

# Piszemy RPSGo-Platformówkę (7) – Interfejs graficzny (sheadovas/poradniki/proj\_platf\_rpg/7-interfejs-graficzny/)

Maj 21, 2017 / proj\_platf\_rpg (sheadovas/category/poradniki/proj\_platf\_rpg/)

Wprowadzamy interfejs do gry

Hej, dzisiaj (głównie) pogadamy sobie o interfejsie graficznym, czyli tzw. UI.

## Słowem wstępu

UI (ang. „user-interface”) w uboższej lub bogatszej postaci posiada każda gra, dzięki niemu użytkownik widzi najbardziej istotne informacje na głównym ekranie gry bez konieczności przeklikiwania się przez różne menu.



(<https://i0.wp.com/szymonsiarkiewicz.pl/wp-content/uploads/2017/05/witcher.jpg>)

Wiedźmin 3



Dobre UI powinno być czytelne już na pierwszy rzut oka, może być przedstawione w postaci tekstowej lub obrazkowej. Przy projektowaniu UI warto zastosować się do ogólnie panujących trendów, tak aby gracz intuicyjnie je rozumiał, a więc np. 2 poziome paski w rogu ekranu w kolorze czerwonym i niebieskim intuicyjnie są rozumiane jako poziom życia i many bohatera.

Gdybyśmy zamienili znaczenie kolorów (tj. niebieski to życie, czerwony to mana), to niepotrzebnie utrudnilibyśmy zrozumienie naszego interfejsu i wprowadzili niepotrzebne zamieszanie u potencjalnego gracza.

Jak wspomniałem już wcześniej, dobre UI to takie, które posiada minimalną ilość wyświetlanych elementów, bowiem nie chcemy dojść do sytuacji gdy ekran gry składa się niemal wyłącznie z wyświetlanych przycisków, wskaźników, które w reultacie sprawią, że gracz nie wiedział co się dzieje na ekranie oraz będzie zadawał pytanie „gdzie właściwie jest ta gra? czy kryje się pod tym gąszczem przycisków?”.



(<https://i0.wp.com/szymonsiarkiewicz.pl/wp-content/uploads/2017/05/crazyui.jpg>)

UI – gra platformowa



W tej części poradnika (tzn. dotyczącej gry platformowej) potraktujemy nieco po macoszemu (nie martwcie się, wrócimy do niego na poważniej przy części dotyczącej RPG). A wynika to z faktu, że gra platformowa nie potrzebuje za dużo.



(<https://i2.wp.com/szymonsiarkiewicz.pl/wp-content/uploads/2017/05/new-super-mario-bros-u-gameplay-2.jpg>)

Super Mario Bros

W obecnym stanie gry wystarczy, że dorobimy wyświetlanie stanu ilości życia, w kolejnej (lub jeszcze kolejnej) części (w sensie: odcinka) tego poradnika – gdy już będziemy jasno wyznaczony cel gry – dodamy jeszcze informacje o ilości zebranych kluczy, itp (a w części dotyczącej RPG dodamy okienka, ekwipunek, dialogi, itd.).

Omawiane zmiany dotyczą commita [e771e45 ([https://github.com/sheadovas/proj\\_platf\\_rpg/commit/e771e455363864f7bfd7a79ca5cd88e860470c13](https://github.com/sheadovas/proj_platf_rpg/commit/e771e455363864f7bfd7a79ca5cd88e860470c13))] i ograniczają się do stworzenia na ekranie obiektu typu *Text* (pochodzącego z Unity) i aktualizowanie jego wartości po otrzymaniu obrażeń przez gracza:

Player.cs

C#

```
12 public override void NotifyDamageTaken()
13 {
14     base.NotifyDamageTaken();
15     m_conditionDeath.CheckConditions();
16
17     GameManager.gm.playerHp.text = stats.hp.ToString() + " HP";
18 }
19
20 protected override void Start()
21 {
22     base.Start();
23
24     if(m_conditionDeath == null)
25     {
26         Debug.LogWarning("Death condition for player is not set!", this);
27     }
28
29     GameManager.gm.playerHp.text = stats.hp.ToString() + " HP";
30 }
```

# Zakończenie

Tak, to koniec jeżeli chodzi o tą lekcję. Jakie UI jest w grach każdy widzi (a jest bardzo proste, więc lekcja jest krótka). Co prawda mógłbym nieco więcej wspomnieć o projektowaniu interfejsu, na urządzenia mobilne, desktopy, ale myślę że lepiej zostawić sobie ten temat na część RPG, gdzie zaimplementujemy dużo więcej ciekawych elementów ;)

Tradycyjnie zapraszam do komentowania, śledzenia bloga przez social-media oraz do śledzenia tej (oraz innych) serii.

W kolejnej części postaramy się skleić stworzone do tej pory elementy w jedną, spójną całość.

*Code ON!*