

Ročníkový projekt

Gymnázium, Praha 6, Arabská 14

Arabská 14, Praha 6, 160 00



Předmět: Programování

Téma: Platform shooter hra

Autor: Vítězslav Procházka

Třída: 2.E

Školní rok: 2022/2023

Vyučující: Mgr. Jan Lána

Třídní učitel: Mgr. Blanka Hniličková

Čestné prohlášení:

Prohlašuji, že jsem jediným autorem tohoto projektu, všechny citace jsou řádně označené a všechna použitá literatura a další zdroje jsou v práci uvedené. Tímto dle zákona 121/2000 Sb. (tzv. Autorský zákon) ve znění pozdějších předpisů uděluji bezúplatně škole Gymnázium, Praha 6, Arabská 14 oprávnění k výkonu práva na rozmnožování díla (§ 13) a práva na sdělování díla veřejnosti (§ 18) na dobu časově neomezenou a bez omezení územního rozsahu.

V Praze dne

Vítězslav Procházka

Poděkování

Rád bych poděkoval panu profesoru Mgr. Janu Lánovi, za vedení mého ročníkového projektu. Dále rád bych rád poděkoval mému tátovi Mgr. Jiřímu Procházkovi, se kterým jsem konzultoval několik problémů, jež se objevily během tvorby mého projektu.

Abstrakt:

Toto je dokumentace k mému ročníkovému projektu na téma platform shooter hra. Hra byla vytvářena pomocí herního enginu Godot a je napsána v jazyce godot script. Cílem hry je projít level s co nejvyšším skóre, to hráč získává za zneškodňování nepřátel. Dále hra umožňuje pohyb hráče v 2D prostředí a střelbu projektilů do nepřátel.

Abstract (English):

This is the documentation for my annual project on the topic of a platform shooter game. The game was created using the Godot game engine and is written in the Godot script language. The objective of the game is to complete level with the highest possible score, which the player earns by eliminating enemies. Additionally, the game allows for player movement in a 2D environment and shooting enemies.

Zadání projektu

- Vytvořte hru s 2D movementem (pohybem) doleva, doprava a možnost výskoku.
- Cílem hráče bude vybavit svou postavu co nejlepší kombinací zbraní a schopností.
- Ty si může koupit po projití levelu.
- Cílem je projít všechny levely.
- Peníze získá za zničení nepřátel.
- Doba na průchod levelu je omezená, protože hráč musí utíkat doprava před nebezpečím, kterému se nedá bránit.
- Hráč může levely opakovat, ale z levelů, ve kterých již byl, může získat jen peníze, které v předchozích průchodech nezískal - Hráč nemůže získat nekonečno peněz.
- Na konci některých levelů se budou nacházet bossové (silnější nepřátelé), které je potřeba zabít, jinak se hráč nedostane dál.

Platforma

- Godot engine
- Godot script

Obsah

1	Úvod	7
2	Vývojové prostředí - Godot	8
3	Funkcionalita hry	9
3.1	Hlavní menu	9
3.2	Gameplay	10
3.2.1	Začátek levelu	10
3.2.2	Pohyb	10
3.2.3	Dash	11
3.2.4	Nepřátelé	11
3.2.5	Střelba	12
3.2.6	Bouře	12
3.2.7	Smrt	13
3.2.8	Cíl	13
4	Řešená problematika	14
4.1	Střelba	14
4.1.1	Rotace zbraně	14
4.1.2	Výpočet rychlosti střelby	15
4.1.3	přidání projektilu do scény	16
4.2	Pohyb	17
4.2.1	Dash	18
4.3	Nepřítel	20
5	Závěr	21
	Použité zdroje	22
	Seznam obrázků	23

1 Úvod

Předmětem tohoto ročníkového projektu je počítačová hra na principu 2D „platform skákaček“. Hra dostala jméno Bobot, protože hlavní postavu ztvárňuje robot. Hra byla vytvořena v herním enginu Godot, kde jsem využil programovacího jazyka godot script. Veškerý vizuální prvky ve hře jsem sám maloval, popřípadě vytvářel efekty pomocí Godotu. Ve hře se nachází 1 level jako ukázka, jak by zhruba mohly levely vypadat v plno hodnotné verzi hry. V levelu se nacházejí nepřátelé za jejichž zničení hráč získává skóre. K likvidaci nepřátel slouží střelba z různých zbraní.

Toto téma jsem si zvolil, protože mě zajímá tvorba her a přišlo mi, že je potřeba začít u něčeho všem známého, zkrátka se nepouštět do něčeho kompletně nového.

Vybral jsem si hru tvořit v herním enginu, protože jsem se nechtěl zabývat s problémy typu detekce kolize a samotného zobrazení jednotlivých objektů. Chtěl jsem se rovnou věnovat obsahu a mechanikám hry. Godot mě oslovil svou jednoduchostí a dobrými ohlasy z komunity herních tvůrců. Hra byla vyvíjena na verzi enginu 3.4 .

2 Vývojové prostředí - Godot

Godot je multiplatformní herní engine, který umožňuje vývojářům vytvářet videohry pro různé operační systémy, jako jsou Windows, Linux, Mac, Android a iOS. Engine je distribuován jako open-source software pod licencí MIT, což znamená, že je zdarma k použití a může být upravován a distribuován komunitou vývojářů.

Godot nabízí ucelené nástroje pro tvorbu her včetně scénářů, fyzikálního simulátoru, animačního editoru, správy zdrojových kódů, tvorby uživatelského rozhraní a mnoha dalších funkcí. Jeho grafický engine podporuje 2D a 3D rendering s možností využití různých technologií, jako jsou například PBR materiály, globální osvětlení, dynamické stíny a postprocessingové efekty.

Godot se stal populárním mezi nezávislými herními vývojáři a komunitou vývojářů díky svému intuitivnímu rozhraní, výkonu a podpoře pro mnoho programovacích jazyků, včetně C++, C#, Python a GDScript, což je jazyk speciálně navržený pro použití s engineem Godot.[1, 2]

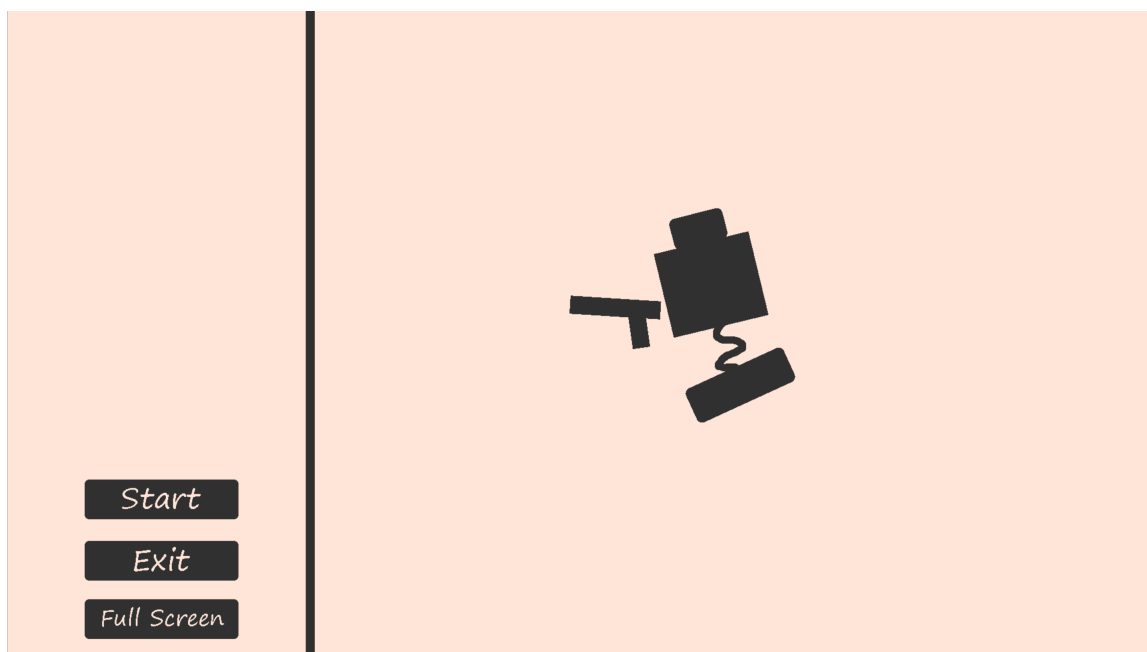


Obrázek 1: Logo Godotu[1]

3 Funkcionalita hry

3.1 Hlavní menu

Na obrázku 2 je vidět hlavní menu hry. To se jako první objeví po spuštění programu. Nachází se zde 3 tlačítka. Tlačítko Start zapne hru, tlačítko Exit hru vypne a tlačítko Full Screen nastaví hru do full screen módu, to se dá také zapnout pomocí kombinace kláves levý „alt + enter“.



Obrázek 2: Hlavní menu

3.2 Gameplay

3.2.1 Začátek levelu

Na obrázku 3 je zobrazen pohled na to co hráč uvidí, když v hlavním menu hry zmáčkne start, tedy to, jak vypadá první pohled na level. Uprostřed se nachází hráč (šedý hranatý robot s hrncem na hlavě), ten je vybaven zbraní, jejíž směr míření je znázorněn fialovou čarou. Dále je zde možno vidět healthbar, který se nachází přímo nad samotným hráčem, je v něm zaznamenán počet bodů zdraví pomocí procent. Vpravo od něj je umístěno počítadlo skóre, jehož hodnota se zvedne při zabití nepřítele.



Obrázek 3: Začátek levelu

3.2.2 Pohyb

Pohyb hráče funguje na bázi vodorovného pohybu doleva nebo doprava. Pro pohyb doleva se používají klávesy: „a“ nebo „šipka doleva“. Pro pohyb doprava se používají klávesy: „d“ a „šipka doprava“. Hráč také může skočit a to i jednou ve vzduchu (možnost double jumpu) skok se aktivuje klávesou: „w“ nebo „šipka na horu“. Na hráče působí gravitace, takže čím déle je ve vzduchu tím rychleji bude padat zpět na zem.

3.2.3 Dash

Dash je další způsob, kterým se hráč může pohybovat. Po jeho aktivaci pomocí klávesy: „shift“ nebo pomocí zmáčknutí pravého tlačítka, je hráč uveden po krátkou chvíli do konstantní rychlosti, která je výrazně rychlejší než hráčova normální rychlost (během dashe na hráče neplatí gravitace). Směr dashe je orientován na ukazatel myši. Dash se stejně jako doublejump dá ve vzduchu použít jen jednou, bohužel jsem nestihl na dash implementovat cooldown, dash se dá na zemi používat neomezeně krát, a dá se během něho střílet.



Obrázek 4: Dash

3.2.4 Nepřátelé

Bohužel kvůli nedostatku času jsem byl schopen vytvořit pouze jeden typ nepřítele a to ducha, který po hráčovi střílí červený laser, to je možné vidět na obrázku 5. Duch se dá pomocí střelby zabít a to tak hráčovi přičte skóre.



Obrázek 5: Nepřítel střílející po hráči

3.2.5 Střelba

Ve hře jsou k dispozici celkem 2 zbraně, jedna která střílí velmi rychle a do více směrů (na obrázku 6) a druhá která je přesnější, pomalejší, ale střílí dvě střely za sebou. Hráč může zbraně měnit pomocí kláves: „1“ a „2“. Hra měla mít obchod, kde si hráč mohl změnit svůj arzenál zbraní, ale bohužel jsem ho nestihl vytvořit.

Vytvořil jsem více zbraní než 2, ale hráč se nemá jak k nim dostat.



Obrázek 6: Střelba ze zbraně

3.2.6 Bouře

Průchod levelu je časově omezen a to formou bouře, která neustále hráče pronásleduje a žene hráče dál směrem doprava dál do levelu. Pokud se hráč nachází v bouři utrpí každý snímek poškození. Bouře je tedy schopna hráče zabít.



Obrázek 7: Hráč umírající v bouři

3.2.7 Smrt

Pokud hráčovy body zdraví klesnou pod 1 bod zdraví umře, to se projeví tak, že se ukáže tzv. „death screen“ s nápisem GAME OVER. Zmáčknutím jakékoliv klávesy se opět spustí level od začátku. Jak death screen vypadá je patrné na obrázku 8.



Obrázek 8: Death screen

3.2.8 Cíl

Cíl je znázorněn žlutým domečkem, do kterého když se hráč dostane hráč vyhraje a je přenesen zpět do hlavního menu.



Obrázek 9: Cíl

4 Řešená problematika

Díky využití herního enginu jsem se nemusel potýkat se základními problémy, které se jinak téměř u každé hry musí vyřešit. Jedná se například o zobrazení spritu (objekt s texturou) a jeho následné rozpohybování, či rotaci. Dále jsem se nemusel zabývat detekcí kolize dvou těles. Všechny dynamické věci, jako je pohyb těles, se dějí kvůli metodě `_process()`, která je volána v každém objektu každý snímek, dále je zde hodnota `delta`, která určuje délku doby od posledního snímku. To se hodí například při výpočtu o jakou vzdálenost musíme posunout pohybující se těleso každý snímek. Hra pak působí plynuleji.

4.1 Střelba

4.1.1 Rotace zbraně

To to je část funkce která je volána každý snímek a otočí zbraň za myši, pokud je zbraň otočena o víc než 90 stupňů je její textura zrcadlena podle osy y, dojde tak realističtějšímu pohybu zbraně.

```
1 func rotate_weapon()->void:
2     #seek mouse position
3     var mouse_pos = get_global_mouse_position()
4
5     #make weapon look at mousposition
6     look_at(mouse_pos) #make weapon look at mousposition
7
8     #calculates rotation of weapon
9     global_rotation = (mouse_pos - global_position).angle()
10
11    #flips weapon
12    if (global_rotation < PI/2 and global_rotation > -PI/2):
13        scale.y = 1
14    else:
15        scale.y = -1
```

4.1.2 Výpočet rychlosti střelby

Zde je rychlostí střelby myšlena velikost intervalů mezi jednotlivými výstřely.

```
1 func calculate_fire_rate()->void:
2
3     #adds value of 1 to actual_fire_state
4     acutal_fire_state = acutal_fire_state + 1
5
6     #when it reaches given fire rate it will fire
7     if (acutal_fire_state == fire_rate) :
8
9         #sets acutal_fire_state to 0 so the cycle can happen
10         again
11         acutal_fire_state = 0
12         shoot()
```

Každá zbraň má pro fire rate nastavenou jinou hodnotu, jedná se o proměnou která určuje, kolik musí uplynout snímků, než zavolá funkci shoot(). Funkce shoot() se stará o samotný jeden výstřel.

4.1.3 přidání projektilu do scény

U přidání projektilu do scény se nastaví jeho otočení, to se spočítá sečtením pomocí součtu úhlu rotace zbraně a v omezeném intervalu náhodně vygenerovaného úhlu, díky tomu je vytvořen rozptyl kulek.[3]

```
1 #prepare rng for randomization
2 func shoot()->void:
3     rng.randomize()
4     #randomize the spread angle in given range
5     spread_angle = rng.randf_range(spread_bottom, spread_top)
6
7     #converts degrees to radians
8     var rad = deg2rad(spread_angle)
9
10    #spawns the projectile
11    var projectile = projectile_scene.instance()
12
13    #sets it to spawn location
14    projectile.global_position = SpawnPosition.
        global_position
15
16    #sets it's rotation and spread angle
17    projectile.global_rotation = global_rotation + rad
18
19    #adds the projectile to the scene
20    get_tree().get_root().get_child(0).add_child(projectile)
```


4.2 Pohyb

To to je funkce která zjistí jakým vodorovným směrem se hráč chce pohybovat, vrátí vektor s Xovou hodnotou 1 nebo -1 podle toho, co hráč zmáčkl za klávesu. [4]

```
1 #returns what direction wants player to go
2 #1 for right -1 for left and 0 for no direction
3 func get_directoin() -> Vector2:
4
5     directionX = Input.get_action_strength("move_right")
6     - Input.get_action_strength("move_left")
7
8     return Vector2(directionX , directionY)
```

Toto je funkce, která počítá novou hráčovu rychlost pro nový snímek, jako parametry dostane starou rychlost, vodorovný směr kterým se hráč chce pohybovat a konstantu rychlosti hráče. To to je první část funkce, která se zabývá horizontálním pohybem. [4]

```
1 #calculates new player velocity
2 func calculate_move_velocity(
3     velocity: Vector2,
4     direction: Vector2,
5     playerSpeed: Vector2
6 ) -> Vector2:
7
8     #horizontal velocity
9     new_velocity.x = playerSpeed.x * direction.x
```

Tohle je druhá část funkce `func calculate_move_velocity()`, tady se vypočítává rychlost skoku a gravitace. Gravitace je každý snímek zvětšována, hráčův pád se tak zrychluje. Dále je tu ošetřen `doublejump`.[\[4\]](#)

```
1      #if player is on floor , dash charge and double jump will
      be recharged
2      if (is_on_floor()):
3          dash_charge = 1
4          doublejump_charge = 1
5
6      #doublejump
7      if( Input.is_action_just_pressed("jump")):
8          if (is_on_floor()):
9              new_velocity.y = calculate_jump_velocity(
              playerSpeed.y)
10             elif(doublejump_charge == 1):
11                 new_velocity.y = calculate_jump_velocity(
                    playerSpeed.y)
12                 doublejump_charge = 0
13                 new_velocity.y += calculate_gravity()
14     return new_velocity
```

4.2.1 Dash

Nejdříve se spočítá jaká bude po dobu dashe rychlost a její směr. K tomu slouží funkce `calculate_dash_velocity()`

```
1 func calculate_dash_velocity():
2     dash.look_at(get_global_mouse_position())
3     return dashspeed * (get_global_mouse_position() -
        global_position).normalized()
```

Dále po aktivaci dashe je zapnut časovač pro dash, pokud časovač běží, je hráčova rychlost nastavována na tu, kterou spočítala fce. `calculate_dash_velocity()`.^[5]

```
1  if (Input.is_action_just_pressed("dash") and dash_charge ==  
    1):  
2      dash.start_dash(dashlength)  
3      dash_velocity = calculate_dash_velocity()  
4      after_dash = true  
5      dash_charge = 0
```

Poslední podmínka je, že po dashi se hráčova vertikální rychlost nastaví na nulu, bez toho by hráč letěl do výšky příliš vysoko, než by jeho rychlost z dashe nepře-násobila ta z gravitace. To je vyřešeno pomocí podmínky, která se nachází v `calculate_move_velocity()`.

```
1  if (after_dash == true):  
2      after_dash = false  
3      new_velocity.y = 0.0
```

4.3 Nepřítel

U nepřítele je nejzajímavější detekce hráčovi pozice a zda-li je hráč v dosahu nepřítele.

```
1 func _on_Area2D_body_entered(body: Node) -> void: #this
    method sets target for it's weapons as player if the
    player entered enemy's range
2     if (body.get_name() == "Player"):
3         player = body
4         for child in get_children():
5             if child.has_method("set_target"):
6                 child.set_target(player)
7
8 func _on_Area2D_body_exited(body: Node) -> void: #this method
    sets target for it's weapons as null if the player left
    enemy's range
9     if (body.get_name() == "Player"):
10        player = null
11        for child in get_children():
12            if child.has_method("set_target"):
13                child.set_target(player)
```

To je vyřešeno pomocí speciálního Godotem nabízeného prvku a tím je Area2D, jedná se objekt který detekuje pozice jiných objektů uvnitř předem zvolené oblasti. Já jsem využil signálů, který Area2D vyšle pokud, do ní nějaký objekt vstoupí nebo ji opustí. Jsem tak schopen poslat zbraním nepřítele pozici hráče, popřípadě ji nastavit na null, pokud se hráč nachází mimo Area2D.

5 Závěr

Projekt se mi úspěšně podařilo uvést do funkčního stavu, který je ukázkou a základem plno hodnotné hry. Bohužel kvůli velmi rozsáhlému zadání jsem nestihl do hry implementovat všechny vytyčené cíle. Hře chybí obsah , jako je větší rozmanitost nepřátel a zbraní. Dále jsem nestihl vytvořit větší počet levelů a do hry zavést obchodu se zbraněmi. Největším problémem pro mě bylo samotné zorientování se v novém vývojovém prostředí a naučení se pracovat s pro mě novými technologiemi.

Při zlepšování vizuálního zážitku ze hry, jako jsou efekty a paralax pozadí (iluze rozdílného pohybu těles, podle jejich vzdálenosti od pozorovatele), jsem využil tyto YouTube tutoriály.[6, 7] Kromě již uvedených zdrojů jsem dále čerpal z rozsáhlé dokumentace Godot enginu, ta mi velmi pomáhala s pochopením jednotlivých prvků, které Godot nabízí, a samotného fungování Godotu.[8]

Použité zdroje

- [1] PCMag. *Godot Review*. 29. led. 2021. URL:
<https://www.pcmag.com/reviews/godot>.
- [2] Godot Engine. *Features - Godot Engine*. URL:
<https://godotengine.org/features/>.
- [3] Agnis NeZvers. *Godot bullet spread / reply to Davies Nzan*. 29. čvc. 2020. URL:
<https://www.youtube.com/watch?v=Gglx2iRDfMk>.
- [4] GDQuest. *Make Your First 2D Game with Godot: Player and Enemy (beginner tutorial part 1)*. Říj. 2019. URL:
<https://www.youtube.com/watch?v=Mc13Z2gboEk>.
- [5] DevWorm. *How to CREATE SMOOTH DASHING for a Platformer in Godot*. 24. čvn. 2022. URL: https://www.youtube.com/watch?v=zLdI0TnD_0w.
- [6] PlayWithFurcifer. *Particle Systems From BEGINNER To PRO / Godot Engine*. 16. lis. 2021. URL:
https://www.youtube.com/watch?v=F1Fyj3Lh_Pc.
- [7] bitbrain. *Parallax Scrolling in under 3 minutes! Godot 3.2 Tutorial*. Břez. 2021. URL: <https://www.youtube.com/watch?v=f8z4x6R7OSM>.
- [8] Godot team. *Godot Docs – 3.4 branch*. URL:
<https://docs.godotengine.org/en/3.4/>.

Seznam obrázků

1	Logo Godotu[1]	8
2	Hlavní menu	9
3	Začátek levelu	10
4	Dash	11
5	Nepřítel střílející po hráči	11
6	Střelba ze zbraně	12
7	Hráč umírající v bouři	12
8	Death screen	13
9	Cíl	13