STŘEDNÉ PRŮMYSLOVÁ ŠKOLA

MLADÁ BOLESLAV

**ROČNÍKOVÁ PRÁCE**

**Mario**



Jakub Kraus

Mladá Boleslav 2022

STŘEDNÍ PRŮMYSLOVÁ ŠKOLA  
MLADÁ BOLESLAV

**ROČNÍKOVÁ PRÁCE**

**Mario**

**Autor: Jakub Kraus  
Studijní obor: 19-23-M/01 Informační technologie  
Vedoucí práce: Jan Till**

Mladá Boleslav 2022

# Obsah Práce

[Obsah Práce 3](#_Toc102328224)

[1 Úvod 6](#_Toc102328225)

[2 Grafický design 7](#_Toc102328226)

[2.1 Textury 7](#_Toc102328227)

[2.2 Animace 7](#_Toc102328228)

[2.3 Tlačítka 8](#_Toc102328229)

[2.4 Pozadí 8](#_Toc102328230)

[3 Hudba 8](#_Toc102328231)

[4 Části kódu 8](#_Toc102328232)

[4.1 Hráč 8](#_Toc102328233)

[4.1.1 Pohyb 9](#_Toc102328234)

[4.1.2 Kolize 9](#_Toc102328235)

[4.1.3 Střílení 10](#_Toc102328236)

[4.2 Nepřítel 10](#_Toc102328237)

[4.2.1 Pohyb 10](#_Toc102328238)

[4.2.2 Kolize 10](#_Toc102328239)

[4.3 Objekty 11](#_Toc102328240)

[4.3.1 Mince 11](#_Toc102328241)

[4.3.2 Vstup do další úrovně 11](#_Toc102328242)

[4.3.3 Překážky 12](#_Toc102328243)

[5 Závěr 13](#_Toc102328244)

[6 Přílohy 14](#_Toc102328245)

[6.1 Seznam obrázků 14](#_Toc102328246)

[6.2 Zdroje 14](#_Toc102328247)

**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

Smyslem této práce bylo vytvořit hru, která se podobá hře Mario. Práce obsahuje všechny základní funkce a mechaniky zmíněného Maria. Vše je napsané v herním enginu Godot (ve verzi 3.4.4), který je volně k použití. Hra je spustitelná v spustitelném formátu exe.

# Grafický design

## Textury

Všechny textury použité ve hře jsou kreslené mnou v programu malování kromě textury hráče, zde je se slovní dohodou použita oříznutá fotka spolužáka Petra Trana a textury pozadí, kde jsem použil fotku moře vyfocenou Petrem Tranem ve Vietnamu. Všechny textury mají určitou výšku a šířku, aby mohli bez problémů zapadnout do scény hry. Určité textury jsou kvůli animacím nakreslené do tzv. tile setů.

Obrázek : Tile set hráče

## Animace

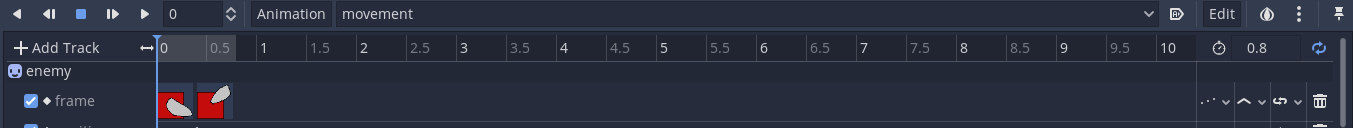
Animace jsou použity pouze u hráče, nepřátel, kolizí s blokem s odměnou, rotující překážkou a masožravou květinou.

Animace hráče a nepřátel požadovaly, aby byli jejich textury dělané do tile setů. Godot umí tyto tile sety předělat na snímky animací, které se poté použijí v uzlech. Byla potřeba zvolit animace typu property track frames. Animace hráče má osm snímků a animace všech nepřátel dva snímky.

Pro animaci rotující překážky sem použil animaci typu property track rotation degrees. Tento druh animace otáčí objekt o daný počet stupňů v časovém úseku.

Animace masožravé kytky je dělaná přes animaci typu property track position. A pro její spuštění je použito nodu AnimationTree.

Kolize s blokem s odměnou potřebovala animaci typu property track position. Po střetnutím s hráčem se blok plynule posune o 10 pixelů nahoru a pak zase dolu.

Všechny animace jsou dělané v nodu typu AnimationPlayer a spuštěné v nodu AnimationTree. 

Obrázek : AnimationPlayer

## Tlačítka

Tlačítka ve hře slouží ke změně jedné scény na druhou nebo k vypnutí hry. Vyskytují se pouze dvě v menu.

Tlačítko new game slouží k zapnutí hry „viz kód 1“.

get\_tree().change\_scene()

kód 1

Tlačítko quit game slouží k vypnutí hry “viz kód 2“.

get\_tree().quit()

kód 2

## Pozadí

Pozadí hry se mění podle úrovně, ve které se právě hráč nachází. Pozadí jsou ve hře 2, kromě menu, a jsou v poměru 1 : 1,7.

# Hudba

Hudba ve hře byla stažena z internetu. Hudba nemá autorská práva a je zdarma k použití. K jejímu přehrání je použit node AudioStreamPlayer2D.

# Části kódu

## Hráč

Hráč je dělaný přes node KinematicBody2D. Hráč je potomek rodiče nodu Actor, který je také KinematicBody2D. Dětí rychlost, gravitaci, směr a detekci kolize.

Pro detekci kolize je použit node Area2D, pro texturu node Sprite, pro kameru Camera2D, pro animace AnimationPlayer a AnimationTree a pro detekci pozice Position2D. Všechny tyto nody jsou potomci Hráče KinematicBody2D.

### Pohyb

Pro zjištění směru je udělaná funkce get\_direction, která od sebe odečítá událost pohybu doleva a událost pohybu doprava a vrací element Vector2 „viz kód 3“, tato funkce se poté používá u funkce pro pohyb.

func get\_direction() -> Vector2:

**return** Vector2(

Input.get\_action\_strength("move\_right") - Input.get\_action\_strength("move\_left"),

-1.0 **if** Input.is\_action\_pressed("jump") and is\_on\_floor() **else** 1.0

)

kód

Pohyb je dělaný přes funkci \_physics\_proccess, která se volá každý snímek hry. Zjišťuje směr pohybu přes zmíněnou funkci get\_direction, detekuje kolize a spouští animace podle směru pohybu.

### Kolize

Pro kolize je udělaný potomek Area2D, který dokáže sám rozpoznávat kolize, které se pohybují na stejné vrstvě. Použil sem signály area\_entered a body\_entered, které jsou unikátní pro zmíněnou Area2D.

Signál area\_entered se stará o zjišťování kolize spodní části hráče a horní části nepřátel pro jejich poražení.

Signál body\_entered se používá pro detekci kolize celého hráče a celého nepřítele. Pokud kolize nastane spustí se funkce die() „viz kód 4“, která hráče zabije a restartuje danou scénu.

func \_on\_EnemyDetector\_body\_entered(body: Node) -> **void**:

die()

kód

### Střílení

Pro střílení projektilů je potřeba sebrat houbičku, po hráčově interakci s houbičkou se spustí část kódu, která dovolí hráčovi střílet projektily.

Pro zjištění směru střel je vytvořená funkce get\_projectile\_direction(), která stejně jako u hráče, odečítá události hráčova pohybu doleva a doprava a ukládá je do proměnné projectile\_direction „viz kód 5“.

func get\_projectile\_direction():

projectile\_direction = Input.get\_action\_strength("move\_left") - Input.get\_action\_strength("move\_right")

kód

Hráč může střílet pouze 10 sekund, k tomuto je použit node typu Timer. Po sebrání houbičky se hráčova velikost zvětší na jeden a půl násobek základní velikosti.

## Nepřítel

Nepřátel se ve hře vyskytují čtyři. Všichni také, stejně jako hráč, jsou potomci nodu Actor a jsou dělaní přes node typu KinematicBody2D. Pro detekci kolize mají vytvořený node Area2D, pro texturu node Sprite a pro animace AnimationPlayer a AnimationTree.

### Pohyb

Pro pohyb je vytvořená funkce \_physics\_process, která násobí rychlost nepřítele s gravitací a proměnou delta „viz kód 6“.

func \_physics\_process(delta: **float**) -> **void**:

\_velocity.y += gravity \* delta

\_velocity.y = move\_and\_slide(\_velocity, FLOOR\_NORMAL).y

kód

### Kolize

O kolize se stará Area2D a její signál body\_entered. Pokud do Area2D vstoupí node, který je ve skupině „Player\_projectile“, tak se spustí funkce die(), která odstraní nepřítele ze hry a přidá hráčovi score „viz kód 7“.

func \_on\_Area2D\_body\_entered(body):

**if** body.is\_in\_group("player\_projectile"):

body.queue\_free()

die()

kód

Pro zjišťování kolizí se solidními objekty, jako třeba zeď, je použita funkce is\_on\_wall(), která je v enginu před vytvořená a stačí ji jenom importovat.

## Objekty

Objekty bez kolizí, která má zastavovat hráče nebo nepřátele jsou ve hře tvořené nodem typu Sprite. Instance s kolizí, která hráče a nepřátele zastaví jsou tvořeny nodem typu KinematicBody2D.

### Mince

Mince jsou tvořené nodem typu Sprite, který je rodičem nodu Area2D.

Zde je opět požito signálu body\_entered, který zjišťuje, zda se node KinematicBody2D na nastavené vrstvě, v tomto případě hráč, dotkl mince. Pokud ano mince se smaže a přidá se Hráčovi score, které se ukládá do jiného nodu s názvem Score, zde je uloženo všechno score, které hráč za úroveň nasbíral.

Pokud hráč umře, nebo vstoupí do jiné úrovně score se nastaví zpátky na nulu.

Obrázek : Mince

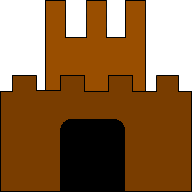
### Vstup do další úrovně

Vstup do další úrovně je opět tvořen nodem typu Sprite a jeho potomkem je node typu Area2D. Pokud se hráč střetne se vstupem do další úrovně score se nastaví na nulu.

Aby vše fungovalo dynamicky tak je ve funkci change\_scene() použita proměnná level, která má začáteční hodnotu 1 a převádí se na řetěz „viz kód 8“. Všechny úrovně ve hře jsou totiž očíslované.

get\_tree().change\_scene("res://src/Levels/Level" + str(Score.level) + ".tscn")

kód



Obrázek : Vstup do další úrovně

### Překážky

Hlavní překážkou ve hře je rotující těleso tvořené nodem KinematicBody2D. Jeho potomky jsou Area2D pro kolize, Sprite pro texturu a AnimationPlayer a AnimationTree pro animace.

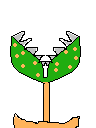
Pro kolize je použito signálu body\_entered. Pokud je podmínka pravdivá spustí se hráčova funkce die().



Obrázek : Rotující těleso

Jako další překážka je masožravá květina, která po kolizi s hráčem spustí jeho funkci die(). Tento node je tvořen přes KinematicBody2D a jeho potomci jsou Area2D pro kolize, Sprite pro texturu a AnimationPlayer pro animaci.

Signal body\_entered() u Area2D se stará o kolizi s hráčem.



Obrázek : Masožravá květina

# Závěr

Vývoj práce se obešel bez větších problému či záseků a vše šlo více méně podle předem stanoveného plánu. Ovšem to neznamená že žádné problémy nenastaly. Jelikož jsem s tímto enginem pracoval poprvé tak sem nevěděl určité funkce a vychytávky enginu. Jeden z problémů byl uchovávání dat jako, třeba score, přes všechny scény hry. Chtěl jsem, aby bylo možné proměnnou scóre upravovat ve všech objektech hry a aby tuto proměnou viděli všechny scény. Vyřešil jsem to vytvořením základního nodu typu Node2D, který jsem poté v editor settings nastavil na autoload, tímto způsobem vše fungovalo bez problémů. Nejhorší částí práce bylo nastavování vrstev a masek objektům. Vrstvy a masky se používají pro tvorbu kolizí, to znamená že vrstva jedna může interagovat s maskou jedna. Godot podporuje pojmenování masek a vrstev, takže jsem si je pojmenoval podle typů objektů, které jsem chtěl, aby se nich objevovali. Kromě těchto problémů nenastali žádné větší potíže. Jsem rád že jsem si práci právě s tímto herním enginem vyzkoušel. Vývoj her mě baví už od dětství a jsem vděčný za tuto zkušenost.

# Přílohy

## Seznam obrázků

[Obrázek 1: Tile set hráče 7](#_Toc102328090)

[Obrázek 2: AnimationPlayer 8](#_Toc102328091)

[Obrázek 3: Mince 11](#_Toc102328092)

[Obrázek 4: Vstup do další úrovně 12](#_Toc102328093)

[Obrázek 5: Rotující těleso 12](#_Toc102328094)

[Obrázek 6: Masožravá květina 12](#_Toc102328095)

## Zdroje

Godot Docs. 3.4 branch [online]. Read the Docs, 2019, [cit. 2022-05-01]. Dostupné z: <https://docs.godotengine.org/en/stable/index.html>

Stack Overflow [online]. [cit. 2022-05-01]. Dostupné z: <https://stackoverflow.com>