

Spis treści

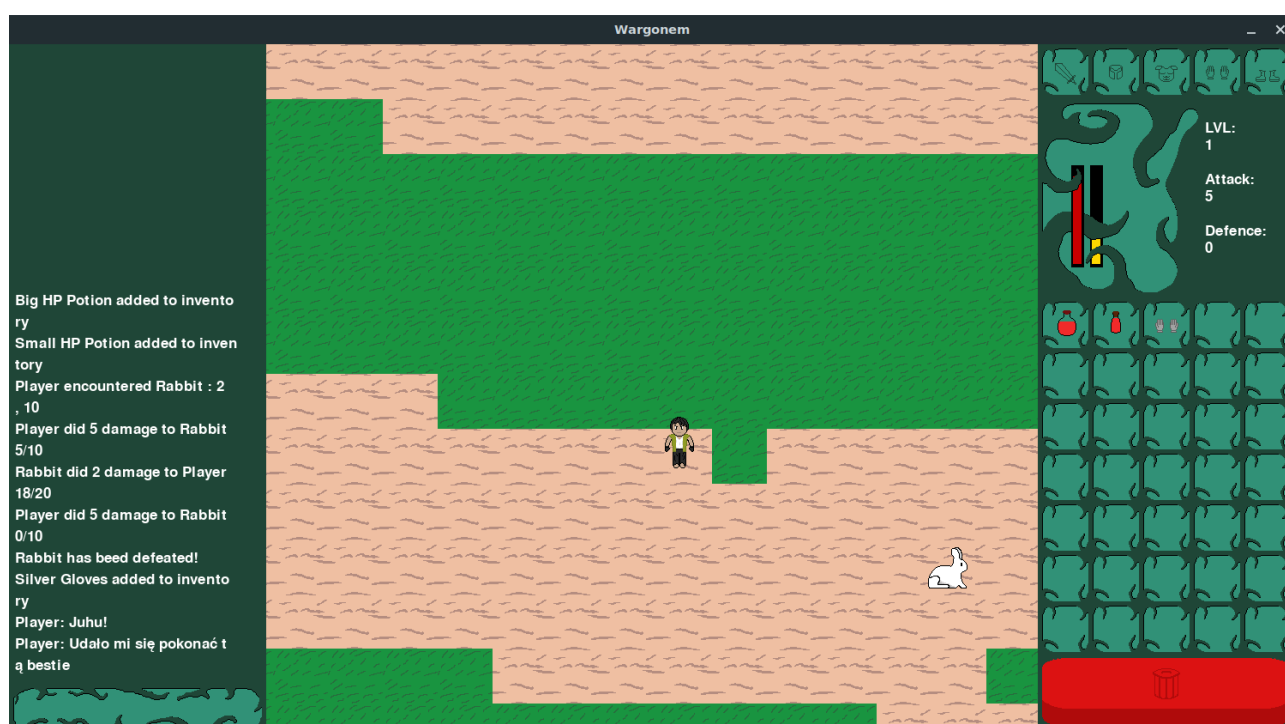
Opis projektu.....	2
Budowa programu.....	4
PyGame.....	4
Chat.....	4
Input.....	4
Pole Tekstowe.....	5
Okno z grą.....	6
Mapa.....	6
Bohater oraz Przeciwnik.....	6
Inwentarz oraz statystyki.....	7
Wyposażenie.....	7
Statystyki.....	8
Inwentarz.....	8
Śmietnik.....	9
Przedmioty.....	9
Pliki z dodatkowymi funkcjami.....	10
LoadFile.py.....	10
color_palette.json.....	10
items.json.....	10
tiles.json.....	10
DB.py.....	10
logs.py.....	10
Słów kilka od autora.....	11
Dodatkowe informacje:.....	11
Zakończenie.....	12

Wargonem

Opis projektu

„Wargonem” jest grą, bazowana graficznie na popularnej grze przeglądarkowej „Margonem”

Gra polega na poruszaniu się za pomocą przycisków W, A, S, oraz D, oraz znajdowaniu królika na mapie za, naciśnięciu na niego LPM w celu wykonania symulacji walki, która jest dokumentowana na polu tekstowym po lewej stronie od okna gry



Postać, którą gramy posiada wiele statystyk, między innymi, zdrowie (HP), czy też atak, które się zwiększają z każdym zwiększeniem się poziomu gracza

Wszystkie statystyki są przedstawione graficznie, bądź w formie liczby po prawej stronie ekranu.

Nasza postać ma możliwość otrzymania z królików przedmioty, które może założyć w celu zwiększenia statystyk oraz ułatwienia dalszych potyczek z królikami

Obrażenia zadane przez królika można wyleczyć za pomocą mikstur, przeciągając je na okno ze statystykami.

Po prawej stronie, na samym dole jest pole do wprowadzania tekstu, po naciśnięciu LPM w tym polu, można wprowadzić tekst, który po wciśnięciu ENTER pojawi się na polu tekstowym wyżej.

Budowa programu

PyGame

Gra Wargonem jest napisana w Pythonie oraz głównie opiera się na bibliotece PyGame.

Pozwala ona na tworzenie powierzchni, kwadratów oraz wgrywania obrazów, które potem są rysowane na ekranie, bądź innych powierzchniach.

Chat

Zaczynając od Lewej strony, pierwszym z trzech widocznych podziałów jest sekcja z wiadomościami tekstowymi oraz wejściem dla komend, bądź wiadomości, którą chce wysłać gracz

Input

Grafika omawianej powierzchni:



Input działa w następujący sposób.

Po naciśnięciu na powierzchnię, ustawiana jest flaga, która blokuje poruszanie się, oraz pozwala programowi czytywać kody ASCII naciśniętych klawiszy do **zmiennej**, z wyjątkiem ENTER oraz BACKSPACE, gdyż służą one kolejno do usunięcia ostatniej litery wpisanego słowa, oraz zaakceptowanie danego tekstu i przekierowanie go do powierzchni z tekstem, do której zaraz przejdziemy.

Jeżeli wysyłana wiadomość zaczyna się od „/”, wtedy wiadomość jest traktowana jako komenda, tym samym pokazuje na powierzchni wyżej jak się zakończyło użycie danej komendy.

Dostępne komendy:

`add <0-17>`

Podanie liczby od 0 do 17 spowoduje pojawienie się przedmiotu o danym numerze ID w inwentarzu bohatera.

`/HP <int>`

wprowadzenie wartości doda, (bądź odejmie, w sytuacji liczby ujemnej) punkty zdrowia bohaterowi.

`/EXP`

Podobnie jak przy komendzie HP, dodaje daną liczbę do ilości punktów EXP bohatera.

`/clear`

Czyści inwentarz postaci, razem z założonymi przedmiotami.

Pole Tekstowe

Pole tekstowe jest powierzchnią, która przetrzymuje w liście wartości przesłane przez input, bądź przez system (np. treści opisujące przebieg walki).

Tekst wypisuje się od dołu do góry, a przechowywane są dane, które mieszczą się na powierzchni pola tekstowego

Okno z grą

Okno z grą mimo swojej prostoty i braku podziału na osiach, ma swoją głębię (dosłownie, gdyż współdziałają powierzchnie, gdzie jedna przykrywa drugą)

Mapa

Mapa jest wgrywana z pliku map1 w katalogu [..]/Wargonem/assets/map/

Plik składa się z praktycznie 3 znaków, a mianowicie ‘p’, ‘g’, oraz każdy inny znak, który nie jest rozpoznawany przez program (także spacja się liczy do tej kategorii)

Znakowi ‘p’ odpowiada plik path.png:



a znak ‘g’ oznacza plik grass.png:



inny znak oznacza obraz o nazwie „blank.png”, który jest pustym obrazem z kanałem alfa, więc jest przezroczystym kwadratem

Mapa która się tworzy jest kwadrat o wymiarach najdłuższa linijka x ilość linii

Bohater oraz Przeciwnik

Są to 2 obiekty pochodzące z podobnych klas.

Mają podobne wartości, ale uznałem że metody są na tyle różne, że chciałem je oddzielić od siebie, aby nie pomylić w tym dla kogo wykonuję daną czynność.

Ogółem są to powierzchnie, które nie są rysowane, ale mają swoje położenie, oraz wartość „image”, która jest wyglądem postaci, bądź przeciwnika

Inwentarz oraz statystyki

Prawa strona ekranu dzieli się praktycznie na 4 części.

Każda z nich jest odpowiedzialna za inne elementy, bądź czynności

Wyposażenie

Wyposażeniem nazywam element na samej górze.

Jest to część gdzie są przedmioty które posiada nasza postać na sobie, tym samym przekazuje statystyki z przedmiotów na bohatera.

Każde miejsce na wyposażenie jest oznaczone ikoną przedmiotu, który tam przynależy:



Nałożenie przedmiotu na nieodpowiednie miejsce jest możliwe, lecz nie zwiększa to statystyk.

Więc można założyć buty na głowę, ale nic to nie da, poza ośmieszeniem się.

Statystyki

Pole na którym jest przedstawieniu wartości , które się kryją w obiekcie bohatera.

Pole na statystyki jest podzielone na 2 części, część która jest obrazem, gdzie w tle są rysowane powierzchnie przedstawiające HP oraz EXP, oraz powierzchnia, na której jest wypisywany tekst.

Grafika do słupków wygląda następująco:



Inwentarz

W inwentarzu znajduje się 35 miejsc na przedmioty.

W tym miejscu przedmioty nic nie robią, ale cały inwentarz pozwala na przechowywanie wartościowych przedmiotów, takie jak mikstury

Ikona pola w inwentarzu:



Śmietnik

Po przeniesieniu przedmiotu ma czerwony przycisk na dole ekranu, trzymany przedmiot zostanie zamieniony przedmiot o ID 0, co jest pustym obrazem bez statystyk.

Dla gry przedmiot o ID 0 jest uważany za puste miejsce

Ikona śmietnika:



Przedmioty

Dokładne informacje na temat przedmiotów znajdują się w pliku items.json w katalogu [..]/Wargonem/assets/

Jest w sumie 19 przedmiotów, gdzie ID == 0 jest dla przedmiotu pustego, ID 1 oraz 2, są dla mikstur, a ID od 3 do 17, są oznacza wyposażenie

(w plikau .json znajduje się tak zwane Jajka Wielkanocna, które można otrzymaj wyłącznie poprzez komendą /add)

Pliki z dodatkowymi funkcjami

Pierwsze pliki znajdują się w katalogu asstest

LoadFile.py

Plik ten jest odpowiedzialny za zwracanie zawartości z plików .json w katalogu assets

color_palette.json

Plik zawiera skróty do wartości RGB kolorów, w formie słownika

items.json

items.json przechowuje informacje o wszystkich przedmiotach, które można uzyskać w grze

tiles.json

Tutaj przechowywane są własności kratek, które są rysowane na mapie

DB.py

Plik ten zawiera klasę bazy danych, którą wykorzystuję do zapisywania oraz ładowania inwentarzu bohatera.

Kolejny plik jest w katalogu Logs

logs.py

Jest tam funkcja, która zapisuje logi, wywołane w czasie działania programu w celu weryfikowania pracy programu, nawet po jego zamknięciu.

Logi są przechowywane w tym samym katalogu, w pliku o nazwie logs

Słów kilka od autora

Dodatkowe informacje:

W końcowym etapie gry zauważyłem że plik z logami miał ponad pół miliona linijek i ważył 30MB.

Sam projekt przeszedł parę zmian w czasie tworzenia. Przykładowo dowiedziałem się że można stworzyć funkcję która zwraca obiekt, przez co nie potrzebowałem „kisić” obiektów w liście i zastanawiać się jak obiekty mają porozumiewać się między sobą. Dzięki temu mogłem korzystać z każdej metody kiedy tylko było to potrzebne

Przyznam że ten projekt jest ciekawej konstrukcji.

Mianowicie plik stary.py, który w sumie nic nie robi poza włączeniem gry. Jest tak z racji że miałem plan aby stworzyć połączenia socketami, więc zrobiłem oddzielny plik, który by się zajął takimi rzeczami, a później przejdzie dalej do etapu włączenia gry.

W plikach gry nadal istnieje plik gumba.png, który użyłem jako tymczasowy obraz dla postaci. Uznałem że zostawię tą grafikę jako smaczek.

Jednak robienie matematycznych wzorów do opisanie położenia elementów w odniesieniu o inne jest jednak czasami trudne

Zakończenie

W tym semestrze po raz pierwszy miałem prawdziwą styczność z językiem Python. Tym samym poznawanie biblioteki pygame było dla mnie progiem do przebrnięcia. Wybrałem duży projekt jak na moje doświadczenie, przez co praca była powolna, a z racji na uciekający czas, ostateczny wynik jest ukrojoną wersją oryginalnego zamysłu.

Ostatecznie mimo tego że w samej grze dzieje się nie wiele jestem dumny ze swojego projektu. Nauczyłem się wiele rzeczy, które chciałem wiedzieć od początku pisania tego programu. W pewnym sensie widać ewolucję sposobu pisania kodu w czasie, gdzie z zaczynałem używać nowych dla mnie własności pythona.

Na koniec chciałem podziękować, za to że Pan Piotr Schneider pozwolił mi wybrać alternatywną drogę tworzenia projektu końcowego. Tak samo chciałem podziękować Damianowi Gradziuk, za pomoc i nauki, które otrzymałem jako początkujący w bibliotece pygame.