerušivě.

CNR4 Avatar Dráček bude splňovat Lorenzova kritéria [10].

CNR5 Moduly budou implementovány v jazyce Java za pomoci knihovny JBox2D.

Požadavky na modul Rovnováha

FR1.1 Páka bude mít na jedné straně závaží podle zadání.

FR1.2 Žák může vytvářet nová závaží z nabídky.

FR1.3 Žák může přesouvat stávající závaží.

FR1.4 Žák může odstraňovat vytvořené závaží.

NR1.1 Páka musí tolerovat odchylku 5 % při porovnání momentů sil.

Požadavky na modul Odklon paprsku

FR2.1 Na hracím poli bude jeden zdroj světla a jeden senzor.

FR2.2 Žák může vytvářet pouze omezené množství hranolů.

FR2.3 Žák může hranoly otáčet a měnit tak úhel lomu paprsku.

FR2.4 Žák může odstraňovat vytvořené hranoly.

Požadavky na modul Odklon paprsku 2

FR3.1 Na hracím poli bude jeden zdroj světla a jeden senzor.

FR3.2 Žák může hranoly otáčet a měnit tak úhel lomu paprsku.

FR3.3 Žák může odstraňovat hranoly.

Požadavky na modul Odklon kuliček

FR4.1 Na hracím poli bude zdroj kuliček a jedna cílová krabice.

FR4.2 Žák může vytvářet libovolné množství odrazných ploch.

FR4.3 Žák může vytvořené plochy odstraňovat.

Požadavky na modul Odklon kuliček 2

FR5.1 Na hracím poli bude zdroj kuliček a jedna cílová krabice.

FR5.2 Žák může zadané plochy odstraňovat.

2.6 Výběr technologií

V této podkapitole rozeberu použitelné technologie a ze zvoleného seznamu vyberu v každé kategorii jednoho adepta.

2.6.1 Herní engine

Pro vývoj mobilních her je nejjednodušší cestou použití existujícího enginu. Ty poskytují řešení pro většinu složitých problémů, jako jsou fyzikální simulace, kreslení scény, či síťování.

Unity Nejčastěji používaným, dle statistik z [15], je Unity. Nespornou výhodou tohoto řešení je možnost vytvářet obsah hry v připraveném editoru, či podpora více platforem. Velikou nevýhodou je nutnost registrace licence pro každého člověka, který by se chtěl na tvorbě obsahu podílet. Navíc by se do projektu Dráček přidala veliká závislost na knihovny třetí strany.

libGDX Další možností je engine libGDX [16], který je sice open-source, tudíž odpadají starosti s pořizováním licencí. Stejně jako Unity podporuje většinu populárních platforem a to jak mobilních, tak desktopových. Na druhou stranu, stále přináší problémy s integrací. Vzhledem k požadavkům na moduly se většina možností tohoto enginu nevyužije.

Jelikož jsem nechtěl přidávat do projektu Dráček mnoho nových závislostí (mnohdy zbytečných) a cílovou platformou je pouze OS Android, vybral jsem cestu tvorby vlastních technologií za pomoci podpůrných knihoven.

2.6.2 Knihovny pro fyzikální simulaci

Box2D Oblíbený 2D fyzikální engine napsaný jazyce C++ a publikovaný pod zlib licencí. Podporuje fyziku tuhých těles, klouby a dokáže rychle řešit kolize. Samotnou knihovnu Box2D používá i engine libGDX [17].

JBox2D Port Box2D do Javy [18].

Chipmunk Cross-platform fyzikální engine, který vyniká svým výkonem (převyšuje Box2D). Je napsaný v jazyce C a má podporu pro většinu dnes používaných platforem [19].

Z vyjmenovaných knihoven jsem vybral JBox2D, jelikož je napsaný v jazyce Java, ve kterém budou implementovány moduly.

2.6.3 Ostatní knihovny

Android SDK Android SDK je velký balík nástrojů pro vývoj aplikací, obsahující mimo jiné knihovny, debugger, obrazy disků jednotlivých verzí Android a ukázkové zdrojové kódy. Vývoj SDK jde ruku v ruce s vývojem samotného Androidu. Knihovny obsahují implementaci veškerých důležitých komponent, jako např. renderer, správce zdrojů, či prostředky pro správu notifikací. Aplikace Dráček i její moduly toto SDK využívají, tudíž kvůli zachování závislostí toto SDK využijí i pro své moduly.

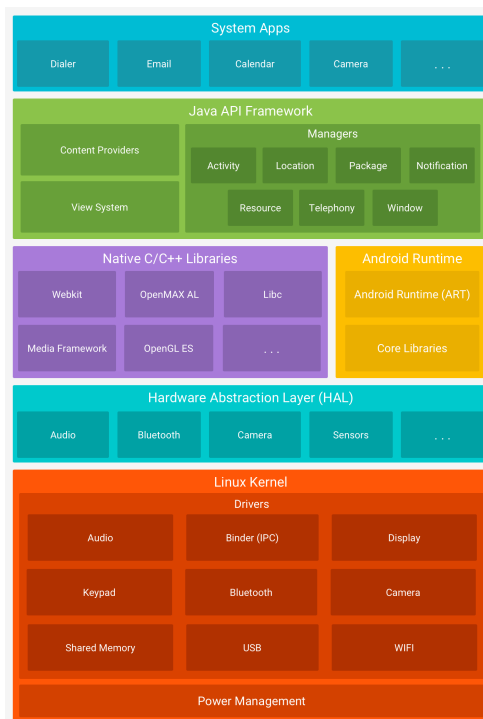
2.6.4 Jazyk

Android podporuje vývoj aplikací v jazycích C/C++ a Java dle [20].

C/C++ C a C++ jsou low-levelové jazyky používané především v aplikacích, kde je výkon největší prioritou. Pro takové případy poskytuje Google Android NDK, pomocí kterého lze přistupovat k nativním C/C++ knihovnám, viz 2.10. Psaní takových aplikací lze zjednodušit frameworky jako například SDL [21].

Java Java je objektově orientovaný jazyk zaručující velkou míru přenositelnosti. Javový kód je nejprve zkompileován do bytecodeu a ten je pak spouštěn na JVM (Java Virtual Machine). Android Java API poskytuje většinu funkcionality Java API. Konkrétně nyní má Android API naimplementovány pouze některé funkcionality z Javy 8.

Vzhledem k výběru SDK Android jsem nucen použít jazyk Java. Každopádně, vzhledem k požadavkům aplikace si nemyslím, že se moduly potýkaly s problémy týkajícími se výkonu.



Obrázek 2.10: Softwarová architektura OS Android, převzato z [1]

Návrh

V této kapitole nejprve rozeberu návrh knihovny společné pro veškeré moduly a v podkapitolách návrh jednotlivých modulů.

3.1 Společná knihovna

Pro návrh společné knihovny jsme se rozhodli společně s kolegy ze dvou důvodů:

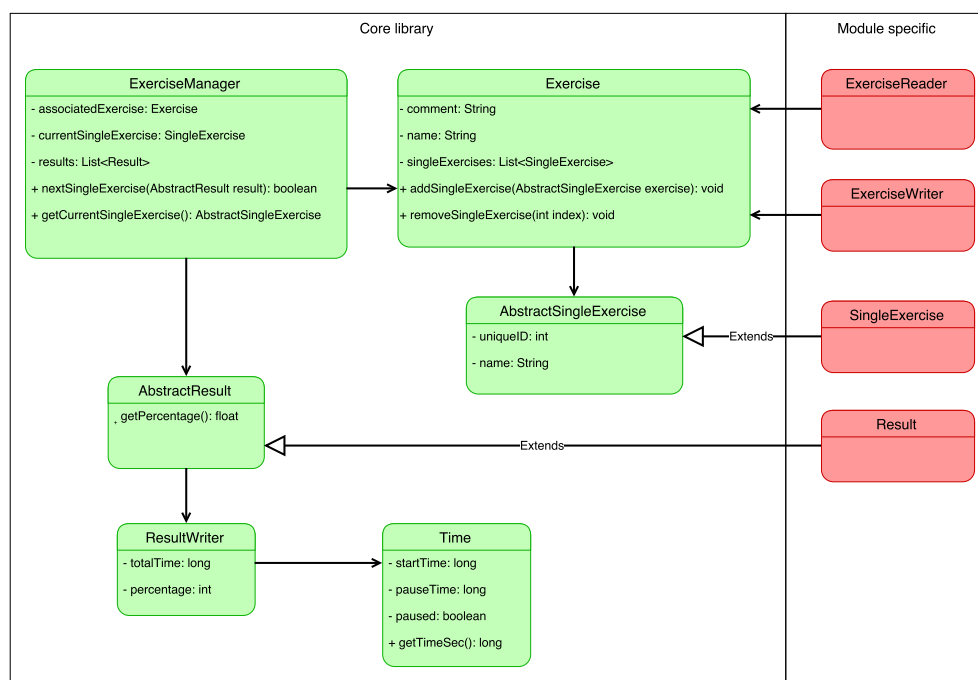
1. Vytvoří se jednotné rozhraní pro komunikaci s klientskou aplikací. Každý modul pak pouze použije vytvořené objekty, čímž si každý z nás ušetří práci.
2. Zjednodušení a zpřehlednění implementace. Komunikace s klientskou aplikací nebude v každém modulu vytvořena jinak, nýbrž jednotně. Což zjednoduší případná budoucí rozšíření, nebo změny.

Při návrhu knihovny jsme vycházeli z existujícího návrhu Michala Bureše [8]. Ten navrhl knihovnu, ze které čerpaly všechny jeho moduly. Při návrhu jsme se rozhodli pro dvouvrstvou architekturu, jelikož pro účely modulů je snazší. Třívrstvá architektura by byla pro účely modulů zbytečnou komplikací, většinu dat spravuje klientská aplikace, modul vždy pouze požádá o data k příslušnému cvičení.

3.1.1 Aplikační vrstva

V této vrstvě je přítomna správa dat a interface pro tvorbu konkrétních zadání pro každý modul. Na diagramu 3.1 lze vidět, že každý modul musí implementovat třídy `ExerciseReader`, `ExerciseWriter` pro načtení, resp. uložení zadání, `SingleExercise` pro uložení dat dílčího cvičení, `AbstractResult` pro uchování výsledků. Mezi jednotlivými cvičeními lze přepínat pomocí třídy `ExerciseManager`, která má rovněž údaje o výsledcích. Třída `ResultWriter` je rovněž v

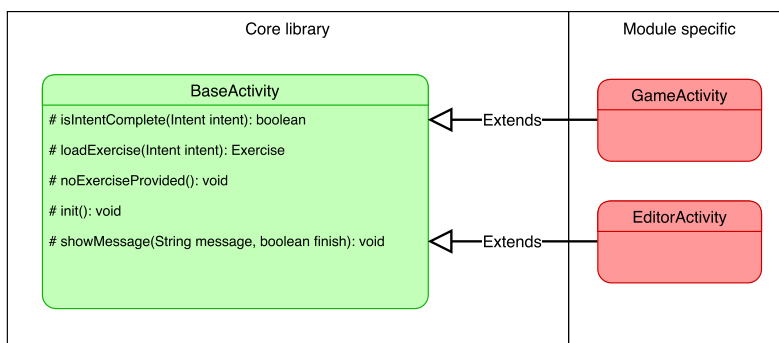
3. NÁVRH



Obrázek 3.1: Třídní diagram aplikační vrstvy

knihovně, jelikož formát celkového výsledku je jednotný. Výsledkem cvičení je vždy procentuální hodnocení a uplynulý čas.

3.1.2 Prezentací vrstva



Obrázek 3.2: Třídní diagram Prezentací vrstvy

V prezentací vrstvě knihovny je pouze abstraktní třída BaseActivity, která zajišťuje pouze základní kontrolu příchozího zadání. Od ní pak v jednotlivých modulech dědí Android Aktivita pro hru a pro editor. Více o Aktivitách v kapitole 3.3.

3.2 Moduly Dráčka III

Každý modul bude mít stejnou skladbu jako moduly z Dráčka II. Moduly se budou skládat z editoru a cvičení. Pomocí editoru je možno vytvořit nové cvičení. Cvičení bude obsahovat více zadání, každé zadání bude mít pouze jeden cíl – splnit fyzikální experiment se zadanými parametry. Cvičení budou zjednodušena natolik, aby je mohli hrát i mladší žáci základní školy.

Chování editorů

- Kliknutí na tlačítko Uložit → uložení dílčího cvičení
- Kliknutí na tlačítko Uložit celek → uložení všech cvičení a ukončení editoru
- Kliknutí na položku v seznamu uložených cvičení → přepnutí na uložené cvičení
- Kliknutí na tlačítko Smazat → smazání vybraného cvičení
- Kliknutí na nové cvičení v seznamu → smazání všech objektů na scéně a příprava pro tvorbu nového dílčího cvičení

3.2.1 Modul Rovnováha

Modul Rovnováha vychází z experimentu rovnováha na páce. Na herní ploše bude přítomna páka s podstavcem. Na jedné z polovin páky bude pevně dané závaží, které musí hráč vyrovnat. Jednotlivá závaží se budou vybírat z nabídky, ve které budou přítomna závaží různých hmotností. Pro snažší manipulaci se závažími je bude aplikace při jejich posunu přichytávat k mřížce. Výsledné skóre se bude počítat podle počtu použitých závaží. Čím menší počet závaží, tím vyšší skóre.

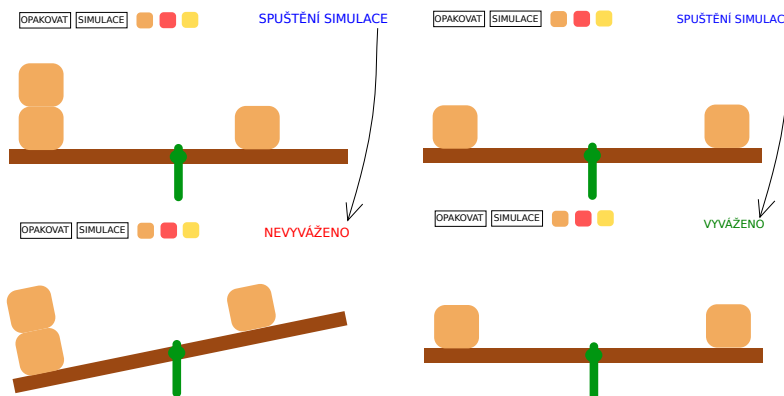
Hlavním problémem při návrhu je vyhodnocení řešení, nabízí se zde hned několik možností. První možností je dotázat se páky, na působící veličiny a vyhodnotit podle nich řešení. Tato možnost je nejjednodušší a lze pomocí ní vyřešit jakékoliv rozmístění závaží. Dokonce by podporovalo různé velikosti a netriviální tvary závaží.

Druhou možností řešení je spočítat jednotlivé momenty na obou stranách páky a porovnat je. Takové řešení je netriviální, přestože známe hmotnosti závaží a jejich rozmístění, musíme počítat i s jejich plochou (nelze je aproximovat jako hmotný bod). Zvolil jsem tedy první možnost.

Dalším problémem je možné obejít řešení. A to například tím, že hráč podepře závažím rameno páky, tudíž se při simulaci páka nevychýlí, ačkoliv nebude vyvážená. Tato situace bude vyřešena zamezením manipulace v prostorech pod pákou. Dalším omezeným prostorem bude rameno páky, kde

3. NÁVRH

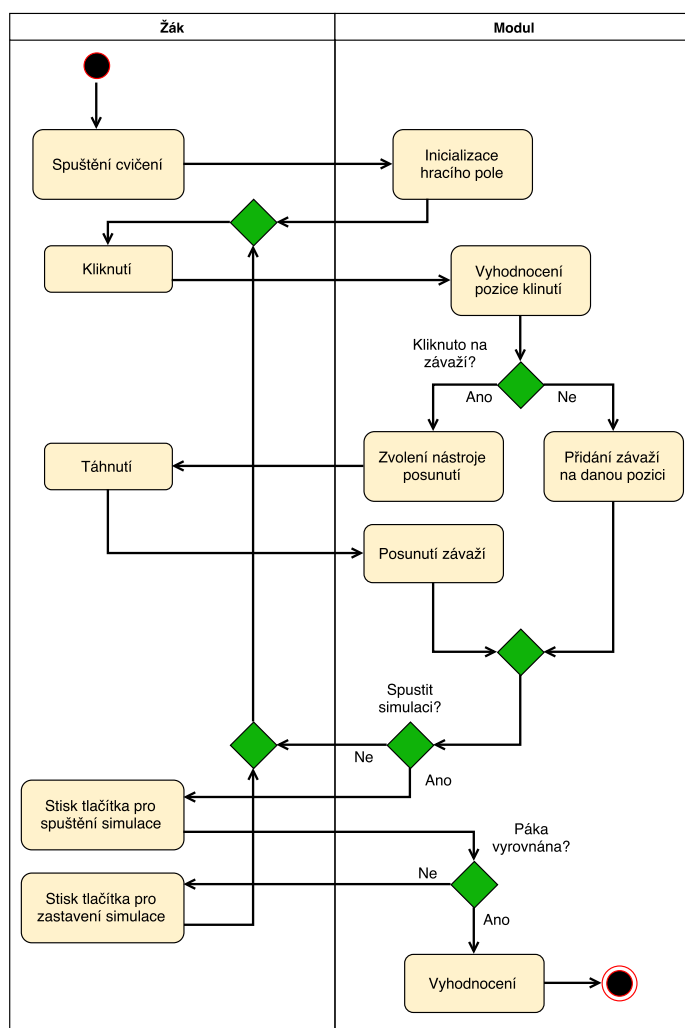
budou umístěna zadaná závaží, hráč nebude smět na toto rameno žádné závaží umístit.



Obrázek 3.3: Wireframe modulu Rovnováha

Chování hry

- Kliknutí do prostoru vlevo nad pákou → přidání nového závaží
- Kliknutí mimo výše zmíněnou oblast → žádná reakce
- Kliknutí na vytvořené závaží a následné táhnutí → přesun závaží
- Táhnutí mimo zvýšenou oblast → zastavení o stanovené hranice
- Kliknutí na zvolené závaží v nabídce → žádná reakce
- Kliknutí na nezvolené závaží v nabídce → zvolení nového závaží
- Kliknutí na tlačítko simulace v editačním módu → spuštění simulace, vyhodnocení
- Kliknutí na tlačítko simulace v simulačním módu → zastavení simulace, vrácení objektů zpět do pozice v jaké byly před spuštěním simulace
- Kliknutí na tlačítko znovu → odstranění vytvořených závaží a návrat všech objektů na původní místa
- Kliknutí na tlačítko zpět → ukončení cvičení
- Přesunutí závaží mimo obrazovku → smazání závaží
- Vyhodnocení jako vyhovující řešení → umožnění přechodu na další úkol
- Vyhodnocení jako nevhovující řešení → vynucení opravy



Obrázek 3.4: Diagram aktivit modulu Rovnováha

Editor Editor bude sdílet s hrou stejný subsystém. Závaží lze umisťovat na obě strany páky. Na pravé straně nalezneme zadání a na levé referenční řešení, podle kterého se vypočítá hodnocení řešení žáka.

3.2.2 Modul Odklon paprsku

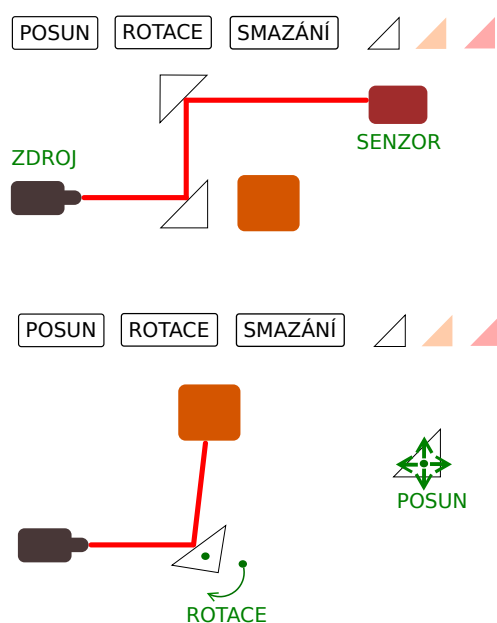
Modul Odklon paprsku spojuje hned několik fyzikálních experimentů z optiky. Cílem bude dostat světelný paprsek ze zdroje do světelného senzoru. Na hrací mapě bude umístěno několik neprůsvitných překážek, okolo kterých hráč musí postavit odrazné plochy tak, aby se paprsek zalomil až do světelného senzoru. Výsledné skóre se počítá podle počtu použitých odrazných hranolů. Čím menší počet hranolů, tím lepší skóre.

3. NÁVRH

Hlavním problémem při návrhu tohoto modulu je kolize paprsku s prostředím. Tento problém spolehlivě vyřeší raycasting, který je implementován v knihovně JBox2D.

Dále bylo potřeba vyřešit přidávání, odebrání a úpravu herních objektů. První možností je zobrazit malé menu po kliku na objekt. Tuto možnost jsem si nezvolil, protože je složitější na implementaci, a hráč si rukou zakrývá možnosti při stisku objektu. Tento způsob nemusí být pro děti optimální.

Druhou možností je vytvořit paletu nástrojů například v horní části obrazovky a mezi jednotlivými nástroji přepínat. Takové rozhraní by bylo dle mého názoru pro děti pochopitelnější.



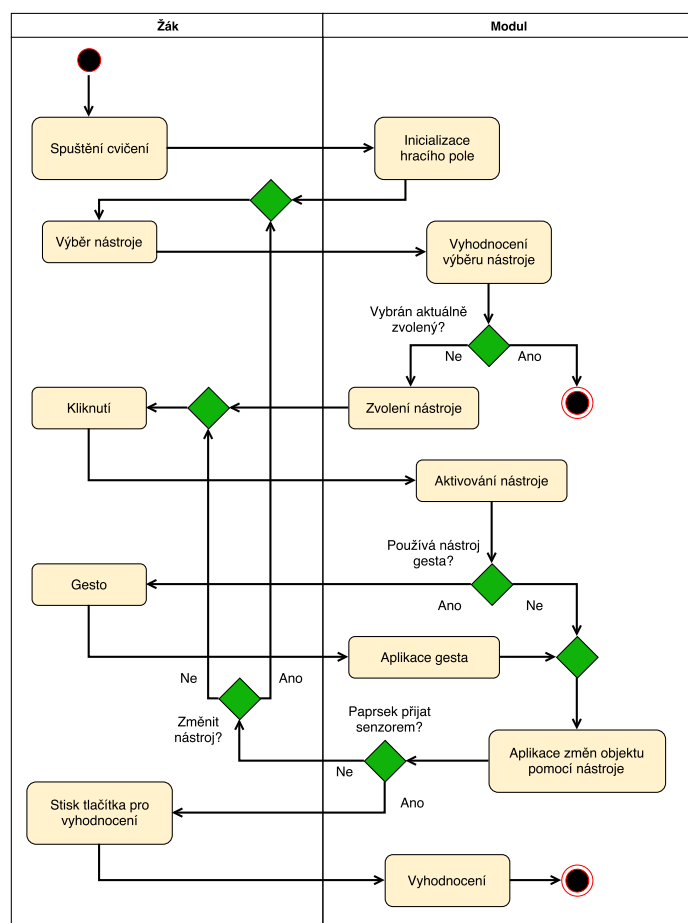
Obrázek 3.5: Wireframe modulu Odklon paprsku

Chování hry

- Kliknutí na hranol v nabídce → vybrání hranolu
- Kliknutí do volného místa po vybrání hranolu z nabídky → přidání vybraného hranolu
- Kliknutí na objekt jiný než hranol v jakémkoliv režimu → žádná reakce
- Kliknutí na hranol s nástrojem otáčení, dále posun → otočení hranolem o úhel daný kruhovou výsečí
- Kliknutí na hranol s nástrojem posunutí, dále posun → posunutí hranolu na danou pozici

- Kliknutí na hranol s nástrojem mazání → smazání hranolu
- Kliknutí na tlačítko otáčení → přepnutí do režimu otáčení
- Kliknutí na tlačítko posunutí → přepnutí do režimu posunutí
- Kliknutí na tlačítko smazání → přepnutí do režimu mazání
- Kliknutí na tlačítko, které přepne na aktuálně zvolený nástroj → žádná reakce
- Kliknutí na zvolený hranol v nabídce → žádná reakce
- Kliknutí na nezvolený hranol v nabídce → zvolení nového hranolu
- Paprsek se dotkne senzoru → umožnění vyhodnocení
- Kliknutí na tlačítko vyhodnotit → vyhodnocení, přechod na další cvičení
- Kliknutí na tlačítko zpět → ukončení cvičení

3. NÁVRH



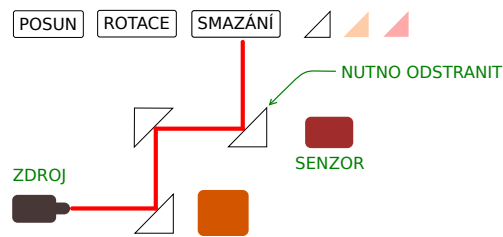
Obrázek 3.6: Diagram aktivit modulů Odklon paprsku

Editor V editoru bude možné umisťovat překážky, hranoly, senzory a zdroje paprsků. Jednotlivé překážky lze libovolně umístit. Pro každé zadání musí platit, že obsahuje alespoň jeden zdroj paprsku a stejné množství senzorů, tudíž editor musí být schopen tuto situaci kontrolovat. Dle počtu hranolů se určí referenční řešení.

3.2.3 Modul Odklon paprsku 2

Modul Odklon paprsku 2 je modifikací předchozího modulu se stejným cílem. V tomto modulu však již budou na mapě rozmístěny hranoly a hráčovým úkolem bude pouze některé z nich odstranit, nebo natočit. Výpočet skóre je shodný s předchozím modulem.

Chování hry Chování hry je podobné s předchozím modulem s výjimkou přidávání nových hranolů, ty již přidávat nelze.



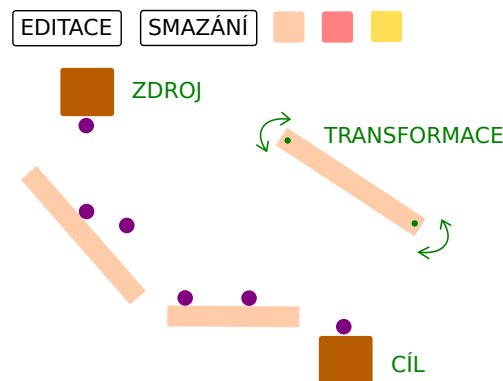
Obrázek 3.7: Wireframe modulu Odklon paprsku 2

Editor Editor je podobný tomu, který je součástí předchozího modulu. Změní se pouze způsob zadávání referenčního řešení. Mimo hranoly sloužící k řešení musí být v zadání i hranoly, které bude muset žák ve cvičení odstranit.

3.2.4 Modul Odklon kuliček

Modul Odklon kuliček demonstruje gravitaci a rázy (srážky) těles. Cílem bude dostat všechny kuličky ze zdroje na vyvýšeném místě do nory umístěné níže. Na mapě bude umístěno několik překážek a hráč musí vytvořit plošky tak, aby se kuličky odrazily do nory. Výsledné skóre se bude počítat podle obsahu odrazných ploch vytvořených hráčem. Čím menší obsah, tím lepší skóre, navíc pro řešení za plný počet bodů musí platit, že do cíle doputují všechny kuličky.

U tohoto modulu je hlavní problém v tvorbě a mazání odrazných ploch. Jejich velikost se při vytváření bude nastavovat gesty. Podle vybraného tvaru se vytvoří fyzikální reprezentace.



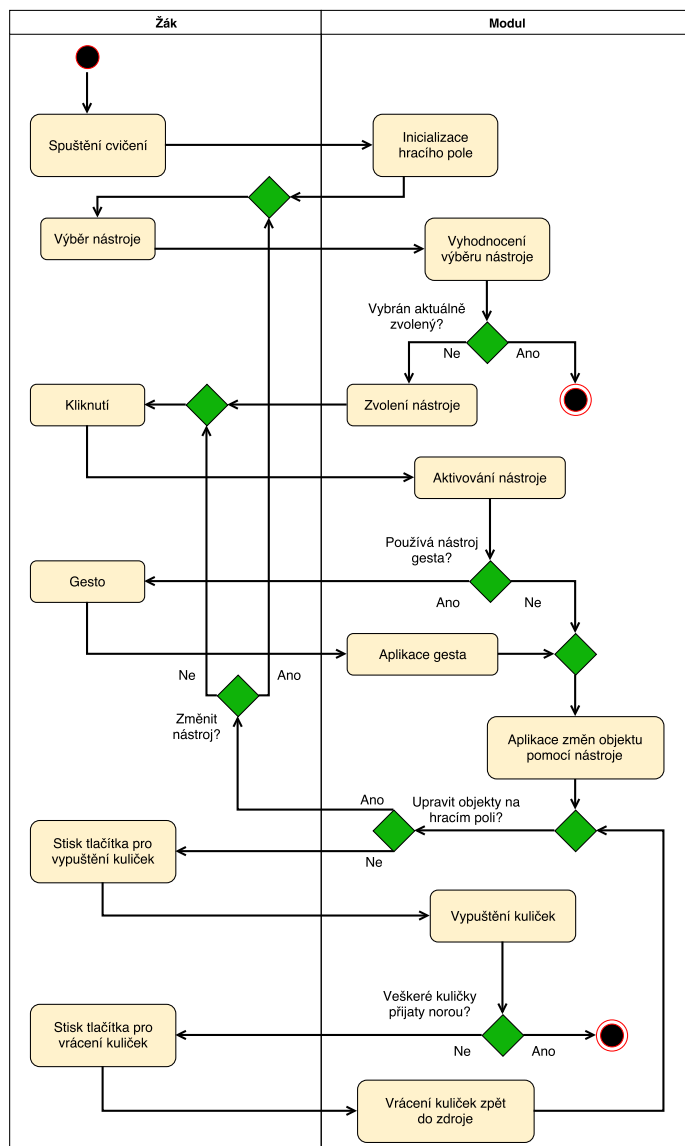
Obrázek 3.8: Wireframe modulu Odklon kuliček

Chování hry

- Kliknutí na tlačítko přidávání → aktivování nástroje přidávání
- Kliknutí na tlačítko odebrání → aktivování nástroje odebrání

3. NÁVRH

- Kliknutí na tlačítko modifikace → aktivování nástroje modifikace
- Kliknutí na tlačítko, které přepne na aktuálně zvolený nástroj → žádná reakce
- Položení prstu do volného místa s nástrojem přidávání → přidání plošky
- Položení prstu na plošku s nástrojem modifikace → začátek úpravy plošky
- Táhnutí prstem od počátečního místa → zvětšování plošky daným směrem
- Táhnutí prstem k počátečnímu místu → zmenšování plošky
- Zvednutí prstu → vytvoření fyzikální reprezentace
- Kliknutí na tlačítko spustit → vypuštění kuliček ze zdroje
- Kulička se dotkne cílové nory → odstranění kuličky
- Kulička spadne z hrací plochy → odstranění kuličky
- Odstranění všech kuliček → vyhodnocení a přechod na další cvičení
- Kliknutí na tlačítko zpět → ukončení cvičení

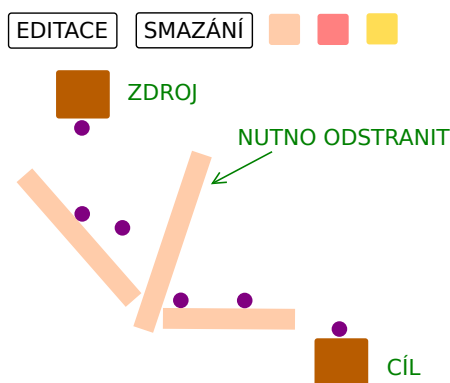


Obrázek 3.9: Diagram aktivit modulů Odklon kuliček

Editor V editoru bude možno přidávat zdroj kuliček, cílovou noru, překážky a odrazné plochy. Překážky se budou přidávat stejným mechanismem jako odrazné plochy. Každé zadání musí mít zdroj kuliček a cílovou noru, tudíž editor musí tuto podmínku kontrolovat. Do UI editoru přibude oproti hře nová nabídka, která bude obsahovat zdroj, noru a překážky.

3.2.5 Modul Odklon kuliček 2

Modul Odklon kuliček 2 je modifikací předchozího modulu se stejným cílem. V tomto modulu budou vytvořeny odrazné plochy a hráč je bude muset pouze odmazávat. Výpočet skóre je shodný s předchozím modulem.



Obrázek 3.10: Wireframe modulu Odklon kuliček 2

Chování hry Chování hry je stejné předchozímu modulu s výjimkou přidávání a úpravy nových plošek, ty už bude možno pouze odmazávat.

Editor V editoru přibude oproti předchozímu modulu možnost vytvářet odrazné plošky, které nebudou v referenčním řešení. Chování je jinak stejné.

3.3 Použité komponenty Android SDK

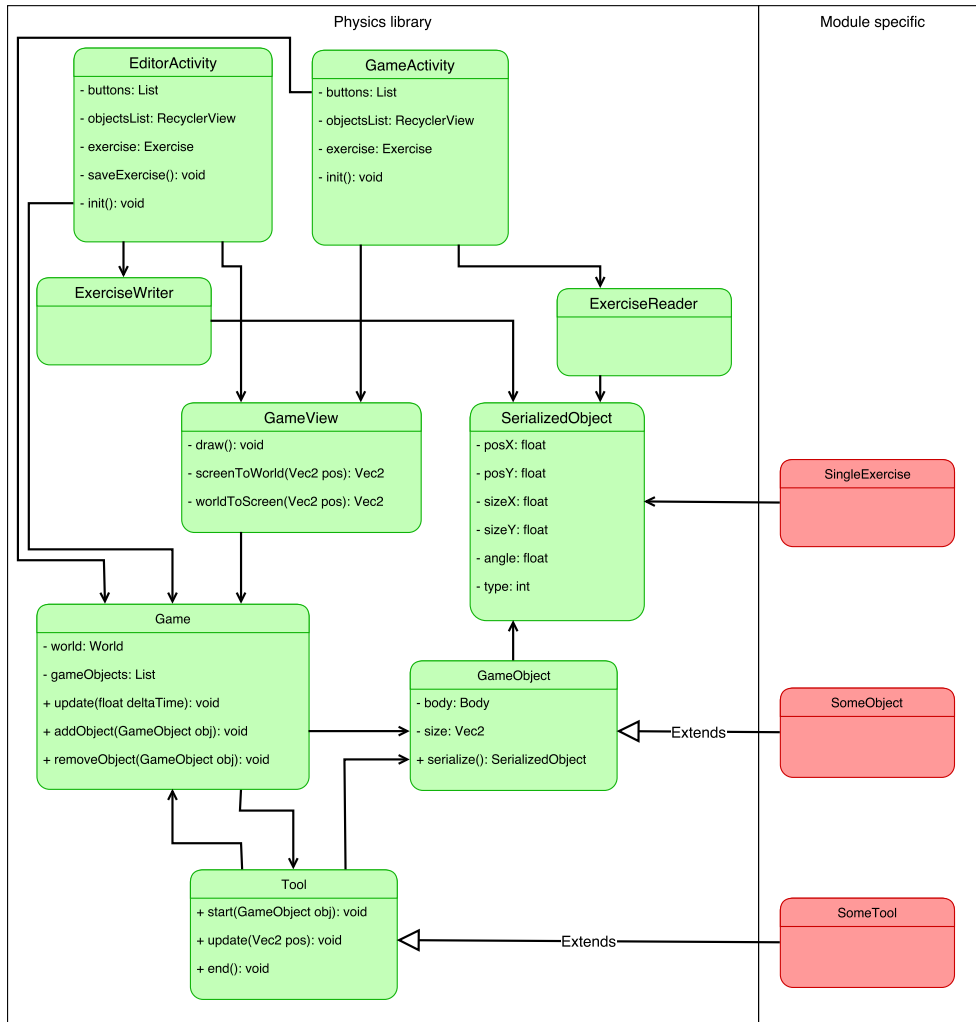
V této sekci popíši použité komponenty Android SDK, které jsem při návrhu využil. Jedná se především o prvky GUI, jelikož se většina herní logiky odehrává v interních třídách a funkcích fyzikální knihovny.

Activity Aktivita má na starosti vytvoření okna, do kterého pak uživatel vkládá UI prvky. Aktivit může být v aplikaci několik, lze mezi nimi přepínat, každopádně, hlavní aktivita musí být zapsána v manifestu jako main.

Button/ImageButton Klasická tlačítka, která na sobě mohou mít buď pouze text, nebo i obrázek. K tlačítkům lze přihlásit listenery, tedy funkce, které se zavolají po stisknutí tlačítka.

Recycler Recycler je UI element zobrazující zadaný seznam objektů. Pro každý zobrazovaný objekt je třeba vytvořit vlastní layout, a pak pomocí vlastní třídy Adapter napojit data uživatele s implementací recycleru.

3.4 Třídní diagramy modulů



Obrázek 3.11: Společné třídy modulů

ExerciseReader Načítá zadání ze zprávy zasláné klientskou aplikací

ExerciseWriter Zpracuje objekty v editoru a uloží je do xml

GameActivity Sbírá data ze vstupů a posílá je třídě Game, nastavuje zadání dle aktuálního SingleExercise

EditorActivity Sbírá data ze vstupů a posílá je třídě Game

GameView Zobrazuje objekty třídy Game, obstarává správné škálování obsahu dle rolišení

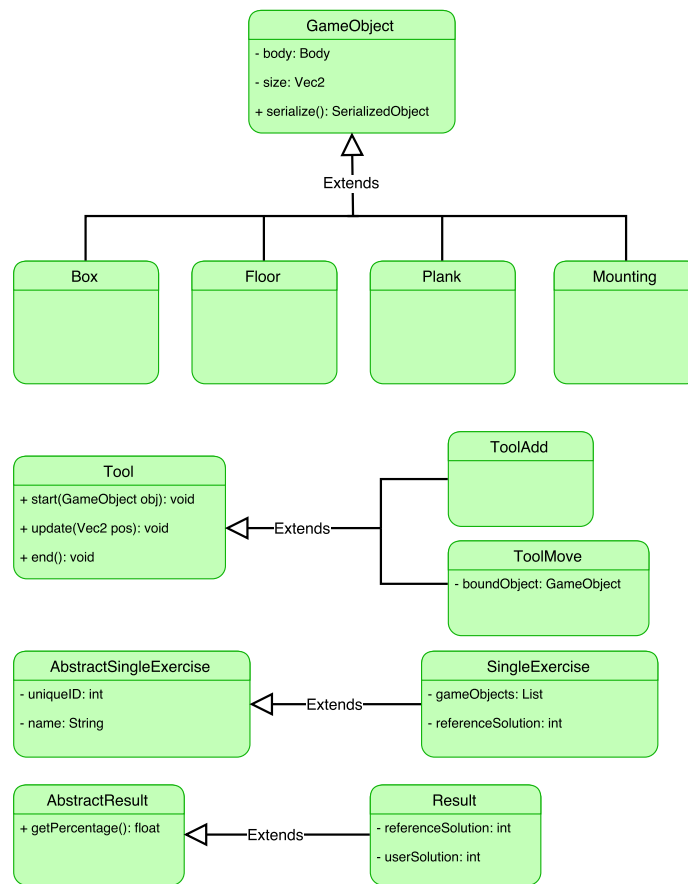
3. NÁVRH

Game Obsahuje logiku modulu – přidávání, odebírání objektů, jejich interakce, reakce na vstupy

GameObject Abstraktní třída obsahující informace a chování jednotlivých herních objektů

Tool Abstraktní třída pro jednotlivé nástroje, pomocí kterých se upravují, přidávají, odebírají objekty

Společné třídy budou přítomny v knihovně pro moduly orientované na výuku fyziky. Moduly tedy budou závislé na této knihovně a knihovně společné pro všechny moduly.



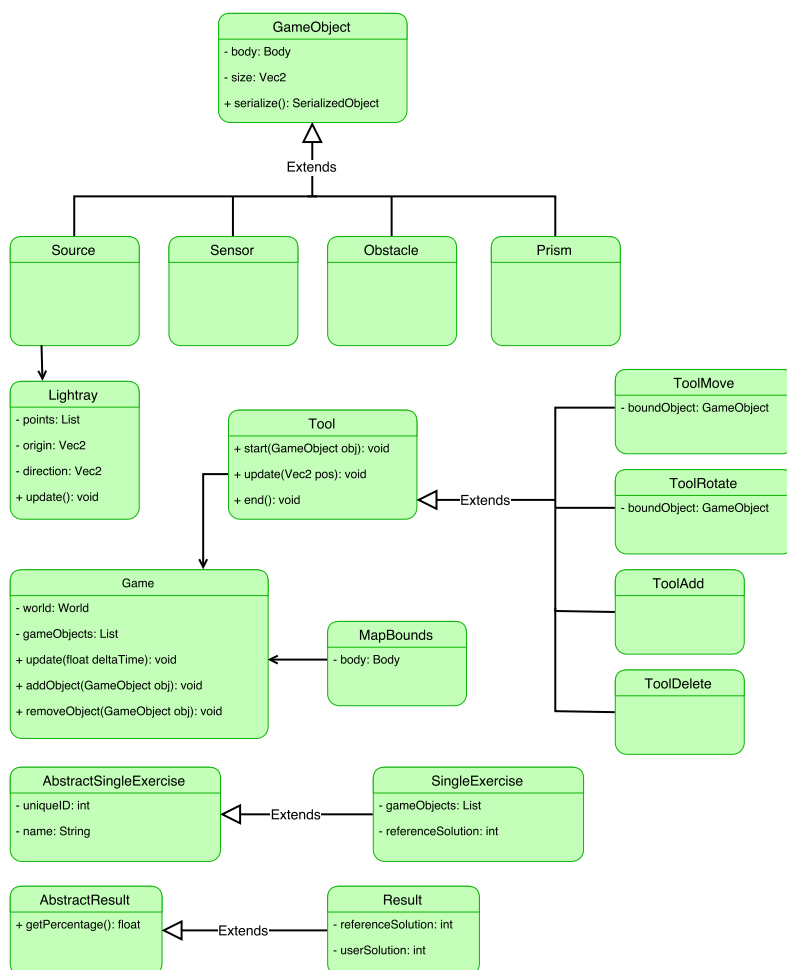
Obrázek 3.12: Třídní diagram modulu Rovnováha

Diagram pro modul Rovnováha

ToolAdd Přidá zvolený objekt do scény

ToolMove Posouvá objekt po scéně

3. NÁVRH



Obrázek 3.13: Třídní diagram modulů Odklon paprsku

Diagram pro moduly odklonu paprsku

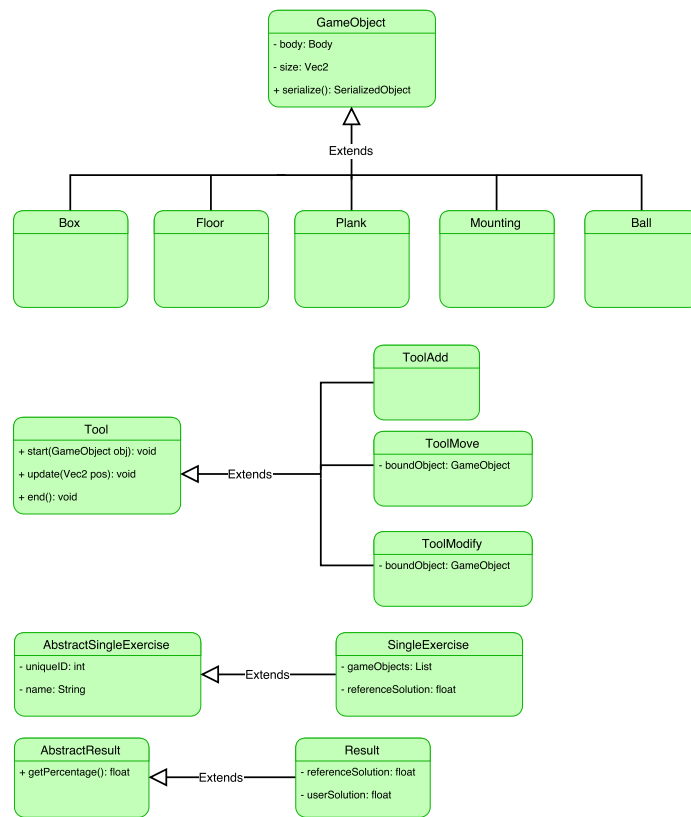
ToolRotate Otočí zvolený objekt o úhel mezi počátečním a koncovým bodem tahu prstu

ToolDelete Smaže zvolený objekt ze scény

MapBounds Ohraničuje scénu neviditelnou stěnou – zabraňuje vystoupení objektů z obrazovky

Lightray Třída reprezentující světelný paprsek – obsahuje body zlomu a barvu

Ostatní třídy mají stejnou funkcionalitu jako předchozí moduly.



Obrázek 3.14: Třídní diagram modulů Odklon kuliček

Diagram pro moduly odklonu kuliček

Source Zdroj kuliček – vytváří objekty typu Ball

Sink Nora pro kuličky – odstraní objekty typu Ball po kontaktu se Sink a zvýší čítač přijatých kuliček

Surface Odrazná plocha pro kuličky

ToolModify Otočí a změní rozměr objektu ve směru tahu objektu

Ostatní třídy mají stejnou funkcionalitu jako předchozí moduly.

Implementace

V této kapitole nejdříve shrnu důležitá rozhodnutí při implementaci, dále popíši instalaci a ovládání v instalační, respektive uživatelské příručce a postup pro modifikaci či rozšíření.

4.1 Společná architektura modulů

Komunikace mezi jednotlivými moduly a klientskou aplikací je pevně specifikovaná, proto jsme se s kolegy Jaroslavem Rybou a Ondřejem Slabým rozhodli vytvořit společnou knihovnu, která bude tuto funkcionalitu zajišťovat. Jako základ jsme použili knihovnu Michala Bureše [8], který ji využíval pro sdílení vybraných tříd mezi moduly. Knihovna nyní zajišťuje zapisování výsledků a provádí základní kontroly přijatého zadání. Mimo to jsme se rozhodli sdílet i vybrané prvky UI, například dialog s výsledným hodnocením cvičení.

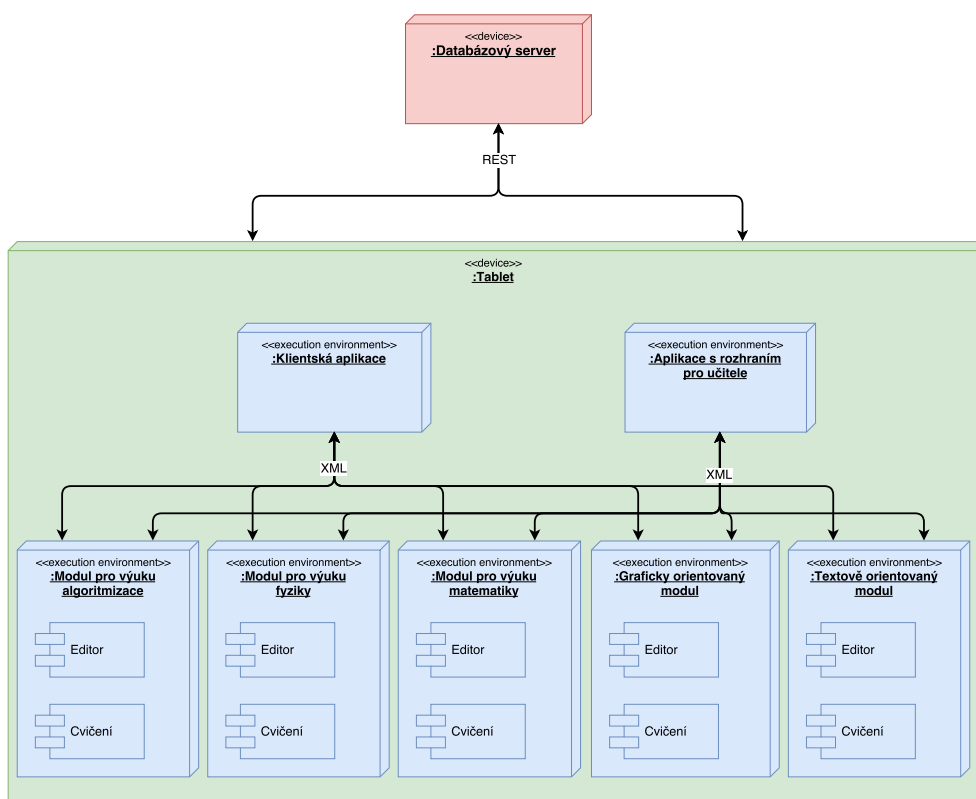
Největší výzvou v implementaci byla integrace fyzikální knihovny. Jelikož v manuálu knihovny doporučují aktualizovat objekty s konstantní periodou, napadlo mě jako první řešení vytvořit časovač, který bude s jistou periodou vyvolávat aktualizaci objektů. V průběhu implementace jsem zjistil, že toto řešení je velmi pracné, protože časovač běží na vlastním vlákne a vzniká tím mnoho kritických sekcí. Později jsem od tohoto řešení upustil, jelikož s rostoucím počtem implementovaných funkcionalit se stále více objevovaly kritické sekce a bohužel i pády aplikace. Tyto nedostatky jsem vyřešil přesunutím vykreslování, herní logiky i aktualizace fyzikálního subsystému na jedno vlákno. Aktualizace fyzikálních objektů se provádí v hlavní smyčce vždy po uplynutí pevně daného časového úseku.

Dále bylo nutné zavést přepočty ze souřadnic v soustavě obrazu do soustavy, ve které bude vypočítáván fyzikální model. V první řadě je třeba přepočítat souřadnice z obrazovky v pixelech do density independant pixels, dále pak do světových souřadnic. Knihovna JBox2D používá metr, kilogram a sekundu jako měrné jednotky. Přepočty z pixelů na metry si lze zvolit dle vlastního uvážení.

4. IMPLEMENTACE

V poslední řadě bylo třeba rozhodnout, jak se bude sdílet zdrojový kód pro moduly. Pro sdílení společné struktury jsem vytvořil knihovnu identickým způsobem jako tu pro sdílení mezi všemi moduly. Jelikož dvojice modulů odklonu kuliček a paprsků budou sdílet velkou část logiky i binárních dat, rozhodl jsem je umístit do jednoho projektu. Každý modul bude mít vlastní aktivitu a rozdíly v herní logice vyřeším přepisováním.

4.2 Schéma nasazení



Obrázek 4.1: Schéma nasazení aplikace Dráček

4.3 Omezení vyvinutého prototypu

Většina navržené funkcionality byla implementována. Všechny požadavky byly splněny, až na avatara, který vzhledem k nedostatku času chybí. Prototypy pouze potřebují vytvořit poutavější UI a přehodnotit používání nástrojů v závislosti na uživatelském testování. Více o uživatelském testování viz kapitola 5.4.

4.4 Instalační příručka

V této podkapitole popíší instalaci vývojového prostředí pro tvorbu nových modulů. Je třeba nainstalovat a přidat do projektu následující:

AndroidStudio Oficiální Android IDE spravované samotným vývojářem platformy Android, Google. Jeho instalaci lze provést instalátorem staženým z webových stránek pro Android vývojáře, či v případě instalace na platformě Linux, stažením z oficiálních repozitářů.

Android SDK Moduly byly naprogramovány na SDK verze 25, pro sestavení je tedy nutné mít nainstalovanou verzi API 25.0.2 a kompilátor gradle verze 2.3.1. Minimálním podporovaným SDK je 21, což odpovídá Android verzi 5.0 (Lollipop).

JBox2D Fyzikální knihovna, jejíž výběr jsem zdůvodnil v analýze. V repozitáři knihovny jsou dostupné jak zdrojové kódy, tak i výsledný jar soubor, který musíme nakopírovat do adresáře **libs** v adresářové struktuře projektu. Dále je třeba přidat závislost pro kompilátor. A to přidáním následující řádky do sekce **dependencies** souboru **build.gradle**.

```
compile files ( 'libs/jbox2d-library.jar ' )
```

Společná knihovna modulů Repozitář knihovny lze nalézt na GitLab FIT ČVUT, stačí jej naklonovat do adresáře s projekty (jedná se o samostatný Android projekt) a přidat následující konfiguraci. Soubor **settings.gradle** by měl vypadat následovně:

```
include ':app', ':corelibrary '
project ( ':corelibrary ' ). projectDir =
    new File ( settingsDir , '../DracekModulesCore/corelibrary ' )
```

Dále do souboru **build.gradle** je třeba přidat závislost, podobně jako tomu je u knihovny JBox2D a zapnout data binding:

```
dataBinding {
    enabled = true
}

dependencies {
    compile project ( ':corelibrary ' )
}
```

Společná fyzikální knihovna Knihovna je dostupná na přiloženém médiu vedle zdrojových souborů modulů. Konfigurace fyzikální knihovny je velmi podobná konfiguraci u společné knihovny. Pokud bude modul využívat i společnou knihovnu, soubor **settings.gradle** by měl vypadat takto:

```
include ':app', ':physicslibrary', ':corelibrary'
project(':physicslibrary').projectDir =
    new File(settingsDir, '../DracekPhysicsCommon/physicslibrary')
project(':corelibrary').projectDir =
    new File(settingsDir, '../DracekModulesCore/corelibrary')
```

Do souboru **build.gradle** je třeba přidat další závislost:

```
compile project(':physicslibrary')
```

4.5 Uživatelská příručka

4.5.1 Instalace

V návrhu klientské aplikace Dráčka se počítá s instalací modulů přímo z aplikace, která by měla být ke stažení z Google Play. Momentálně je ovšem klientská aplikace ve fázi prototypu a tato funkcionality podléhá testování, proto se musí moduly instalovat manuálně.

Při manuální instalaci je nutné nejprve zkontrolovat verzi systému Android. Podporovány jsou verze od 5.0 (Lollipop). Dále je třeba povolit instalaci aplikací z neznámých zdrojů, což lze provést v nastavení systému. Dále již stačí otevřít apk soubor modulu a potvrdit instalaci. Každý modul obsahuje jedno testovací cvičení a přístup do editoru. Bez klientské aplikace se cvičení vytvořená v editoru ani výsledky neukládají.

4.5.2 Společné ovládání při tvorbě zadání

Editor pro tvorbu zadání lze spustit z aplikace pro učitele. Objekty ve scéně lze vytvářet stejným způsobem jako ve cvičení. Po výběru objektu z nabídky v horní části obrazovky se objekt umístí kliknutím na libovolné místo. Po sestavení cvičení je nutné ověřit podmínku splnitelnosti, tedy je-li cvičení splnitelné. Podmínka splnitelnosti se v každém modulu liší. Například v modulu odklonu paprsku musí být na obrazovce umístěn zdroj i senzor. Jakmile je podmínka splněna je možné vytvořené cvičení uložit, a to tlačítkem *Uložit*. Nově vytvořené cvičení se přidá do seznamu cvičení v levé části obrazovky. Mezi jednotlivými cvičeními lze přepínat kliknutím na číslo v seznamu. Smazání vybraného cvičení se provede tlačítkem *Smazat*. Celá sada cvičení se uloží tlačítkem *Uložit celek*. Po stisku tlačítka pro uložení se zadání odešle aplikaci pro učitele, která jej uloží na server. Po uložení se editor ukončí.



Obrázek 4.2: Obrazovka editoru modulu Rovnováha

4.5.3 Modul Rovnováha

V modulu *Rovnováha* je cílem vyrovnat závaží umístěné na pravé straně pomocí co nejmenšího počtu závaží na straně levé, která lze vybírat z nabídky v horní části obrazovky. Po kliknutí na závaží je možné jej vytvořit kliknutím do prázdného místa. Kliknutím na již existující závaží a následným táhnutím se objekt přesune na pozici dotyku. Závaží, které spadne mimo obrazovku bude smazáno. Kontrolu vyvážení lze spustit i zrušit tlačítkem *Simulace*. Pokud je závaží vyrovnáno, objeví se na obrazovce ohodnocení řešení, které lze přijmout, nebo si jej vylepšit. Počet pokusů není penalizován. Pokud je nutné cvičení spustit znovu, lze tak učinit stiskem tlačítka *Opakovat*.

Nástroje

Přidání Po vybrání závaží z nabídky jej lze přidat do scény kliknutím na volné místo.

Posunutí Po kliknutí a následném tažení závaží jej lze posunout.



Obrázek 4.3: Modul Rovnováha

Editor

Při tvorbě zadání je třeba přidat závaží na obě strany páky. Na pravé straně bude umístěno zadání a na straně levé referenční řešení, podle kterého se bude hodnotit řešení žáka. Vyrovnání páky lze zkontrolovat stisknutím tlačítka *Simulace*. Pokud bude páka vyrovnána je možno dílčí cvičení uložit.

4.5.4 Modul Odklon paprsku

V modulu *Odklon paprsku* je cílem zalomit paprsek tak, aby mířil do senzoru světla. Zalomat paprsek lze pomocí hranolů, které vytváříme dle nabídky v horní části obrazovky. Při řešení úkolů budou překážky zastíňovat paprsek. Okolo nich je třeba rozestavět odrazné hranoly. Přidání a úpravu hranolů popíší dále. Jakmile je paprsek namířen na senzor, umožní se vyhodnocení řešení, které spustíme tlačítkem *Vyhodnotit*. Opět můžeme řešení i po vyhodnocení bez penalizací upravit.

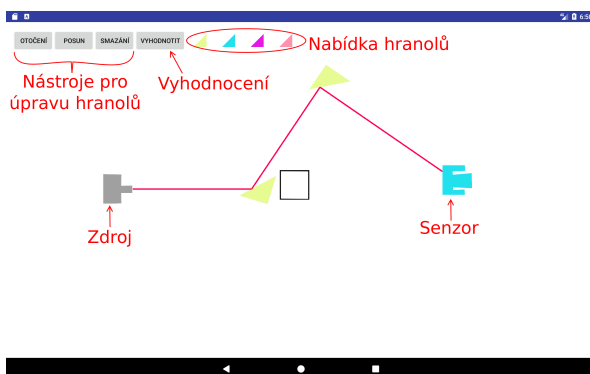
Nástroje

Přidání Po vybrání objektu z nabídky jej lze přidat do scény kliknutím na volné místo.

Posunutí Po kliknutí a následném tažení objektu jej lze posunout.

Otočení Po kliknutí na objekt a posunutím prstu okolo hranolu jej lze otočit.

Smazání Po kliknutí na objekt jej lze smazat.



Obrázek 4.4: Modul Odklon paprsku

Editor

Tvorba zadání je opět velmi podobná samotnému cvičení. Každé zadání musí mít zdroj světla a senzor. Jakmile budou objekty na scéně rozestavěny

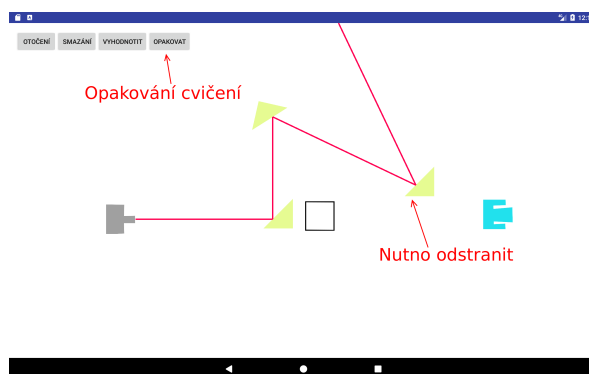
tak, že paprsek míří na senzor, lze spustit vyhodnocení. Vyhodnocení se provede stiskem tlačítka *Vyhodnotit*. Pokud bude zadání korektní, můžeme ho uložit. Počet odrazných hranolů bude brán jako referenční řešení.

4.5.5 Modul Odklon paprsku 2

Modul *Odklon paprsku 2* je pouze modifikací předchozího modulu. Cíle jsou stejné, nyní však není možné hranoly přidávat, lze je pouze odebírat, nebo s nimi otáčet.

Nástroje

Smazání Po kliknutí na hranol jej lze smazat.



Obrázek 4.5: Modul Odklon paprsku 2

Editor

Editor je taktéž podobný předchozímu. Po kontrole zadání (paprsek míří na senzor a stisknuto tlačítka *Vyhodnotit*) je nutné přepnout editor z režimu referenčního řešení kliknutím na tlačítka *Referenční řešení*. Dále přidat další hranoly, či upravit stávající. Žák tak bude muset některé odstranit, nebo otočit, aby dosáhl cíle. Po úpravách stačí cvičení uložit stiskem tlačítka *Uložit*.

4.5.6 Modul Odklon kuliček

V modulu *Odklon kuliček* je cílem dostat do nory všechny kuličky, které vypustí zdroj. Nora je tvaru písmene „U“, tudíž kuličky do ní spadnou pouze shora (pokud není nijak natočená). Pro tvorbu cesty slouží odrazné plošky, u kterých lze měnit velikost (pouze v jednom směru). Po sestavení cesty je možné vyzkoušet řešení stiskem tlačítka *Spuštění*. Pokud do nory dorazí všechny kuličky, řešení se ohodnotí dle referenčního řešení. Řešení se řídí dle součtu obsahu všech plošek. Čím menší obsah, tím lepší hodnocení.

4. IMPLEMENTACE

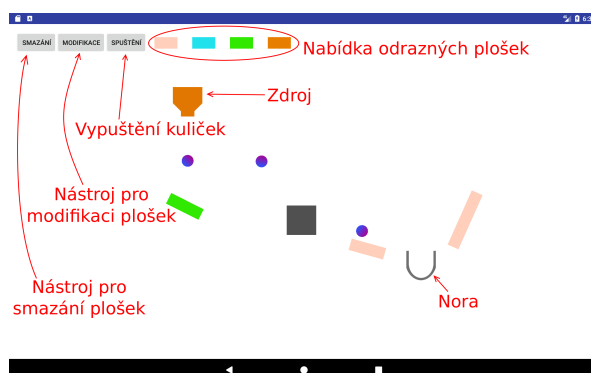
Nástroje

Přidání Po vybrání objektu z nabídky jej lze přidat do scény kliknutím na volné místo.

Modifikace Po kliknutí na objekt a posunutím prstu se objekt protáhne, či zkrátí ve směru pohybu prstu.

Smazání Po kliknutí na objekt jej lze smazat.

Posunutí Po kliknutí a následném tažení objektu jej lze posunout. Nástroj je přítomen pouze v editoru.



Obrázek 4.6: Modul Odklon kuliček

Editor

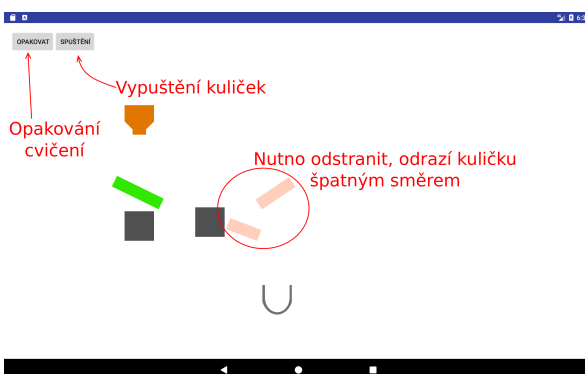
Editor má velmi podobné použití jako u předchozích modulů. Po přidání zdroje, nory a odrazných plošek lze otestovat splnitelnost kliknutím na tlačítko *Spuštění*. Cvičení je možné uložit, pokud všechny kuličky doputují do nory. Vytvořené odrazné plochy se započítávají jako referenční řešení.

4.5.7 Modul Odklon kuliček 2

Modul *Odklon paprsku 2* je modifikací předchozího modulu. Cíle jsou stejné, avšak žák může odrazné plošky pouze mazat.

Nástroje

Smazání Po kliknutí na hranol jej lze smazat.



Obrázek 4.7: Modul Odklon kuliček 2

Editor

Editor je téměř identický s předchozím. Po vytvoření a kontrole zadání (všechny kuličky spadly do nory a stisknuto tlačítko *Vyhodnotit*) je nutné přepnout editor z režimu referenčního řešení kliknutím na tlačítko *Referenční řešení*. Dále je nutné přidat další odrazné plochy.

4.6 Návod změny grafického obsahu

Pro změnu textur objektů stačí pouze nahradit soubory v adresáři:

```
app/src/main/res/drawable/
```

Pro změnu velikosti objektů je nutno změnit konstanty v jednotlivých potomcích třídy `GameObject`. Vzhled a rozložení tlačítek lze měnit v editoru `AndroidStudio`. Jednotlivé layouty nalezneme v adresáři:

```
app/src/main/res/layout/
```

Testování

Vzhledem k povaze navrhnutých aplikací jsem se rozhodl moduly testovat unit testy, manuálním testováním a v poslední řadě uživatelským testováním dětmi ze základní školy.

5.1 Unit testy

Pomocí jednotkových (unit) testů jsem se rozhodl testovat funkcionalitu důležitých tříd a jejich funkcí. Google doporučuje testovat třídy, které nemají žádné, nebo pouze jednoduché závislosti na Android pomocí takzvaných lokálních unit testů. Tyto testy se spouští na stanici vývojáře. Dále existují instrumented testy, kterými se testují třídy se závislostmi na Android. Instrumented testy jsou naopak spouštěny na fyzických zařízeních, nebo v emulátorech.

Využil jsem pouze lokálních unit testů, jelikož při instrumented testech kompilátor hlásil chyby v použité knihovně JBox2D. K testování jsem použil Googlem doporučené frameworky JUnit a Mockito [22]. Testy mi pomohly s nalezením kritických chyb, které se týkaly funkčnosti, ale i chyb v návrhu. Mezi takové prohřešky patřily především závislosti na Android komponenty, jelikož kvůli nim se testy nedařilo zkompileovat.

5.2 Integrační testy

Integrační testy kontrolují komunikaci mezi komponentami. Vzhledem k tomu, že jsem integroval fyzikální knihovnu přímo do herní logiky, testuji integraci i v unit testech. Takový způsob integrace však není žádoucí, jelikož zakomponování knihovnických funkcí do logiky znemožňuje jednoduchou záměnu v případě zjištění problémů s knihovnou.

5.3 Systémové testování

Při systémovém testování se aplikace ověřuje jako celek. Po celou dobu implementace jsem prováděl testování modulů za účelem ověření funkcionality dle požadavků. Testy zahrnovaly logování důležitých hodnot a průchod aplikací nasazené na fyzickém stroji a emulátoru.

Mezi další systémové testy patřilo zkušební nasazení klientské a adminské aplikace s moduly. Obě aplikace však vyžadovaly úpravy. Musela se změnit adresa serveru, která byla pevně zakomponována ve zdrojovém kódu. Dále bylo nutné změnit zasílání hesla uživatele z hashe na plaintext, jelikož to tak server vyžadoval. Upravený kód lze nalézt na Gitlab FIT ČVUT. Více o úpravách serveru napsal ve své práci Ondřej Slabý [2].

Každý modul má navíc vytvořenou aktivitu pro simulaci jádra, která spouští cvičení s testovacím zadáním a editor.

5.4 Uživatelské testování

Nejdůležitějšími testy jsou uživatelské. Jelikož automatické testy kontrolují správnost implementace dle návrhu, ale nedokáží zhodnotit, jestli jsou pro uživatele přívětivé. Bude-li chtít uživatel aplikaci používat. Zejména je třeba provádět tento typ testů u dětí, protože mají jiné nároky než dospělí.

Vzhledem k tomu, že testovacími subjekty budou žáci základní školy, kteří musí být uvolněni z výuky, budou se testovat všechny moduly v jeden den. Z tohoto důvodu jsme se rozhodli vytvořit jednotné materiály pro testování. Testování začne vstupními dotazníky, z kterých zjistíme, jaký má žák vztah k jednotlivým předmětům ve škole a jaké má zkušenosti s ovládáním tabletu. Výsledek dotazníku určí, které moduly bude žák testovat. Dále bude testování pokračovat dle scénářů, které slouží moderátorovi k popisu jednotlivých modulů a postup při řešení zadání. Po projití scénářů žák s moderátorem vyplní výstupní dotazník, ve kterém popíše své dojmy z testovaných modulů.

Materiály k testování byly vytvořeny pomocí Google Docs, které umožňují dotazníky vyplňovat elektronicky a výsledky exportovat do spreadsheetu. Složka s materiály je dostupná zde: <https://goo.gl/iD0aB9>

5.5 Výsledky testování

Po dohodě s vedoucím práce uživatelské testování proběhne z organizačních důvodů až po odevzdání práce. Výsledky budou dostupné v termínu obhajoby práce.

Závěr

Cílem práce bylo analyzovat stávající řešení problematiky, navrhnout vhodné řešení a dle návrhu jej implementovat a otestovat. Při analýze jsem zjistil, že aktuálně dostupné aplikace neodpovídají požadavkům, protože nejsou přizpůsobeny žákům prvního stupně základních škol. Základní školy vyučují dle Rámcového vzdělávacího programu pro základní vzdělávání. V závislosti s rámcovým vzdělávacím programem, jsem navrhl pět zásuvných modulů aplikace Dráček včetně editorů zadání, které budou přizpůsobeny dětskému publiku. Fyzikální simulace jsem implementoval pomocí knihovny JBox2D, a to především kvůli jednoduchosti integrace na OS Android. Veškeré moduly byly otestovány automatickými a systémovými testy, avšak uživatelské testy proběhnou až po odevzdání práce.

Při automatických a systémových testech jsem narazil na následující problémy:

- Fyzikální knihovna je pevně propojena s herní logikou, v případě zjištění komplikací nebude jednoduché fyzikální knihovnu zaměnit
- Fyzikální model v modulu Odklon kuliček není dostatečně odladěn, při vypuštění většího počtu kuliček se aplikace značně zpomalí
- GUI není poutavé

Na projektu Dráček již pokračují další studenti, kteří pracují na zdokonalení stávající struktury. Budoucnost projektu vidím v odstranění všech zmíněných nedostatků serveru, klientské aplikace a aplikace s rozhraním pro učitele. Navíc by mohly moduly sledovat více metrik, v současné době se získává pouze výsledné skóre a čas řešení. Dále je třeba navrhnout jednotný grafický vzhled, který by vzbuzoval dojem uceleného projektu, protože momentálně jsou moduly vzhledově velmi různorodé.

Jsem přesvědčen, že po odstranění všech nedostatků bude projekt Dráček důstojným soupeřem již existujících aplikací.

Literatura

- [1] Google Inc.: *The Android software stack*. [online]. [cit. 2017-01-01]. Dostupné z: https://developer.android.com/guide/platform/images/android-stack_2x.png
- [2] Slabý, O.: *Zásuvné moduly aplikace Dráček III - výuka základů algoritmi-zace*. České vysoké učení technické v Praze, Fakulta informačních technologií, 2017.
- [3] Ryba, J.: *Zásuvné moduly aplikace Dráček III - výuka matematiky*. České vysoké učení technické v Praze, Fakulta informačních technologií, 2017.
- [4] Vaščák, V.: *Fyzika ve škole LITE – Aplikace pro Android ve službě Google Play*. [online]. [cit. 2017-03-11]. Dostupné z: <https://play.google.com/store/apps/details?id=air.cz.moravia.vascak.physicsatschool>
- [5] TechSophia: *Jdu do školy! Výuková hra CZ – Aplikace pro Android ve službě Google Play*. [online]. [cit. 2017-03-11]. Dostupné z: <https://play.google.com/store/apps/details?id=cz.techsophia.schoolreadiness>
- [6] Muggr: *PhyWiz - Physics Solver – Aplikace pro Android ve službě Google Play*. [online]. [cit. 2017-03-11]. Dostupné z: <https://play.google.com/store/apps/details?id=xyz.muggr.phywiz.calc>
- [7] Orbital Nine Games: *Brain It On! - Physics Puzzles – Aplikace pro Android ve službě Google Play*. [online]. [cit. 2017-02-20]. Dostupné z: <https://play.google.com/store/apps/details?id=com.orbital.brainiton>
- [8] Bureš, M.: *Výuková aplikace Dráček II - zásuvné moduly II*. České vysoké učení technické v Praze, Fakulta informačních technologií, 2016.
- [9] Mazel, M.: *Dragon II - plugins I*. České vysoké učení technické v Praze, Fakulta informačních technologií, 2016.

- [10] Podaný, P.: *Animovaný avatar pro výukovou aplikaci Dráček*. České vysoké učení technické v Praze, Fakulta informačních technologií, 2016.
- [11] Národní ústav pro vzdělávání: *RVP pro základní vzdělávání*. [online]. [cit. 2017-03-13]. Dostupné z: <http://www.nuv.cz/t/rvp-pro-zakladni-vzdelavani>
- [12] Reichl, J.; Všeticka, M.: *Moment síly vzhledem k ose otáčení :: MEF*. [online]. [cit. 2017-02-20]. Dostupné z: <http://fyzika.jreichl.com/main.article/view/84-moment-sily-vzhledem-k-ose-otaceni>
- [13] Reichl, J.; Všeticka, M.: *Lom (refrakce) světla :: MEF*. [online]. [cit. 2017-03-15]. Dostupné z: <http://fyzika.jreichl.com/main.article/view/438-lom-refrakce-svetla>
- [14] Reichl, J.; Všeticka, M.: ****Rázy (srážky) těles :: MEF*. [online]. [cit. 2017-03-15]. Dostupné z: <http://fyzika.jreichl.com/main.article/view/52-razy-srazky-teles>
- [15] Unity Technologies: *Unity - Fast Facts*. [online]. [cit. 2017-01-02]. Dostupné z: <https://unity3d.com/profiles/unity3d/themes/unity/images/company/pr/unity-vision-mobile-graph.jpg>
- [16] Badlogic Games: *libgdx*. [online]. [cit. 2017-01-02]. Dostupné z: <https://libgdx.badlogicgames.com/features.html>
- [17] Catto, E.: *Box2D | A 2D Physics Engine for Games*. [online]. [cit. 2017-03-05]. Dostupné z: <http://box2d.org/>
- [18] Murphy, D.: *JBox2D: A Java Physics Engine*. [online]. [cit. 2017-01-02]. Dostupné z: <http://www.jbox2d.org/>
- [19] Howling Moon Software: *Chipmunk2D Physics*. [online]. [cit. 2017-03-05]. Dostupné z: <http://chipmunk-physics.net>
- [20] Google Inc.: *Platform Architecture | Android Developers*. [online]. [cit. 2017-01-01]. Dostupné z: <https://developer.android.com/guide/platform/index.html>
- [21] Lantinga, S.: *Simple DirectMedia Layer - Homepage*. [online]. [cit. 2017-01-04]. Dostupné z: <https://libsdl.org/>
- [22] Google Inc.: *Building Local Unit Tests | Android Developers*. [online]. [cit. 2017-05-05]. Dostupné z: <https://developer.android.com/training/testing/unit-testing/local-unit-tests.html>

Seznam použitých zkratk

UI User interface

GUI Graphical user interface

SDK Software development kit

XML Extensible markup language

Obsah přiloženého média

	readme.txt	stručný popis obsahu média
	exe	adresář se spustitelnou formou implementace
	src	
	impl	zdrojové kódy implementace
	thesis	zdrojová forma práce ve formátu L ^A T _E X
	text	text práce včetně příloh
	BP_Štěpán_Jaroslav_2017.pdf	text práce ve formátu PDF
	annexes	přílohy práce