STŘEDNÉ PRŮMYSLOVÁ ŠKOLA

MLADÁ BOLESLAV

**ROČNÍKOVÁ PRÁCE**

**World’s Hardest Game**



Štěpánka Klimešová

Mladá Boleslav 2022

STŘEDNÍ PRŮMYSLOVÁ ŠKOLA  
MLADÁ BOLESLAV

**ROČNÍKOVÁ PRÁCE**

**World’s Hardest Game**

**Autor: Štěpánka Klimešová  
Studijní obor: 19-23-M/01 Informační technologie  
Vedoucí práce: Lukáš Mázl**

Mladá Boleslav 2022

# Obsah Práce

[Obsah Práce 3](#_Toc102327382)

[1 Úvod 6](#_Toc102327383)

[2 Grafický design 7](#_Toc102327384)

[2.1 Textury 7](#_Toc102327385)

[2.2 Animace 7](#_Toc102327386)

[2.3 Tlačítka 7](#_Toc102327387)

[2.4 Pozadí 8](#_Toc102327388)

[3 Hudba 8](#_Toc102327389)

[4 Části kódu 8](#_Toc102327390)

[4.1 Hráč 8](#_Toc102327391)

[4.1.1 Pohyb 9](#_Toc102327392)

[4.1.2 Kolize 10](#_Toc102327393)

[4.2 Projektil 10](#_Toc102327394)

[4.3 Mince 11](#_Toc102327395)

[4.4 Vstup do další úrovně 11](#_Toc102327396)

[4.5 Závěr 12](#_Toc102327397)

[4.6 Zdroje 12](#_Toc102327398)

**Prohlášení**

Prohlašuji, že jsem svou ročníkovou práci vypracovala samostatně a použila 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 2D hru ve stylu hry World’s Hardest game. Všechny mechaniky a základní funkce jsou zde tedy velmi podobné.

Základní mechaniky jsou pohyb hráče, pohyb projektilů, sbírání mincí a vstup do další úrovně. Hráč má za cíl splnit všechny úrovně s, pokud co možno největším počtem mincí.

K vytvoření této hry jsem použila herní engine Godot ve verzi 3.4.4, který je zdarma k použití. Hru lze pomocí Godotu uložit a spustit přes formát spustitelného souboru exe.

# Grafický design

## Textury

Všechny textury a grafické prvky jsem nakreslila v grafickém programu Gimp a také jsem k práci použila grafický tablet. Jednotlivé prvky jsou kreslené ve 2D zobrazení a mají nastavené přesné a neměnné rozměry.

Textura hráče je z důvodu animace pohybu nakreslená do tzv. tile setu. Více grafických prvků v jednom obrázku, každý obrázek je jeden snímek animace.

## Animace

Animace ve hře je použita pouze u hráče. Pro animaci hráče byl vytvořený takzvaný uzel AnimationPlayer, což je datový kontejner, který může obsahovat více animací, které mohou automaticky přecházet jedna do druhé.

K dosažení plynulého pohybu stačí dva klíčové snímky. V této hře však byla na pohyb do určitého směru použita animace o čtyřech snímcích v časovém rozestupu nula celá dvě vteřiny.

Animaci byla dána do smyčky a poté uložena. Uložená animace byla vložena do scény AnimationTree a pomocí grafu byl na ose x a ose y nastavený směr hráče podle vstupu zmáčknutých kláves uživatelem. Spuštění animace viz zdrojový kód č.1.

$AnimationTree.set("parameters/Walk/blend\_position", velocity)

zdrojový kód č. 1

## Tlačítka

Tlačítka jsou ve hře použita dvě, a to pro spuštění a pro ukončení hry. Obě tlačítka jsou ve scéně Menu, která se spustí na začátku hry a kdykoliv hráč zmáčkne klávesu escape během hry. Pro tlačítka je použitý TextureButton node, do kterého lze vkládat vlastní textury.

Jedna z použitých vlastností TextureButton je Hover a Pressed, kde je vložena odlišná textura. To způsobí změnu textury tlačítka při kliknutí nebo při přejetím myši přes tlačítko.

Na obou tlačítkách je připojený signál pressed(), který zajistí změnu scény po kliknutí na tlačítko. Změna scény viz zdrojový kód č.2.

func \_on\_start\_pressed():

get\_tree().change\_scene("res://level\_0.tscn")

zdrojový kód č. 2

## Pozadí

Pro grafické pozadí hry jsou zde použity textury tzv. tile sety o velikosti 64 x 64 pixelů, jež byly uloženy do TileMapy a poté postupně vkládány do pozadí hry.

Hrací plocha se tedy sestává z malých čtvercových grafických obrazů označovaných jako dlaždice rozložené v mřížce.

# Hudba

Hudba bez autorských práv, která hraje v pozadí hry, byla stažena na internetu a vložena do scény Music. V této scéně byl vytvořen node AudioStreamPlayer, kde byla hudba nahrána a zdrojovým kódem viz zdrojový kód č.3 spuštěna.

func \_process(delta):

if $AudioStreamPlayer.playing == false:

$AudioStreamPlayer.play()

pass

zdrojový kód č. 3

# Části kódu

## Hráč

Pro hráče je zde použitý node KinematicBody2D což je speciální typ tělesa, který je určený k ovládání uživatelem.

### Pohyb

Godot umožňuje v nastavení projektu zvolení vstupu, který bude po uživateli vyžadován k ovládání hráče. Pro nastavení vstupu se zvolí fyzická klávesa, která se potvrdí, ve zdrojovém kódu propojí a nastaví směr. Viz zdrojový kód č.4.

**if Input.is\_action\_pressed("ui\_left"):**

**velocity.x -= 1**

zdrojový kód č. 4

Pro samotný pohyb je použita funkce get\_input(), ve které se zkontrolují čtyři vstupy, které uživatel může stisknout a přidá se k nim směr podle osy x nebo osy y.

Aby se předešlo zrychlení pohybu při stisknutí dvou kláves najednou, je zde použita normalizace proměnné velocity. Normalizací proměnné dosáhneme toho, že se délka nastaví na hodnotu jedna a poté ji vynásobíme požadovanou rychlostí uložené v proměnné speed.

Viz zdrojový kód č.5.

**func get\_input():**

**velocity = Vector2()**

**if Input.is\_action\_pressed("ui\_left"):**

**velocity.x -= 1**

**if Input.is\_action\_pressed("ui\_right"):**

**velocity.x += 1**

**if Input.is\_action\_pressed("ui\_up"):**

**velocity.y -= 1**

**if Input.is\_action\_pressed("ui\_down"):**

**velocity.y += 1**

**velocity = velocity.normalized() \* speed**

zdrojový kód č. 5

### Kolize

Hráč má přidaný node CollisionShape2D, u kterého je nastavený tvar obdélníku o velikosti hráče, aby kolize na hráči přesně seděla a mohla detekovat kolize s ostatními objekty.

Je zde také přidaný node Area2D, ke kterému je připojený signál on\_projectile\_area\_entered zapsaný ve funkci, viz zdrojový kód č.5, zajišťující resetování úrovně po střetu projektilu a hráče.

Tato Area2D je přidaná do vytvořené skupiny player z důvodu více se vyskytujících stejných nodů. Ve zdrojovém kódu je název skupiny zapsán do funkce, ve které je podmínka pro vykonání kódu, pokud se node nachází ve skupině. U ostatních nodů, které do této skupiny nespadají, se kód nevykoná. Viz zdrojový kód č.6.

**func \_on\_projectile\_area\_entered(area):**

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

**get\_tree().reload\_current\_scene()**

zdrojový kód č. 6

## Projektil

Jsou zde naprogramované dva druhy projektilů, a to projektil s vertikálním směrem a projektil se směrem horizontálním.

Oba projektily mají naprogramovaný směr a určenou rychlost, viz zdrojový kód č.11.

**func \_process(delta):**

**var motion = Vector2(0,1) \* speed**

**position += motion \* delta**

zdrojový kód č. 7

K projektilům jsou připojené signály detekující kolizi s objektem ohraňující hrací pole. V případě kolize jsou projektily naprogramované změnit směr na opačný ke stávajícímu. Viz zdrojový kód č.8.

**func \_on\_vertical\_projectile\_body\_entered(body):**

**speed \*= -1**

zdrojový kód č. 8

## Mince

Sbírání mincí je na podobném principu jako kolize hráče s projektilem. Mince je ve skupině coin a je nastavená jako Area2D s kolizí, u které je připojený signál coin\_area\_entered. O přepočítávání mincí se stará funkce uložená ve skriptu Count jmenující se update\_coin, která přepočítává vytvořenou proměnnou Coin. Viz zdrojový kód č.9.

**var Coin = 0**

**func update\_coin(var delta):**

**Coin += delta**

zdrojový kód č. 9

Ve skriptu Coin je připojený zmíněný signál, skupina a použitá funkce ze skriptu Count. U funkce Count je zadaná hodnota jedna tzn. hráči se bude po sebrání mince vždy připočítávat plus jedna k celkovému počtu sebraných mincí. Viz zdrojový kód č.10.

**func \_on\_Node2D\_area\_entered(area):**

**if area.is\_in\_group("coin"):**

**queue\_free()**

**Count.update\_coin(1)**

zdrojový kód č. 10

## Vstup do další úrovně

Další úroveň se spustí, jakmile hráč vstoupí do oblasti Area2D, kde se spustí kód viz zdrojový kód č.11

**func \_on\_nextLevel\_area\_entered(area):**

**get\_tree().change\_scene("res://level\_"+str(int(get\_tree().current\_scene.name) + 1 ) + ".tscn")**

zdrojový kód č. 11

Po vstoupení hráče do oblasti další úrovně se spustí funkce s připojeným signálem a změní se scéna.

Scény se mění popořadě a úrovně proto musí být správně pojmenovány.

# Závěr

Vytváření práce šlo z velké části podle plánu a problémy skoro žádné nebyly. S herním enginem Godot jsem pracovala poprvé a spoustu věcí jsem se během vývoje hry naučila.

Grafické prvky a textury zabrali na vývoji hry nejspíše nejvíce času. Psaní kódu nebyl problém díky dokumentaci, která je ke Godotu dostupná. Celkově to byla dobrá zkušenost a jsem ráda, jak se hry vydařila.

# Přílohy

## 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>

Stackoverflow [online]. 2019 [cit. 2022-05-01]. Dostupné z: <https://stackoverflow.com/questions/69860324/how-could-i-create-a-music-file-using-godot>