

ZÁPADOČESKÁ UNIVERZITA V PLZNI

FAKULTA PEDAGOGICKÁ

KATEDRA VÝPOČETNÍ A DIDAKTICKÉ TECHNIKY

**POROVNÁNÍ POČÍTAČOVÝCH HER
ZAMĚŘENÝCH NA VÝUKU PROGRAMOVÁNÍ**

BAKALÁŘSKÁ PRÁCE

PETR TOBIÁŠ

Informatika se zaměřením na vzdělávání

Vedoucí práce: Mgr. Jan Král

Plzeň, 2017

Prohlašuji, že jsem bakalářskou práci vypracoval samostatně
s použitím uvedené literatury a zdrojů informací.

V Plzni, 12. dubna 2017

.....
vlastnoruční podpis

Poděkování

Rád bych poděkoval Mgr. Janu Královi za odborné vedení, cenné rady a vstřícnost při vytváření bakalářské práce.

OBSAH

SEZNAM ZKRATEK A POJMŮ	3
ÚVOD	4
CÍLE PRÁCE	6
1 POPIS HLAVNÍCH ZNAKŮ HER PRO VÝUKU PROGRAMOVÁNÍ	7
1.1 DEFINICE HRY	7
1.2 HLAVNÍ ZNAKY HER PRO VÝUKU PROGRAMOVÁNÍ	9
1.3 FUNKCE HRY	10
1.4 KRITÉRIA VÝBĚRU	11
1.5 HRY NESPLŇUJÍCÍ KRITÉRIA.....	11
2 POPIS VYBRANÝCH POČÍTAČOVÝCH HER ZAMĚŘENÝCH NA VÝUKU PROGRAMOVÁNÍ.....	13
2.1 HUMAN RESOURCE MACHINE.....	13
2.1.1 Popis hry	13
2.1.2 Zaměření hry.....	14
2.1.3 Uživatelská přívětivost.....	14
2.1.4 Možnost nápovědy	14
2.2 CODEMONKEY	15
2.2.1 Popis hry	15
2.2.2 Zaměření hry.....	15
2.2.3 Uživatelská přívětivost.....	15
2.2.4 Možnost nápovědy	16
2.3 CODECOMBAT	17
2.3.1 Popis hry	17
2.3.2 Zaměření hry.....	18
2.3.3 Uživatelská přívětivost.....	18
2.3.4 Možnosti nápovědy	19
2.4 CODINGAME.....	20
2.4.1 Popis hry	20
2.4.2 Zaměření hry.....	20
2.4.3 Uživatelská přívětivost.....	20
2.4.4 Možnosti nápovědy	21
2.5 CHECKIO.....	22
2.5.1 Popis hry	22
2.5.2 Zaměření hry.....	22
2.5.3 Uživatelská přívětivost.....	22
2.5.4 Možnosti nápovědy	24
2.6 EMPIRE OF CODE	25
2.6.1 Popis hry	25
2.6.2 Zaměření hry.....	25
2.6.3 Uživatelská přívětivost.....	25
2.6.4 Možnosti nápovědy	26
2.7 CODE KINGDOMS.....	27
2.7.1 Popis hry	27
2.7.2 Zaměření hry.....	27
2.7.3 Uživatelská přívětivost.....	27
2.7.4 Možnosti nápovědy	29
2.8 TIS-100.....	30

2.8.1	Popis hry	30
2.8.2	Zaměření hry.....	30
2.8.3	Uživatelská přívětivost.....	30
2.8.4	Možnosti nápovědy	31
2.9	SCREEPS.....	32
2.9.1	Popis hry	32
2.9.2	Zaměření hry.....	32
2.9.3	Uživatelská přívětivost.....	32
2.9.4	Možnosti nápovědy	33
2.10	CODE WARRIORS.....	34
2.10.1	Popis hry	34
2.10.2	Zaměření hry.....	35
2.10.3	Uživatelská přívětivost.....	35
2.10.4	Možnosti nápovědy	36
3	POROVNÁNÍ VYBRANÝCH HER	38
3.1	JAZYK HRY.....	38
3.2	ZAMĚŘENÍ HRY	38
3.2.1	Začátečníci	38
3.2.2	Pokročilí	39
3.3	PROGRAMOVACÍ JAZYKY.....	40
3.3.1	Forma programovacího jazyka	40
3.3.2	Způsob zadávání	42
3.4	POMĚR HRY A PROGRAMOVÁNÍ.....	42
4	NÁVRH ROZŠIŘUJÍCÍCH PŘÍKLADŮ PRO VYBRANOU HRU.....	46
4.1	VÝBĚR HRY.....	46
4.2	NÁVRH PŘÍKLADU	47
4.3	REALIZACE PŘÍKLADU	48
	ZÁVĚR.....	50
	RESUMÉ.....	51
	SEZNAM LITERATURY	52
	SEZNAM OBRÁZKŮ, TABULEK, GRAFŮ A DIAGRAMŮ	53
	PŘÍLOHY.....	I

SEZNAM ZKRATEK A POJMŮ

FPS (First Person Shooter) – podžánr akčních her, ve které hráč hraje z pohledu své postavy, za kterou hraje

RPG (Role-Playing Game) – žánr hry, ve které se hráč vtěluje do fiktivní postavy a hraje s ní podle daných pravidel

Vývojové prostředí – program, do kterého se zapisují programy pomocí programovacích jazyků, většinou obsahuje vestavěný překladač a další užitečné funkce

Překladač – program, který přeloží vytvořený kód programovacího jazyka do strojového kódu, který je následně spuštěn a vykonán, při překladu může upozornit na chyby

Level – herní úroveň, může se jím rozumět část hry nebo úroveň samotného hráče, podle čísla levelu si může hráč zjistit, jak daleko už ve hře pokročil nebo jak si stojí mezi ostatními hráči

Tutoriál – výuková část hry, která má za úkol seznámit hráče se všemi funkcemi

Mod – (česky modifikace) rozšíření hry nebo programu o funkce, které se v základu hry nevyskytují, většinou pocházejí od lidí, kteří se nepodíleli na vývoji

Engine – jádro hry, které nabízí knihovny pro herní funkce, na jednom enginu může být postaveno více různých her

Trial verze – omezená verze plné hry, většinou nabízí nízký počet šivelů nebo nabízí časově omezený přístup ke hře

Shareware – typ programu, který je volně šiřitelný po internetu, ale uživatel musí dát autorovi nějakou odměnu stanovenou v licenci, shareware může poskytovat i trial verzi na určitý čas

Úvod

Téma této práce bylo zvoleno na základě zájmu o počítačové hry, které nejsou jen běžnou odpolední kratochvílí, ale dokážou rozvinout schopnosti hráče, především jeho logické myšlení. Tyto hry nejen vzdělávají svého hráče, ale také udržují jeho mysl bdělou a neustále ji cvičí a posilují. Fungují na lidský mozek podobně jako posilovna na svaly. Stejně jako ne každé cvičení je opravdu efektivní na konkrétní svaly v těle, ne každá hra rozvíjí schopnosti hráče a bystří jeho mysl tak, jako jiné. Právě proto se tato práce zabývá jejich porovnáváním.

Od logických her se dostaneme až ke hrám zaměřeným na výuku programování, které jsou založené na stejném principu a navíc obohacené o prvky důležité při vytváření programů, například algoritmizace nebo syntaxe skutečných programovacích jazyků. Zaměření těchto her nemusí být pouze na začátečníky, nýbrž i na pokročilé.

Do této práce bylo vybráno deset her, které nabízejí široké spektrum přístupů k výuce programování. Byly vybrány tak, aby se od sebe lišily a nabízely vždy něco, co ostatní hry nenabízí, ať už z hlediska zaměření na hráče podle věku nebo nabídky programovacích jazyků. I když se na trhu nacházejí desítky různých her, jejich přístupy k výuce programování se často podobají a liší se pouze tématem a grafickým rozhraním. Počet vybraných her v této práci je dostačující vzhledem k celkové nabídce, při čemž všechny splňují kritéria, která byla zvolena za účelem dodržení tématu práce.

Výběr her zpočátku probíhal pomocí internetových vyhledávačů a obchodního portálu Steam, který nabízí široké spektrum her všech žánrů. Jakmile jsem získal počáteční přehled o nabídce her, začal jsem některé z nich zkoumat. O tom, které hry zařadím do výsledné práce, jsem se rozhodoval na základě vlastních zkušeností, které jsem získal hraním hry a po přečtení odborných i amatérských recenzí. Některé hry mě přivedly k jiným hrám, některé jsem zkoumal na serveru YouTube a četl jsem články na internetu, které se zabývaly stejnou tematikou. Testoval jsem i hry, o kterých jsem se dozvěděl od přátel, spolužáků ze školy i učitelů. Dozvěděl jsem se i o serveru Hour of Code, který hry se stejnou tematikou sdružuje a nabízí zdokonalení v programování pro každého.

Všechny tyto metody výběru mě dovedly ke konečnému výsledku této práce, veškeré informace obsaženy v popisech her a v jejich srovnání pocházejí z mých vlastních zkušenos-

tí, s recenzemi a názory ostatních hráčů jsem porovnával až své vlastní poznatky a vyhýbal jsem se jakýmkoliv hodnocením her, které by mohly ovlivnit můj úsudek a samotný přístup k vybrané hře.

Tato práce nemá za úkol hodnotit, nýbrž pouze porovnávat vybrané hry a poskytnout nezájatý pohled na nabídku her zaměřených na výuku programování.

Součástí této práce je také praktický příklad, ve kterém jsem navrhl a naprogramoval svůj vlastní rozšiřující příklad pro vybranou hru. I na tuto část jsem si zvolil kritéria, podle kterých jsem postupoval.

CÍLE PRÁCE

Výběr vhodných her

- Stanovení kritérií pro výběr – tato kritéria jsou blíže popsána v první kapitole

Porovnání vybraných her z hlediska programovacích jazyků, zaměření podle věku a zkušeností hráče a poměru hry a programování

- Pro dosažení tohoto cíle musí být každá hra podrobně prozkoumána a musejí být vzaty v potaz názory dalších hráčů, kteří danou hru hráli. Následně musejí být vybrané hry mezi sebou porovnány a vypsány všechny jejich rozdíly.

Poskytnutí návodu a možného srovnání pro výběr hry

- Tento cíl souvisí s předchozím cílem, jelikož jeho dosažení závisí na stejných faktorech.

Návrh rozšiřujícího příkladu pro vybranou hru

- Výběr hry
- Návrh příkladu
- Realizace příkladu

1 POPIS HLAVNÍCH ZNAKŮ HER PRO VÝUKU PROGRAMOVÁNÍ

1.1 DEFINICE HRY

„Hra je aktuální činnost volená i uskutečňovaná zcela svobodně, není osobně vědomým ‚prostředkem k cíli‘, ale spíše ‚cílem sama o sobě‘.“ (Čačka, 2000, s. 150)

„Existuje ovšem celá řada pokusů o definici hry a o vysvětlení její podstaty, jejího vzniku a účelu. Nejčastěji se zdůrazňuje, že hra je činnost (fyzická nebo psychická), která je vykonávána jenom proto, že je libá a že přináší dítěti (i dospělému, pokud si ještě dovede hrát) uspokojení sama o sobě, bez vnějšího uloženého cíle, ať je to činnost sama o sobě příjemná, nebo i výrazně nepříjemná (např. dolování kamenů při stavbě „domů“ nebo dlouhé čekání v křoví bez pohybu při hře na schovávanou).“ (Langmeier a Krejčířová, 1998, s. 100)

Obě definice zmiňují, že hra by měla přinášet uspokojení sama o sobě. Výuková hra by měla k tomu ještě přinést nějaké znalosti, aniž by narušila smysl hry samotné. Jak píše profesor Josef Maňák (2003, s. 127):

„Hra ve výuce má své opodstatnění, neboť edukační proces by měl rozvíjet sociální, kognitivní, kreativní, tělesné, volní a estetické kompetence žáků. Role pro učitele je náročná, aby přirozeně spojil hru s učením. Didaktická hra obsahuje výrazný seberealizační prvek v oblasti poznávacích činností. Je to hra s pravidly, tudíž má účinek i výchovný. Didaktickou hru lze vymezit jako takovou seberealizační aktivitu jedinců nebo skupin, která svobodnou volbu, uplatnění zájmů, spontánnost a uvolnění přizpůsobuje pedagogickým cílům. Didaktická hra si zachovává většinu znaků hraných činností.“

Podle slovníku cizích slov je didaktika synonymem teorie vyučování a vzdělávání. Didaktická hra by tedy měla vzdělávat a zároveň si zachovat všechny prvky, které definují hru. S pomocí hry lze vzdělávat žáky ve výuce zábavným způsobem. V případě, že už člověk není účastníkem výuky nebo se chce pomocí hry vzdělávat mimo školu, může vyhledat výukovou hru. Definice učení jsou následující:

„Učení je získávání zkušeností a utváření jedince v průběhu jeho života. Naučené je opakem vrozeného.“ (Čáp, 1993)

„Učení je proces utváření způsobilostí, jež umožňují subjektu, který prošel učením, chovat se novým způsobem a podat jiný výkon, než jakého byl schopen na počátku učení.“
(Gagné, 1975)

Pokud spojíme obě definice, můžeme získat definici výukové hry. Výukovou hru bych definoval jako hru, která přináší hráči uspokojení a seberealizaci a zároveň mu předá zkušenosti, aby mohl podat jiný výkon, než jakého byl schopen na počátku hraní hry.

Dnešní hry, které se zaměřují na výuku programování, dokazují, že vzdělávat se lze i zábavně. Nejenže jsou natolik kreativní, že se svými prvky velmi podobají běžným hrám, některé navíc dovolují hráči, aby přispěl k rozvoji obsahu těchto her a tím rozšířil povědomí o programování i mezi ty, kteří ho ve škole nikdy nestudovali, a jejich zaměstnání ho nijak nevyžaduje.

Právě díky tomu, že se od běžných her liší stále méně, rozšiřují se základny jejich hráčů a přicházejí stále nové a nové hry, které vzdělávají a zároveň pobaví. Velký zájem přivádí nové kreativní programátory, kteří mají zájem o vytváření her zaměřených na výuku programování. Her, které pomáhají v mimoškolním vzdělávání, poskytují praktické zkušenosti těm, kteří je potřebují (například během studia programování) a přinášejí zábavu všem, kteří rádi testují svou mysl logickými úkoly.

Prvkem, který se bude lišit i nadále v každé výukové hře, jsou nějaké skutečné znalosti. V případě běžných her veškeré znalosti hráče potřebné ke splnění úkolů vyplývají ze samotného hraní hry, ve zkušenostech s ovládním a s prvky samotné hry. Dalo by se říct, že i úplný amatér by dokázal najít způsob, jakým v určité hře dosáhnout úspěchu. U výukových her získává hráč dovednosti užitečné i mimo samotnou hru a ovládním těchto znalostí je nezbytné pro další postup ve výukové hře.

Čím se tedy liší například od běžného výukového programu? Ve výukovém programu nebo například elektronickém kurzu naleznete výukové materiály v psané podobě, případně video přednášku univerzitního profesora. Obě formy mají za účel předat uživateli informace potřebné ke splnění dalších úkolů nebo testů.

V průběhu každého výukového programu je každý člověk sám za sebe, ve hře se může stát někým jiným. Ať už je to rytíř ve fiktivním světě plném nadpřirozených bytostí nebo vládce své vlastní kolonie, hráč může použít vlastní fantazii a přijmout podmínky herního svě-

ta, ve kterém se právě nachází. V takovém světě se může hráč cítit jako někdo s neomezenými možnostmi a získá tak větší motivaci překonávat své vlastní limity. Může jít o získání většího počtu hvězdiček za splnění levelu nebo skok v žebříčku všech hráčů na lepší pozici. U výukových programů směřuje uživatel pouze ke splnění všech podmínek, postavení mezi ostatními účastníky kurzu nebo uživateli programu nehraje příliš velkou roli.

Definice Jana Průchy (1998) rozděluje výukové programy do třech kategorií:

1. S akcentem na obsahový efekt - systém pečlivě vybraných a promyšleně uspořádaných témat, která by měla být předmětem vyučování a učení.
2. S akcentem na cílový aspekt - systém hierarchicky uspořádaných, vnitřně konzistentních, dobře vymezených a relativně kontrolovatelných požadavků, jejichž vyučování a učení má směřovat.
3. S akcentem na regulativní aspekt - systém optimálně spojující vnější řízení a autoregulaci žákovu učení tak, aby respektoval zvláštnosti konkrétních žáků, poznatky o učení a strategiích jeho řízení.

Lze odvodit, že výukové programy mají jediný účel, člověka něco naučit. Z porovnání definic vyplývá, že hra se více soustředí na potřeby člověka a výukový program na obsah.

1.2 HLAVNÍ ZNAKY HER PRO VÝUKU PROGRAMOVÁNÍ

Hlavním znakem je přítomnost nějakého programovacího jazyka a samotného programovacího procesu. Proces může probíhat formou ručního zadávání kódu nebo skládání grafických prvků. Programovací jazyk pak může být skutečný (JavaScript, Java, Python...) nebo uměle vytvořený pro danou hru.

Dalším znakem je skutečnost, že veškerý váš postup závisí na znalostech programování. V případě her, které učí programovat od úplných začátků, se znalosti potřebné ke splnění úkolu rovnají postupu ve hře a nikdy se neobjeví nic, co by hráč zatím neměl ovládat, pokud nejde o nějaký bonus. Ostatní, které využívají znalosti, jež už si hráč přinesl z dřívějších, brání v postupu každému, kdo nesplní daný úkol. Nelze to nijak obejít, vše závisí na schopnostech hráče. Hry obvykle nabízí nějakou formu nápovědy pro případ nouze.

Třetím znakem je využitelnost získaných dovedností i mimo danou hru. Dokonce i mimo hry obecně. Běžné hry mohou rozvíjet reflexy hráče, strategické nebo logické myšlení. V simulátorech vlaků či letadel se může hráč seznámit s fungováním těchto strojů. Obvykle si hráč tyto zkušenosti přenáší pouze do jiných her stejného typu. V případě her zaměřených na výuku programování lze využít zkušenosti zcela mimo herní sféru, pokud se ne-

jedná o naprogramování vlastní hry nebo alespoň její části. Některé hry tento typ výuky přímo nabízejí. Nemusí se jednat pouze o samotné programování, nýbrž i o řešení logických úkolů, které výukové hry učí, využitelných v běžném životě.

Dalším neméně důležitým znakem tohoto typu her je rozvíjení algoritmického myšlení. Pro každý běžný postup v lidském životě existuje algoritmus, i když o tom běžně nepřemýšlíme. Pro zápis algoritmů existují konkrétní pravidla a vzniklá struktura se nazývá diagram. Programátor přenáší tyto algoritmy do programovacího jazyka. Hráč si může ve hře vyzkoušet popsat na první pohled běžnou činnost algoritmem v různých formách (kód nebo grafické prvky). Díky těmto zkušenostem se může posléze naučit přemýšlet o složitějších problémech odlišným způsobem a hledat v nich algoritmy vedoucí k jejich vyřešení.

1.3 FUNKCE HRY

Pro začátečníky

Hra určená hráčům, kteří nemají žádné zkušenosti s programováním. Vzdělává hráče v programování od nejjednodušších základů a vše musí pečlivě vysvětlit. V takovém případě se minimálně hlavní část hry zaměřuje na jednu výukovou linii a provází hráče po předem definované cestě. Úkoly bývají ze začátku tak jednoduché, že jednotlivá řešení se příliš neliší, některé hry mají zavedený systém hodnocení každého úkolu podle počtu řádků kódu nebo jiných aspektů, nabízejí tím hráči možnost stále se zlepšovat a zároveň ho učí neplýtvat pamětí počítače, když by zpracovával složité algoritmy. Jak jsou hry propracované a jestli není tato vlastnost pouze hlídání počtu řádků bez důsledného prověření, zda opravdu není delší kód lepší z hlediska využití paměti, už záleží na konkrétní hře.

Pro pokročilé

Hra určená hráčům, kteří už mají s programováním zkušenosti a ovládají více než všeobecné základy. Hra nabízí hráčům zdokonalení v programovacích dovednostech a nevysvětluje základní programové struktury. Na rozdíl od her pro začátečníky neposkytuje hráči žádné pevné vedení, hráč má mnohem větší možnost výběru, jakým směrem se ve hře vydá. Může si vybírat obtížnost úkolů podle svých schopností a sdílet svá řešení s ostatními hráči, jelikož úkoly nabízí větší spektrum možných řešení. Tato svoboda nebývá u všech her úplná (například CheckIO nebo TIS-100). Hráč se vzdělává úplně sám jen

tím, že hru hraje, není zde natolik patrné zaměření na výuku. Tyto hry musí splňovat definici učení a rozvinout hráčovy schopnosti, čehož lze jednoduše dosáhnout postupným zvyšováním obtížnosti nabízených úkolů.

U obou kategorií platí, že hra by měla plnit relaxační a seberealizační účel a poskytovat jistou svobodu pro hráče. Zároveň z hlediska pedagogického přinést nové znalosti a definovat jistá pravidla.

1.4 KRITÉRIA VÝBĚRU

Přítomnost programovacího jazyka

Jazyk musí být ve hře přítomen v jakékoliv podobě. Může se vyskytovat pouze jako zjednodušená verze skutečného jazyka nebo jako jazyk speciálně vytvořený pro hru. Může se tam nacházet i v podobě grafických prvků, pokud jsou pojmenovány jako příkazy a nejsou jen v podobě obrázku. Účelem kritéria je vybrat hry, které ukazují programování v jeho skutečné podobě (i když třeba jen zjednodušeně).

Postup ve hře musí záviset na znalosti programování

U her pro začátečníky je postup závislý pouze na tom, co hra hráče v danou chvíli naučí, žádné jiné znalosti nejsou vyžadovány. Nesmí být možné dosáhnout nejvyšší možné úrovně bez použití jakékoliv formy programování.

Významný rozdíl od ostatních vybraných her

Hry nesmí fungovat na stejném základu s rozdílným grafickým rozhraním. V případě, že se hry podobají, musí se lišit alespoň v jednom zásadním aspektu, který se týká programování nebo způsobu výuky. Tento rozdíl se netýká programovacích jazyků, tj. několik her může fungovat na jazyku JavaScript a přesto nabízet úplně jiné možnosti a formu výuky.

1.5 HRY NESPLŇUJÍCÍ KRITÉRIA

Na trhu se kromě her obsažených v této práci nachází celá řada dalších her, které nespĺnily kritéria výběru v jednom či více bodech. V největší míře neobsahují tyto hry žádný konkrétní programovací jazyk a způsob programování má podobu skládání grafických prvků. Tyto prvky jsou pouze v podobě obrázku a na první pohled nepřipomínají skutečný kód. Tyto hry stále plní většinu funkcí her zaměřených na výuku programování, především rozvoj algoritmizace a logického myšlení. Fungují jako startovní bod pro úplné začátečníky,

kteří nechtějí pracovat s psanou formou kódu, jelikož jim psané příkazy nic neříkají nebo o takovou formu programování vůbec nemají zájem. Hráči těchto her si mohou zlepšit své logické myšlení i bez zájmu o programování jako takové. Stejně tak mohou tyto hry přivést k zájmu o programování všechny, kteří se s ním zatím neseťkali.

Významné rozdíly hrají roli u her, které jsou založené na stejném základu, nalezneme v nich stejné úkoly, pouze se liší grafickým rozhraním, nabízejí hráči jiný svět s jinými postavami, ale nenabízí žádné nové možnosti z hlediska programování. Grafické rozhraní hry pro výběr her nebylo důležité, i když je u každé hry popsáno.

Na trhu se nacházejí i hry, které se týkají spíše počítačových sítí, a hráč se vtělí do role hackera. Pomocí rafinovaných metod proniká do cizích serverů a krade z nich peníze nebo důležité informace. Ke své práci používá příkazový řádek, případně grafické rozhraní. Tyto hry se principem podobají hrám pro výuku programování, naučí hráče, jak se pracuje s IP adresami, předvedou názornou ukázkou některých síťových útoků a zábavnou metodou předají hráči znalosti, které může využít i mimo danou hru (nezákonnou aktivitu nepodněcují). Hráč se zde s programováním setkává pouze okrajově, programy, které používá k prolomení zabezpečení serverů, už jsou ve hrách hotové.

O počtu her, které se nacházejí na trhu, nelze mluvit v konkrétních číslech, poněvadž stále vznikají nové a využívají pro výuku různé principy. Komunita hráčů má také na vývoj vliv, praktická část této práce dokazuje, že i běžní hráči mohou vytvářet vlastní příklady. Když ponecháme kritéria stranou a zaměříme se na výukové hry se zaměřením na programování v jakékoliv míře, mluvíme o desítkách různorodých her a velkých možnostech výběru.

2 POPIS VYBRANÝCH POČÍTAČOVÝCH HER ZAMĚŘENÝCH NA VÝUKU PROGRAMOVÁNÍ

2.1 HUMAN RESOURCE MACHINE

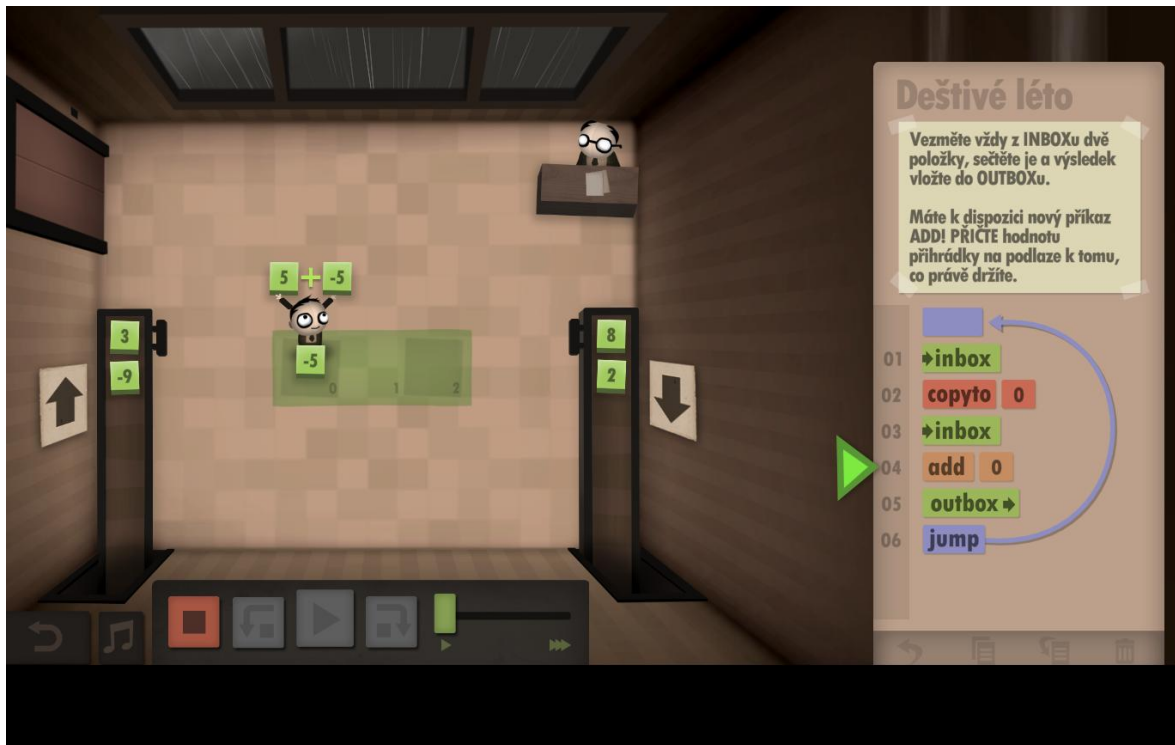
2.1.1 POPIS HRVY

- Jazyk: čeština.
- Dostupnost: shareware.
- Platforma: PC.

Tato hra není primárně propagována jako výuková, nýbrž jako logická hra pro pobavení a zlepšení logického myšlení, definici výukové hry vyplývající z předchozí kapitoly však splňuje. V jednotlivých levelech nabízí logický problém řešitelný sérií příkazů v grafickém formátu. S každou další úrovní je problém náročnější. Hráč začíná s minimem příkazů, které stačí pro základní problémy a postupně s vyšší náročností přicházejí další.

Hra nabízí příběh zaměstnance, který při splnění úkolů postupuje do vyšších pater obří kancelářské budovy. V každém levelu dostane hráč sérii čísel, občas i znaků (vstup), kterou použije pro zadaný úkol (např. násobení bez použití funkce násobení) a výsledky pošle na výstup. Každý level nabízí bonus za použití omezeného počtu příkazů a počtu kroků zaměstnance při plnění úkolu.

Při plnění úkolu lze procházet jednotlivé příkazy a sledovat postup zaměstnance na obrazovce. Lze procházet programem krok po kroku, vrátit se o krok zpět a úplně na začátek. Plnění lze pomocí posuvníku urychlit nebo zpomalit podle potřeby.



Obrázek 1: Human Resource Machine: Průběh programu

2.1.2 ZAMĚŘENÍ HRY

Tato hra je vhodná pro úplně začátečníky, kteří nikdy neprogramovali. Výzvu v ní ovšem najdou i zkušení programátoři. Nepoužívá se zde žádný konkrétní jazyk, celé plnění úkolů je řešeno grafickými prvky, které lze skládat za sebou do připraveného formuláře. Každý nový prvek je ve hře popsán a jeho funkce ukázána na příkladu. Každý úkol představuje logický problém, pro jehož řešení je potřeba sestavit univerzální algoritmus. Tyto algoritmy jsou v přenesené formě využitelné pro většinu známých programovacích jazyků.

2.1.3 UŽIVATELSKÁ PŘÍVĚTIVOST

Hra nabízí přívětivé grafické zpracování a přehledné rozhraní. Potěší i krátké meziscény a situování hry do kancelářského prostředí, kde hráč svými algoritmy zlepšuje administrativní úkoly firmy a za splnění úkolu je vždy povýšen. Všechny prvky jsou na obrazovce dobře viditelné a jejich použití je zřejmé. Každý nový příkaz je dobře vysvětlen a předveden na příkladu.

2.1.4 MOŽNOST NÁPOVĚDY

V každém levelu máte svého „vedoucího“, který vám na požádání vysvětlí, v čem spočívá váš úkol a nabídne vám i příklad výstupu. Pro samotné splnění úkolu musí hráč využít pouze své vlastní schopnosti, zobrazení výsledku není možné během ani po splnění levelu.

2.2 CODEMONKEY

2.2.1 POPIS HRY

- Jazyk: některé části přeložené do češtiny, zbytek angličtina.
- Dostupnost: trial zdarma, plná verze za poplatky.
- Platforma: webový prohlížeč.

Hráč v této hře pomocí kódu ovládá malou opičku a pomáhá jí plnit úkoly. Začíná s jednoduchými výzvami a v dalších levelech se obtížnost postupně zvyšuje. Každý level může být splněn na jednu, dvě nebo tři hvězdy v závislosti na efektivitě kódu a počtu zabraných řádek. Během hry lze využívat jednoduché předdefinované funkce odpovídající úkolu ve hře (např. pro sebrání banánu použijeme funkci `grab()`). Dalším prvkem je zde tečková notace využívaná při ovládání jiných zvířat než opičky.

2.2.2 ZAMĚŘENÍ HRY

Už na první pohled můžeme zjistit, že hra je zaměřena pro žáky základních škol, kteří se poprvé chtějí seznámit s programováním. Tato hra nevyžaduje žádné zkušenosti a postupem přidává nové prvky a učí s nimi hráče zacházet.

Hra je přímo zaměřena pro výuku a nabízí cenové balíčky pro třídy nebo celé školy. Do hry se mohou registrovat také učitelé. Používá se zde jazyk CoffeeScript podobný JavaScriptu používaný primárně ve webových aplikacích.

Na vyšších úrovních hra učí také použití cyklů s pevným počtem opakování nebo s podmínkami na začátku a na konci. Nechybí zde ani proměnné využívané především pro zjednodušení zápisu. S dalším postupem přicházejí i jednoduché funkce a jejich volání.

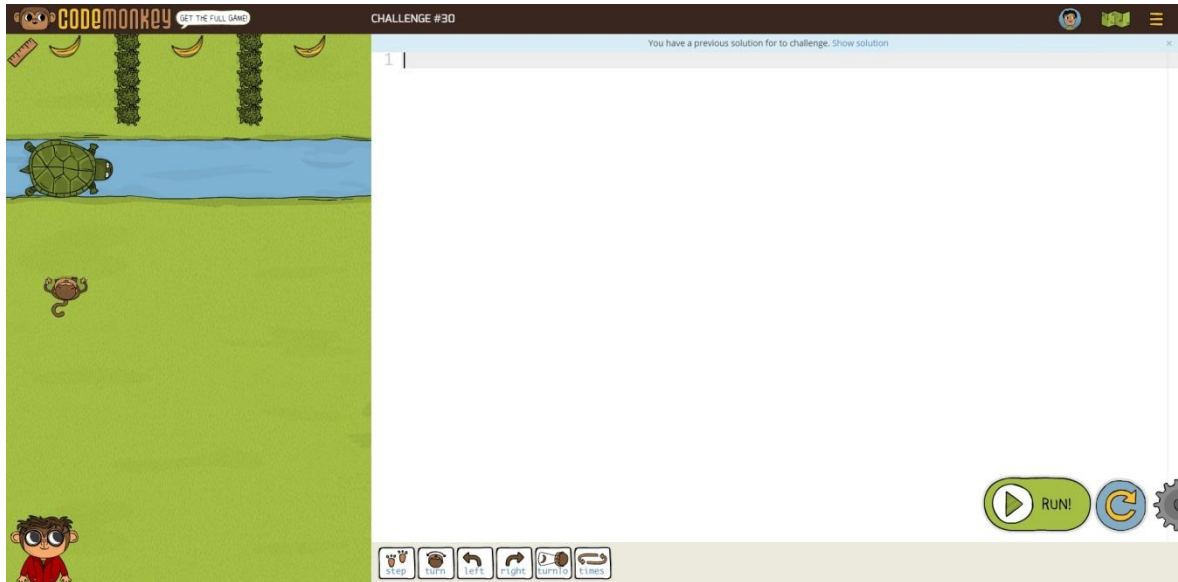
```
1 step 10
2-3.times ->
3   ... step 10
4   ... step -10
5   ... turtle.step 10
```

Obrázek 2: CodeMonkey: CoffeeScript

2.2.3 UŽIVATELSKÁ PŘÍVĚTIVOST

Tato hra je svým grafickým rozhraním příjemná a velmi jednoduchá na ovládání. Každý program ovládá kreslenou opičku nebo jiná zvířátka. Ovládací prvky jsou na obrazovce

dobře viditelné a řádně popsané. Samotný kód lze psát ručně do připraveného formuláře nebo „naklikat“ pomocí grafických prvků (s ručním dopisováním čísel). Měření vzdálenosti mezi opičkou a jejím cílem nebo měření úhlu, o který se má otočit, může být pro některé hráče náročnější a vyžaduje trochu cviku.



Obrázek 3: CodeMonkey: Uživatelské rozhraní

2.2.4 MOŽNOST NÁPOVĚDY

Hra nápovědu nenabízí, pouze tipy v případě, že kód není v pořádku, které se ovšem nemusejí týkat vašeho problému.

2.3 CODECOMBAT

2.3.1 POPIS HRY

- Jazyk: čeština (kód: angličtina).
- Dostupnost: základní verze hry zdarma, premium placené.
- Platforma: webový prohlížeč.

Tato velmi propracovaná hra přenesení hráče do fantasy prostředí a poskytne mu vlastní přizpůsobitelnou postavu. Hráč za každý splněný úkol získává zkušenosti a postupně se zvyšuje jeho úroveň, odemykají se mu nové předměty, které přinášejí vylepšení postavy i funkce, které může použít v kódu. Hra nabízí možnost přidat se do klanu, získávat odměny za úspěchy i nakupovat speciální diamanty vyměnitelné za nové předměty. Lze ji zařadit do žánru RPG.



Obrázek 4: CodeCombat: Inventář

V každém levelu dostane hráč zadaný úkol a pro jeho vyřešení musí napsat kód. Každý úkol obsahuje několik bodů, které musí pro úspěch splnit. Postupně se zvyšuje obtížnost a zároveň se objevují nové možnosti a programové konstrukce. Při plnění vidí pohyb svého hrdiny na obrazovce, kde se zobrazuje jeho trasa, veškeré útoky, protivníci a životy. Po úspěšném splnění všech bodů může kód odeslat, získat zkušenosti, nové předměty, odměny, drahokamy a především odemknout další level.

Hra nabízí i multiplayer, kde se v několika odlišných módech hráč utká s ostatními hráči. Může hrát buď se svým bojovníkem, nebo dostane všechny dostupné metody, aby záleželo pouze na kvalitě jeho kódu.

2.3.2 ZAMĚŘENÍ HRY

Do této hry se lze pustit i bez znalosti programování, vhodná je pro začátečníky, pokročilé programátory i každého, kdo chce zkusit nový přístup k RPG hře. Na výběr je zde ze 4 jazyků: Python, JavaScript, CoffeeScript a Lua (jazyk pro skriptování her).

Hra je přímo zaměřena na výuku programování a nabízí možnost zaregistrovat se jako běžný hráč, student nebo učitel. Hra je bez věkového omezení, což dokazuje i možnost zaregistrovat se jako rodič. Účelem hry je vyvolat v lidech zájem o programování tím, že kód přímo ovládá jejich postavu ve hře a výsledek je tak okamžitě viditelný.

2.3.3 UŽIVATELSKÁ PŘÍVĚTIVOST

Úkoly se dělí do několika různých světů, které se liší vzhledově a podle toho, jaké programové konstrukce se zde nacházejí. Hráč si vybírá pomocí grafického rozhraní a poté se dostane na mapu s úkoly, která se na první pohled zdá poněkud nepřehledná. Při základním postupu se lze orientovat podle velké žluté šipky. Hlavní a vedlejší úkoly se odlišují barevně, nesplněné označuje červená vlaječka a zamčené jsou šedé a létají kolem nich tečky vytvářející dojem ohně.



Obrázek 5: CodeCombat: Mapa postupu ve hře

Uvnitř úkolu dostane hráč nejprve zadání úkolu a vysvětlení toho, co může použít (především při získání nového předmětu). Poté lze spatřit po levé straně animaci, která ukazuje veškeré dění a pohyb bojovníka řízeného zadaným programem. Samotný program lze spatřit po pravé straně včetně seznamu metod, které může hráč použít. Po rozklinutí metody se zobrazí, jak se metoda používá a jakou má funkci. Samotný kód lze napsat ručně.

Celé grafické rozhraní je provedeno v příjemných barvách a animace jsou provedeny jednoduše kreslenou grafikou s neškodnou formou násilí, které by omezovalo některou věkovou skupinu. Veškeré ovládací prvky jsou dobře viditelné a jejich funkce jsou zřejmé na první pohled.



Obrázek 6: CodeCombat: Uspořádání grafického rozhraní

2.3.4 MOŽNOSTI NÁPOVĚDY

Během výukových levelů dostane hráč popis úkolu, který musí splnit a struktura kódu je naznačena pomocí komentářů ve formuláři. Při psaní kódu se objevuje seznam metod, které obsahují zadané znaky, umožňují tak rychlejší zadávání. Po kliknutí na tlačítko „Nápověda“ lze popis úkolu znovu zobrazit a rovněž zjistit několik dalších vodítek včetně vysvětlení konstrukce, která je zapotřebí. Při získání nového předmětu je celá úroveň zaměřena na seznámení hráče s možnostmi tohoto předmětu a ukazuje přesně, co je třeba udělat. Nápověda k předmětu se rovněž zobrazí vždy po rozkliknutí předmětu v úpravě bojovníka včetně popisu schopností, které bojovník s tímto předmětem získá a funkcí, které může použít.

2.4 CODINGAME

2.4.1 POPIS HRY

- Jazyk: angličtina.
- Dostupnost: zdarma.
- Platforma: webový prohlížeč.

Po vstupu do hry lze spatřit dvě hlavní kategorie – procvičování a soupeření. První z nich nabízí několik úkolů rozdělených podle obtížnosti. Po jejich splnění dostává hráč zkušenosti a zvyšuje se jeho úroveň. Na výběr jsou jak úkoly hry, tak úkoly vytvořené samotnými hráči. V druhé kategorii soupeří proti ostatním hráčům, dostane určitý čas na splnění úkolu a vyhrává ten, kdo ho splní první. Hra nabízí jak stálé módy, tak i velké turnaje, které se konají vždy v určitý den a čas. Hráč se do nich musí nejprve zaregistrovat.

Kromě dvou hlavních kategorií lze vytvářet vlastní úkoly. Tyto úkoly lze vytvářet jako komunitní příklady nebo příklady do soutěže více hráčů zvané ‚Clash of Code‘, kde se skupina nejvýše osmi hráčů utká v souboji o to, kdo napíše funkční kód pro zadaný úkol nejrychleji nebo v nejkratší podobě.

2.4.2 ZAMĚŘENÍ HRY

Tato hra není určena pro úplné začátečníky, spíše jako místo, kde si procvičit svoje znalosti jazyka a zdokonalit se. Nabízí široké spektrum jazyků – Java, C++, Python, PHP, Ruby, JavaScript, Perl a mnoho dalších. Jazyky jsou zde používány v jejich skutečné podobě, v jaké se používají ve vývojových prostředích třetí strany. Funguje jako velká komunita programátorů, nachází se zde mnoho diskuzí jak o jednotlivých úkolech, tak i obecné. Nabízí i možnost vytvářet úkoly a stát se tak někým, kdo pomáhá ostatním zdokonalit se v programování. Lze tak hru hrát a zároveň do ní přispívat, to vše zcela zdarma.

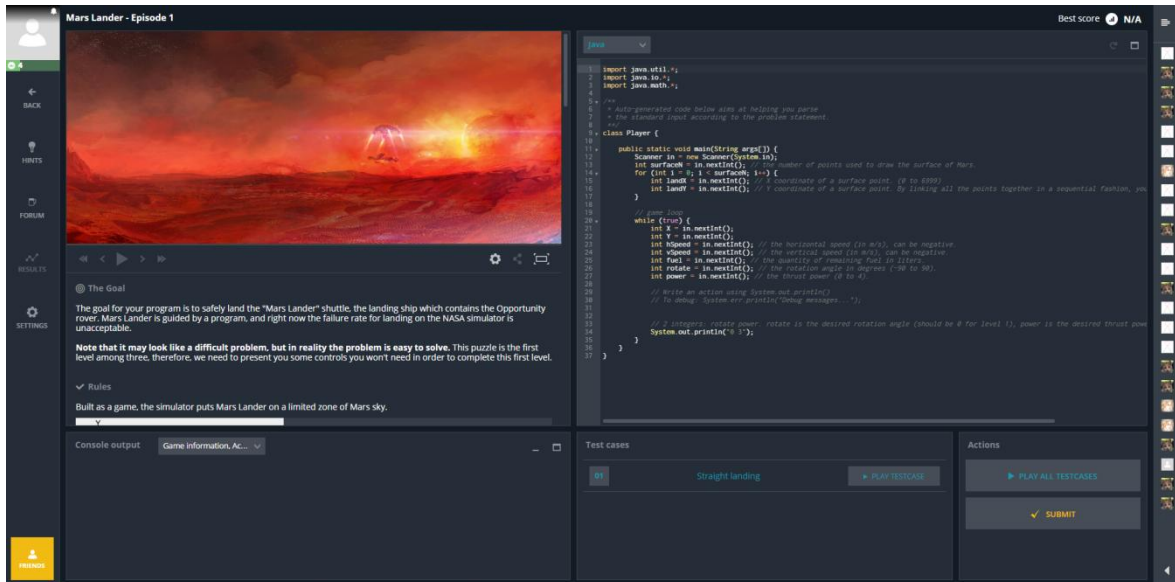
2.4.3 UŽIVATELSKÁ PŘÍVĚTIVOST

Rozhraní této hry je velmi jednoduché, na hlavní obrazovce si vyberete úkol, což vás přeměruje k detailnímu popisu problému, který je potřeba vyřešit a jaké oblasti si při plnění procvičíte.

Vzhled uživatelského rozhraní se liší úkol od úkolu. V některých se nachází animace, která ukazuje výsledek vašeho programu, jiné obsahují pouze detailní popis úkolu včetně vzoro-

vých výstupů. Pochopit, jak má program fungovat, nemusí být jednoduché, záleží na zkušenostech v této hře, na zkušenostech s programováním i na kvalitě zadání úkolu.

Grafické ovládací prvky jsou přehledně umístěny a jejich funkce je zřejmá. K orientaci v úsecích kódu, které už jsou připraveny, slouží komentáře v kódu. Kód se píše ručně a funguje zde skutečný překladač, který ukazuje chyby.



Obrázek 7: CodinGame: Plnění příkladu

Jednotlivé úkoly jsou nápadité a zajímavě řešené. Pro každého je dostupná diskuze, kde se lze poradit o postupu při plnění úkolu. Po odeslání nějakého řešení lze prohlédnout řešení ostatních hráčů v jazycích, které použili.

2.4.4 MOŽNOSTI NÁPOVĚDY

Při plnění některých úkolů máte možnost zobrazit si řešení v pseudokódu a skutečné řešení ve vámi zvoleném jazyce. Jiné nenabízejí řešení vůbec.

2.5 CHECKIO

2.5.1 POPIS HRY

- Jazyk: angličtina (v jednotlivých úkolech možnost popisu v češtině).
- Dostupnost: zdarma.
- Platforma: webový prohlížeč.

Po vstupu do hry se hráč nachází na mapě jednotlivých ostrovů, na kterých se nacházejí úkoly, jejichž splnění otevírá přístup na další místa. Po rozkliknutí ostrova nebo jednoho z jeho přístupových bodů se objeví seznam úkolů s označením obtížnosti a prvky, se kterými se v úkolu pracuje (čísla, cykly, podmínky atd.).

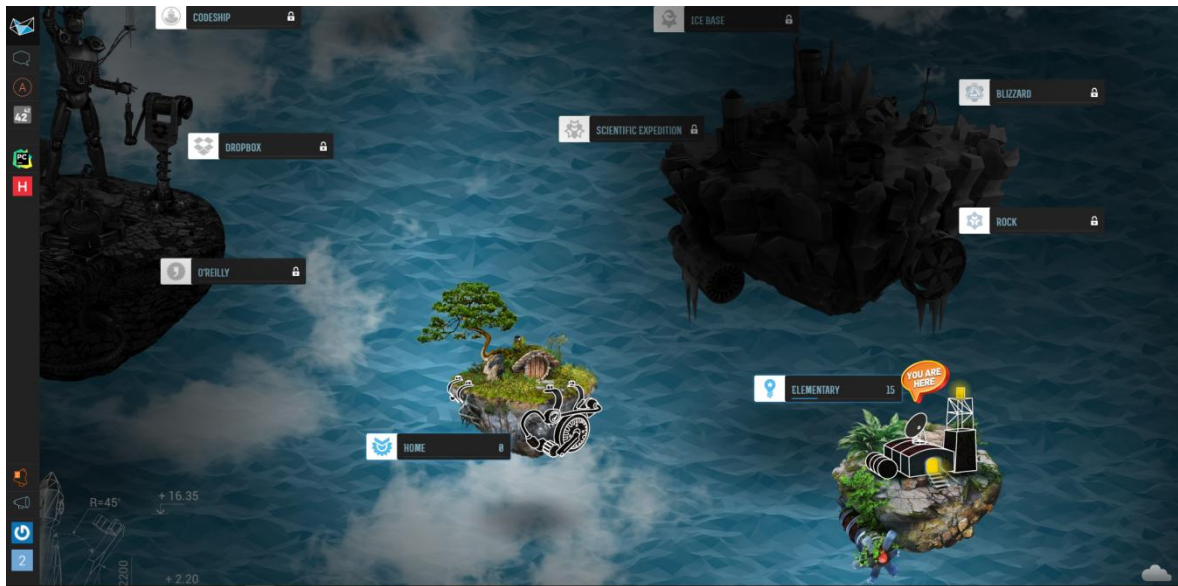
Po rozkliknutí daného úkolu se objeví detailní popis, ukázka výstupu a možnost nápovědy. Při samotném plnění úkolu se objeví formulář pro zadání kódu, který je následně přeložen a spuštěn podobně jako v jakémkoliv jiném vývojovém prostředí třetí strany. Pro kontrolu má hráč možnost zadat vlastní vstup, spustit předdefinovaný vstup nebo kód rovnou odeslat, pokud si je jistý jeho funkčností. Po splnění úkolu lze zobrazit řešení ostatních hráčů nebo publikovat vlastní řešení pro další hráče.

2.5.2 ZAMĚŘENÍ HRY

Hra je určena pro každého, kdo má zkušenosti s programovacími jazyky Python a JavaScript. Nabízí zdokonalení programátorových znalostí pomocí hry v dobře vybaveném formuláři s překladačem, který dokáže vypsat chyby.

2.5.3 UŽIVATELSKÁ PŘÍVĚTIVOST

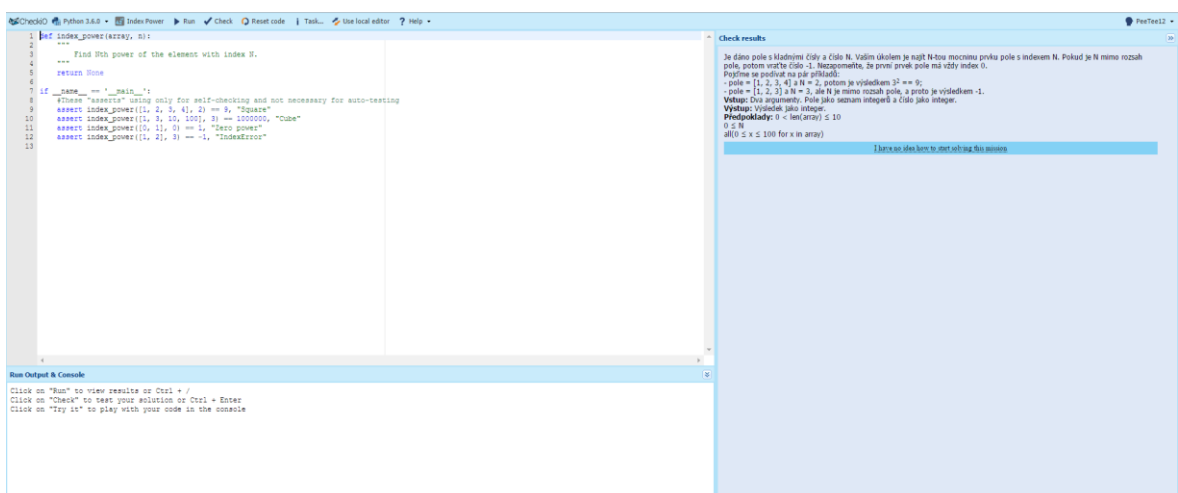
Uživatelské rozhraní je velmi prosté. Na herní mapě lze spatřit jednotlivé ostrovy s jejich jmény, z nichž odemčené jsou v plných barvách a zamčené šedé s malým zámečkem vedle zámku.



Obrázek 8: CheckIO: Herní mapa

Po vybrání ostrova lze vybrat z jednotlivých úkolů, kde jsou přehledně vidět označení obtížnosti, konstrukce, se kterými se hráč při řešení úkolu setká a v případě, že existuje český překlad, tato informace se také nachází přímo zde ve výběru. Po výběru úkolu se objeví obrazovka s popisem úkolu, vstupy a požadovanými výstupy. Pokud zadání úkolu existuje v několika jazycích, lze si libovolně vybrat z jejich seznamu.

Formulář pro tvorbu programu je též velice prostý, všechny prvky jsou na obrazovce dobře vidět a jejich funkce je zřejmá. Hráč si může vybrat mezi několika možnostmi spuštění, překladač vypíše případné chyby.



Obrázek 9: CheckIO: Tvorba programu

2.5.4 MOŽNOSTI NÁPOVĚDY

Už z detailního zadání úkolu má hráč možnost prohlédnout si nápovědu, kterou dodává autor úkolu. Postupně kliknutím na tlačítko nápovědy se objevují detailnější informace o postupu plnění úkolu. Po rozkliknutí všech nápověd dostává hráč možnost přejít na diskusní fórum a požádat o pomoc. Tyto nápovědy lze následně vidět i v samotném řešení úkolu.

2.6 EMPIRE OF CODE

2.6.1 POPIS HRY

- Jazyk: angličtina.
- Dostupnost: zdarma.
- Platforma: webový prohlížeč.

V této hře se stanete majitelem své vlastní kolonie, která se může kdykoliv stát terčem útoku a zároveň sama útočit pomocí vojáků. Jako ve všech hrách tohoto typu lze stavět nové budovy, vylepšovat stávající, těžit a sbírat suroviny, vytvářet vojáky a zbraně, útočit na jiné hráče, ničit jejich budovy a krást jim suroviny.

Útočit na jiné hráče lze ve třech kategoriích:

1. Trénink – hráč má možnost otestovat své strategie a sílu vojska před vkročením do skutečné bitvy.
2. Kampaň – v několika desítkách misí dostává hráč možnost projít celou herní mapou, získávat zkušenosti a suroviny.
3. Dobývání – v této kategorii se hráč může postavit jiným skutečným hráčům, zničit cizí kolonie a ukrást jim suroviny.

Od jiných her se liší dvěma podstatnými rozdíly. K vylepšování budov je potřeba splnit úkol naprogramováním algoritmu pro řešení zadaného problému. Pro útoky vojáků lze napsat vlastní algoritmus jejich chování v bitvě a pro obranu taktéž (např. pro obranná děla).

2.6.2 ZAMĚŘENÍ HRY

Hra je určena pro každého, kdo má zkušenosti s programovacími jazyky Python a JavaScript. Nabízí zdokonalení programátorových znalostí pomocí hry v dobře vybaveném formuláři s překladačem, který dokáže vypsát chyby. Možnost ovládat pomocí kódu útok i obranu umožňuje rozvíjet strategické myšlení.

2.6.3 UŽIVATELSKÁ PŘÍVĚTIVOST

Hra nabízí na pohled příjemnou grafiku s ovládacími prvky ve tvaru šestiúhelníku. Při rozkliknutí jednotlivých budov se objeví nabídka s možnostmi zobrazení informací o budově, vylepšení budovy pomocí dělníků nebo pomocí splnění úkolu a také možnost zbourání. Některé možnosti se liší podle typu budovy. Při najetí na tyto ikony se nezobrazí žádný

text, hráč si musí nejprve vyzkoušet, které tlačítko k čemu slouží, jejich funkce nemusí být podle obrázku každému jasná.



Obrázek 10: Empire of Code: Kolonie

Ve formuláři pro tvorbu kódu nalezneme po pravé straně srozumitelné zadání a testovací vstupy, po levé straně prostor pro kód a nahoře tlačítka pro překlad programu, kde se vypíší případné chyby, pro kontrolu zadaného programu v případě, že ho hráč chce odeslat a získat vylepšení (v případě neúspěchu opět vypíše překladač chyby) a tlačítko pro uložení rozpracovaného úkolu, ke kterému se lze kdykoliv vrátit. Uložení na disk hra neumožňuje, kopírování kódu ano. Využití externího vývojového prostředí není zakázáno.

2.6.4 MOŽNOSTI NÁPOVĚDY

Náповěda k této hře existuje ve formě diskusního fóra, kde mohou hráči požádat o pomoc při problémech s kódem nebo hrou samotnou.

2.7 CODE KINGDOMS

2.7.1 POPIS HRY

- Jazyk: angličtina.
- Dostupnost: trial verze zdarma, plná verze placená.
- Platforma: webový prohlížeč.

Tato hra umožňuje vytvářet své vlastní mody do počítačové hry Minecraft. Nabízí více než 80 kurzů a k nim výuková videa. Jednotlivé kurzy jsou rozděleny podle věku a obtížnosti. Cílem každého kurzu je vytvořit jeden mod do hry Minecraft pomocí programovacího jazyka Java.

Každý hráč dostane k dispozici svůj vlastní server hry Minecraft, který mu dovolí otestovat mody, které vytvořil. Server může využívat libovolně, jak pro sebe, tak i pro více hráčů.

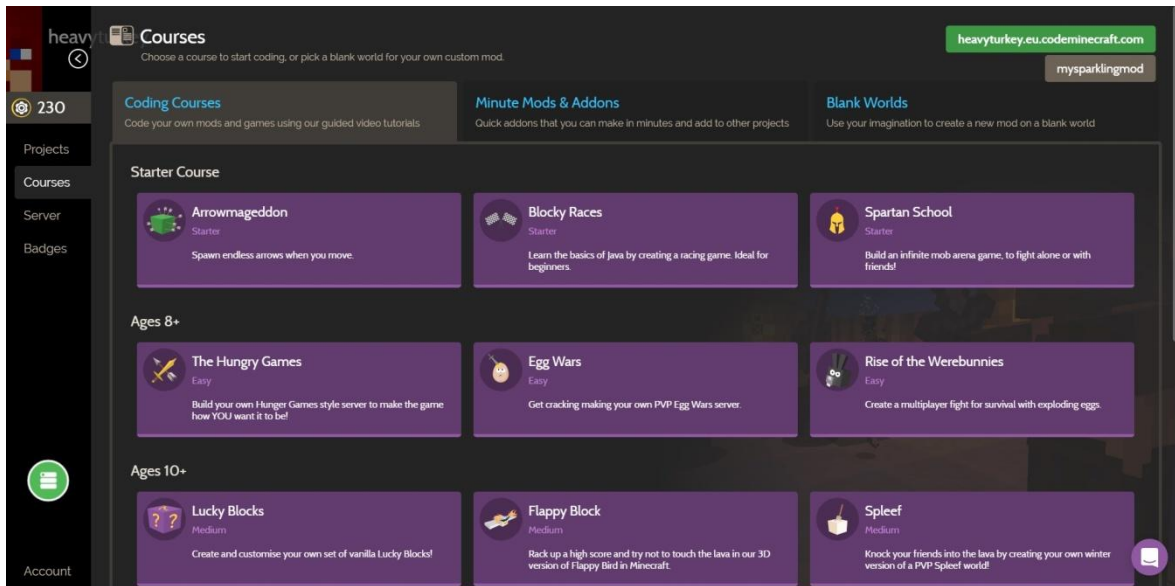
Za úspěchy dostává hráč odznaky.

2.7.2 ZAMĚŘENÍ HRY

Hra je určena pro hráče od 8 let a její kurzy jsou rozděleny podle věku. Vyučuje jazyk Java skrz skládání metod s využitím funkcí pojmenovaných přesně podle toho, k jakému účelu jsou využity. Hlavním účelem je výuka tvorby modů do hry Minecraft a cílí na každého, kdo tuto hru hraje a chtěl by se naučit, jak ji rozvíjet.

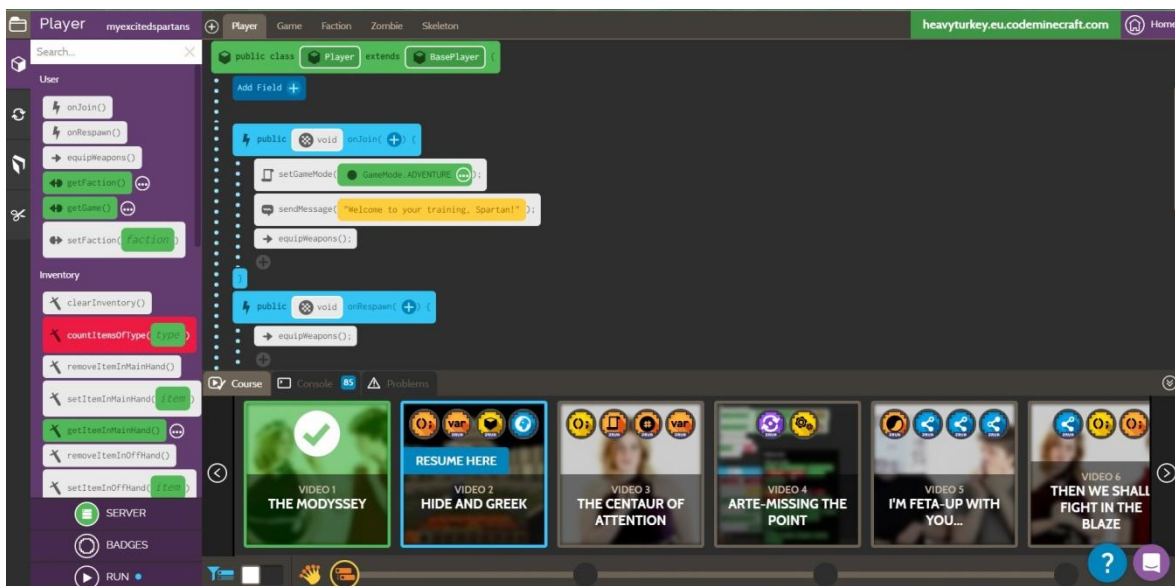
2.7.3 UŽIVATELSKÁ PŘÍVĚTIVOST

Hlavní obrazovka je rozdělena na čtyři hlavní sekce: projekty, kurzy, server a odznaky. Další možností je úprava účtu hráče vespod obrazovky. Veškeré ovládací prvky jsou velké, dobře viditelné a barevně odlišené.



Obrázek 11: Code Kingdoms: Nabídka kurzů

Po výběru kurzu se vždy objeví popis úkolu a úvodní video. Po vstupu do něj vidíme formulář pro zadávání kódu, který lze přepínat na čtyřech úrovních, přičemž ta první činí z kódu zcela grafické skládání prvků bez nutnosti cokoli zapisovat, vyjma hodnot proměnných případně názvů metod, které lze ovšem vybrat ze seznamu, zde záleží zcela na hráči. Poslední úroveň pak promění kód na zadávaný z klávesnice. V každé úrovni lze přidávat metody ze seznamu viditelného po levé straně. V tomto seznamu lze vyhledávat potřebné metody, ty jsou zde rozděleny podle kategorií.



Obrázek 12: Code Kingdoms: Prostředí tvorby modů

Formulář umí rozpoznávat chyby v kódu a překladač umí vyhodnotit problémy a vypsát je do konzole. Ve spodní části obrazovky lze sledovat výuková videa a zároveň sledovat postup plnění úkolu.

2.7.4 MOŽNOSTI NÁPOVĚDY

V úvodním kurzu je hráč prováděn krok po kroku tím, co musí udělat, aby dosáhl vytvoření programu. Veškerý postup je navíc ukázán a vysvětlen ve výukovém videu, což platí pro každý kurz této hry. Při tvorbě lze zobrazit nápovědu s ukázkou kódu po kliknutí na tlačítko otazníku. Formulář pro tvorbu kódu nabízí kontextovou nápovědu.

2.8 TIS-100

2.8.1 POPIS HRY

- Jazyk: angličtina.
- Dostupnost: shareware.
- Platforma: PC.

Hráč se stává programátorem zvláštního stroje, který zdědil po svém strýčkovi. Jelikož je tento stroj rozpracovaný, musí přijít na to, jak tento stroj dokončit.

Ve skutečnosti je tato hra sérií logických úkolů, jež je potřeba vyřešit pomocí programovacího jazyka Assembler. Úkoly jsou postupně těžší a těžší a jejich plněním se odemykají další. V každém úkolů lze najít dvanáct uzlů, z nichž každý pojme pouze omezený počet řádků kódu.

K dispozici jsou tři kategorie:

Hlavní kategorie, kde jsou zadané úkoly, a odemčení dalších vyžaduje vytvoření určitého počtu příkladů.

Druhá kategorie nabízí hráči vytvořit si nový program, kde nalezne vstupy a jen podle nich musí vytvořit program, který je zpracuje do požadovaných výstupů.

Třetí kategorie je hráči vytvořená bonusová kampaň, která se odemyká až po dokončení základní kampaně.

Všechny vytvořené programy se ukládají do složky ve formátu .txt.

2.8.2 ZAMĚŘENÍ HRY

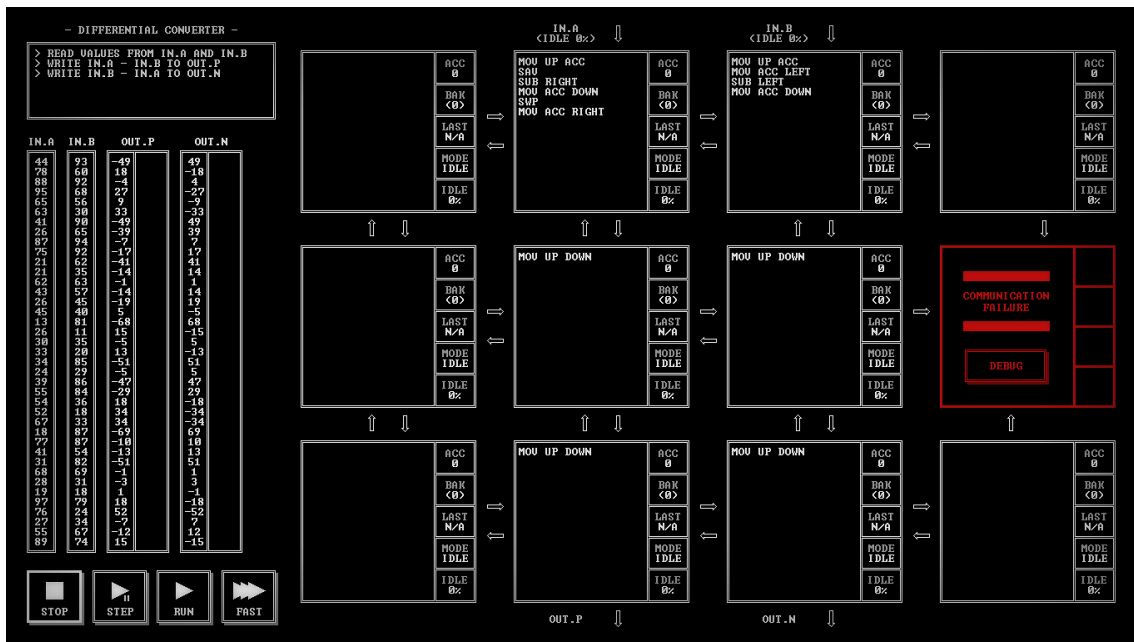
Hra je určena jak pro začátečníky, tak pro pokročilé. Manuál ke hře vysvětlí vše potřebné i úplným začátečníkům. Pro pochopení paralelního fungování všech uzlů je potřeba algoritmického myšlení, hra je tedy spíše pro studenty středních škol.

Pracuje se zde s programovacím jazykem Assembler ve zjednodušené podobě. Jeho pomocí lze naprogramovat cykly i podmínky a pracovat s matematickými operacemi.

2.8.3 UŽIVATELSKÁ PŘÍVĚTIVOST

Tato hra nenabízí příliš možností, prostředí je jednoduché na pochopení a z hlediska grafického rozhraní je prostá. Vlevo nahoře lze spatřit popis úkolu, pod ním vstupy a očeká-

vané výstupy. Všechny vstupní a výstupní uzly jsou viditelně označeny. V průběhu programu lze vidět, která instrukce je právě prováděna a jaké výstupy program tiskne.



Obrázek 13: TIS-100: Tvorba programu

Při programování hra kontroluje špatně zadané instrukce a na každou z nich upozorní. Provádění programu lze procházet krok po kroku, ale pouze vpřed. Pro úpravy je potřeba provádění vždy zastavit a vrátit na začátek.

2.8.4 MOŽNOSTI NÁPOVĚDY

Veškeré informace potřebné k hraní této hry lze nalézt v příloženém manuálu, včetně seznamu instrukcí a vysvětlení všech programových konstrukcí. Při tvorbě programů si hráč musí poradit sám.

2.9 SCREEPS

2.9.1 POPIS HRY

- Jazyk: angličtina.
- Dostupnost: shareware.
- Platforma: PC, webový prohlížeč.

Zde se stává hráč majitelem kolonie organismů, které potřebují jeho pomoc při přežití. Musí se zde starat nejen o získávání surovin, nýbrž i o ochranu před protihráči, kteří se usídlí v jeho blízkosti. Tito hráči jsou skuteční lidé a může přežít pouze nejsilnější – v tomto případě nejsilnější programátor. Činnost organismů závisí čistě na programech, které hráč naprogramuje. Cílem hry je přežít, dokud je hráč ve hře nováčkem, žádný jiný hráč na něj nemůže zaútočit.

Hra probíhá na externích serverech, tudíž kolonie žije vlastním životem i v době, kdy právě nemáte hru zapnutou. Veškerá činnost probíhá podle zadaných programů, k jejichž tvorbě lze použít nejen formulář obsažený ve hře, nýbrž i externí vývojové prostředí podle výběru hráče.

2.9.2 ZAMĚŘENÍ HRY

Tato hra se zaměřuje především na hráče, kteří už mají zkušenosti s programováním. Jediným jazykem využívaným ve hře je JavaScript, který zde není vyučován od základů, používá se pouze jako prostředek pro hraní a hra slouží pro procvičení daného jazyka. Pro samotnou výuku jazyka nabízejí vývojáři interaktivní kurz a podrobnou dokumentaci ke hře.

2.9.3 UŽIVATELSKÁ PŘÍVĚTIVOST

Hlavní barvou uživatelského rozhraní je šedá, celá vizuální stránka hry je poměrně tmavá a nedoprovází ji žádná hudba. Rozložení prvků na herní obrazovce není nijak nepřehledné, veškeré důležité prvky jsou umístěny do dvou hlavních sekcí (po pravé straně a dole). Samotný průběh hry se odehrává pomocí animace v herním okně, kdy hráč může vidět práci svojí kolonie, grafika je velmi úsporná, pouze 2D zobrazení.



Obrázek 14: Screenshot: Hlavní obrazovka

Pro samotné zadávání kódu lze využít možnosti vývojového prostředí třetí strany. Formulář obsažený ve hře neobsahuje kontrolu syntaxe, pouze vrací chyby v konzoli, pokud nějaké nastanou. Hráč má možnost tvořit programy a dělit je do jednotlivých modulů (které lze brát jako třídy) nebo zadávat příkazy přímo do konzole.

2.9.4 MOŽNOSTI NÁPOVĚDY

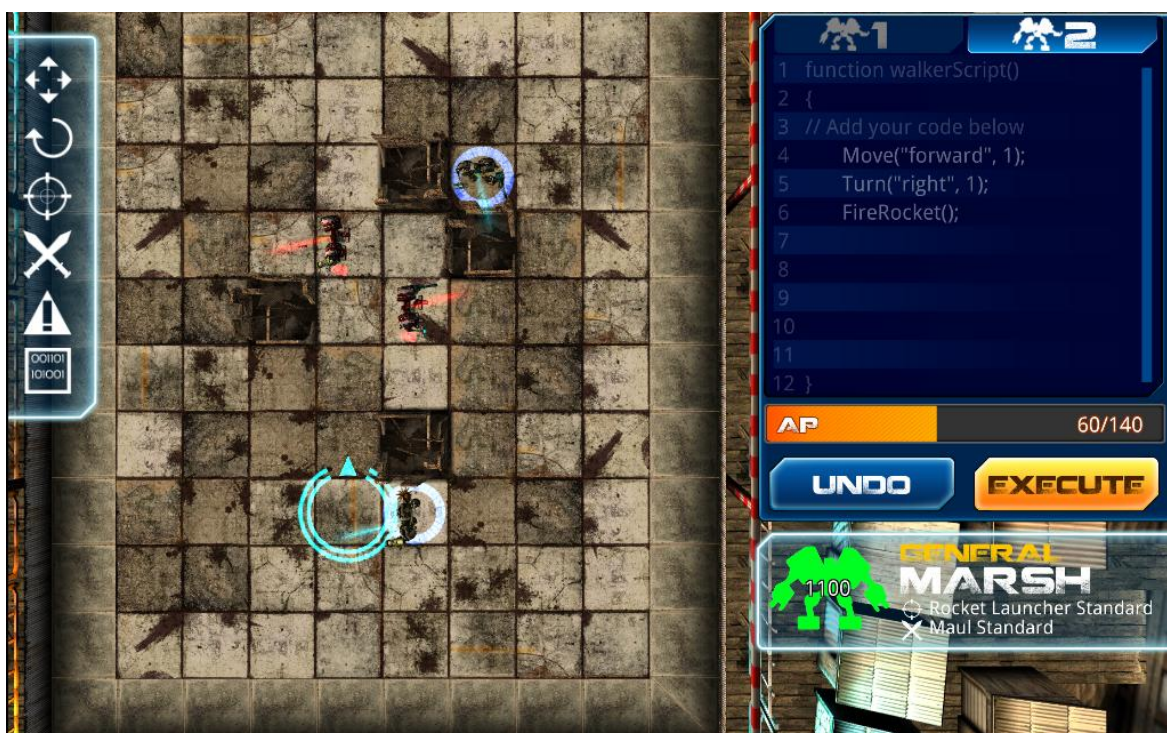
Pro vytváření programů lze použít externí vývojové prostředí, které může obsahovat určitou formu nápovědy. Samotná hra nabízí úvodní tutoriál pro vytváření skriptů a dokumentaci ke všem použitelným metodám.

2.10 CODE WARRIORS

2.10.1 POPIS HRY

- Jazyk: angličtina.
- Dostupnost: zdarma.
- Platforma: webový prohlížeč.

Hra z prostředí bojových robotů, jejichž pohyby a útoky hráč ovládá kódem. Cílem každého levelu je zničit všechny nepřátelské roboty nebo zničit jejich zdroj energie. Souboj se odehrává prostřednictvím tahů v aréně podobné šachovnici, po níž se mohou roboti pohybovat volně, pokud na daném políčku neleží žádná překážka. Hra nabízí čtyři úrovně lišící se pouze podle poskytnuté nápovědy pro hráče ve formě automatického doplňování zadaného kódu do formuláře. Nejnižší level umožňuje zadávat kód pouze pomocí grafického rozhraní.



Obrázek 15: Code Warriors: Programování útoku

Hodnocení hry je prováděno pomocí tří hvězdiček, které jsou udělovány za způsob splnění úkolu. Hlavním faktorem jsou tzv. „akční body“, kterých má každý hráč omezený počet a při plnění každého levelu by jich hráč měl spotřebovat co nejméně. Od toho se odvíjí způsob, jakým lze provádět jednotlivé úkoly, počet řádků kódu nemusí být přímo úměrný počtu spotřebovaných akčních bodů a naopak. Hodnocení se liší podle jednotlivých módů.

Hra obsahuje v každé úrovni tyto módy:

Trénink: Jeden výukový level, který hráče provede plněním a ukáže mu, jak hra funguje.

Souboj: Hráč se utká s nepřátelskými roboty a musí je porazit s pomocí co nejméně akčních bodů.

Bludiště: Úkolem hráče je zde najít nejkratší cestu ke zdroji energie a ten poté zničit. Význam hrají použité řádky kódu.

Rozhodnutí: Hra poskytne úsek kódu a hráč musí rozhodnout, jestli s jeho pomocí bojovník splní úkol, či nikoliv. Možnosti jsou pouze ano / ne.

Debugging: V zadaném kódu je chyba a hráč ji musí najít a opravit. V nejnižší úrovni není tento mód aktivní, jelikož v něm nelze zapisovat kód ručně.

Stealth: Robot musí projít přes arénu, aniž by ho některý z protivníků zahlédl a zničit zdroj energie. Hvězdičky se udělují podle spotřebovaných akčních bodů.

2.10.2 ZAMĚŘENÍ HRY

Hra používá jazyk JavaScript, příkazy pro bojovníky jsou ve formě předdefinovaných funkcí s parametry či bez nich. Nelze zde používat podmínky, cykly ani žádné jiné funkce, než ty nabízené. Pro pohyb robotů a hraní hry jsou nabízené funkce dostačující, pro výuku programování se toho zde nachází velmi málo. Hra se tudíž hodí pro začátečníky, kteří nemají žádnou zkušenost s programováním a získané znalosti se hodí pro další vzdělávání v daném jazyce, pro vytvoření vlastního programu mimo hru nikoliv.

2.10.3 UŽIVATELSKÁ PŘÍVĚTIVOST

Tato hra běží na enginu Unity a nabízí propracovanou grafickou stránku včetně doprovodné hudby. Hra nabízí několik arén, které se liší svým vzhledem, veškeré pohyby a útoky robotů jsou vidět v každém tahu a liší se svou animací. Jejich pohyby jsou nicméně trochu nemotorné a každý tah působí zdlouhavě bez možnosti přeskočit jeho provádění. Ovládací prvky jsou dobře viditelné, jejich označení není matoucí. Při provádění tahu vidí hráč, které příkazy se v danou chvíli provádějí.



Obrázek 16: Code Warriors: Provádění tahu

Samotný formulář pro zadávání kódu působí trochu zanedbaně, v případě opravy některého příkazu je obtížné umístit kurzor na správné místo pomocí myši. Během přepínání mezi jednotlivými bojovníky se občas nezachová zadaný kód, především když opakujeme vybraný level znovu nebo kód významnějším způsobem upravujeme, tj. mazání příkazů nebo úprava parametrů.

2.10.4 MOŽNOSTI NÁPOVĚDY

Pro každou ze čtyř úrovní hra nabízí úvodní tutoriál, kde vysvětlí princip hry a způsob ovládání bojovníků vztahující se k dané úrovni. Během hry lze získat nápovědu ke každému prvku na herní obrazovce pomocí tlačítka „?“ v pravém horním rohu obrazovky. Při chybě v zápisu zobrazí překladač kódu chybu včetně čísla řádky, na kterém se chyba objevila. Řádek se rozsvítí červeně.

Úroveň 1: Kód pro ovládání bojovníků nelze zapisovat ručně, veškerý kód se zadává pomocí tlačítek na levé straně obrazovky a označením políčka na v aréně. Příklad: pro pohyb na určité políčko je nutno kliknout na ikonu pohybu a vybrat jedno ze zeleně označených políček v aréně. Příslušný řádek kódu se do formuláře vyplní automaticky. Popis útoku lze zjistit po stisknutí tlačítka „?“ v pravém horním rohu obrazovky.

Úroveň 2: Kód lze od této úrovně zadávat ručně a možnost zadávat ho pomocí ikon pro ovládání bojovníka mizí. Formulář nabízí automatické doplnění zadávaného příkazu zmáčknutí klávesy *mezerník*. Po kliknutí na ikonu ovládání bojovníka se od této úrovně zobrazí nápověda daného příkazu.

Úroveň 3: Pro metody nabízí formulář automatické dokončení, pro parametry nikoliv.

Úroveň 4: K zapisovanému kódu nenabízí hra žádnou formu automatického dokončování.

3 POROVNÁNÍ VYBRANÝCH HER

3.1 JAZYK HRY

Human Resource Machine – čeština, angličtina, němčina, francouzština, ruština...

CodeMonkey – čeština, angličtina, němčina, francouzština, ruština, španělština...

CodeCombat – čeština, angličtina, němčina, francouzština, ruština, španělština...

CodinGame – angličtina, francouzština

CheckIO – angličtina, některá zadání úkolů v různých jazycích

Empire of Code - angličtina

Code Kingdoms - angličtina

TIS-100 - angličtina

Screeps - angličtina

Code Warriors - angličtina

3.2 ZAMĚŘENÍ HRY

Uvedená vhodnost podle úrovně dosaženého vzdělání je pouze orientační a představuje nejnižší doporučenou hranici. S výjimečně nadanými jedinci se nepočítá.

3.2.1 ZAČÁTEČNÍCI

CodeMonkey

Hra se hodí pro děti na základních školách. Výuka programování probíhá od úplných základů a velmi zvolna přidává nové konstrukce jako jsou podmínky nebo cykly. Veškeré techniky jsou dobře vysvětleny a vždy má hráč možnost si je vyzkoušet na lehkých příkladech. Celá hra je zabalena do velmi příjemného grafického rozhraní a hráč zde svým programem ovládá malou opičku, která sbírá banány.

CodeCombat

Zaměření pro žáky základní školy i studenty středních škol. Vyučuje programování od úplných začátků s možností volby programovacího jazyka. Programovací formulář a ovládání celé hry je zde o něco složitější, žáci prvního stupně by s ním mohli mít s pochopením problémy. Veškeré použité programové konstrukce jsou ve hře dobře vysvětleny a

v začátcích nabízí hra celé řešení. Programové konstrukce jsou dávkovány postupně, začíná se vždy jednoduchými a dobře vysvětlenými úkoly.

Code Kingdoms

Ve hře jsou kurzy rozděleny podle věku. Ty nejsnazší jsou pro hráče od osmi let. Tato hranice platí pouze pro anglicky mluvící národy, jelikož výuková videa, která každý kurz doprovázejí obsahují pokročilou angličtinu v psané i mluvené podobě.

Code Warriors

Pro žáky základní školy i studenty středních škol. Hra nenabízí žádné složité programové konstrukce. V celé hře se používají stejné funkce, některé z nich závisí na tom, jaké zbraně roboti používají.

Human Resource Machine

Pro studenty středních škol. Programování vyučuje od základů, používá speciální jazyk formou skládání grafických prvků. Hráč začíná s několika málo prvky, které postupně přibývají a s nimi přicházejí i náročnější úkoly. Zvyšování obtížnosti není tak přívětivé jako výše popsaných her a relativně náročné matematické úlohy přicházejí už po několika levelech. Tato obtížnost už by pro žáky základní školy nebyla vhodná, záleží hodně na logickém i algoritmickém myšlení.

TIS-100

Pro studenty středních škol. Hra pracuje se zjednodušenou verzí programovacího jazyka Assembler. Obtížnost příkladů se zvyšuje velmi rychle, hra se zaměřuje především na logické myšlení a algoritmizaci, princip jazyka Assembler zde zůstává zachován. Přístup k výuce je zde podobný jako u Human Resource Machine, úkoly jsou hlavně matematicky založené. I když jsou grafické rozhraní a způsob zadávání kódu na první pohled zcela odlišné, několik podobností lze najít (např. matematické funkce a podmínky).

3.2.2 POKROČILÍ

CheckIO, Empire of Code, CodinGame

Pro studenty vysokých škol, případně studenty středních škol s dobrým základem v konkrétních jazycích. Všechny tři hry předkládají logické úlohy a požadují po hráči jeho vlastní řešení v konkrétním programovacím jazyce. Kromě CodinGame, kde si může hráč vybrat

příklad podle vlastního uvážení, dávkuje hry příklady postupně podle obtížnosti, možnost výběru je zde omezená. V případě CheckIO si může vybrat kterýkoliv příklad, které nabízí ostrovy, jež má hráč odemčené. V Empire of Code záleží na budovách, které má hráč postavené ve své kolonii a které chce rozšiřovat. Všechny tři hry rozdělují příklady podle obtížnosti, předkládají hráči, na jaké programové konstrukce při řešení příkladu narazí a každý příklad je řádně popsán, včetně vstupů, které bude vytvořený program zpracovávat.

Screeps

Pro studenty vysokých škol se zkušenostmi v programovacím jazyce JavaScript. Tato hra neobsahuje žádné konkrétní příklady, způsob, jakým se hráč bude starat o svou kolonii, je závislý čistě na jeho rozhodnutí. Hra obsahuje tutoriál, který hráče naučí, jak hra funguje a naučí ho základy programů a příkazů, které lze ve hře využít. Veškeré další možnosti jsou popsány v dokumentaci hry.

3.3 PROGRAMOVACÍ JAZYKY

3.3.1 FORMA PROGRAMOVACÍHO JAZYKA

Skutečný programovací jazyk

CodinGame

Kterýkoliv jazyk, který hra nabízí, vypadá skutečně a lze ho použít i mimo prostředí hry. V případě jazyka Java lze spatřit importované knihovny, deklarace třídy i rozdělení kódu na metody. Vstup se do programu obvykle načítá pomocí knihovny *Scanner*, která čte vstup v textové podobě, který zde zajišťuje sama hra. Sekce kódu s instancí této knihovny bývá většinou v příkladu připravena předem. Výše uvedené prvky se objevují v každém podporovaném jazyce v podobě, kterou jazyk vyžaduje. Všechny výstupy musí být vytisknuty na obrazovku pomocí *System.out.print()*.

Code Kingdoms

Hra podporuje pouze jazyk Java a všechny mody do hry Minecraft musí být napsány v tomto jazyce. Kód je rozdělen na třídy, které lze libovolně vytvářet i pojmenovávat (záleží na podmínkách daného kurzu). Bez ohledu na podobu, kód by šel využít i mimo herní prostředí. Z pochopitelných důvodů by kód ve vývojovém prostředí třetí strany nefungoval jako ve hře Minecraft.

CheckIO, Empire of Code, CodeCombat

Tyto hry používají dva jazyky – JavaScript a Python. Code Combat používá navíc CoffeeScript a Lua, nicméně forma zadávání je stejná. Veškeré potřebné vstupy jsou zadávány pomocí speciální metody, která je už v kódu připravená.

Screeps

Od výše uvedených her se liší tím, že obsahuje pouze JavaScript, kód lze použít i mimo herní prostředí, stejně jako v případě Code Kingdoms by ovšem neplnil svou funkci. Jednotlivé části kódu lze zadávat i jednotlivě pomocí konzole, tato funkce se využívá pouze pro příkazy, které se nemusí opakovat.

CodeMonkey

Využívá jazyk zvaný CoffeeScript, což je skutečný programovací jazyk založený na JavaScriptu. Pro zjednodušení využívá některé konstrukce inspirované Pythonem a Ruby. Nabízí „čistší“ funkcionální zápis (Malý, 2010). Tento jazyk je úplný a nabízí podmínky i cykly.

Zjednodušení

Code Warriors

Používá zjednodušenou verzi JavaScriptu, kde se pohyb a útoky robotických bojovníků ovládají pomocí speciálních funkcí, jejichž tělo nelze nikde spatřit. Nejedná se o příkazy, nýbrž opravdu o funkce, jelikož se zde zapisují atributy, případně prázdné závorky. Nevyskytují se zde žádné cykly ani podmínky.

Human Resource Machine

Program se skládá z grafických bloků, vlastní programovací jazyk. V programu lze použít cykly, podmínky i matematické operace. Podmínky jsou zde natvrdo určeny (např. pokud je výsledek nula) a fungují jako skoky v programu, tj. pokud je podmínka splněna, program přeskočí na požadované místo. Tyto podmínky lze využít jako cykly a skočit například na začátek programu, dokud není podmínka splněna.

TIS-100

Hra využívá zjednodušenou verzi programovacího jazyka Assembler. Místo přesouvání dat z registrů je lze přesouvat pouze nahoru, dolů, doleva a doprava. Počet instrukcí je zde také výrazně zmenšený., avšak některé jsou vytvořeny speciálně pro hru. Kromě těchto změn a omezení se jazyk chová jako ve skutečnosti, podporuje jak podmínky, tak cykly.

3.3.2 ZPŮSOB ZADÁVÁNÍ

Human Resource Machine

Program se skládá z grafických prvků, které obsahují název funkce. Ručně se nic nezapisuje.

Code Warriors

V nejnižší úrovni se kód zadává pouze pomocí myši. Po kliknutí na příkaz se ve formuláři vpravo objeví funkce. Při pohybu vybírá hráč políčko na herní mapě, podle toho, kam se chce s bojovníkem pohnout.

CodeMonkey

Obsahuje možnost kód vepsat ručně nebo pomocí grafických prvků. Číselné hodnoty se musí vždy zapisovat ručně.

Code Kingdoms

Umožňuje zadávat kód pomocí myši, přičemž se nepoužívají grafické prvky ve stejné podobě jako ve výše zmíněných hrách. Jedná se zde o interaktivní verzi jazyka Java, kde se veškeré metody a programové konstrukce vybírají ze seznamů. Způsob zadávání lze změnit na ruční.

CodinGame, CodeCombat, Screeps, TIS-100, CheckIO, Empire of Code

Kód se zadává pouze ručně.

3.4 POMĚR HRY A PROGRAMOVÁNÍ

Důležitým aspektem při porovnávání her je poměr herního prvku (tj. grafické rozhraní, možnost ovládat fiktivní postavu, příběh hry apod.) a prvku programování (tj. tvorba kódu, překlad programu, kontrola chyb v kódu apod.). Liší se také přístup k úspěchům ve hře, zda jde pouze o to, kolik toho hráč naprogramoval nebo také o to, jak postupuje hrou se svou fiktivní postavou nebo fiktivní kolonií. Tato část je v porovnání s ostatními nejvíce detailní, aby čtenář pochopil, o co se v nich jedná.

CodinGame

V této se hráč setkává pouze s programováním, neovládá žádnou postavu, nežije ve virtuálním světě, každý zde hraje sám za sebe, může se utkat s ostatními hráči a poměřit s nimi své znalosti. Postupným plněním úkolů získává hráč zkušenosti, kterými ovšem není samotný obsah hry ovlivněn, jde pouze o srovnání s ostatními hráči a získávání úspěchů.

TIS-100

Hráč na první pohled hraje sám za sebe a programuje ve velmi prostém grafickém rozhraní. Na pozadí se nicméně skrývá příběh, který odhaluje pravdu o skutečné funkci zařízení, jež pomocí svých programů hráč opravuje.

CheckIO

Mírné rozšíření k samotnému programování nabízí CheckIO, které přidává k plnění úkolů objevování nových ostrovů a odemykání nových úkolů. Opět hraje hráč sám za sebe a postupným plněním úkolů, které postupně zvyšují obtížnost, postupuje ve hře, dokud neobjeví celý herní svět, neodemkne a nesplní všechny úkoly.

Empire of Code

Stejně jako CheckIO pracuje na principu plnění úkolů, u kterých se postupně zvyšuje obtížnost a jejich plnění odemyká nové prvky hry. Na rozdíl od ní ovšem přidává zásadní herní prvek, což je budování vlastní kolonie. Hráč z pohledu druhé osoby vidí vlastní městečko, které postupně rozšiřuje s pomocí herních peněz a zkušeností. Rozšiřování budov závisí na znalostech programování hráče. Jako další podstatný prvek jsou zde bitvy s ostatními hráči. Lze naprogramovat vlastní útoky a obranu, kdy útoky probíhají pomocí vojáků, kteří útočí na konkurenční kolonii a obrana za pomoci vojáků a speciálních obranných budov, které lze rovněž programovat.

Screeps

O vlastní kolonii pečuje hráč i ve hře Screeps. Zde však narazíme na několik zásadních odlišností. Za prvé, grafické rozhraní je zde výrazně prostší a programovací část mnohem výraznější. Hráč zde programuje veškerý pohyb a práci svých organismů v kolonii - zvaných ‚Creeps‘. Od toho se odvíjí i druhý zásadní rozdíl, hráč se nestará o město, nestaví budovy, vytváří pouze body na herní mapě a cesty. Pro představu by šla tato hra srovnat

s kolonií mravenců. Třetím zásadním rozdílem je skutečnost, že kolonie si i po odhlášení hráče žije vlastním životem a pracuje přesně podle vytvořených programů. Mezi podobné prvky patří možnost ocitnout se pod útokem a zaútočit vlastními organismy na nepřítele. Cílem hry je růst kolonie a přežití.

Code Kingdoms

Tato hra je stejně jako v jiných aspektech trochu speciálním případem. Samotná hra žádné herní prvky neobsahuje, nicméně její hlavní součástí je hra Minecraft, kde lze použít všechny naprogramované mody a běžně s nimi hrát. Přímo v Code Kingdoms jsou pro hráče připraveny úspěchy a herní zkušenosti.

CodeMonkey

Hráč zde do jisté míry přebírá roli malé opičky, případně jiných zvířátek. Hra s postupem času nabízí hráči stále složitější úlohy a veškerá řešení lze sledovat přenesená do pohybu opičky v příjemné animaci určené především pro děti. V této hře opět více převládá programování než hra, nicméně její stylizace působí mnohem příjemněji, i když neobsahuje žádné skutečně herní prvky, veškerý děj a postup se zakládá pouze na vytvořených programech.

Code Warriors

Tato hra působí mnohem méně jako výuka programování než předchozí hry. Hráč zde skutečně „hraje hru“ a programování pouze využívá jako ovládací prvek. Jak moc obtížné je samotné programování, to už záleží na vybrané obtížnosti, hra samotná zde převládá. Hráč bojuje s vlastním týmem robotických bojovníků proti týmu nepřítele, buď počítače, nebo skutečného hráče. Jak bude souboj probíhat, to závisí stále jen na vytvořených programech.

Human Resource Machine

Jako první z výše popsaných her nabízí skutečný příběh a hráč vidí svou postavu, jejíž osud má ve svých rukou. Jako zaměstnanec obří firmy se hráčova postava musí potýkat se stále obtížnějšími zadáními ze strany vedení a řešení závisí na vytvořeném algoritmu složeného z grafických prvků. I když hra neobsahuje žádný skutečný jazyk, posloupnost grafických prvků se dá považovat za skutečný program, jehož plnění vidí hráč na obrazovce pokaždé, když ho spustí. Jeho postava v tuto chvíli začne pracovat se zadanými čísly, přenášet je

z místa na místo a výsledky odesílat. Matematických operací se zde nachází dostatek. Podstatným rozdílem jsou zde meziscény, které popisují herní příběh, hráč se může do své postavy více vžít a získat motivaci. Během každého levelu může hráč požádat o nápovědu svého nadřízeného, který sedí za stolem a na pracovníka dohlíží.

CodeCombat

Na tuto hru lze nahlížet jako na běžnou hru, která mimo jiné zvládá učit programování. Hráč zde získává svou vlastní postavu, kterou může v průběhu hry rozvíjet, získávat a dokupovat pro ni nové vybavení, upravovat vzhled a sledovat její cestu celou hrou složenou z několika různých světů, které se postupně odemykají během hraní. Postava hráče získává zkušenosti a s každým novým předmětem dostává hráč možnost použít nové metody. Veškerý pohyb postavy, její útoky a schopnosti se ovládají pomocí vytvořeného programu.

4 NÁVRH ROZŠIŘUJÍCÍCH PŘÍKLADŮ PRO VYBRANOU HRU

4.1 VÝBĚR HRY

Zvolená hra: CodinGame

Pro výběr hry byla rozhodující tato kritéria:

Dostupnost pro běžného uživatele vytvořit dodatečný obsah

Hra musí nejen nabízet možnost vytvořit dodatečný obsah, musí to nabízet skutečně každému běžnému hráči.

Tvorba příkladu pro hru, nikoliv za pomoci hry

Některé hry jako například Code Combat nabízejí pro hráče možnost vytvořit vlastní hru za pomoci jejich vývojového prostředí a navíc poskytnout potřebné znalosti. V praktické části této práce nicméně nejde o to pouze využít některou z her pro vytvoření konkrétního příkladu, nýbrž o vytvoření takového příkladu, který by mohli běžní hráči, kteří danou hru hrají skutečně absolvovat a získat z něj nějaké nové znalosti.

Návrh i realizace

Posledním kritériem bylo, aby mohl být daný příklad skutečně vytvořen a zpřístupněn ostatním hráčům.

Všechna výše zmíněná kritéria splnila pouze jediná z testovaných her - CodinGame. Tato hra obsahuje širokou škálu příkladů pro hráče, z nichž část vytvořili i sami uživatelé. Principem hry je zadat hráči určitou výzvu a dát mu určitý čas na jeho vyřešení. V případě kategorie pro procvičování je tento čas neomezený, v případě soutěžní kategorie nazývané ‚Clash of Code‘ dostávají hráči přesně pět minut na vytvoření svého řešení a vítězí ten, který to stihne nejrychleji. Vytvoření příkladu je možné právě pro tyto dvě kategorie.

Byla vybrána kategorie ‚Clash of Code‘, kde jsou příklady jednodušší a hráči v ní soupeří o to, kdo vytvoří program nejrychleji. Cílem bylo, aby příklad zvládli studenti středních škol, kteří mají jen základní znalosti o jazyce Java a procvičili si při jeho řešení podmínky a cykly.

4.2 NÁVRH PŘÍKLADU

Při návrhu byl jako inspirace použit článek na blogu hry CodinGame, který byl sepsán za účelem podat pomocnou ruku každému, kdo by chtěl vymyslet svůj vlastní příklad, ale nemá zkušenosti s jeho vytvářením.

Článek doporučuje každému, kdo neví, jak s vytvářením svého příkladu začít, zamyslet se nad svými koníčky, zájmy a vytvořit příklad, který se týká něčeho, co ho opravdu baví, jelikož s dobrým tématem se bude bavit nejen člověk při vytváření, ale i hráč, který bude zadaný úkol plnit.

Jako další přicházejí testy pro daný příklad. Jinými slovy, tvůrce musí připravit takové testovací vstupy, aby prověřil funkčnost zadaného kódu i v případech, které nejsou běžné. Hráči mají při plnění úkolů relativní svobodu, musejí ho pouze plnit ve vybraném jazyce a výstup musí mít podobu výpisu na obrazovku, veškeré ostatní metody mají hráči plně ve svých rukou. Každý dobrý příklad by měl předcházet „natvrdo“ zadaným výstupům, program by měl být skutečně univerzální pro jakýkoliv zadaný vstup. Pomocí zadaných testovacích vstupů si může tvůrce přizpůsobit náročnost vytvářeného příkladu. Lze tím zjednodušit příklad tak, aby byl zvládnutelný v daném čase, ale zároveň nabídl hráčům adekvátní výzvu (např. hráči ušetří čas, když nemusí ošetřit dělení nulou).

Hra požaduje od každého tvůrce jeho vlastní řešení příkladu, aby se předešlo situaci, kdy je příklad skutečně nesplnitelný, ať už z hlediska samotného problému nebo konkrétních zadaných vstupů. Toto řešení může být zadáno v jakémkoliv z nabízených jazyků. Vývojáři CodinGame ho požadují čistě pro ověření splnitelnosti zadání, pro hráče to nicméně představuje možnost přizpůsobit zadání problému nebo vstupy podle toho, jak dlouho jemu samotnému zabralo vytváření daného řešení. Pro ‚Clash of Code‘ je zásadní, aby příklad šel vyřešit v časovém limitu nejvýše patnácti minut.

Prezentace příkladu může být zásadní pro to, jestli si ho hráč vybere nebo. Zajímavý název nebo malý příběh, který zasazuje příklad do reálné situace, může vytvořit z běžného problému lákavou výzvu. Hráč by ze zadání měl pochopit, co se po něm chce a zároveň získat motivaci to splnit. Na dobrém zadání závisí i skutečnost, jestli bude příklad přijat vývojáři i komunitou, aby se mohl stát součástí hry.

Po vytvoření příkladu a řešení každému tvůrci pomůže zpětná vazba. Článek doporučuje před odesláním příkladu do hry požádat o jeho vyřešení kamaráda, sourozence nebo kolegu. Pokud by tvůrce neměl tuto možnost, dalším způsobem, jak získat zpětnou vazbu, je projít si řešení problému na internetu a ověřit si, zda neexistuje lepší řešení nebo nějaký kontrolní vstup, který ho nenapadl. Kombinace obojího je z pochopitelných důvodů nejlepší.

Při vytváření příkladu v této práci byly nejprve posuzovány zájmy autora i užitečnost příkladu. Nakonec hrály roli především příklady, které byly vytvářeny na hodinách programování na Katedře výpočetní a didaktické techniky při Fakultě pedagogické i na Katedře informační a výpočetní techniky při Fakultě aplikovaných věd Západočeské univerzity. Mezi těmito příklady se objevily i převodníky soustav, které plně vyhovovaly požadavkům na složitost příkladu. Aby to nebyl jen obyčejný převodník soustav a mohl prospět i lidem bez zájmu o informační technologie, byl nakonec zvolen převodník římských číslic do desítkové soustavy.

Příklad byl vytvořen v jazyce Java. Při vytváření nebyly použity žádné složité datové struktury, pouze jednoduchý cyklus *for* a podmínková konstrukce *switch*. Jako proměnné byly využity celočíselný typ *integer*, znak *char* a řetězec *string*. Při ověřování jiných řešení, které byly hledány na webu *Stackoverflow*, byla nalezena řešení s použitím seznamu nebo využívající dvě metody. V řešení této práce byla použita pouze jedna metoda a základní konstrukce, které odpovídají pokročilejším znalostem studentů na střední škole. Tyto znalosti se nemusí týkat pouze Javy, nýbrž i jiných jazyků, ve kterých lze v CodinGame řešit příklad.

4.3 REALIZACE PŘÍKLADU

Program přečte ze vstupu římské číslo a uloží ho do proměnné typu *string*. Tato proměnná je v cyklu *for* procházena znak po znaku. Cyklus má tolik iterací, kolik znaků je načteno. V první iteraci konstrukce *switch* ověřuje, zda se v řetězci vyskytuje konkrétní písmeno (tj. X, L, M...). V případě výskytu se ke konečnému výsledku typu *integer* přičte konkrétní hodnota. Výsledek je na začátku cyklu nulový.

V dalších iteracích už je konstrukce *switch* složitější. Kromě přičítání konkrétní hodnoty k výsledku také ověřuje předcházející znak, který snižuje hodnotu daného znaku (napří-

klad $X = 10$, $IX = 9$). V případě, že se před ověřovaným znakem nachází konkrétní znak, který sníží jeho hodnotu, přičte se ke konečnému výsledku nižší číslo.

V případě, že se ve vstupu vyskytne znak IX , výsledek v tomto případě bude číslo 9. Při procházení cyklem se nejprve přičte číslo 1 a v další iteraci se po kontrole předcházejícího znaku přičte pouze 8. Program díky tomuto ověření předcházejícího znaku nemusí kontrolovat znaky IX , CM apod. zvláště a je tak kratší. Vše se navíc odehrává v jediné metodě.

ZÁVĚR

Hru samotnou lze definovat několika způsoby a v kombinaci s definicí učení lze popsat výukové hry se zaměřením na programování. Hry se liší od výukových programů možnostmi relaxace, seberealizace a soustředí se na potřeby samotného hráče namísto obyčejného předávání informací. Pro výukové hry je typické předat hráči potřebné znalosti zábavnou formou, která ho psychicky nevyčerpá. Zkušenosti si může do hry už přinést, ale vždy by měl odcházet s více.

Hry se dělí na začátečníky a pokročilé, podle toho také přistupují k hráči a volí formu výuky. První kategorie vede hráče mnohem pevněji a na každou novou programovou konstrukci mu dává více času. Druhá kategorie dává hráči větší volnost, obtížnost se zvyšuje mnohem rychleji bez dlouhého vysvětlování daného problému. Hra se dá považovat za výukovou, pokud obsahuje širokou škálu obtížností a dává možnost hráči získat nové dovednosti.

Na trhu se nachází desítky her s různými přístupy k výuce a programovacím jazykům. Některé neobsahují žádné skutečné jazyky, přesto zlepšují algoritmické a logické myšlení. Do této práce jsem zvolil pouze hry, kde se nachází programovací jazyk, a tudíž probíhá výuka programování tak, jak v reálném životě vypadá (včetně zjednodušené podoby). V práci jsem popsal i další podoby her, které se týkají spíše počítačových sítí a tzv. hackerů. Během svého průzkumu jsem zjistil, že hry s podobnou tematikou stále přibývají a jejich nabídka je velká už dnes. Na vývoji se mohou podílet profesionálové i běžní uživatelé, ať už formou rozšiřujících příkladů nebo práce na hrách s otevřeným kódem.

Při průzkumu vybraných her byly zjištěny odlišnosti v zaměření her, uživatelském rozhraní a možnostech nápovědy. Veškeré aspekty byly ve větších detailech porovnány a rozděleny do několika kategorií – zaměření z hlediska věku a zkušeností, přístupu k programovacím jazykům a poměru herní a programovací části.

V praktické části se podařilo vytvořit příklad, který mohou zvládnout studenti středních škol se základy programovacího jazyka Java, případně jiného, který pro ně zároveň představuje výzvu. Příklad umožňuje několik možných řešení a záleží jen na hráči, jakou cestu si zvolí. Zvolená hra CodinGame umožní sdílet jeho řešení s ostatními hráči.

RESUMÉ

The main goal of this work was to compare chosen games and give the reader an idea about the market and what the games have to offer. This goal was achieved by getting familiar with available games, finding people who made similar research and coming up with a set of criteria to find games that will help make this work a benefit for readers.

Main focus of the examination of games was on main focus of the game, user experience and help possibilities. The comparison is more thorough. What was compared is how the games work with which age group and experience, how the games work with programming languages and how the game components prevail over the programming components and vice versa.

In practical part criteria to pick a game for my own expansion were chosen. The expansion was designed for students of high school with basic knowledge of Java or other programming language. The expansion was programmed in Java.

In the work all goals were achieved. The readers were given a possibility to compare games and the work also contains some information other similar games on the market. The work should give readers useful information about educational games with special focus on programming.

SEZNAM LITERATURY

ABZ.cz: *slovník cizích slov*. [Online] [Citace: 5. Duben 2017.] Dostupné z: <http://slovník-cizich-slov.abz.cz/web.php/slovo/didaktika>.

ČÁČKA, Otto. *Psychologie duševního vývoje dětí a dospívajících s faktory optimalizace*. Brno: Doplněk, 2000. ISBN 80-7239-060-0.

ČÁP, Jan. *Psychologie výchovy a vyučování*. Praha: Karolinum, 1993. ISBN 80-7066-534-3.

GAGNÉ, Robert M. *Podmínky učení*. 1. vyd. Praha: SPN, 1975. 287 s. Pedagogická teorie a praxe.

LANGMEIER, Josef a Dana KREJČÍŘOVÁ. *Vývojová psychologie*. Praha: Grada, 1998. Psyché (Grada). ISBN 80-7169-195-X.

MALÝ, Martin. CoffeeScript: řádně oslazený JavaScript. *Zdroják*. [Online] 2. prosince 2010 [Citace: 9. Duben 2017.] Dostupné z: <https://www.zdrojak.cz/clanky/coffeescript-radne-oslazeny-javascript/>.

MAŇÁK, Josef a Vlastimil ŠVEC. *Výukové metody*. Brno: Paido, 2003. ISBN 80-7315-039-5.

PRŮCHA, Jan. *Moderní pedagogika*. Praha: Portál, 1998. 488 s. ISBN 978-80-7367-503-5.

SPŠE - Uchazeč SŠ - studijní obory. *SPŠE*. [Online] [Citace: 5. duben 2017.] Dostupné z: <http://www.spse.cz/obory-ss.php#top>.

Zaměření Programování a vývoj softwaru. *SŠINFIS*. [Online] [Citace: 5. duben 2017.] Dostupné z: <http://www.ssinfis.cz/index.php/2012-12-17-11-01-02?layout=edit&id=523>.

SEZNAM OBRÁZKŮ, TABULEK, GRAFŮ A DIAGRAMŮ

Obrázek 1: Human Resource Machine: Průběh programu	14
Obrázek 2: CodeMonkey: CoffeeScript	15
Obrázek 3: CodeMonkey: Uživatelské rozhraní	16
Obrázek 4: CodeCombat: Inventář	17
Obrázek 5: CodeCombat: Mapa postupu ve hře.....	18
Obrázek 6: CodeCombat: Uspořádání grafického rozhraní	19
Obrázek 7: CodinGame: Plnění příkladu	21
Obrázek 8: CheckIO: Herní mapa	23
Obrázek 9: CheckIO: Tvorba programu	23
Obrázek 10: Empire of Code: Kolonie	26
Obrázek 11: Code Kingdoms: Nabídka kurzů.....	28
Obrázek 12: Code Kingdoms: Prostředí tvorby modů	28
Obrázek 13: TIS-100: Tvorba programu	31
Obrázek 14: Screeps: Hlavní obrazovka	33
Obrázek 15: Code Warriors: Programování útoku	34
Obrázek 16: Code Warriors: Provádění tahu.....	36

PŘÍLOHY

- Dostupnost
 - Human Resource Machine
 - Běžná cena \$9.99
 - Možnost zakoupit na Steam, GOG, Humble Store
 - CodeMonkey
 - Trial verze zdarma
 - \$5.95 / měsíčně pro domácnosti
 - \$49 / ročně pro domácnosti
 - \$129 / ročně pro domácí školy
 - Pro třídu a školu na dotaz
 - CodeCombat
 - Základní verze hry obsahuje přes 100 úrovní a je zdarma
 - \$9.99 / měsíčně za Premium verzi
 - Premium obsahuje přes 330 úrovní, výuku tvorby her a tvorby webových stránek v HTML, CSS a jQuery
 - CodinGame
 - Zcela zdarma
 - CheckIO
 - Zcela zdarma
 - Empire of Code
 - Zcela zdarma
 - Code Kingdoms
 - Trial verze na 30 dní zdarma
 - €69.99 / ročně
 - TIS-100
 - Běžná cena \$6.99
 - Možnost zakoupit ve službě Steam, GOG, Humble Store
 - Screeps
 - Běžná cena \$14.99
 - Možnost zakoupit ve službě Steam, Humble Store
 - Code Warriors
 - Zcela zdarma

- Řešení rozšiřujícího příkladu

```
import java.util.Scanner;

public class RomanToDEC {

public void convert(String roman){
    Scanner sc = new Scanner();
    roman = sc.next();
    int dec = 0;
    for (int i = 0; i < roman.length(); i++){
        if (i == 0){
            switch(roman.charAt(i)){
                case 'I': dec += 1;
                break;
                case 'V': dec += 5;
                break;
                case 'X': dec += 10;
                break;
                case 'L': dec += 50;
                break;
                case 'C': dec += 100;
                break;
                case 'D': dec += 500;
                break;
                case 'M': dec += 1000;
                }
        } else {
            switch(roman.charAt(i)){
                case 'I': dec += 1;
                break;
                case 'V':
                if (roman.charAt(i-1) == 'I'){
                    dec += 3;
                } else {
                    dec += 5;
                }
                break;
                case 'X':
                if (roman.charAt(i-1) == 'I'){
                    dec += 8;
                } else {
                    dec += 10;
                }
                break;
                case 'L':
                if (roman.charAt(i-1) == 'X'){
                    dec += 30;
                } else {
                    dec += 50;
                }
                break;
                case 'C':
                if (roman.charAt(i-1) == 'X'){
                    dec += 80;
                } else {
                    dec += 100;
                }
                break;
                case 'D':
                if (roman.charAt(i-1) == 'C'){
                    dec += 300;
                } else {
                    dec += 500;
                }
                break;
                case 'M':
                if (roman.charAt(i-1) == 'C'){
```

```
        dec += 800;
    } else {
        dec += 1000;
    }
    break;
}
}
}
System.out.println(dec);
}
}
```