

Piszemy RPSGo-Platformówkę (8) – Diabeł tkwi w szczegółach (sheadovas/poradniki/proj_platf_rpg/8-diabel-tkwi-w-szczegolach/)

Maj 28, 2017 / proj_platf_rpg (sheadovas/category/poradniki/proj_platf_rpg/)

Wprowadzamy kilka szczegółów do gry...

Hej, witam Was w pierwszym z dwóch wpisów w których skupimy się na ulepszaniu dotychczas napisanych mechanik do gry platformowej.

Omawiane zmiany dotyczą wersji do commita [f88f9cd (https://github.com/sheadovas/proj_platf_rpg/commit/f88f9cd10bdf216503e0b36838306d4b74a04610)] (najlepiej zmiany widać [tutaj (https://github.com/sheadovas/proj_platf_rpg/compare/0abaf23...f88f9cd)]). Jednocześnie pominę omawianie kodu, który jest na/po-prawą działania niektórych featerów (ich objaśnienie znajduje się w szczegółowym opisie konkretnych commitów).

Diabeł tkwi w szczegółach

Jak powszechnie wiadomo szczegóły nie pozostają bez znaczenia, także w kontekście informatyki / gier. Same systemy, które wymyślimy jakby nie były genialne, tak bez odpowiedniego dopracowania na pewno pozostaną niezauważone przez przeciętnego użytkownika.

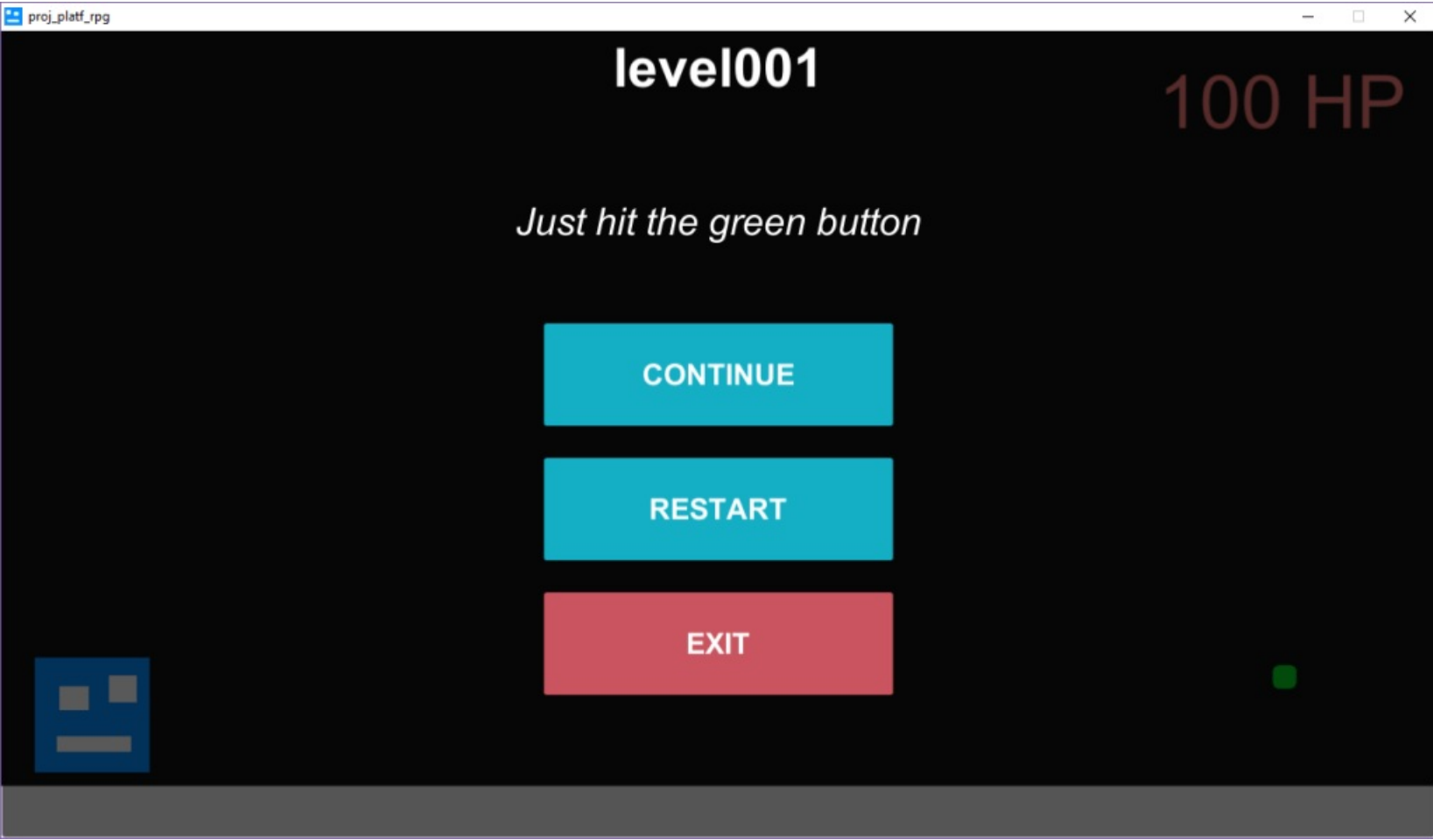
Musimy też pamiętać, że gra powinna być wygodna dla grającego, tzn. całość powinna być dość intuicyjna (np. sterowanie) lub łatwe w „złapaniu” pewnych intuicji.

Menu pauzy (końca poziomu)

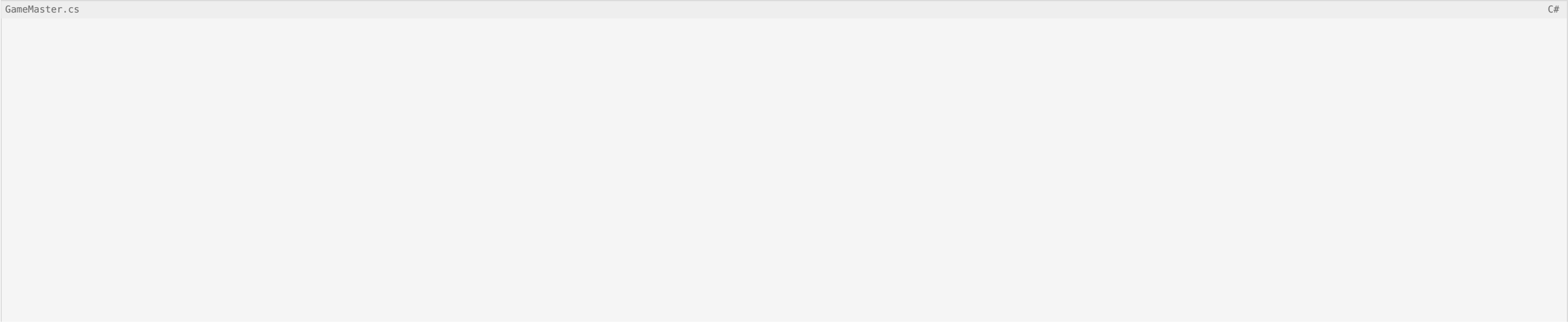
Gra powinna także posiadać wyraźnie przerywniki pomiędzy kolejnymi poziomami, misjami podczas których gracz może odetchnąć lub przerwać grę (fajnie opowiedzieli o tej mechanice ludzie z [Extra Credits (https://www.youtube.com/user/ExtraCreditz)]).

Z tego powodu stworzymy sobie wewnętrzne menu, z poziomu którego będziemy mogli:

- 1. Zapauzować grę / przejść do następnego poziomu jeżeli ukończyliśmy poziom.
- 2. Zobaczyć aktualny cel poziomu.
- 3. Zrestartować poziom.
- 4. Wyjść z gry



(<https://i2.wp.com/szymonsiarkiewicz.pl/wp-content/uploads/2017/05/001-startmenu.png>)
Menu na uruchomienie poziomu



```

99  /* UI */
100 public void ShowMenu()
101 {
102     Time.timeScale = 0;
103     if(m_isGameOver && SceneManager.GetActiveScene().buildIndex + 1 == SceneManager.sceneCount)
104     {
105         m_continue.interactable = false;
106     }
107
108     m_gameMenuParent.SetActive(true);
109 }
110
111 public void HideMenu()
112 {
113     Time.timeScale = 1.0f;
114     m_gameMenuParent.SetActive(false);
115 }
116
117 public void OnClickedContinue()
118 {
119     if(!m_isGameOver)
120     {
121         HideMenu();
122     }
123     else
124     {
125         int current = SceneManager.GetActiveScene().buildIndex;
126         SceneManager.LoadScene(current + 1);
127     }
128 }
129
130 public void OnClickedRestart()
131 {

```

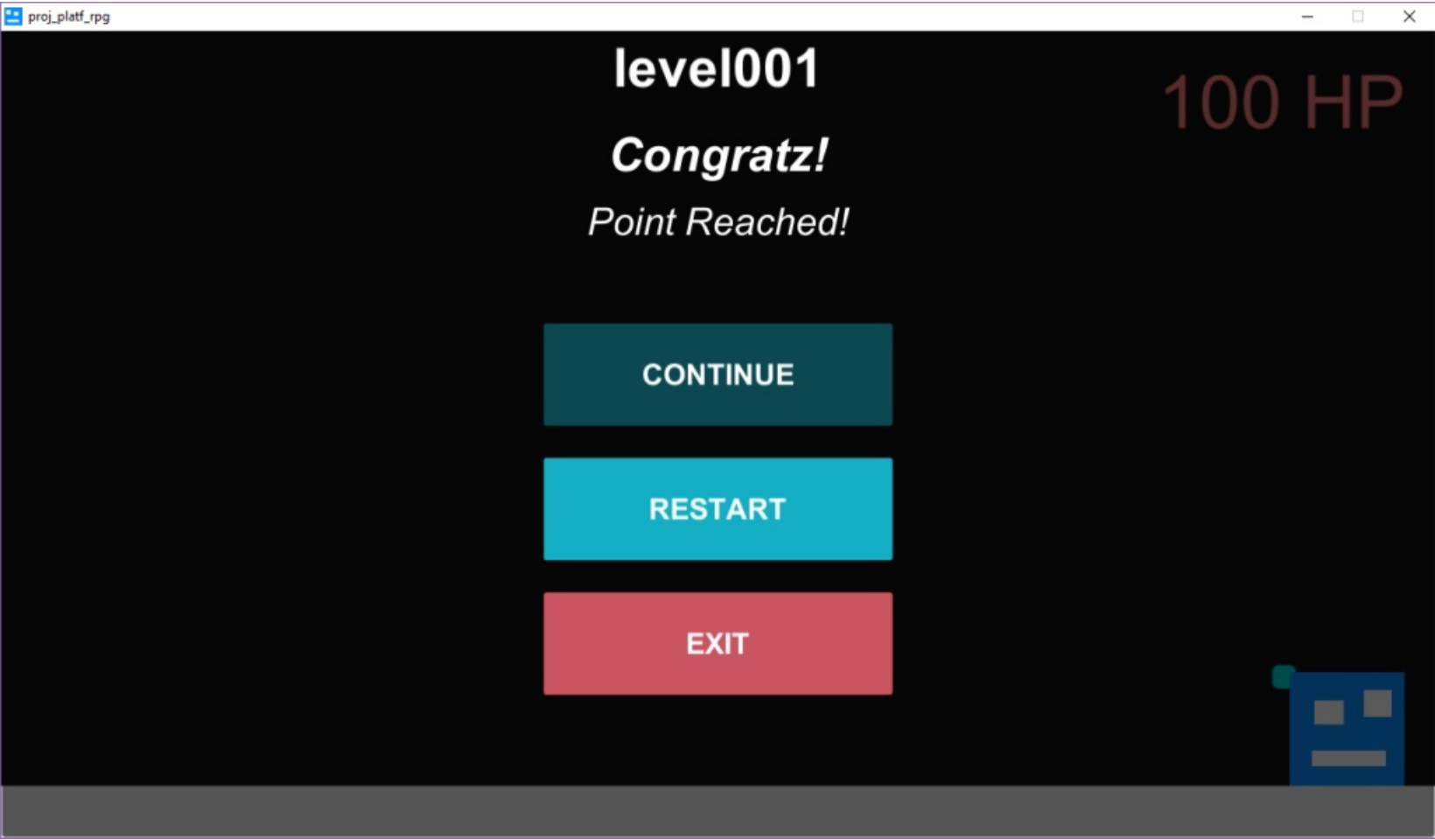
W tym celu stworzyłem nakładkę która w momencie aktywacji pauzuje całą rozgrywkę oraz zależnie od stanu rozgrywki (pauza / koniec gry / brak kolejnych poziomów) jest w stanie odpowiednio zareagować.

```

GameManager.cs
59 public void NotifySuccess(GameOverCondition cond)
60 {
61     NotifyGameOver(true, cond.GetProgressInfo());
62 }
63
64 public void NotifyFailure(GameOverCondition cond)
65 {
66     NotifyGameOver(false, cond.GetProgressInfo());
67 }
68
69 public void NotifyGameOver(bool success, string message)
70 {
71     if (!success)
72         m_continue.interactable = false; // deaktywacja buttona
73
74     string title = (success ? "Congratz!" : "EPIC FAIL!");
75     m_isGameOver = true;
76
77     m_gameStatus.text = title;
78     m_missionDetails.text = message;
79
80     Invoke("ShowMenu", 0.5f);
81 }

```

Dodatkowo finalnej zmianie uległy metody z cyklu *Notify* które teraz potrafią w sposób nie-konsolowy zareagować na wygraną / przegraną gracza.



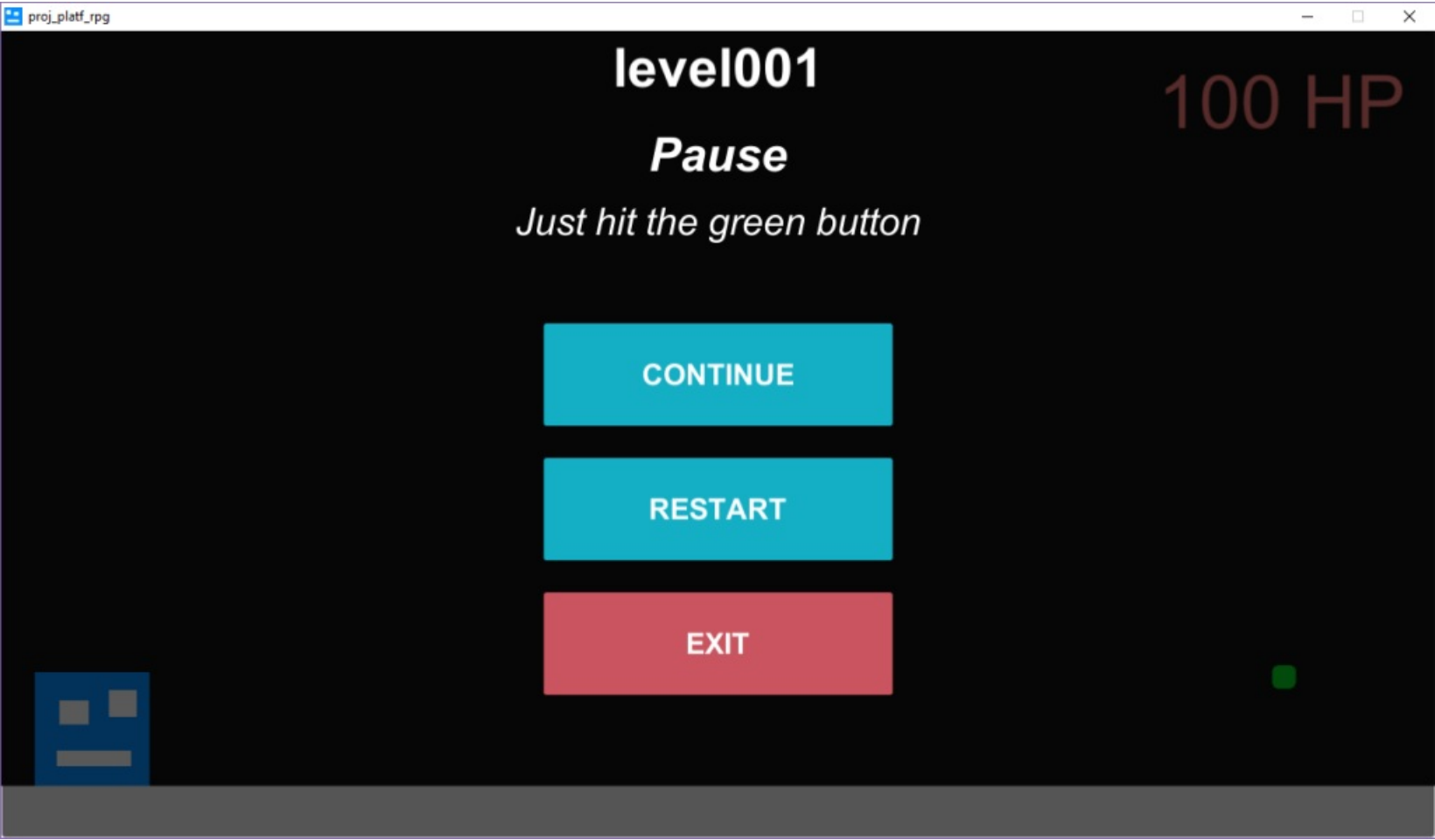
(<https://i0.wp.com/szymonsiarkiewicz.pl/wp-content/uploads/2017/05/002-finish.png>)
Menu końca gry

Nowe poziomy

Skoro doszedł nam feature jakim jest płynne przejście pomiędzy poziomami, to warto go wykorzystać tworząc nowe poziomy. Od teraz chcemy aby każdy poziom cechował się następującą sekwencją:

1. Wyświetlenie menu poziomu (cel gry)
2. Przejście do właściwej gry
3. (opcjonalne) wyświetlanie menu pauzy
4. Menu zakończenia gry

Dodane przeze mnie poziomy (dostępne do obejrzenia w [demie (https://github.com/sheadovas/proj_platf_rpg/releases/tag/1.6)]) mają charakter samouczkowy i na pewno nie stanowią żadnego wyzwania. Ale jak być może wspominałem, przedstawiam tutaj głównie mechanizmy do tworzenia gier, a nie zajmuję się pisaniem pełnoprawnej gry ;)



(<https://i2.wp.com/szymonsiarkiewicz.pl/wp-content/uploads/2017/05/003-pause.png>)

Manu pauzy

Siłą rzeczy w tym paragrafie nie ma jak pokazać kodu bo całe poziomy powstały przy użyciu Unity Edytora, nowe mechanizmy które wraz z nimi doszły omówię w kolejnym podrozdziale.

Tutaj chciałbym jedynie pokrótce wspomnieć, że przy tworzeniu gier należy zwrócić uwagę na sposób projektowania poziomów: pierwsze powinny mieć charakter samouczkowy (mają uczyć nowych mechanik), a kolejne powinny te nowe mechaniki wykorzystywać i być coraz trudniejsze. Nie zaszkodzi też dodawać nowych mechanik wraz z przebiegiem rozgrywki tak aby utrzymać poziom nowych elementów i zaniepokojenie gracza.



(https://i2.wp.com/szymonsiarkiewicz.pl/wp-content/uploads/2017/05/RAYMAN_ORIGINS_2.jpg)
Rayman Origins jako przykład różnorodnej gry

Nowe elementy rozgrywki

Jak wspomniałem wraz z nowymi poziomami doszły nam nowe elementy rozgrywki.

Najprostszą jest śmierć od upadku w przepaść, która jest zwykłym triggerem pod główną zawartością każdego poziomu gry:

FalloffTrigger.csC#

```
1 using UnityEngine;
2
3 public class FalloffTrigger : MonoBehaviour
4 {
5     private void OnTriggerEnter2D(Collider2D collision)
6     {
7         del(collision.gameObject);
8     }
9
10    private void OnCollisionEnter2D(Collision2D collision)
11    {
12        del(collision.gameObject);
13    }
14
15    private void del(GameObject obj)
16    {
17        switch (obj.tag)
18        {
19            case "Player":
20                GameManager.gm.NotifyGameOver(false, "Gravity is brutal :(");
21                break;
22
23            default:
24                if (obj.transform.parent == null)
25                {
26                    Debug.Log("Falloff: Destroying: " + obj.name);
27                    Destroy(obj);
28                }
29                break;
30        }
31    }
32 }
```

Innym elementem jest możliwość wygrania (konkretnego) poziomu w przypadku gdy wszyscy przeciwnicy zostaną zniszczeni. W tym celu musimy zliczać stworzonych przeciwników:

Enemy.cs

C#

```
1 public class Enemy : PlayableCharacter
2 {
3     public static int spawnedCount = 0;
4
5     protected override void Start()
6     {
7         base.Start();
8         spawnedCount++;
9     }
10
11    protected override void OnDeath()
12    {
13        base.OnDeath();
14        spawnedCount--;
15        GameManager.gm.NotifyEnemyDeath();
16    }
17
18    private void OnDestroy()
19    {
20        OnDeath();
21    }
22 }
```

Oraz stworzyć warunek końca gry:

ConditionNoEnemies.cs

C#

```
1 using System;
2
3 public class ConditionNoEnemies : GameOverCondition
4 {
5     private string m_text = "Enemies left: ";
6     private string m_status;
7
8     public override string GetProgressInfo()
9     {
10         if (Enemy.spawnedCount > 0)
11         {
12             m_status = m_text + Enemy.spawnedCount;
13         }
14         else
15         {
16             m_status = "All enemies are dead!";
17         }
18         return m_status;
19     }
20
21     protected override bool isFailure()
22     {
23         return false;
24     }
25
26     protected override bool isSuccess()
27     {
28         return (Enemy.spawnedCount <= 0);
29     }
30 }
```

Podsumowanie

To tyle jeżeli chodzi o ten wpis, zmian było sporo – głównie po stronie wizualnej/grywalnej, dlatego gorąco zachęcam do pobrania [dema (https://github.com/sheadovas/proj_platf_rpg/releases/tag/1.6)] i wypróbowania zmian samodzielnie.

Update 1: właśnie zauważyłem, że demo albo jest na wersji z mini bugiem, albo z dodaną tylko jedną sceną (a więc dostępny jest do ogrania tylko 1 poziom z 3). Poprawioną wersję demo wrzucę na dniach ;)

Update 2: wrzuciłem dobrą wersję, link jest ten sam (paczka z numerkiem 1.6.1).

Tradycyjnie zapraszam do komentowania, śledzenia bloga przez social media i do zajrzenia także do innych prowadzonych przeze mnie serii ;)

W kolejnej części zajmiemy się finalnymi szlifami związanymi z częścią z pisania gry platformowej ;)

Update 2.5: Z racji, że jednak nie będę zmieniał grafik na tym etapie (bo w sumie nie ma zbytniej potrzeby), a część platformowa przedstawia to co chciałem pokazać, to nie będzie kolejnej części. Nawiążę do tej decyzji przy kolejnej części.

Code ON!