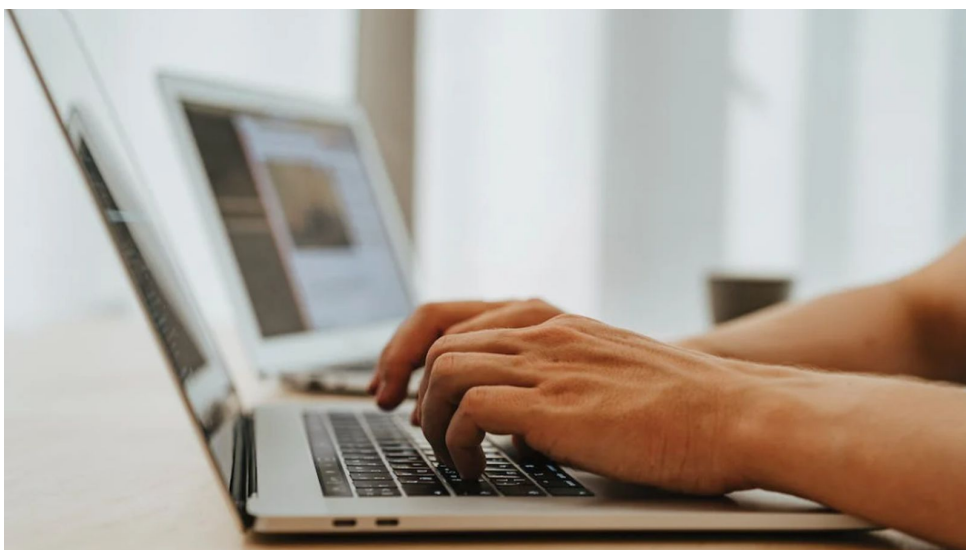


ZÁVĚREČNÁ STUDIJNÍ PRÁCE

dokumentace

2D hra - JumpAdventure



Autor: Filip Kožaný
Obor: 18-20-M/01 INFORMAČNÍ TECHNOLOGIE
se zaměřením na počítačové sítě a programování
Třída: IT4
Školní rok: 2024/25

Poděkování

Děkuji panu učiteli Mgr. Markovi Lucnemu a Ing. Petru Grussmanovi za rady

Prohlášení

Prohlašuji, že jsem závěrečnou práci vypracoval samostatně a uvedl veškeré použité informační zdroje.

Souhlasím, aby tato studijní práce byla použita k výukovým a prezentačním účelům na Střední průmyslové a umělecké škole v Opavě, Praskova 399/8.

V Opavě 1. 1. 2025

.....
Podpis autora

Abstrakt

Tato práce se zaměřuje na návrh a vývoj 2D plošinové hry, která je postavena na jednoduchém a zábavném herním mechanismu – skákání. Hráč ovládá postavu, jejímž cílem je překonávat různé překážky, zdolávat platformy a dosáhnout konce jednotlivých úrovní. Herní design klade důraz na přesné načasování, pohyb a rostoucí obtížnost, která motivuje hráče k zlepšování svých dovedností.

Hra zahrnuje několik úrovní, které jsou postupně složitější, s dynamickými prvky, jako jsou pohyblivé platformy, nebezpečné pasti a interaktivní objekty. Důležitým aspektem je také vizuální styl, který vytváří poutavé prostředí, a soundtrack, který podporuje atmosféru hry.

Cílem projektu je vytvořit zábavnou a návykovou hru, která bude přístupná široké škále hráčů, od příležitostných hráčů až po nadšence, kteří hledají výzvy. Hra demonstruje základní principy herního designu a poskytuje vhled do tvorby 2D herních mechanik.

Klíčová slova

2D plošinovka, skákání, herní design, pohyblivé platformy, obtížnost, herní mechaniky, vizuální styl, herní vývoj, interaktivní objekty, zábavná hra.

Abstract

This project focuses on the design and development of a 2D platformer game built around a simple and engaging gameplay mechanic: jumping. The player controls a character whose goal is to overcome various obstacles, traverse platforms, and reach the end of each level. The game design emphasizes precise timing, movement, and progressively increasing difficulty, encouraging players to improve their skills.

The game features multiple levels that become more challenging over time, with dynamic elements such as moving platforms, hazardous traps, and interactive objects. A key aspect of the project is the visual style, which creates an immersive environment, along with a soundtrack that enhances the game's atmosphere.

The objective of the project is to create an enjoyable and addictive game that is accessible to a wide range of players, from casual gamers to enthusiasts seeking challenges. The game demonstrates fundamental principles of game design and provides insight into the development of 2D game mechanics.

Keywords

2D platformer, jumping, game design, moving platforms, difficulty, game mechanics, visual style, game development, interactive objects, fun game.

Obsah

Úvod	3
1 TEOTETICKÁ ČÁST	5
1.1 Úvod	5
1.2 Co je 2D hra	5
1.3 Co je Godot engine	5
1.4 Co je Krita	6
2 TECHNOLOGIE	9
2.1 Godot	9
2.2 Krita	11
3 Ukázka kódů	15
3.1 Ukázka kódu pohybu postavy	15
3.2 Ukázka kódu Menu	16
3.3 Ukázka kódu úvodního intra	16
3.4 Ukázka kódu nastavení	17
3.5 Možná vylepšení	19

ÚVOD

Tato dokumentace se věnuje návrhu, vývoji a implementaci 2D plošinové hry, která staví na základní herní mechanice skákání. Cílem projektu je vytvořit hru, která je jednoduchá na ovládání, ale zároveň nabízí postupně rostoucí obtížnost a výzvy, jež udrží zájem hráče po celou dobu hraní.

Hra je navržena jako návyková zábava pro široké spektrum hráčů, od příležitostných hráčů hledajících relaxaci, až po zkušenější uživatele toužící po zdokonalení svých reflexů a přesnosti. Projekt si klade za cíl nejen zprostředkovat herní zážitek, ale také demonstrovat praktické využití herních mechanik, designových prvků a technologických postupů při tvorbě 2D her.

V dokumentaci jsou popsány všechny klíčové aspekty vývoje hry – od počáteční analýzy a návrhu herního konceptu, přes implementaci herních mechanik a grafiky, až po testování a optimalizaci. Projekt rovněž zdůrazňuje důležitost herní estetiky a zvukového doprovodu, které dohromady vytvářejí poutavé herní prostředí.

Dokumentace slouží jako průvodce nejen pro pochopení celého procesu vývoje, ale také jako zdroj inspirace pro podobné projekty v oblasti herního designu.

1 TEOTETICKÁ ČÁST

1.1 ÚVOD

1.2 CO JE 2D HRA

2D hra je videohra, která se odehrává ve dvourozměrném prostoru, kde jsou herní objekty reprezentovány na rovině s osami X (vodorovná osa) a Y (svislá osa). Na rozdíl od 3D her, které zahrnují také hloubkovou osu (Z), 2D hry používají ploché grafické prvky a jednoduchou perspektivu.

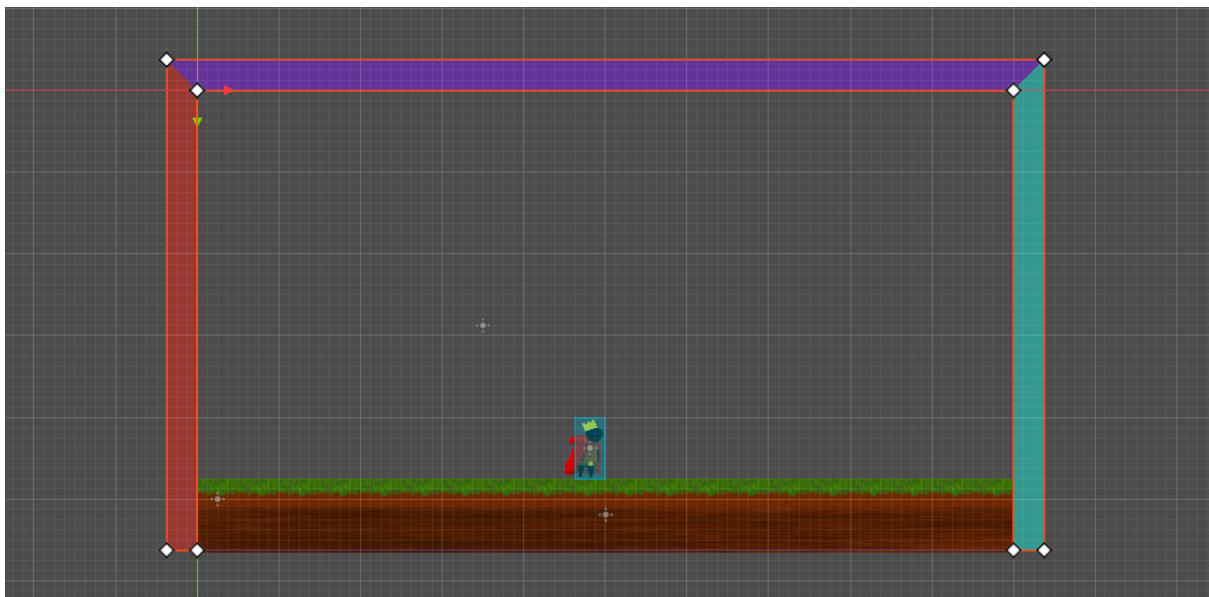
Hratelnost v 2D hrách se obvykle odehrává z pohledu shora (top-down), z boku (side-scroller), nebo izometrického pohledu, kde objekty vypadají jako trojrozměrné, ale zachovávají si omezení 2D prostoru. Mezi nejběžnější typy 2D her patří plošinovky, adventury, puzzle hry, akční hry, arkádové tituly nebo strategické hry.

1.3 CO JE GODOT ENGINE

Godot Engine je open-source herní engine určený pro vývoj 2D i 3D her. Je známý svou flexibilitou, jednoduchostí a tím, že je zdarma k použití, a to i pro komerční projekty. Godot je vyvíjen komunitou a financován především prostřednictvím darů na platformách jako Patreon nebo GitHub Sponsors.

- Podpora 2D a 3D her
- Scene System
- Scripting
- Multiplatformní podpora
- Integrovaný editor
- Open Source
- Aktivní komunita

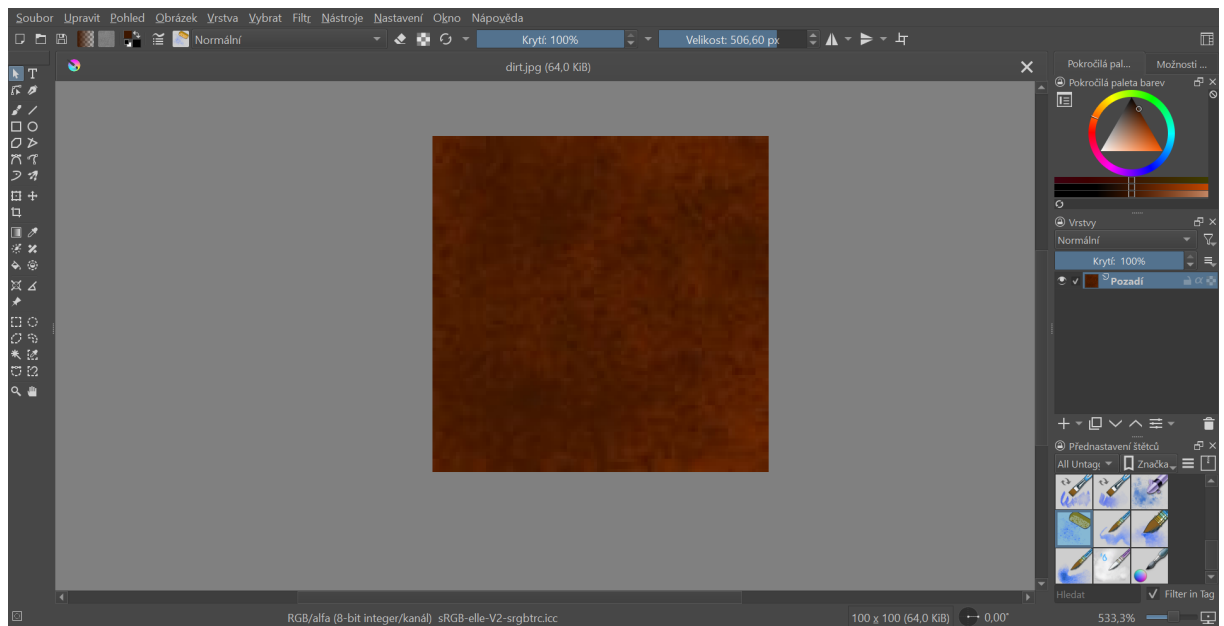
- Lehký a efektivní



1.4 CO JE KRITA

Krita je open-source a zdarma dostupný software určený především pro digitální malbu, kreslení a tvorbu ilustrací. Je vysoce oblíbený mezi profesionálními i amatérskými umělci díky svým výkonným funkcím, snadnému použití a pravidelným aktualizacím. Krita je vyvíjena komunitou pod licencí GNU General Public License (GPL).

- Přizpůsobitelný pracovní prostor
- Rozsáhlá podpora štětců
- Pokročilé nástroje pro vrstvy
- Animace
- Podpora barevných modelů
- Otevřenost a přizpůsobitelnost
- Komunita a výuka



Kdo používá Kritu?:

- Ilustrátory a konceptuální umělce
- Tvůrce komiksů
- Digitální malíře
- Animátory hledající bezplatný, ale výkonný nástroj

2 TECHNOLOGIE

2.1 GODOT

2.1.1 Jádro a programovací jazyky

- GDScript: Primární skriptovací jazyk inspirovaný Pythonem, navržený specificky pro Godot.
- C#: Godot podporuje integraci s jazykem C# (prostřednictvím Mono), což ocení vývojáři z prostředí .NET.
- C++: Samotný engine je napsán v C++ a pokročilí uživatelé mohou vytvářet nativní moduly pomocí GDNative.
- VisualScript: Nástroj pro vizuální skriptování, vhodný pro ty, kteří preferují práci bez psaní kódu.
- Rust, Python a další: Pomocí pluginů nebo GDNative je možné použít i další jazyky.

2.1.2 Grafika

- Rendering:
 - Godot používá dva odlišné renderovací enginy:
 - * OpenGL ES 3.0: Pro modernější zařízení a pokročilé grafické efekty.
 - * OpenGL ES 2.0: Pro starší zařízení a jednodušší grafiku, zejména v 2D.
 - Vulkan: Od verze Godot 4.0 nabízí engine plnou podporu Vulkan API pro lepší výkon a moderní grafické funkce.
- PBR (Physically Based Rendering): Podporuje realistické materiály včetně odrazů, stínů a osvětlení.
- Shader Language: Umožňuje psát vlastní shadery pro pokročilé vizuální efekty.

2.1.3 2D technologie

Godot nabízí jedny z nejlepších nástrojů pro 2D vývoj:

- Přesné vykreslování 2D bez aliasingu.
- Správa 2D fyziky, včetně podpory Box2D a vlastního fyzikálního enginu.
- Pokročilý systém animací a editor.

2.1.4 Fyzika

- Bullet Physics Engine: Používá se pro simulaci fyziky v 3D.
- Built-in physics: Godot má vlastní fyzikální engine pro 2D a jednoduché 3D simulace.

2.1.5 Zvuk

Godot obsahuje plně integrovaný audio engine:

- Podpora 2D a 3D prostorového zvuku.
- Efekty, jako jsou dozvuk (reverb) a ekvalizér.
- Streamování zvuku pro hudbu a velké audio soubory.

2.1.6 Multiplatformní podpora

Godot umožňuje export her na různé platformy:

- PC: Windows, macOS, Linux.
- Mobilní zařízení: Android, iOS.
- Web: HTML5 (WebAssembly).
- Konzole: PlayStation, Xbox, Nintendo Switch (často s doplňkovými nástroji).

2.1.7 Správa paměti a výkon

Resource Loader/Saver: Optimalizované načítání a ukládání herních dat.

Godot Packed Scene: Efektivní systém pro ukládání a načítání herních objektů.

2.1.8 UI a UX technologie

Godot UI systém: Nabízí bohaté nástroje pro tvorbu uživatelských rozhraní s podporou adaptivních layoutů a skinování.

Scalable Vector Graphics (SVG): Podpora pro vektorovou grafiku.

2.1.9 Síťování

Godot podporuje síťové protokoly, včetně WebSocket, TCP, a UDP.

Nabízí možnosti pro tvorbu multiplayerových her, včetně synchronizace hráčů a RPC (Remote Procedure Call).

2.2 KRITA

2.2.1 Jádro a programovací jazyky

- C++: Krita je napsaná primárně v C++, což zajišťuje vysoký výkon a nízkou spotřebu paměti.
- Qt Framework: Krita využívá knihovnu Qt pro tvorbu uživatelského rozhraní a správu grafických funkcí. Qt je multiplatformní, což umožňuje snadný běh Krita na Windows, macOS a Linuxu.

2.2.2 Grafický engine

- OpenGL: Krita využívá OpenGL pro akceleraci vykreslování a zajištění plynulého chodu i při práci s velkými plátny nebo složitými efekty.
- Canvas Graphics Engine: Krita má vlastní vykreslovací systém optimalizovaný pro práci s vysokým rozlišením a vrstvami.

2.2.3 Barevné modely a správa barev

- Podpora barevných modelů: Krita pracuje s různými barevnými modely, včetně RGB, CMYK, Lab, a Grayscale.
- ICC Profily: Krita podporuje správu barev prostřednictvím ICC profilů, což je důležité pro tisk a přesné zobrazení barev.
- HDR podpora: Krita umí pracovat s vysokým dynamickým rozsahem (HDR), což je ideální pro moderní monitory a profesionální práci.

2.2.4 Systém vrstev

- Bitmapové vrstvy: Základní vrstva pro kreslení.
- Vektorové vrstvy: Pro práci s vektorovou grafikou.
- Masky a filtry: Krita umožňuje používat masky a nelineární filtry přímo na vrstvách.

2.2.5 Štětce a efekty

- Brush Engine: Krita má výkonný systém pro štětce s podporou simulace tradičních médií (vodové barvy, olej, pastelky apod.).
- Custom Brush Settings: Každý štětec může být přizpůsoben do detailů, včetně tlaku, textury a dynamiky.
- Shader Brushes: Použití OpenGL shaderů pro speciální efekty na štětce.

2.2.6 Animace

- Krita má vestavěné nástroje pro tvorbu 2D animací, včetně časové osy a onion skinning.
- Podpora exportu do formátů jako MP4, GIF nebo obrázkové sekvence.

2.2.7 Podpora souborů

- Krita podporuje mnoho formátů, včetně PSD (Photoshop), PNG, TIFF, JPEG a vlastní formát .kra.
- OpenRaster: Krita je součástí iniciativy OpenRaster pro standardizaci formátů pro digitální kreslení.

2.2.8 Platformy a rozšíření

- Multiplatformní: Krita běží na Windows, macOS, Linuxu a některých mobilních zařízeních (např. Android tablety).
- Python Scripting: Krita podporuje skriptování v Pythonu pro automatizaci úloh a tvorbu vlastních nástrojů.
- Pluginy: Krita umožňuje rozšiřovat funkce pomocí pluginů, např. pro práci s texturami nebo speciálními efekty.

2.2.9 Kompatibilita hardwaru

- Tablety a stylusy: Krita podporuje většinu grafických tabletů, včetně Wacom, Huion a dalších, s funkcemi jako tlak, náklon a otočení.
- 4K a HiDPI monitory: Krita je optimalizovaná pro vysoká rozlišení.

2.2.10 Open Source

Krita je vyvíjena jako open-source projekt pod licencí GPL (GNU General Public License), což znamená, že její kód je veřejný a může být upravován komunitou.

3 UKÁZKA KÓDŮ

3.1 UKÁZKA KÓDU POHYBU POSTAVY

```
1  extends CharacterBody2D
2
3
4  const SPEED = 300.0
5  const JUMP_VELOCITY = -400.0
6
7
8  func _physics_process(delta: float) -> void:
9      >| # Add the gravity.
10     >| if not is_on_floor():
11         >| velocity += get_gravity() * delta
12
13     >| # Handle jump.
14     >| if Input.is_action_just_pressed("ui_accept") and is_on_floor():
15         >| velocity.y = JUMP_VELOCITY
16
17     >| # Get the input direction and handle the movement/deceleration.
18     >| # As good practice, you should replace UI actions with custom gameplay actions.
19     >| var direction := Input.get_axis("ui_left", "ui_right")
20     >| if direction:
21         >| velocity.x = direction * SPEED
22     >| else:
23         >| velocity.x = move_toward(velocity.x, 0, SPEED)
24
25     >| move_and_slide()
```

3.2 UKÁZKA KÓDU MENU

```
1 extends Control
2
3 ▾ func _ready() -> void:
4     > $VBoxContainer/START.grab_focus()
5
6
7
8
9 ▾ func _on_start_pressed() -> void:
10    > get_tree().change_scene_to_file("res://Scenes/Level0ne.xml.tscn")
11
12
13 ▾ func _on_options_pressed() -> void:
14    > get_tree().change_scene_to_file("res://Scenes/options.tscn")
15    >
16
17
18
19 ▾ func _on_quit_pressed() -> void:
20    > get_tree().quit()
```

3.3 UKÁZKA KÓDU ÚVODNÍHO INTRA

```
1 extends Node2D
2
3
4 # Called when the node enters the scene tree for the first time.
5 ▾ func _ready() -> void:
6
7     > $AnimationPlayer.play("Fade in")
8     > await get_tree().create_timer(6).timeout
9     > $AnimationPlayer.play("Fade out")
10    > await get_tree().create_timer(3).timeout
11    > get_tree().change_scene_to_file("res://Scenes/menu.tscn")
12
```

3.4 UKÁZKA KÓDU NASTAVENÍ

```
1  extends Control
2
3
4
5  ▼ func _on_volume_value_changed(value: float) -> void:
6    >|  AudioServer.set_bus_volume_db(0,value)
7
8
9  ▼ func _on_mute_toggled(toggled_on: bool) -> void:
10   >|  AudioServer.set_bus_mute(0,toggled_on)
11
12
13 ▼ func _on_resolutions_item_selected(index: int) -> void:
14 ▼ >|  match index:
15 ▼ >|  >|  0:
16   >|  >|  >|  DisplayServer.window_set_size(Vector2i(1920,1080))
17 ▼ >|  >|  1:
18   >|  >|  >|  DisplayServer.window_set_size(Vector2i(1600,900))
19 ▼ >|  >|  2:
20   >|  >|  >|  DisplayServer.window_set_size(Vector2i(1280,720))
21
22 ▼ func _on_back_menu_pressed() -> void:
23   >|  get_tree().change_scene_to_file("res://Scenes/menu.tscn")
24
```


ZÁVĚR

V této práci jsem se zaměřil na vytvoření základního herního konceptu zahrnujícího mechaniku skákání a systém výběru úrovní. Tyto prvky představují klíčové komponenty pro zábavný a uživatelsky přívětivý herní zážitek. Implementace herní fyziky, plynulého pohybu a přehledného uživatelského rozhraní pro výběr úrovní byly hlavními výzvami, které se mi podařilo úspěšně překonat.

Mechanika skákání byla optimalizována tak, aby byla intuitivní a přístupná pro hráče různých úrovní dovedností. Systém výběru levelů umožňuje hráčům snadno navigovat mezi úrovněmi, což podporuje plynulý průběh hry.

3.5 MOŽNÁ VYLEPŠENÍ

- Rozšíření herních mechanik
- Vylepšení vizuální stránky
- Multiplayer a sociální prvky

LITERATURA

- [1] OpenAI, ChatGPT, Jazykový model umělé inteligence. Dostupné z: <https://openai.com/chatgpt>, [cit. 2025-01-02].
- [2] YouTube, Godot Engine - Complete Game Dev Tutorials, Playlist vytvořená uživatelem Godot Tutorials. Dostupné z: <https://www.youtube.com/playlist?list=PL9FzW-m48fn1iR6WL4mjXtGi8P4TaPIAp>, [cit. 2025-01-02].
- [3] Godot Engine, Oficiální repozitář na GitHubu. Dostupné z: <https://github.com/godotengine/godot>, [cit. 2025-01-02].