



2D RPG HRA

Dokumentace k ročníkové práci

Autor: Patrik Poklop

Třída: 3ITB

Vedoucí práce: Mgr. Michal Beneš

2021/2022

Prohlášení

Prohlašuji, že jsem ročníkovou práci na téma „2D RPG HRA“ vypracoval samostatně a s použitím uvedené literatury a pramenů.

V (název obce, kde podepisuji) dne

.....

Poděkování

Chtěl bych poděkovat Mgr. Michalovi Benešovi za vedení mé ročníkové práce, cenné rady a odborný dohled. Děkuji také Mgr. Vratislavu Medřickému za cenné rady.

Anotace

Jedná se o 2D RPG HRA jménem Páťasy vytvořené podle mé fantasmie. Dokumentace se skládá z rešerše, kde jsou obrázky, podle kterých jsem se inspiroval, z technologií, se kterými jsem pracoval. V praktické části mám produktivizaci, do níž jsem vybral věci z práce, na které jsem nejvíce pyšný a uvedl jsem zde popis, jak by se měla hra hrát. Na závěr tu jsou nějaké obrázky ze hry.

Klíčová slova

gameplay, krajina, survival, útok, život, boj, příroda, lesy, jezera, nepřátelští mobové

Obsah

Úvod	6
1 Rešerše	7
2 Technologie	9
3 Praktická část.....	10
3.1 Produktizace	10
3.2 Popis pro uživatele	16
Závěr	17
Použitá literatura	18
Seznam obrázků	19
Obsah média.....	21

Úvod

Rozhodl jsem se ve své práci udělat 2D RPG hru a budu se inspirovat Final Fantasy I. a WOW. Vybral jsem si to, protože se od mala zajímám o hry a jejich grafiku. Hru udělám ve 2D platformě ve Windows. Jedná se o single-playerovou hru, již jsem pojmenoval Pátasy. Hra je vytvořená na základě mé fantasie.

Jako engine jsem si vybral Unity Engine, který podporuje c# a přišel mi ideální. Obrázky jsem dělal a upravoval v Pixlr editoru.

Na začátku je herní menu. Hráč si má možnost pojmenovat postavu. Hra je doprovázená hudbou. Je vytvořena mapa, na které se vyskytují lesy, rostliny, cesty, kopce, zámek, jeskyně, jezera a hlavně příroda, ve které se bude pohybovat hráč a narážet na různé příšery. Každá příšera je jinak silná. Úkolem hráče je zlikvidovat nepřátele. Příšery jsou rozmístěny po celé mapě náhodně. Ve hře jsou i efekty.

Postava má svou animaci. Nahoře vpravo je minimapa, která ukazuje polohu. Dolní bar, kde jsou výběry pro útok, staty postav a příšer, se kterými se bude bojovat, jako je množství hp nebo many. Při zaútočení se ukáže typ spellu. Hráč útočí mečem. Ve hře se sbírá score za zabíjení nepřátel. Hra se dá ukládat a nahrávat. Hráč si bude moc doplňovat život lektvary a vylepšovat si svoje staty za nasbírané body za zabíjení nepřátel u obchodníka.

1 Rešerše



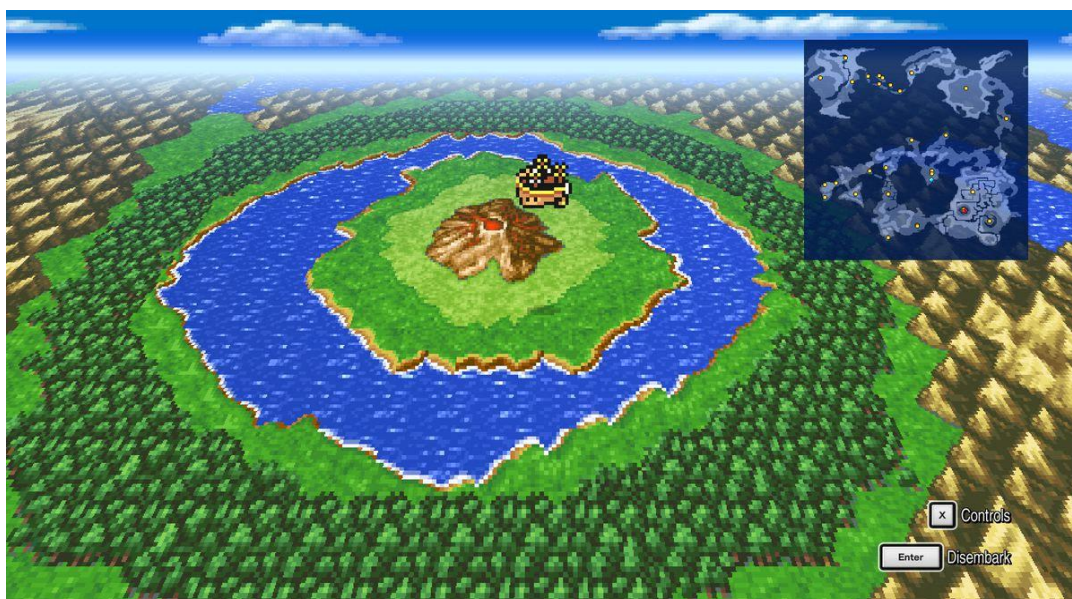
Fotka souboje ze hry FF I. - Ukázka toho, jak se bojuje s nepřátelskými postavami. HP staty na postavách. Možnost přetočit souboj. Útoky probíhají střídavě, jednou zaútočí hráč a potom nepřítel.



Logo hry FF I. – Zde je vidět název hry a pod názvem je obraz z FF1.



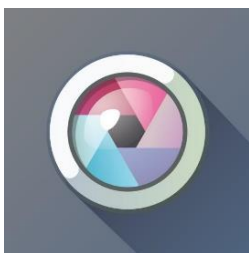
Fotka ze hry FF I. – Fotka gameplaye ze hry Final Fantasy I. Minimapa na které se ukazuje, kde se hráč zrovna nachází. Zámek, do kterého se dá jít. V hře jsou tlačítka, například pro nápovědu nebo pro hraní.



Fotka velkého kusu mapy s ostrovem, který má ve středu sopku. - Na ostrovem je loď, která přes něj prolétá a v ní cestuje hráč.

2 Technologie

Pixlr



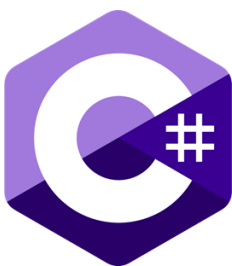
Pixlr je sada obrazových nástrojů založených na principu cloudu, vytvořená ve Švédsku, zahrnující řadu editorů fotografií. Umožňuje jak jednoduché, tak i pokročilé úpravy obrázků.

Unity



Unity je multiplatformní herní engine vyvinutý společností Unity Technologies. Unity poskytuje možnosti vývoje pro 2D i 3D hry libovolného žánru a zaměření. Kromě grafického prostředí pro tvorbu, podporuje také tvorbu skriptů v jazyce C#. Dnes funguje na více než 15 platformách. Jeho hlavním konkurentem (především na trhu her pro PC a konzole) je Unreal Engine.

CSharp



C# je vysokoúrovňový objektově orientovaný programovací jazyk vyvinutý firmou Microsoft zároveň s platformou .NET Framework. Používám ho pro programování v unity.

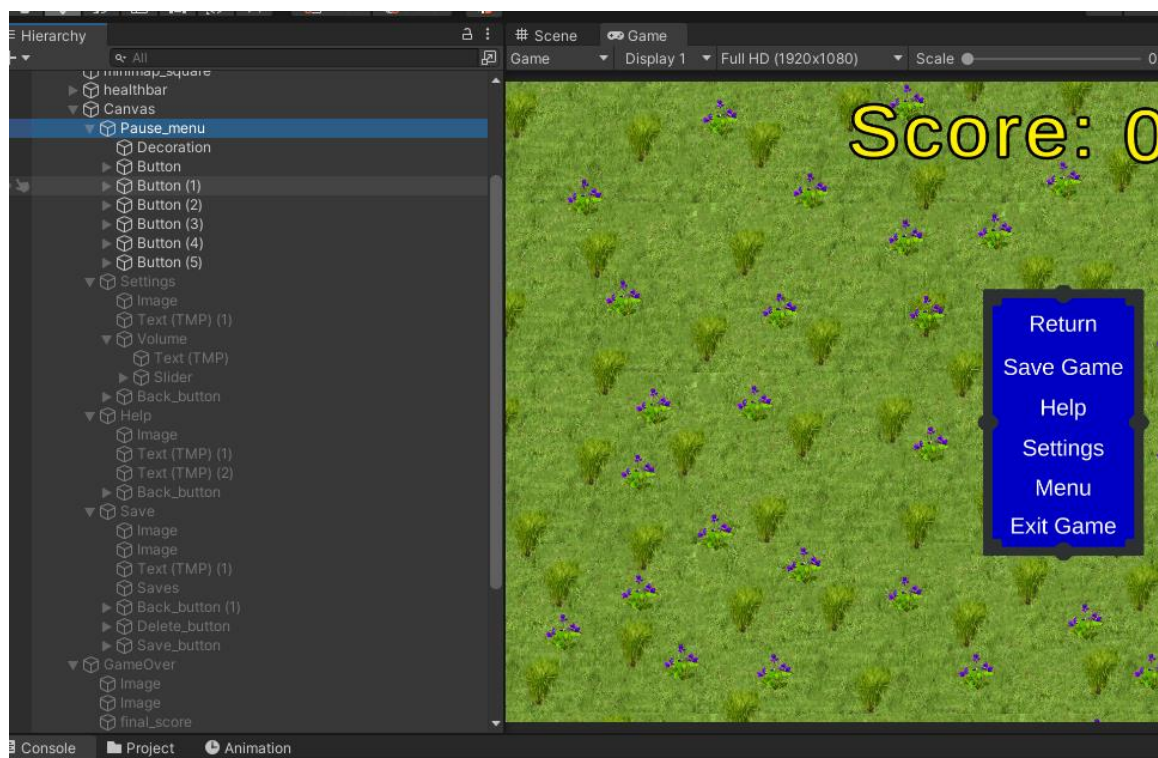
Visual Studio Code



Visual Studio Code je editor zdrojového kódu vyvíjený společností Microsoft pro operační systémy Windows, Linux a macOS.

3 Praktická část

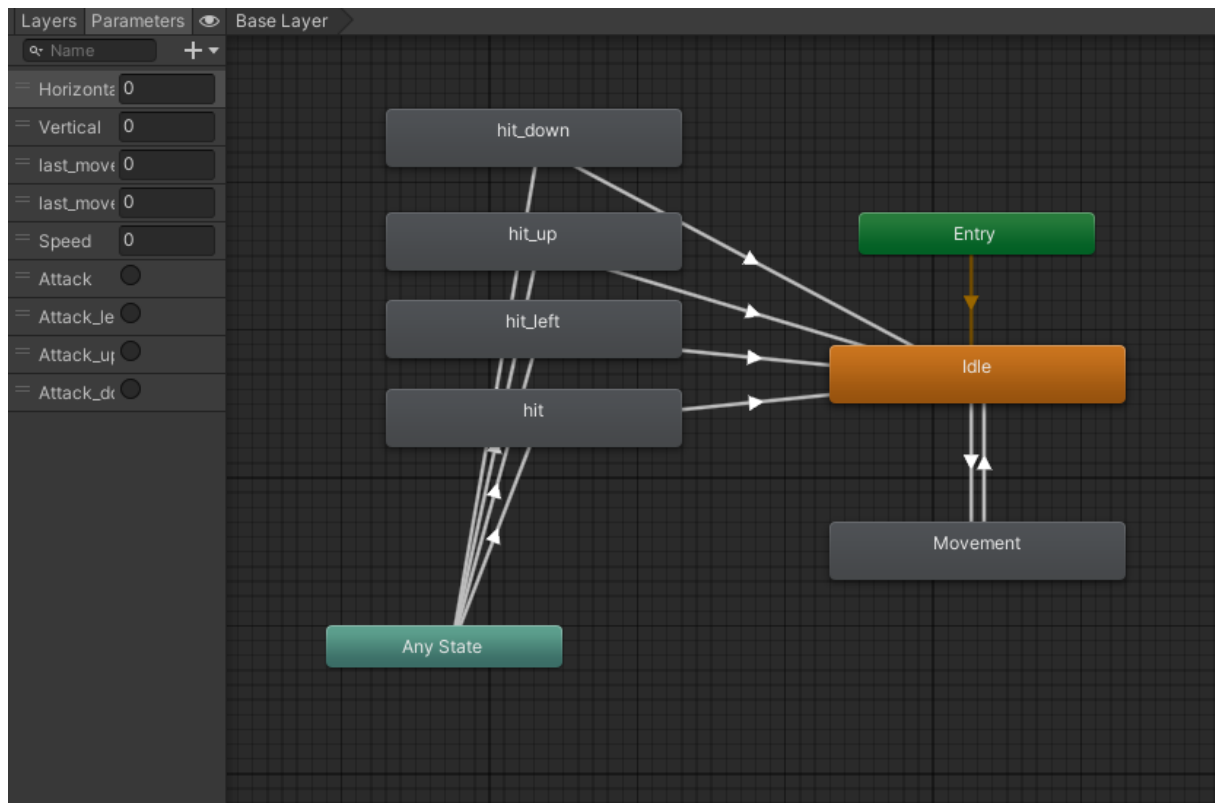
3.1 Produktizace



Vytváření nabídky – Menu nabídku ve hře jsem vytvořil obrázky. Dělal jsem spoustu tlačítek, texty a vyladoval jsem to možnostmi.



Animace boje – Zde jsem vytvořil v Pixlr editoru animace boje hráče tím, že jsem postavě dal meč a udělal jsem obrázky podle toho, jak útočí.



Animace – Zde jsem v animátoru v Unity engineu vytvářel způsob animace hráče ze 12 různých animací. Idle slouží k tomu, když se hráč zastaví a má 4 animace podle toho, na jakou stranu se zrovna hýbe, a to i movement, když se hráč hýbe. Čtyři attack trigger se spouštějí podle toho, na jaké straně zrovna hráč je, hráč také při útoku stojí. Speed float slouží k nastavení způsobu a rychlosti přechodu. Horizontal a Vertical floaty jsou u movementu. Last move floaty jsou u Idle, když hráč stojí.

```

1  using System.Collections;
2  using System.Collections.Generic;
3  using UnityEngine;
4  using UnityEngine.AI;
5  using TMPPro;
6  using UnityEngine.SceneManagement;
7  using UnityEngine.UI;
8  using UnityEngine.EventSystems;
9
10 public class PlayerMovement : MonoBehaviour
11 {
12     public float moveSpeed = 5f;
13     public Rigidbody2D rb;
14     public Animator animator;
15     private Vector2 movement;
16     public SpriteRenderer sprite;
17
18     private int player_damage;
19     public TextMeshProUGUI playerName;
20     private int health = 100;
21     private int current_health;
22     public TextMeshProUGUI healthText;
23
24     private int Score = 0;
25     public TextMeshProUGUI scoreText;
26     public TextMeshProUGUI finalScoreText;
27     public TextMeshProUGUI damageAmountText;
28     public Animator damageTextAnimator;
29
30     public TextMeshProUGUI pointText;
31     private int points = 0;
32
33     public health_bar HealthBar;
34     public GameObject redplayer;
35     public GameObject GameOver;
36     public GameObject Damage_count;
37     public GameObject pause_menu;
38
39     public GameObject Audio_manager;
40     public AudioSource walk1;
41     public AudioSource walk2;
42     private AudioSource player_audio;
43     public LayerMask enemyLayers;
44     public float attackRange = 0.5f;
45     public Transform attack_point;
46
47     public LayerMask NPClayer;
48     public float clickRange;
49     public Transform clickPoint;
50     public GameObject shopFable;
51
52     private GameObject hit;
53     private EnemyGenerator enemies;
54
55     private float latest_attack;
56     private readonly float time_between_attack = 2f;
57     private bool attack;
58
59     void Start()
60     {
61         current_health = health;
62         //HealthBar = GameObject.Find("health bar object").GetComponent<health_bar>();
63         player_audio = Audio_manager.GetComponent<AudioSource>();
64         HealthBar.SetMaxHealth(health);
65         scoreText.text = Score.ToString();
66         healthText.text = health.ToString() + " hp";
67         attack = true;
68         enemies = GameObject.Find("Enemies").GetComponent<EnemyGenerator>();
69     }
70
71     void Update()
72     {
73         movement.x = Input.GetAxisRaw("Horizontal");
74         movement.y = Input.GetAxisRaw("Vertical");
75
76         animator.SetFloat("Horizontal", movement.x);
77         animator.SetFloat("Vertical", movement.y);
78         animator.SetFloat("Speed", movement.sqrMagnitude);
79     }

```

```

80 if(Input.GetAxisRaw("Horizontal") == 1 || Input.GetAxisRaw("Horizontal") == -1 || Input.GetAxisRaw("Vertical") == 1 || Input.GetAxisRaw("Vertical") == -1){
81     animator.SetFloat("last_move_h", Input.GetAxisRaw("Horizontal"));
82     animator.SetFloat("last_move_v", Input.GetAxisRaw("Vertical"));
83     /*if(time.time - latest_walk >= time_between_walk){
84         latest_walk = Time.time;
85         StartCoroutine(walk());
86     }*/
87 }
88
89 if (Input.GetKeyDown(KeyCode.Mouse0) && Time.time - latest_attack >= time_between_attack || Input.GetKeyDown(KeyCode.Space) && Time.time - latest_attack >= time_between_attack){
90     player_damage = Random.Range(30, 50);
91     latest_attack = Time.time;
92     if (EventSystem.current.IsPointerOverGameObject() || (attack == false)){
93         Debug.Log("Clicked on the UI");
94     }
95     else{
96         if(movement.x == 1 || animator.GetFloat("last_move_h") == 1){
97             animator.SetTrigger("Attack");
98         }
99         else if(movement.x == -1 || animator.GetFloat("last_move_h") == -1){
100             animator.SetTrigger("Attack_left");
101         }
102         else if(movement.y == -1 || animator.GetFloat("last_move_v") == -1){
103             animator.SetTrigger("Attack_up");
104         }
105         else if(movement.y == 1 || animator.GetFloat("last_move_v") == 1){
106             animator.SetTrigger("Attack_down");
107         }
108     }
109     player_audio.Play();
110     Collider2D[] hitEnemies = Physics2D.OverlapCircleAll(attack_point.position, attackRange, enemyLayers);
111
112     foreach(Collider2D enemy in hitEnemies)
113     {
114         Debug.Log("We hit " + enemy.name);
115         hit = GameObject.Find(enemy.name);
116         hit.GetComponent<Hit>().TakeDamage(player_damage);
117         int mob_health = hit.GetComponent<Hit>().GetHealth();
118         if(mob_health <= 0){
119             StartCoroutine(destroy());
120         }
121     }
122 }
123
124 Collider2D[] clickNPC = Physics2D.OverlapCircleAll(clickPoint.position, clickRange, NPCLayer);
125 foreach(Collider2D npc in clickNPC)
126 {
127     Debug.Log("NPC detected");
128     if(Vector2.Distance(transform.position, GameObject.Find(npc.name).transform.position) < 40 && (Input.GetKeyDown(KeyCode.Mouse1))){
129         shopTable.SetActive(true);
130     }
131     else if(Vector2.Distance(transform.position, GameObject.Find(npc.name).transform.position) >= 40){
132         Debug.Log("not");
133         shopTable.SetActive(false);
134     }
135 }
136
137 void FixedUpdate()
138 {
139     if (current_health <= 0){
140         StartCoroutine(dead());
141     }
142     else if(animator.GetCurrentAnimatorStateInfo(0).IsName("hit") || animator.GetCurrentAnimatorStateInfo(0).IsName("hit_down")
143     || animator.GetCurrentAnimatorStateInfo(0).IsName("hit_up") || animator.GetCurrentAnimatorStateInfo(0).IsName("hit_left")){
144         StartCoroutine(if_hit());
145     }
146     else{
147         if(Input.GetAxisRaw("Horizontal") == 1 && Input.GetAxisRaw("Vertical") == 1 || Input.GetAxisRaw("Horizontal") == 1 && Input.GetAxisRaw("Vertical") == -1
148         || Input.GetAxisRaw("Horizontal") == -1 && Input.GetAxisRaw("Vertical") == 1 || Input.GetAxisRaw("Horizontal") == -1 && Input.GetAxisRaw("Vertical") == -1){
149             movement /= (float)1.3;
150         }
151         rb.MovePosition(rb.position + movement * moveSpeed * Time.fixedDeltaTime);
152     }
153 }
154
155 IEnumerator walk(){
156     walk1.volume /= 5;
157     walk1.Play();
158     yield return new WaitForSeconds((float)0.2);
159     walk1.Stop();
160     walk2.volume /= 5;
161     walk2.Play();
162     yield return new WaitForSeconds((float)0.1);
163     walk2.Stop();
164     walk1.volume *= 5;
165     walk2.volume *= 5;
166 }
167
168 IEnumerator if_hit(){
169     yield return new WaitForSeconds(1);
170     movement.x = 0;
171     movement.y = 0;

```

```

173 IEnumerator dead(){
174     yield return new WaitForSeconds(1);
175     movement.x=0;
176     movement.y=0;
177     yield return new WaitForSeconds(6);
178 }
179 IEnumerator destroy()
180 {
181     yield return new WaitForSeconds(2);
182     Destroy(hit);
183 }
184
185 IEnumerator damage_count()
186 {
187     Damage_count.SetActive(true);
188     damageTextAnimator.Play("Fade");
189     yield return new WaitForSeconds((float)1.8);
190     Damage_count.SetActive(false);
191 }
192
193 public void TakeDamage(int damage)
194 {
195     current_health -= damage;
196     StartCoroutine(Red());
197
198     if (current_health <= 0){
199         StartCoroutine(WaitTime1());
200         StartCoroutine(WaitTime2());
201         StartCoroutine(WaitTime3());
202         current_health = 0;
203         attack = false;
204         Debug.Log ("Dead");
205     }
206     damageAmountText.text = "-" + damage;
207     healthText.text = current_health.ToString () + " hp";
208     StartCoroutine(damage_count());
209     HealthBar.SetHealth(current_health);
210 }
211
212 public void Score_up(int health)
213 {
214     Score += health;
215     scoreText.text = Score.ToString();
216     StartCoroutine(Score_color_blink());
217 }
218 public void Point_up(){
219     points += 1;
220     pointText.text = points.ToString();
221 }
222 public void Points_down(int amount)
223 {
224     points -= amount;
225     pointText.text = points.ToString();
226 }
227 public int Return_points()
228 {
229     return points;
230 }
231 IEnumerator Red()
232 {
233     sprite.color = new Color (255, 0, 0, 255);
234     yield return new WaitForSeconds((float)0.4);
235     sprite.color = new Color (255, 255, 255, 255);
236 }
237 IEnumerator Score_color_blink()
238 {
239     for(int i = 0; i <= 1; i++)
240     {
241         scoreText.color = Color.green;
242         yield return new WaitForSeconds(1);
243         scoreText.color = new Color(255, 240, 0, 255);
244     }
245 }
246 IEnumerator WaitTime1()
247 {
248     yield return new WaitForSeconds(1);
249     sprite.color = new Color (255, 0, 0, 255);
250     pause_menu.GetComponent<PauseMenu>().EscDisable();
251 }
252
253 IEnumerator WaitTime2()
254 {
255     yield return new WaitForSeconds(3);
256     GameOver.SetActive(true);
257     finalScoreText.text = "Your score: " + Score.ToString();
258 }

```

```

259 IEnumerator WaitTime3()
260 {
261     yield return new WaitForSeconds(8);
262     SceneManager.LoadScene(SceneManager.GetActiveScene().buildIndex-1);
263     GameOver.SetActive(false);
264     Score = 0;
265     sprite.color = new Color (255, 255, 255, 255);
266     pause_menu.GetComponent<PauseMenu>().EscEnable();
267     attack = true;
268     Debug.Log("Scene loaded");
269 }
270 public void GetName(string name){
271     playerName.text = name;
272 }
273 public int GetHealth(){
274     return current_health;
275 }
276 public int GetScore(){
277     return Score;
278 }
279 public string ReturnName(){
280     return playerName.text;
281 }
282
283 public void SavePlayer(string name)
284 {
285     GameObject player = GameObject.FindGameObjectWithTag("Player");
286
287     enemies.Detect_actual_enemy_names_and_positions();
288
289     PlayerMovement player_stats = player.GetComponent<PlayerMovement>();
290     SaveSystem.SavePlayer(player, player_stats, enemies, name);
291 }
292 public void LoadPlayer(string name)
293 {
294     Player_data data = SaveSystem.LoadPlayer(name);
295
296     playerName.text = data.name;
297     Score = data.score;
298     scoreText.text = Score.ToString();
299     current_health = data.health;
300     healthText.text = current_health.ToString () + " hp";
301     HealthBar.SetHealth(current_health);
302     points = data.points;
303     pointText.text = points.ToString();
304     Vector3 position;
305     position.x = data.position[0];
306     position.y = data.position[1];
307     position.z = data.position[2];
308     transform.position = position;
309
310     Debug.Log("Hra nactena");
311     string jmena = "";
312     foreach(var enemy_name in data.enemy_names){
313         jmena += enemy_name + ", ";
314     }
315     Debug.Log("Enemies Generated "+ jmena + "Length: " +data.enemy_names.Length);
316
317     Vector3[] enemy_string_positions = new Vector3[data.active_enemies_locations.Length];
318     for(int i=0;i<data.active_enemies_locations.Length;i++){
319         if (data.active_enemies_locations[i].StartsWith "(" && data.active_enemies_locations[i].EndsWith (")")) {
320             data.active_enemies_locations[i] = data.active_enemies_locations[i].Substring(1, data.active_enemies_locations[i].Length-2);
321         }
322         string[] s = data.active_enemies_locations[i].Split(',');
323         float[] VectorPosition = new float[s.Length];
324         Vector3 result = new Vector3();
325         for (int j = 0; j < s.Length;j++){
326             VectorPosition[j] = float.Parse(s[j], System.Globalization.CultureInfo.InvariantCulture);
327             result.x = VectorPosition[0];
328             result.y = VectorPosition[1];
329             result.z = 1;
330         }
331         enemy_string_positions[i] = result;
332     }
333
334     string pozice = "";
335     foreach(var enemy_vector in enemy_string_positions){
336         pozice += enemy_vector + ", ";
337     }
338     Debug.Log("Positions Generated "+ pozice + "Length: " +data.active_enemies_locations);
339     enemies.Delete_all_enemies();
340     enemies.Generate_new_enemies_by_count(data.enemy_names,enemy_string_positions);
341
342 }
343
344 void OnDrawGizmosSelected()
345 {
346     if(attack_point == null){
347         return;
348     }
349     Gizmos.DrawWireSphere(attack_point.position,attackRange);
350 }

```

Hráč – Tento kód s celkem asi 350 řádky obsahuje pohyb hráče, animaci, život, staty, zvukové efekty, souboj hráče a ukládání v jednom skriptu. Hodně práce jsem si dal s animacemi pohybu a útoku, jako je např. načasování útoku, aby vypadaly dobře. Do tohoto kódu vstupují textová pole pro život, pro souboj, pro množství damage, pro scóre, text ukazující množství ubraného damage s animací. Vstupují do něj také objekty, které se ukazují při smrti jako je GameOver a potom mě to pošle zpět do menu. Je tam detekce pohybu nepřítele, kdy script detekuje nepřítele a přes název stringu najde objekt a potom se připojí na něm k všude stejnému scriptu Hit a ubere danému nepřítelovi damage. Má animaci i zvuk útoku. Jsou zde funkce, které přičítají nebo vrací scóre. Nakonec má i udělané ukládání a načítání, kdy se dají načíst uložená data hráče a nepřátel a uložit aktuální data o hráči a nepřátelích. Přidal jsem funkce, přes které se dají přičítat a vracet body.

3.2 Popis pro uživatele

Ovládání pohybu je W,A,S,D. Útočí se klikem myši nebo mezerníkem. Hráč chodí po mapě a zabíjí moby a jeho úkolem je nasbírat co nejvíc scóre. Stisknutím escapu se zobrazí herní nabídka.

Závěr

Projekt byl pro mě celkem náročný. Nastavoval a designoval jsem menu, mnoho objektů, doplňků a také herní menu a nabídku. Hodně času jsem také věnoval obrázkům. Vyplnil jsem mapu velkým množstvím objektů jako jsou květiny, tráva, stromy, hrad, jezero, cesta, kopce. Tyto objekty jsem v editoru fotografií buď vytvořil, vystřihnul, nebo upravil. Udělal jsem ikonu hry. Přidal jsem několik textů, které se umí automaticky updatovat. Texty jsem dělal přes textmesh zadarmo. Vytvořil jsem HP bary nad hráčem a nepřáteli. Dále jsem jim udělal stat, který ukazuje jejich aktuální HP. Při útoku se nad hráčem nebo nepřítelem ukáže ubrané množství HP s animací. Udělal jsem jim slider, který je vyplněný podle aktuálního množství HP. Přidal jsem hudbu do menu a do hry, která se dá ovládat pomocí sliderů v nastavení, které ukazují i procenta. Udělal jsem přechod z menu do hry.

Při startu hry se dá hráč libovolně pojmenovat. Přidal jsem zvukové efekty, jako je chůze, seknutí nebo kliknutí. Nahoře je score text, při zabití se přičtou body a ukáže se to v textu. Přidal jsem více nepřátel, podle jejich druhu jsem jim v určitém úseku vygeneroval náhodné množství HP. Udělal jsem víc kolizí po mapě, dále jsem udělal pohyb a animace nepřátel ze stažených spritů z internetu, pohyb a animace hráče a animaci jeho útoků. Nepřátelé mají náhodný pohyb a také určitý čas, kdy změni svůj směr. Když se hráč přiblíží příliš blízko k nepříteli, tak na něj nepřítel zaútočí. Po celé mapě se generují přes script random nepřátelé na random pozicích. Boss umí udeřit do hráče bleskem, vytvoří line mezi hráčem a bossem a nad Bossem bar s bleskem. Udělal jsem vrstvy, které jsou přímo pro nepřátele, nebo pro hráče. Některé animace, jako je útok nebo pohyb Bosse, jsem si vytvářel sám. Udělal jsem souboj hráče s nepřítelem, hráč může udeřit jakéhokoli nepřítele na mapě. Dále jsem vytvořil načasování útoku, kdy hráč nepřítele i nepřítel hráče může udeřit až za určitý čas po útoku. Při úderu jsem udělal efekt, že se na půl sekundy začervení.

Vpravo nahoře jsem vytvořil minimapu, kde je hráč označen zeleným puntíkem a nepřátelé červeným puntíkem. Ve hře jsem také udělal ukládání hry, kdy stav hry, jako jsou informace o hráčovi nebo o počtu nepřátel a jejich pozicích, se dá uložit do souboru a zase potom načíst. Uložené hry se zobrazují přes script, který zobrazí a vytvoří seznam všech uložených her. Uložené hry se i dají mazat, a když zvolím nějaké uložení, tak se zabarví. Při smazání nebo uložení se i nahoře ukáže text, že se hra smazala nebo uložila. Ve hře mám udělaného obchodníka, u kterého se dají vylepšovat sklily nebo kupovat lektvary na život. Celkově jsem udělal přes 20 různých skriptů a každý z nich slouží k nějaké funkci nebo funkcím ve hře.

Mohu říci, že jsem se svým projektem celkově spokojený. V tomto projektu jsem se naučil programovat v CSharpu a dělat Unity Enginu. Do budoucna bych chtěl, aby vznikla i nějaká česká RPG hra. Za to, co jsem byl schopen vytvořit, chci poděkovat mým vyučujícím, jim za to vděčím, panu učiteli Jakubu Pokornému, Michalu Benešovi a Vratislavu Medřickému.

Použitá literatura

1.cs.wikipedia.org: *Pixlr (Editor Fotografií)* [online]. 2008 [cit. 2022-07-03]. Dostupné z: <https://cs.wikipedia.org/wiki/Pixlr>

2.cs.wikipedia.org: *Unity (Herní engine)* [online]. 2005 [cit. 2022-07-03]. Dostupné z: [https://cs.wikipedia.org/wiki/Unity_\(herní_engine\)](https://cs.wikipedia.org/wiki/Unity_(herní_engine))

3.cs.wikipedia.org: *C# (Programovací jazyk)* [online]. 2000 [cit. 2022-07-03]. Dostupné z: https://cs.wikipedia.org/wiki/C_Sharp

4.cs.wikipedia.org: *Visual Studio Code (Editor zdrojového kódu)* [online]. 2015-4-28 [cit. 2022-07-03]. Dostupné z: https://cs.wikipedia.org/wiki/Visual_Studio_Code

Seznam obrázků



Ukázka ze hry



Ukládání hry



Game over



Souboj s bosem

Obsah média

Médium obsahuje:

- Projekt Páťasy
- Páťasy build spustitelný ve windows.
- Dokumentaci k Páťasy
- Prezentaci k Páťasy