STŘEDNÍ PRŮMYSLOVÁ ŠKOLA  
MLADÁ BOLESLAV

**ROČNÍKOVÁ PRÁCE**

Tadeáš Karban

Mladá Boleslav 2022

STŘEDNÍ PRŮMYSLOVÁ ŠKOLA  
MLADÁ BOLESLAV

**ROČNÍKOVÁ PRÁCE**

**Autor: Tadeáš Karban  
Studijní obor: 18-20-M/01 Informační technologie  
Vedoucí práce: Jan Till**

Mladá Boleslav 2022

Obsah

[1 Úvod 6](#_Toc102331283)

[2 Obsah Práce 7](#_Toc102331284)

[2.1 Programovací Jazyk 7](#_Toc102331285)

[2.1.1 JavaScript 7](#_Toc102331286)

[2.1.2 HTML 7](#_Toc102331287)

[2.1.3 CSS 8](#_Toc102331288)

[2.1.4 WebStorm 8](#_Toc102331289)

[2.2 Postup při tvořeni hry 8](#_Toc102331290)

[2.2.1 Canvas 9](#_Toc102331291)

[2.2.2 Vytvářeni Objektu a Mapa 9](#_Toc102331292)

[2.2.3 Vytvořeni Uživatele 10](#_Toc102331293)

[2.2.4 Ghost 10](#_Toc102331294)

[2.2.5 Kolize 11](#_Toc102331295)

[2.2.6 Start hry 11](#_Toc102331296)

[2.2.7 Konec Hry 12](#_Toc102331297)

[2.2.8 Reset hry 13](#_Toc102331298)

[2.3 Použití obrázky v Projektu 13](#_Toc102331299)

[2.3.1 Painet.net 13](#_Toc102331300)

[3 Závěr 14](#_Toc102331301)

[4 Přílohy 14](#_Toc102331302)

[4.1 Seznam obrázků 14](#_Toc102331303)

[4.2 Zdroj 14](#_Toc102331304)

**Prohlášení**

Prohlašuji, že jsem svou ročníkovou práci vypracoval samostatně a použil jsem pouze podklady (literaturu, projekty, SW atd.) uvedené v přiloženém seznamu.

Nemám závažný důvod proti zpřístupňování této ročníkové práce v souladu se zákonem č. 121/2000 Sb., o právu autorském, o právech souvisejících s právem autorským a o změně některých zákonů (autorský zákon) v platném znění.

V Mladé Boleslavi dne podpis:

# Úvod

Cílem této práce bylo vytvořit Webovou hru za použiti Javascriptu. Vytvářeni hry  
není snadné, protože je nutná znalost v základech v programovaní a je třeba mít zkušenosti s prací textur, obrázků. Většina těchto obrázku se vytváří v nástrojích určených pro jejich tvorbu třeba paint.net.

Důvod zvoleni teto práce je, že tvorba her je zajímavá a je možné získat spousta znalosti a zkušenosti z různých oblasti, jako například programováním. Tyto znalosti se můžou hodit při uplatnění v budoucnosti. Při prací této hra byla, použito i dalších jazyků HTML, CSS.

Zvoleni této práce bylo také použít vzor originální hrou PacMan, která se už objevila na internetu ještě minulem století. PacMan se také stal nejpopulárnější hrou na světě. Stala se symbole pro mnoha her.

Práce je strukturována tak, že nejdříve jsou vysvětleny základní pojmy z oblasti programovaní jejichž znalost je nutná pro pochopení problematiky této práce.

Další kapitola se věnuje tvorby projektu z pohledu návrhu, podle kterého by se poté měla hra vytvářet. Je zde popsána základní myšlenka, principy a pravidla. Současně jsou zde popsány způsoby, jakými se hra ovládá, a také základní herní mechanismy.

V závěrečné kapitole je vytvářená hra popsána z pohledu výsledného produktu.

# Obsah Práce

V následující části jsou vysvětleny základní pojmy, z oblasti programovaní

## Programovací Jazyk

Pod pojmem programovací jazyk rozumíme prostředek pro zápis programů, jež mohou být provedeny na počítači. V tomto smyslu je programovací jazyk komunikačním nástrojem mezi uživatelem počítače

### JavaScript

JavaScript[[1]](#footnote-1) je multiplatformní, objektově orientovaný, událostmi řízeny skriptovací jazyk. Patří do rodiny C/C++/Java, ale JavaScript je od těchto jazyku zásadně odlišný sémanticky. Nejčastější použiti Javascriptu v současnosti zahrnuji. Webové stránky, Serveru, je možné spouštět v operačních systémech Windows pomoci programu Windows Script Host.  
JavaScript je soubor se zdrojovým kódem, který představuje určitou sérii příkazu nebo algoritmus. V souvislostech s hrami mohou pracovat se vstupy od uživatele, detekce kolizi, grafickým vzhledem, umělou inteligenci, uživatelském rozhraní a mnoha dalšími prvky hry.

### HTML

Celým[[2]](#footnote-2) názvem Hypertext Markup Language hlavním účel, kterému se používá k tvorbě Webových stránek, které jsou propojeny hypertextovými odkazy. HTML je hlavním jazykem pro vytvářeni stránek v systému World Wide Web. Vývoj HTML byl ovlivněn vývojem webových prohlížečů, která zpětné ovlivňoval definic jazyka. Dříve se HTML používal i ke stylovaní.

### CSS

Celým[[3]](#footnote-3) názvem Cascading Style Sheets, je speciální programovací jazyk stylů je to nástavba značkovacích jazyků, která umožňuje autorům popsat vzhled jejich dokumentů  
Cílem CSS je oddělit popis struktury dokumentace od definice vzhledu. Pomoci stylu lze definovat formátovací vlastnost pro každý logický prvek dokumentu.

### WebStorm

WebStorm je sada nástrojů pro tvorbu webu, WebStorm patří do integrované vývojového prostředí, které umožňuje uživatelům se orientovat prací s Javascriptem.

Obrázek 1: WebStorm [online]. In: . [cit. 2022-05-01]. Dostupné z: https://www.wikidata.org/wiki/Q4053021#/media/File:WebStorm\_Icon.svg

## Postup při tvořeni hry

Hra je 2D a skládá se z několika tříd. Úkolem hráče při této hře je sebrat všechny body   
a nesmí se během toho nechat chytit od nepřítele. Každá třída exportuje jen do jedné třídy a to main.js.

### Canvas

Canvas představuje plochu, na které jsou umístěny prvky uživatelského rozhraní (UI),  
Canvas byl použít takového důvodu že je nejlehčí na manipulaci vykresleni objektu na web a také se používá nejčastěji k vytvářeni 2d her. Ve hře je poukázáno na to, aby se vykreslila postava i nepřítel i stěny. Při vytvoření canvas stačí do HTML pomoci párových tagu do nich canvas a Javascriptu stačí ho propojit a přidělit mu nějaké rozměry1.

**const** canvas = document.querySelector('canvas');

**const** c = canvas.getContext('2d');

canvas.height = 480;

canvas.width = 960;

Zdrojový kód č. 1

U canvasu je dobře znát jak se používá a to že začíná v levým horním rohu a to v praxi znamená při sčítaní osy X jde doleva a při odčítaní osy X jde doprava, to samé platí při osy Y ale tam je to nahoru a dolu

### Vytvářeni Objektu a Mapa

Ve třídě Boundary se použil konstruktor, konstruktor je speciální metoda pro vytvoření a inicializaci instance objektu této třídy, v konstruktoru se nastavila jenom barva a v té samé třídě se použila voidu draw pro vytvoření objektu. Mapa se vyskytuje ve dvourozměrném poli, a to pomocí znaků, který každý z nich znamená něco jiného. Aby se z pole načet jeden znak z jednoho řádku je potřeba použití dvou forEach. Mapa se vykresluje na canvas pomoci switche, který rozlišuje, jestli se bude jednat o objekt nebo tečku. Pro vytvoření teček na mapě je hodně podobná jako na objektu, ale vytváří se pomoci metody arc.

draw(){

c.fillStyle = **this**.color

c.fillRect(**this**.position.x,**this**.position.y,**this**.width,**this**.height)

}

Zdrojový kód č. 2

### Vytvořeni Uživatele

K vytvořeni třídy “hráč“ se použil konstruktor, do kterého se použila pozice, rychlost a radius a pomoci void draw se vytvořil uživatel. Ve třídě main se zavolá class hráče a ustanoví se počáteční hodnota. Pohyb uživatele je pomoci kláves WASD, které po stisknutí klávesy se uživateli spustí nekonečné přičítaní nebo odečítaní, dokud nezmění směr, nebo se nenarazí do kolize, ale kdyby se stalo, že se kolize nenačetla, pohyb byl nekonečný, aby se uživatel mohl pohybovat musí se dat do class hráč, void update.

update () {

**this**.draw()

**this**.position.x += **this**.velocity.x

**this**.position.y += **this**.velocity.y

**if** (**this**.MaxOpen < 0 || **this**.MaxOpen > 0.75 )**this**.MinOpen = -**this**.MinOpen

**this**.MaxOpen += **this**.MinOpen

}

Zdrojový kód č. 3

### Ghost

Ve třídě ghost byl použit konstruktor jako u hráče, ale navíc bylo přidáno Image na rozdíl od hráče byly použitý obrázky, voidu draw se zkrátil jenom na jeden řádek. Ve třídě main se zavolá class ghost, při použití se musí dát do pole pro více vytvoření ghosts. Pohyb mají duchové jednoznačný, nemohou se otočit, ani zastavit, ale pouze dokážou na křižovatce změnit svůj směr kterým se vydají. Pomoci Math.Random se uloží do direction se pak použije u Switche, který rozhodne, kterým směrem se ghost vydá.

**if** (collisions.**length** > ghost.prevCollisions.**length**) {

ghost.prevCollisions = collisions

}

**if** (JSON.**stringify**(collisions) !== JSON.**stringify**(ghost.prevCollisions)) {

**const** pathways = ghost.prevCollisions.**filter**((collision) => {

**return** !collisions.includes(collision)

})

**const** direction = pathways[Math.**floor**(Math.**random**() \* pathways.**length**)]

Zdrojový kód č. 4

### Kolize

Ve třídě main se použila funkce, která vrací return, v níž je podmínka pro zjištění kolize, jestli entita narazila od obdélníku, nebo jestli se vedle něj prochází. Pokud narazí přímo do obdélníku tak se entitě změní údaje na nulu a úplně se zastaví. Na druhou stranu, pokud se entita nachází vedle obdélníku tak se nezastaví, dokud entita nezmění svůj směr nebo nenarazí do obdélníku. U postavy je nastaveno, že se kliknutím klávesy změní směr. Pokud není kolem postavy žádná odbočka, tak se tento příkaz nestane. Pokud se ale tato postava blíží k odbočce, tak dodělá svůj příkaz, kde se nachází a pak se tento příkaz provede.

**const** circleWidth = ({entity, rectangle}) => {

**const** padding = Boundary.width / 2 - entity.radius - 1

**return** (entity.position.y - entity.radius + entity.velocity.y <= rectangle.position.y + rectangle.height + padding &&

entity.position.x + entity.radius + entity.velocity.x >= rectangle.position.x - padding &&

entity.position.y + entity.radius + entity.velocity.y >= rectangle.position.y - padding &&

entity.position.x - entity.radius + entity.velocity.x <= rectangle.position.x + rectangle.width + padding)

}

Zdrojový kód č. 5

### Start hry

Za start hry zajišťuje tlačítko „Star Game“ které je propojené na funkci. Funkce v tomto případě spustí požadavek pro spuštění metody pro animace, která je propojena s funkcí pro pohyb celé hry.

### Konec Hry

Pro ukončení téhle hry jsou pouze dva způsoby. První možnost je ta, že duchové chytí hráče a ta druhá, že hráč sebere všechny tečky.

#### Chycení hráčem

Tady to ukončení hry se stane jenom když se hráčská postava dotkne enemy a tím se provede podmínka, provedení téhle podmínky se neresetuje všechno jenom se vrátí všechny entity na startovací pozici. Hráč má pouze 3 životy. Pokud se dostanou na nulu, tak se resetuje všechno skóre na nulu a obnoví se na mapě všechny tečky a entity se přesunou na startovací pozici.

**if** (

Math.**hypot**(ghost.position.x - pacman.position.x,

ghost.position.y - pacman.position.y) < ghost.radius + pacman.radius

)

Zdrojový kód č. 6

#### Sebraní všech teček

Ukončení hry je tehdy, když hráč sebere všechny tečky, které jsou uložené v poli. Když je pole prázdné tak se ukončí hra pomocí jednoduché podmínky, která kontroluje, jestli hráč přejel přes tečku a pomocí metody splice se odstraňují tečky z canvasu a při tom se přičítá skóre.

**if** (Math.**hypot**(dot.position.x - pacman.position.x,

dot.position.y - pacman.position.y) < dot.radius + pacman.radius) {

dots.splice(i, 1)

score1 += 10}

Zdrojový kód č. 7

### Reset hry

Restart hry lze použít u „konce hry”. Pro reset hry se používá funkce, která se volá jen tehdy, když se zmáčkne tlačítko nebo jen částečný reset, když má uživatel více životů. Celý reset vrátí stránku do původního stavu. Částečný reset jen u pozice postav.

## Použití obrázky v Projektu

Obrázky pro tento projekt byly vytvořeny a nebyly použity z internetu. Vytváření obrázku byl použit program jménem Painet.net.

### Painet.net

Painet.net[[4]](#footnote-4) je bezplatný software pro úpravu fotografie. Byl napsán v jazyce C#. Je založen na jednoduché verzi pro přidávání module, která umožňuje obohatit funkce softwaru. V tomto projektu se pomoci programu vyrobily duchové, které jsou transparentní.

# Závěr

Vytvořil jsem hru pro jednoho hráče v JavaScript. Odehrává se ve 2D. Při vývoji, jsem se musel spoustu věcí naučit, třeba jako je vytvoření samotné hry. Vytvořit takovou hru není problém, protože na internetu je hodně podobných her. Díky tomuto projektu jsem měl možnost získat mnoho nových znalostí a zkušeností, které se budou hodit do budoucna pro vytváření dalších projektů. U takové hry si je dobré promyslet u čeho začít nejdříve. Práce začala vytvořením canvasu. Dále jsem vytvořil objekty(mapu) a herní postavu. Konec této práce směřovala k vytvoření kolizí mezi hráčem a mapou, také bylo potřeba vymyslet pohyb nepřátel (duchů) a kolize mezi duchem a hráčem.

# Přílohy

## Seznam obrázků

[Obrázek 1: WebStorm [online]. In: . [cit. 2022-05-01]. Dostupné z: https://www.wikidata.org/wiki/Q4053021#/media/File:WebStorm\_Icon.svg 8](#_Toc102318463)

## Zdroj

HTML. *Wikipedie* [online]. [cit.2022-04-30]. Dostupné z: <https://cs.wikipedia.org/wiki/Hypertext_Markup_Language>

JavaScript. *Wikipedie* [online]. [cit. 2022-04-29]. Dostupné z: <https://cs.wikipedia.org/wiki/JavaScript>

*Painet.net* [online]. [cit. 2022-05-01]. Dostupné z: <https://cs.frwiki.wiki/wiki/Paint.net>

CSS. *Wikipedie* [online]. [cit. 2022-05-01]. Dostupné z: https://cz.wikipedia.org/wiki/CSS

1. JavaScript. *Wikipedie* [online]. [cit. 2022-04-29]. Dostupné z: https://cs.wikipedia.org/wiki/JavaScript [↑](#footnote-ref-1)
2. HTML. *Wikipedie* [online]. [cit.2022-04-30]. Dostupné z: https://cs.wikipedia.org/wiki/Hypertext\_Markup\_Language [↑](#footnote-ref-2)
3. CSS. *Wikipedie* [online]. [cit. 2022-05-01]. Dostupné z: https://cz.wikipedia.org/wiki/CSS [↑](#footnote-ref-3)
4. *Painet.net* [online]. [cit. 2022-05-01]. Dostupné z: https://cs.frwiki.wiki/wiki/Paint.net [↑](#footnote-ref-4)