

Maturitní práce

Worms are back! In your browser

**Jan Habas
Pardubický kraj**

Pardubice 2021

Maturitní práce

Worms are back! In your browser

Autor: Jan Habas

Škola: DELTA - Střední škola informatiky a ekonomie, s.r.o.
Ke Kamenci 151, 530 03 Pardubice

Kraj: Pardubický kraj

Konzultant: RNDr. Jan Koupil Ph.D.

Pardubice 2021

Zadání maturitního projektu

Způsob zpracování, cíle práce, pokyny k obsahu a rozsahu práce:

Žák vytvoří online hru podobnou davným titulům Worms nebo Scorched Earth. Tato hra poběží v internetovém prohlížeči

1. Žák podnikne rešerši za účelem nalezení existujících vhodných nástrojů k realizaci hry podobné zmíněným vzorům. Vlastní sada používaných pravidel i grafický vzhled budou záviset na volbě autora.
2. Žák připraví algoritmy nutné ke generování krajiny, výpočtu místa dopadu, zásahu a poškození a zdokumentuje je.
3. Žák naprogramuje vlastní hru s využitím výše uvedených algoritmů a zveřejní ji.
4. Žák provede testování uživatelského rozhraní a hry samotné s několika uživateli ze svého okolí, získá od nich zpětnou vazbu, zdokumentuje ji a na jejím základě upraví uživatelské rozhraní

Stručný časový harmonogram (s daty a konkretizovanými úkoly):

- *Září*: vyhledávání vhodných nástrojů, existujících řešení, příprava fyzikálního modelu
- *Říjen-listopad*: Návrhy algoritmů pro použití ve hře a jejich oddělené testování
- *Prosinec*: Kompletace vlastní hry
- *Leden*: Testování uživatelského rozhraní, sběr dat
- *Únor-březen*: Úpravy hry a rozhraní na základě dat z testů, opakované testy, práce na dokumentaci projektu

Poděkování

Chtěl bych poděkovat RNDr. Janu Koupilovi Ph.D. za vedení mého projektu a velikou ochotu z jeho strany, v tom, cokoliv vysvětlit, popřípadě poradit co zlepšit nebo za vysvětlení, které řešení je optimální.

Anotace

Cílem mého projektu bylo vytvořit dělostřeleckou hru ve stylu Worms, která bude hrána v prohlížeči. Hra by měla v rámci možností aplikovat reálné fyzikální zákony (vítr, gravitace apod.)

Klíčová slova

Worms, internetová hra, fyzika, JS, P5, Matter.js

Annotation

The goal of my project was to create an artillery game in the style of Worms, which will be played in a browser. The game should apply the real laws of physics (wind, gravity, etc.) as far as possible.

Annotation keywords

Worms, internet game, physics, JS, P5, Matter.js

Obsah

1. Úvod	1
2. Motivace	1
3. Výběr technologie	2
4. Technologie	2
4.1 Matter.js.....	2
4.2 P5.js.....	2
4.3 Polygon tools js	2
4.4 Decom.js	2
4.5 D3.js	3
5. Tvoření hry.....	3

5.1 Svět	3
5.2 Wormové.....	5
5.3 Ovládání.....	6
5.4 Tímy	9
5.5 Voda	9
5.6 Vítr	10
5.7 Zbraně.....	10
5.8 Prak (Slingshot)	11
5.9 Destrukce	12
5.9 Audio.....	13
6. Web	13
6.1 Základní rozložení stránky	13
6.2 Kreslení mapy	14
7. Vložení mapy.....	15
7.1 Ovládací prvky	15
8. Instruktažní videa.....	16
9. Závěr	16
Reference.....	17
Seznam Obrázku	18

1. Úvod

Worms je dnes už kultovní hrou, která byla vydána roku 1995 vývojářským týmem jménem Team17. Ale je určitě důležité zmínit, že první hra žánru artillery games byla hra Artillery duel z roku 1982. I přes to, že hlavní inspirací pro tento projekt byla série her Worms jsou zde věci, které jsem se rozhodl udělat jinak. [1]



Obrázek 1 Ukázka jednoho z dílů worms

V žánru Artillery Game se hráči pomocí postaviček (tanky, červíci atd.) snaží zasáhnout nepřátelské cíle a dostat jejich zdraví na nulu, nebo se je pokouší odhodit z mapy za pomoci síly exploze. V tomto vám samozřejmě brání mapa, vítr a gravitace. Po výstřelu prvního hráče začne hrát hráč druhý, a tak se střídají, dokud jeden z týmů není eliminován.

Hlavní myšlenkou mého projektu bylo dostat tento styl hry do prohlížeče, kde pomocí lokálního multiplayeru, což znamená, že dva hráči hrají na jedné klávesnici a střídají se po kolech.

Mojí inspirací pro tento maturitní projekt nebyla pouze hra Worms, ale i ostatní hry žánru. Hlavní změnou oproti předloze je především způsob střelení, který zde je vytvořený na způsob výstřelu z praku.

2. Motivace

Moje motivace pro tento projekt byla taková, že jsem si vždy chtěl zkusit, jaké je to být herní vývojář. Přes to, že se nejedná o hru, která by byla nějak velká, či měla propracovanost velkých her, které je možné získat na různých platformách.

3. Výběr technologie

Pro simulaci fyziky jsem se rozhodl použít knihovnu Matter.js. Vybral jsem ji proto, že mi pro mou práci přišla vhodná a oproti ostatním jako je Phaser nebo Box2d byla přesnější a optimalizovanější [2]. K tomu, abych mohl pracovat s polygony, si engine Matter.js vyžádal knihovnu s názvem decomp.js. Tato knihovna je zde tedy vložena, ale není prakticky využívána mimo Matter.js. I přes to že Matter.js má svou funkci render, rozhodl jsem se pro využití P5.js, což je externí knihovna pro vykreslování. Důvodem pro mé rozhodnutí byla výkonová nenáročnost zmiňované knihovny. Výsledkem je, že jsem získal jednoduchou a výkonem nenáročnou cestu k vykreslování hry. Jako poslední je použit Polygon Tools. Jedná se o nástroj, který jsem využíval na operace s polygony.

4. Technologie

4.1 Matter.js

Matter.js je knihovna, kterou v roce 2014 začal vyvíjet Liam Brummitt jako nástroj pro simulaci fyziky [3].

Matter.js je fyzický engine určený pro prohlížeče psaný pouze v JavaScriptu. Tato knihovna slouží k simulování 2D fyziky a je vhodná pro vývoj her a fyzikálních simulací. Umí v základu mnoho funkcí, jako je například: vytvoření těles a přiřazení různých vlastností, jako je plocha, hmotnost nebo hustota. Také simuluje srážky sil, gravitaci a tření, jak pro tělesa nebo dokonce třeba i pro vzduch.

4.2 P5.js

P5.js je JS open-source knihovna pro základní vykreslování tvarů. [4] Hlavní myšlenkou knihovny je, že se snaží jednoduchostí zpřístupnit co největší vizuální modifikaci celé stránky. Jedna z největších výhod knihovny je to, že je nenáročná, takže pro vytváření hry je vhodná. Knihovna jako taková umí kromě základních tvarů typu trojúhelník, čtverec nebo kruh, vykreslovat také složitější objekty, jako jsou polygony nebo třeba vykreslení obrázků.

4.3 Polygon tools js

Polygon tools js je malá knihovna sloužící k operaci s polygony od jejich vytvoření až po jejich sčítání a odčítání [5]. Knihovna byla využita zejména u tvoření určitých částí mapy a její destrukce.

4.4 Decom.js

Jedná se o port knihovny Poly Decom, kterou napsal Mark Penner. Knihovna nabízí dva algoritmy [6]. Jeden, který je více optimalizovaný, ale pomalý, a druhý, který je méně přesný ale rychlý. Tato knihovna se zde nachází, protože úzce spolupracuje s naším Matter.js enginem a jeho prací s polygony.

4.5 D3.js

D3.js je knihovna JavaScript pro manipulaci s dokumenty na základě dat. Tato knihovna se hodí na práci s daty a v projektu je použita na úpravu stránky pro kreslení mapy. [7]

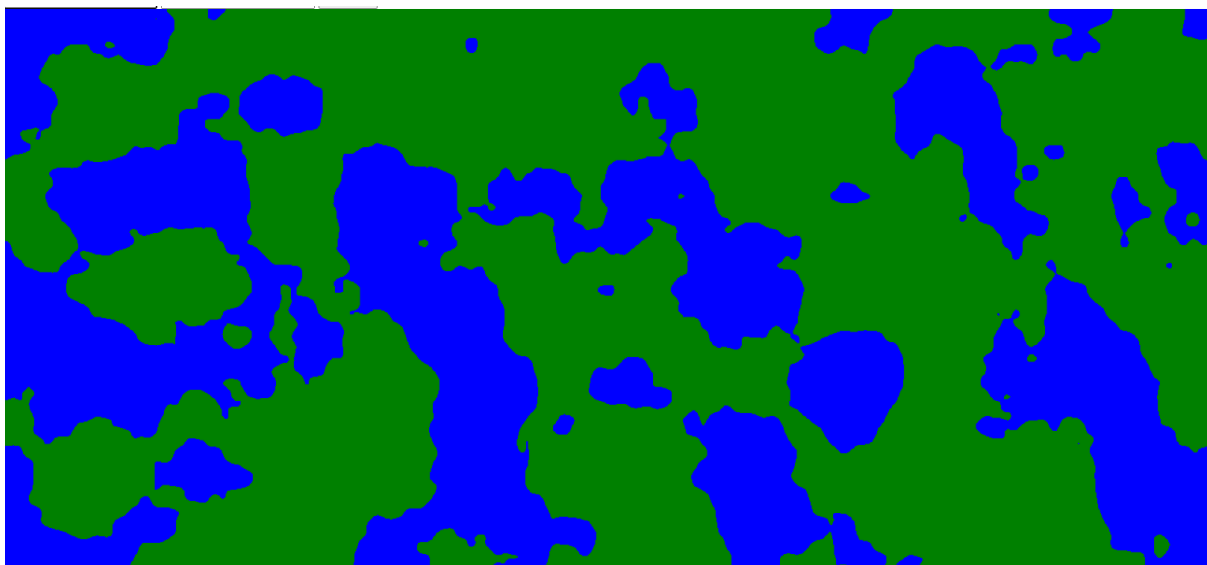
5. Tvoření hry

5.1 Svět

Klíčovým faktorem je pro engine takzvaný world. V něm uchovává své objekty a tvoří jejich reálnou simulaci, jako jsou srážky, klouzaní atd.

Po prozkoumání několika možností, jak vytvořit mapu, včetně tvorby náhodné mapy metodou generování několika překrývajících se elips nebo vytvoření náhodné mapy pomocí Perlinova šumu. [8]

Nakonec jsem se rozhodl vytvořit mapu tak, že nakreslím polygony a ty poté do světa vložím.



Obrázek 2 Druhý pokus o vytvoření mapy

K dosažení do samotné mapy, jsem si obstaral externí kód [9], který umí načrtnout polygony a modifikoval jsem ho pro své využití takovým způsobem, abych si mohl jednotlivé součásti mapy kreslit. Modifikace spočívala v tom, že jsem stránku upravil, aby v ní uživatel mohl načrtnout celou mapu a také si označit místa kde chce, aby se Wormové rozmístili.

Největším problém nastal v tu chvíli, kdy jsem tyto součásti vkládal do samotného enginu. Bylo nutné data dále upravit pro správný vstup do Matter.js. Poté je zase bylo důležité dostat ve správném formátu do P5.js, abych mohl polygony vykreslit. Původně jsem chtěl

využít funkci render, kterou má v sobě samotný engine Matter.js. Využití této funkce bylo pro projekt nereálné, protože nenabízí takovou škálu možností pro práci na projektu, jako P5.js. Dále byla funkce render příliš náročná na výkon a po vykreslení celého projektu by byla hra nehratelná kvůli její výkonnostní náročnosti.

Vytvořil jsem tudíž třídu Map do jejíž konstruktéru jsem vložil textové souřadnice samotného polygonu, jako další souřadnici na ose X a jako poslední souřadnici na ose Y. Poté jsem pomocí funkcí změnil souřadnice takovým způsobem, aby je byl schopný Matter.js převést na polygony. Zde bylo nutné vložit do projektu externí knihovnu Decomp.js, protože jí knihovna Matter.js využívala jako pomocnou knihovnu na práci s polygonem. Ve finále jsem z konečných dat vytvořil těleso v Matter.js, které funguje jako fyzická část polygonu neboli mapy. Ve vykreslování jsem potom musel pomocí výpočtu vypočítat správnou souřadnici polygonu. Pokud by tomu tak nebylo, tak by nastala situace díky, které by polygon v pozdější fázi hry měnil svoji polohu.

Proto bylo nutné vymyslet tento vzorec:

$$\text{minXpo} - \text{minXpřed} = \text{rozdílX}$$

$$\text{minYpo} - \text{minYpřed} = \text{rozdílY}$$

Výsledek jsem uložil do proměnné, kterou jsem potom ve vykreslení přičetl k souřadnicím, které jsem bral z původních souřadnic před upravením. Ty potom vykreslily polygon na místě, kde se nacházel v enginu Matter.js.



Obrázek 3 Konečná verze mapy

5.2 Wormové

Samotné wormy jsem tvořil tak, že jsem v Matter.js vytvořil malý obdélník, který hráč poté ovládá pomocí klávesnice a myši. Obdélník byl vybrán, protože má větší stabilitu oproti polygonu, který by byl ve tvaru polygonu worma. Polygon ve tvaru worma by se převracel a tím by se hra stala nehratelná.

Chvilí, kdy se worm stává statický je, když se chystá zaútočit. Chvilí po ukončení kola se také stává statickým, nikoliv hned po konci. Pokud worma trečí projektil, když nehraje, bude to na něj působit a může to zapříčinit jeho posun. Na worma také neúčinkuje síla větru. Je vůči jeho síle naprosto imunní.

Samotné vyobrazení worma funguje jako malá animace, kde se střídají dva obrázky a worm, tak při pohybu hýbe ocasem. Při pohybu se také natočí worm podle toho, na jakou stranu se zrovna hráč rozhodl jít. Pokud skáče, tak se pustí animace natažení ocasu. V případě, že je worm odhozen explozí, tak se opět načte animace.

Když wormovo zdraví klesne na nulu a méně, nebo se worm dostane mimo mapu či spadne do vody. Promění se jeho proměnná `alive` na `false`. Tím se jeho pozice nastaví mimo mapu. Objekt se stane statickým a hra se už o worma zbytek hry nestará. Hra také poté už kolo daného worma přeskakuje.



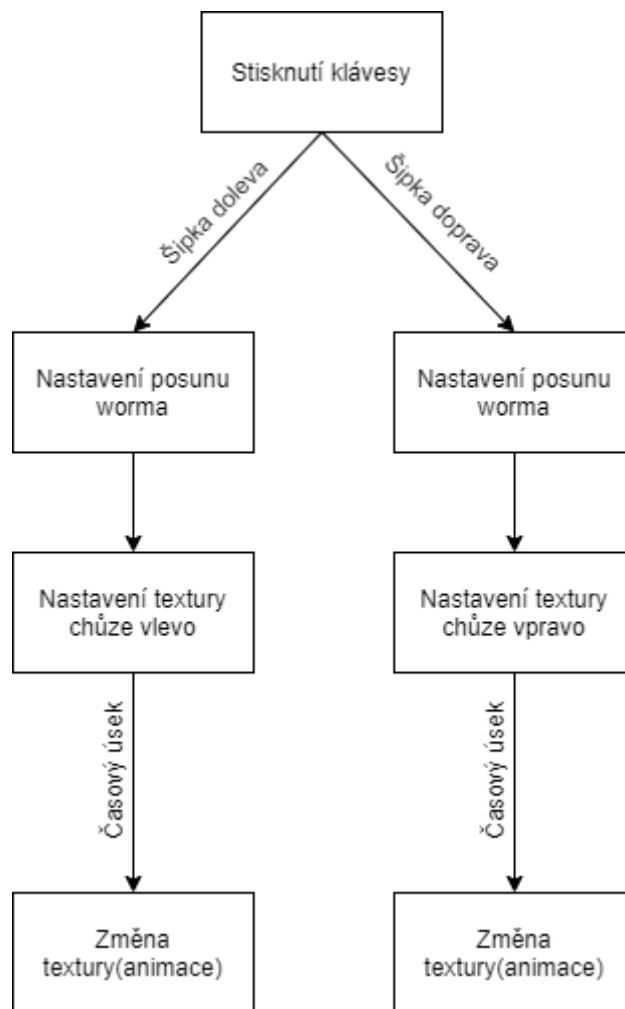
Obrázek 4 Samotný worm

5.3 Ovládání

Ovládání bylo vytvořeno pomocí funkce `checkKey`. Ovládání je část třídy `worm`, kde se nejdříve zkontroluje, jestli tento worm zrovna hraje, zda už útočil a v poslední řadě, jestli nemá zrovna vypnutou klávesnici.

Ovládání umožňuje pomocí `shiftu` se přepnout do útočného módu, což znamená, že se vytvoří prak a worm tak může zaútočit. V tomto módu je worm statický. Poté pomocí číslic může hráč přepnout svou zbraň. Při výběru zbraně se do proměnné `weapon` vytvoří nová třída zbraně.

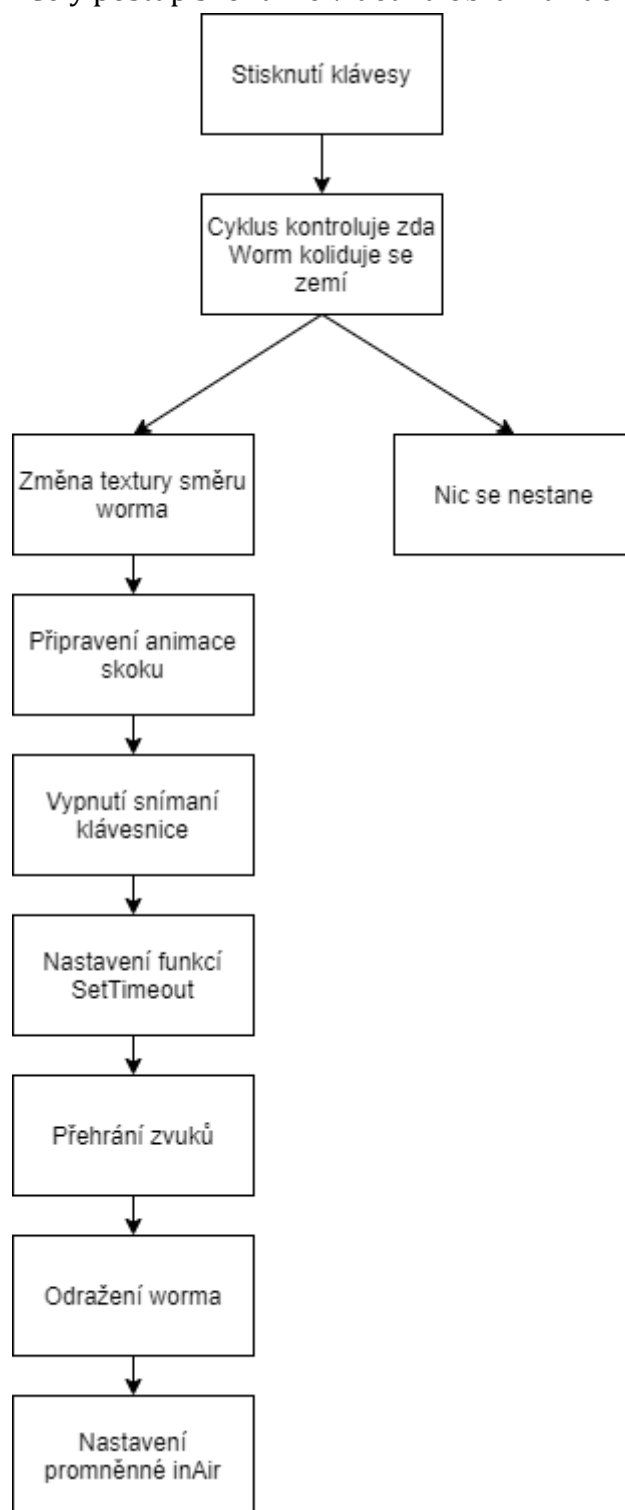
Pohybovat se hráč ve hře může pomocí šipek. Co se odehrává po jejich stisknutí je vysvětleno na obrázku níže.



Obrázek 5 Programový diagram chůze

Při odrazu se na 2 sekundy vypne jakékoliv snímání klávesnice pro hráče, aby nemohlo dojít k tomu, že by se hráč odrážel i ve vzduchu. Vypnutí snímání klávesnice je nutné také proto, že by jinak mohl worm lézt po stěnách, protože by se do kolize mohl dostat už ve vzduchu při nárazu do stěny. Po dopadu na zem se proměnná, která určuje, zda je worm ve vzduchu, opět nastaví na false.

Celý postup skoku lze vidět na obrázku zde:

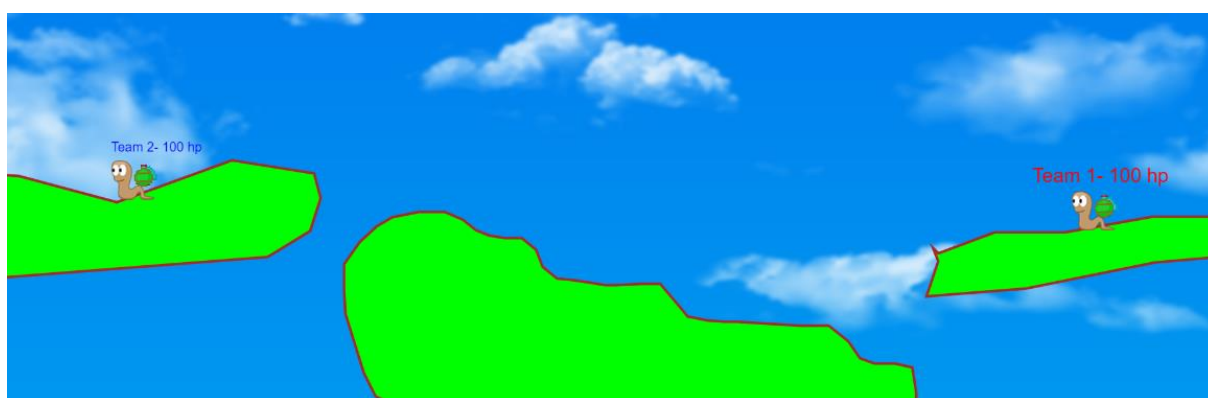


Obrázek 6 Programový diagram skoku

5.4 Týmy

Týmy se na mapě nachází dva. Každý tým se skládá ze dvou až čtyř wormů. Každý z nich má nad sebou zobrazený počet životů, barevné odlišení od druhého týmu a v neposlední řadě má také své jméno. Nad wormem co zrovna hraje je vyobrazena červená šipka. Pokud worm spadne z mapy, zemře, a to i v případě, kdy se dotkne vody. Každý v týmu má v základu 100 bodů života, pokud tato hodnota spadne pod jedničku tak worm zemře. Tým, ve kterém se už nenachází žádný worm prohrává. Poté se zobrazí oznámení a hra se restartuje.

Zasáhne-li worm nešťastnou náhodou svého spolubojovníka nebo sebe, je zasažen zraněn právě tak, jako by byl zraněn nepřítel.



Obrázek 7 Rozlišení samotných týmů před zavedením jmen

Na obrazovce se také nachází graficky vyobrazený bar s celkovým počtem životů wormů v týmu. Znázornění životů reprezentují pruhy v dané barvě týmu. Pokud nějaký člen z týmu utrpí zranění, tak se pruh zkrátí.



Obrázek 8 Grafický ukazatel životů

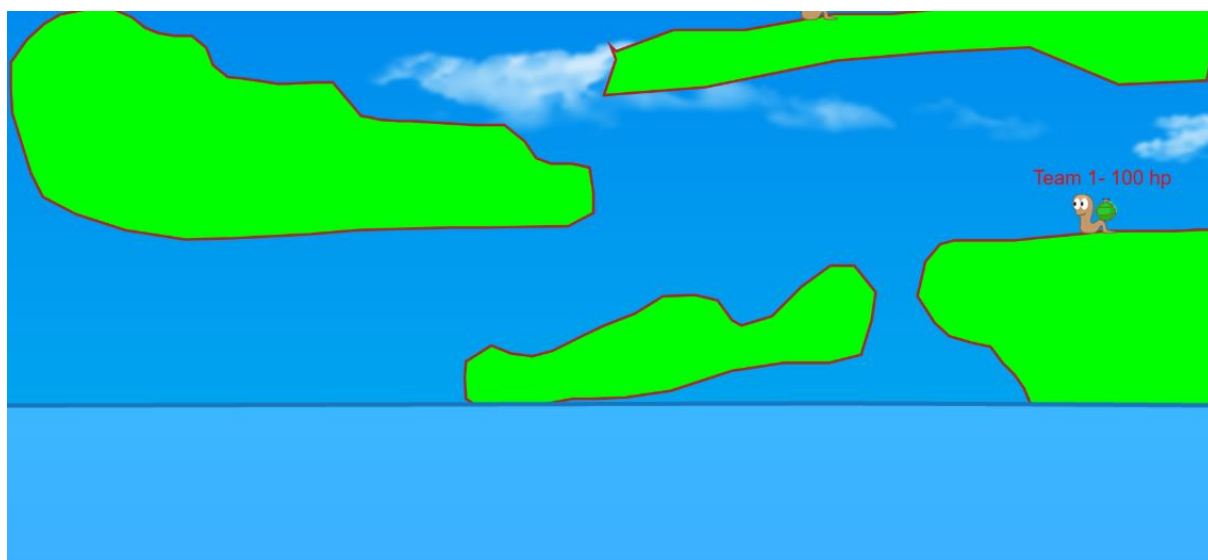
5.5 Voda

Voda je implementací velice jednoduchá, ale i tak praktická v tom, jak do samotné hry přidává element tlaku, který je na hráče kladen.

Voda je třída, která se vytvoří po určitém čase od začátku hry, poté se zde opakovaně volá funkce pro zvednutí hladiny to znamená, že každých x sekund se voda zvedne o x pixelů.

Voda se zvedá o 1 pixel v cyklu, dokud se nezvedne o danou výšku a díky tomu vykreslení působí jako animace.

Vykreslení funguje na principu takovém, že začnu vyobrazovat texturu vody, která se postupně posouvá nahoru. V Matter.js voda není nijak reprezentována, funguje pouze jako textura v P5.js. O správné fungování, se stará program pomocí výpočtů, které jsou zapsány v externím skriptu.



Obrázek 9 Voda v praxi

5.6 Vítr

Vítr ve hře působí jen na granáty, to znamená, že pokud vyhodíme granát tak na něj působí síla větru. Tato síla ho buď zpomalí, nebo zrychlí, záleží, na jakou stranu letí, a jakým směrem vítr fouká. Každý daný časový úsek vítr mění náhodně svou hodnotu.

5.7 Zbraně

Jako zbraně ve hře slouží granáty. Jsou zde dva druhy granátů. Jeden je časovaný granát, který vybuchne za dva a půl sekundy po vyhození a druhým je kontaktní granát, který při kontaktu se zemí vybuchne. Tento granát nepoškodí mapu, ale způsobuje větší poškození a působí větší odhození při zasažení daného cíle. Snažil jsem se plusy a mínusy granátu vyvážit tak, aby jeden nebyl silnější než druhý.

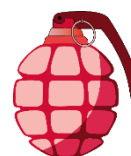
Granát v Matter.js je definovaný jako objekt kruhu, ale dokud není granát vržen, tak se do světa Matter.js nevytvoří a funguje tak pouze jako textura. Tato textura v samotném světě není, a nevztahuje se na ní tudíž zatím žádná fyzika. Objekt se v Matter.js vytvoří až při zapnutí bojového postavení worma. Granát poté pomocí třídy Slingshot připevním nad worma a nastavím, aby byl worm statický.



Obrázek 10 Programový diagram vŕbuchu granátu



Obrázek 11 Ukázka textury klasického granátu



Obrázek 12 Ukázka textury kontaktního granátu

5.8 Prak (Slingshot)

Prak ve hře slouží jako inovace oproti původním worms, kde střelba fungovala tak, že hráč vybral směr, kam chce házet, a poté podržením mezerníku přidal sílu hodu.

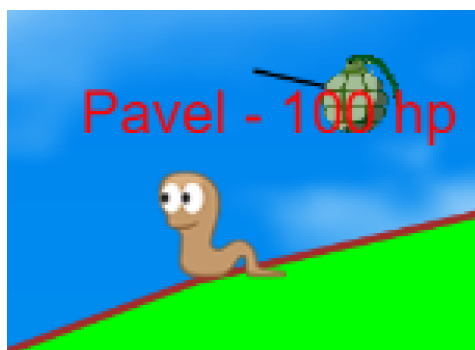
V této hře vrh granátem funguje tak, že po zmáčknutí dané klávesy se proměnná throw ve třídě granát nastaví na true, čímž se vytvoří fyzické těleso granátu a společně s ním se vytvoří i objekt třídy slingshot.

Vytvořením objektu slingshot se těleso granátu připevní nad worma, který se při házení nemůže hýbat. Hráč má možnost začít útok tím, že pomocí myši natáhne granát a poté ho pustí.

Jako první bylo nutné zajistit, aby při zmáčknutí myši hra poznala, zdali je hráč kurzorem uvnitř kruhu. To jsem ošetřil s použitím vzorce [10]

Dále bylo nutné ošetřit, aby se hráči od vrhnutí granátu do jeho exploze vyplo snímání myši. Pokud by se myš nevypla hráč by mohl granát přesouvat po jeho vypuštění z praku.

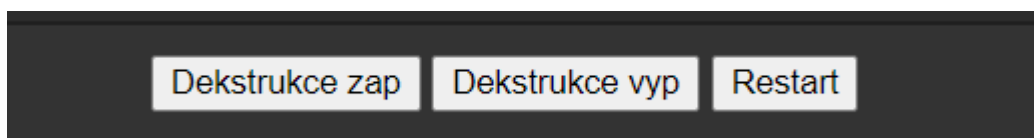
Jednou z věcí, které bylo nutné ošetřit, je aby měl slingshot nějaké délkové omezení. Proto jsem ve funkci show třídy slingshot nastavil kontrolu vzdálenosti granátu od středu praku. V případě, kdy se granát v ose X nebo Y vzdálil více jak o 30 pixelů, tak jsem nastavil, aby lano mělo fixní délku a tuhost, a nebylo ho možné dál natáhnout.



Obrázek 13 Ukázka samotného praku

5.9 Destrukce

Destrukce ve hře je napsána v třídě Destroy. Podle toho, zda uživatel nastaví zapnutí či vypnutí destrukce, pomocí tlačítek, na html stránce, tak se destrukce vypne či zapne. Nastavení se ukládá s pomocí cookies.



Obrázek 14 Nastavení destrukce

Pokud je destrukce zapnutá, program vytvoří objekt třídy Destroy. V konstruktoru se vytvoří polygon výbuchu. Tento objekt je potom přidán do světa Matter.js a dán mimo mapu.

Ve funkci destroyPol je vstupními parametry zasažený polygon a souřadnice výbuchu. Poté se odečte polygon výbuchu a zasažený polygon, následný rozdíl pak tvoří nově

upravený (poškozený) polygon. Poté je zde vzorec, který polygon narovná tak, aby nedošlo k jeho posunu oproti původnímu polygonu.

$$\text{posunCentraX} = \text{NovéCentrumX} - \text{StaréCentrumX}$$

$$\text{posunCentraY} = \text{NovéCentrumY} - \text{StaréCentrumY}$$

Poté tyto proměnné odečtu při určení pozice nového(poškozeného) tělesa.

5.9 Audio

Ve hře se nachází několik zvukových efektů. Od výbuchu granátu až po zvuky, které vydávají samotní wormové. Zvukové efekty jsou zde pro výbuch granátu, chůzi, řev worma pokud utrpí poškození, nebo pokud si worm jen tak mluví. Každý z těchto zvuků má více variací, které se náhodně střídají.

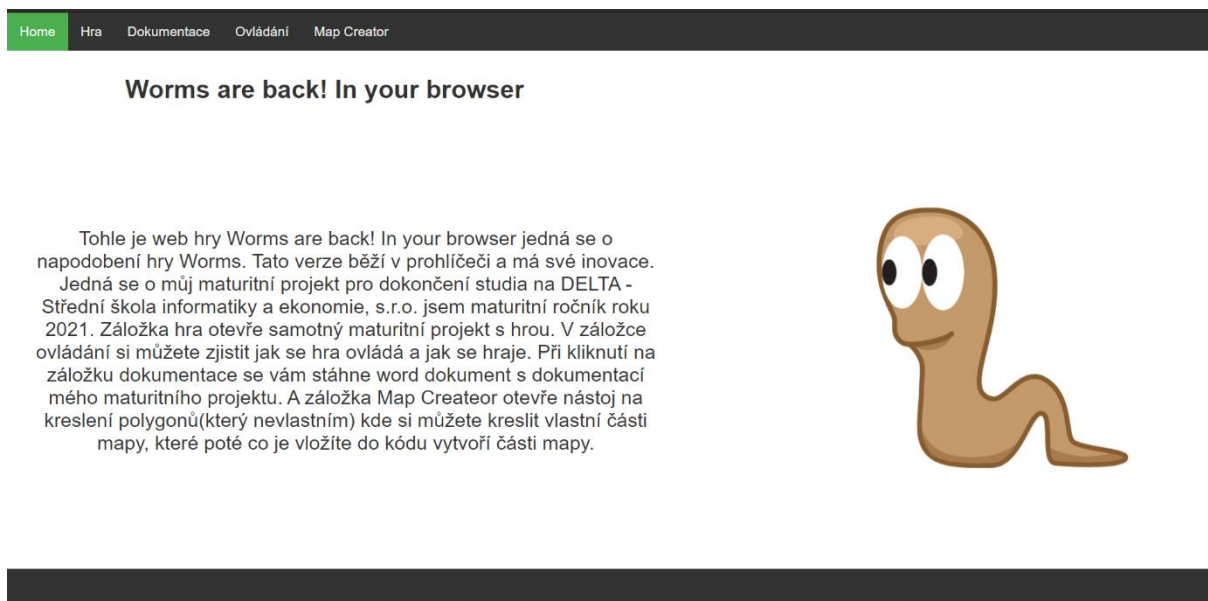
6. Web

Na webové stránce se nachází základní informace k mému projektu. Od ovládání až po editor mapy.

6.1 Základní rozložení stránky

Základní rozložení stránky je v zeleno-černé barvě, nahoře se nachází základní rozcestník stránky.

- Home – indexová stránka webu a nachází se na ní základní informace o samotném projektu.
- Hra – zde se nachází samotná hra. Nahoře v liště se nachází také tlačítko pro restart a zapnutí či vypnutí destrukce, nebo pro samotné vložení mapy.
- Dokumentace – v této části je možné stáhnout samotnou dokumentaci.
- Ovládání – zde je uvedeno samotné ovládání hry.
- Map Creator – na této stránce je možné načrtnout si vlastní mapu, u které si uživatel nastaví i pozice, kde se mají vygenerovat wormové. Poté zde může tuto mapu exportovat v podobě textového souboru.



Obrázek 15 Rozložení webové stránky

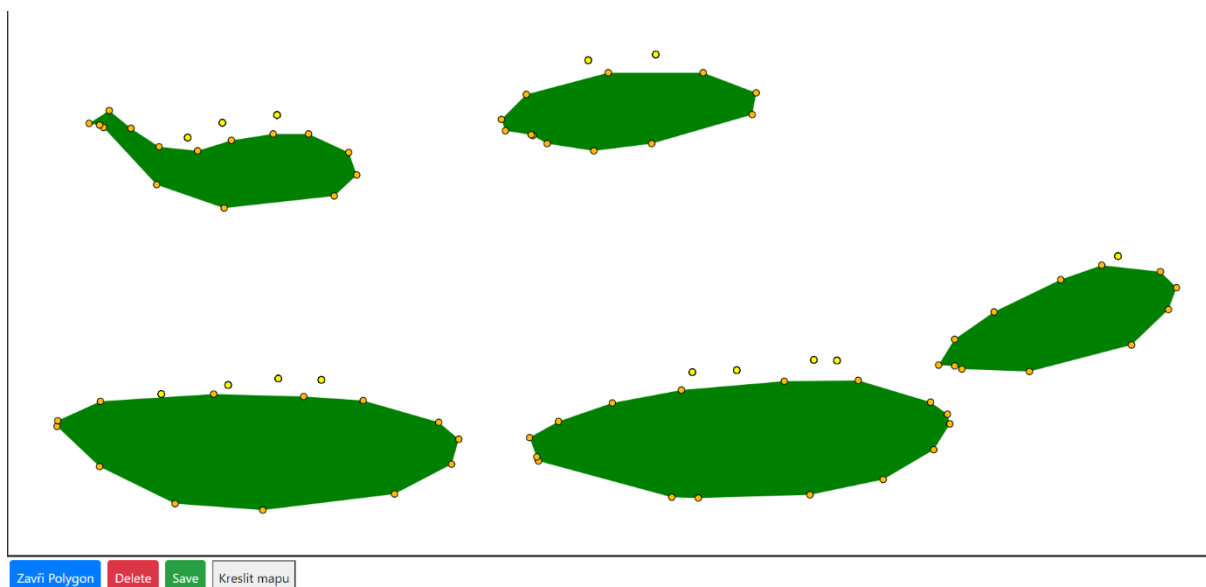
6.2 Kreslení mapy

Mapu lze nakreslit na kartě Map Creator. Tato stránka slouží jako editor mapy, kde si uživatel nakreslí svojí vlastní mapu a označí na ní místa, kam poté hra vloží samotné wormy.

Když už je uživatel s mapou a pozicemi Worms spokojen, stačí jen aby zmáčkl tlačítko Save a prohlížeč stáhne textový soubor. V tomto souboru jsou napsaná data v šabloně, kterou si posléze může ve hře nahrát.

Dalšími z ovládacích prvků jsou

- Delete – smaže aktuálně rozkreslený polygon.
- Zadat pozice Wormů – pokud na tento čudlík uživatel klikne přepne se kreslení mapy na určování pozic, kam se wormové mohou vygenerovat.
 - Zelené polygony znázorňují jednotlivé části mapy.
 - Žluté kroužky pozice, kde se Wormové mohou vygenerovat.



Obrázek 16 Editor Mapy

7. Vložení mapy

Vložení Mapy je poté jednoduché stačí jen najít soubor, který byl stažen, kliknout na tlačítko s vložit soubor a nahrát daný soubor s uloženou mapou a pozicemi.

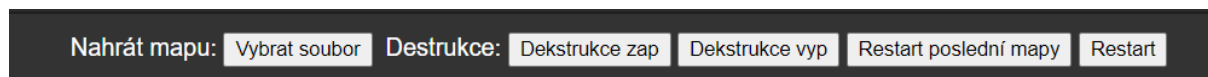
Pokud je soubor nějak poškozený, nebo se hře nepovede mapu nahrát uživatel obdrží okénko s příslušnou chybovou hláškou, a hra se restartuje se základní mapou.



Obrázek 17 Ukázka nahrané mapy

7.1 Ovládací prvky

Po zapnutí hry je možné za pomoci tlačítek, buď restartovat původní mapu, vypnout či zapnout destrukci, nebo je zde input pro vložení vlastní mapy skrze textový soubor.



Obrázek 18 Ovládací prvky

8. Instruktažní videa

Pomocí Loom [11] jsem nahrál v prohlížeči dvě krátká videa, která jsem umístil na hlavní stránku mého projektu. Tato videa složí jako instruktážní videa. Prvním video vysvětlí uživateli, jak se hra hraje, a jak se střílí z praku ve hře. Druhé video je vysvětlení toho, jak si uživatel může nakreslit a poté vložit vlastní mapu.

9. Závěr

Podařilo se mi vytvořit základní část hry, která je plně hratelná a je připravená na webových stránkách. Na těchto stránkách jsem vytvořil Map Creator (z externí stránky co jsem upravil) díky kterému může uživatel vytvořit vlastní mapu. Prostřednictvím tohoto aspektu dostává hra velký rozměr a nestává se tak repetitivní a je možné jí upravovat. K tomu aby hra nebyla repetitivní napomáhají i prvky, které hra umí generovat náhodně, jako jsou pozice wormů (odvozené od pozic co zadá uživatel) a nebo vítr.

Na začátku března jsem poslal hru svým spolužákům ať jí otestují. Ohlas byl kladný a veškeré nedostatky, na které si testeři stěžovali se mi povedlo ze hry odstranit.

Zdrojový kód je nyní volně dostupný na této stránce:

<https://github.com/LastShock/WormsMatterJS>

Reference

- [1] „Worms (series) - Wikipedia,“ [Online]. Available: [https://en.wikipedia.org/wiki/Worms_\(series\)](https://en.wikipedia.org/wiki/Worms_(series)). [Přístup získán 18 9 2020].
- [2] „Porovnání Matter js,“ [Online]. Available: https://www.reddit.com/r/javascript/comments/d40odd/askjs_best_js_physics_library_for_2019/. [Přístup získán 15 9 2020].
- [3] „matter.js,“ [Online]. Available: <https://brm.io/matter-js/>. [Přístup získán 11 10 2021].
- [4] „P5.js,“ [Online]. Available: <https://p5js.org/>. [Přístup získán 1 10 2020].
- [5] „polygon-tools,“ [Online]. Available: <https://www.npmjs.com/package/polygon-tools>. [Přístup získán 12 12 2020].
- [6] „Poly-decomp.js,“ [Online]. Available: <https://github.com/schteppe/poly-decomp.js/>. [Přístup získán 15 11 2020].
- [7] „D3.js,“ [Online]. Available: <https://d3js.org/>. [Přístup získán 17 10 2020].
- [8] „Perlinův šum - Wikipedia,“ [Online]. Available: https://cs.wikipedia.org/wiki/Perlin%C5%AFv_%C5%A1um. [Přístup získán 30 9 2020].
- [9] „Stránka na kreslení polygonů,“ [Online]. Available: <https://gist.github.com/RiseupDev/b07f7ccc1c499efc24e9>. [Přístup získán 21 10 2020].
- [10] „Vzorec pro kruh,“ [Online]. Available: <https://math.stackexchange.com/questions/198764/how-to-know-if-a-point-is-inside-a-circle>. [Přístup získán 25 11 2020].
- [11] „Loom for chrome,“ [Online]. Available: <https://www.loom.com/>. [Přístup získán 10 3 2021].
- [12] „Ukládání txt souboru,“ [Online]. Available: <https://robkendal.co.uk/blog/2020-04-17-saving-text-to-client-side-file-using-vanilla-js>. [Přístup získán 5 3 2021].

- [13 „Smazání posledního characteru ve stringu,“ [Online]. Available:
] <https://www.geeksforgeeks.org/how-to-read-a-local-text-file-using-javascript/>. [Přístup
získán 19 1 2021].
- [14 „Obrázek hry worms,“ [Online]. Available:
] https://cdn.alza.cz/Foto/LegendFoto/photos/MVE000930_1.jpg. [Přístup získán 1 3
2021].
- [15 „JS arrow P5,“ [Online]. Available:
] <https://stackoverflow.com/questions/44874243/drawing-arrows-in-p5js>. [Přístup získán
20 2 2021].
- [16 „Detekce Mobilu,“ [Online]. Available:
] <https://stackoverflow.com/questions/11381673/detecting-a-mobile-browser>. [Přístup
získán 15 2 2021].
- [17 „čtení txt souboru,“ [Online]. Available: [https://www.geeksforgeeks.org/how-to-read-a-](https://www.geeksforgeeks.org/how-to-read-a-local-text-file-using-javascript/)
] [local-text-file-using-javascript/](https://www.geeksforgeeks.org/how-to-read-a-local-text-file-using-javascript/). [Přístup získán 1 3 2021].
- [18 „Cookies Mazání StackOverflow,“ [Online]. Available:
] <https://stackoverflow.com/questions/179355/clearing-all-cookies-with-javascript>.
[Přístup získán 1 3 2021].

Seznam Obrázku

Obrázek 1 Ukázka jednoho z dílu worms	1
Obrázek 2 Druhý pokus o vytvoření mapy	3
Obrázek 3 Konečná verze mapy	5
Obrázek 4 Samotný worm.....	6
Obrázek 5 Programový diagram chůze.....	7
Obrázek 6 Programový diagram skoku	8
Obrázek 7 Rozlišení samotných týmů před zavedením jmen.....	9
Obrázek 8 Grafický ukazatel životů.....	9
Obrázek 9 Voda v praxi	10
Obrázek 10 Programový diagram vůbuchu granátu	11
Obrázek 11 Ukázka textury klasického granátu.....	11
Obrázek 12 Ukázka textury kontaktního granátu.....	11
Obrázek 13 Ukázka samotného praku.....	12
Obrázek 14 Nastavení destrukce.....	12

Obrázek 15 Rozložení webové stránky.....	14
Obrázek 16 Editor Mapy	15
Obrázek 17 Ukázka nahrané mapy	15
Obrázek 18 Ovládací prvky	16