

[RElog] – Oel (Pompowacze) port z C64 (część 1) (sheadovas/artykuly/relog/oel-port/relog-oel-pompowacze-port-z-c64-czesc-1/)

Mar 09, 2017 / Oel Port (sheadovas/category/artykuly/relog/oel-port/)

Wstęp

Hej, witam Was w kolejnym RElog’u. Tym razem razem postanowiłem zabrać się za coś co miałem zamiar zrobić jeszcze rok temu, a mianowicie postanowiłem się podjąć portem gry [Oel (https://www.c64-wiki.de/wiki/Hauptseite)] z [Commodore64 (https://pl.wikipedia.org/wiki/Commodore_64)] na coś nowszego zgodnie z tym co zaproponował *arhon* w swoim [odcinku (https://www.youtube.com/watch?v=41ZYJ5QctG0)] o tej grze.

Dla niewtajemniczonych: *Oel* jest grą od niemieckiego studia, wydaną w latach 80 na platformę *Commodore64*, a sama gra została napisana w [BASIC’u (https://en.wikipedia.org/wiki/BASIC)] (będzie to istotna informacja w dalszej części wpisu). Sama gra polega na stworenie imperium naftowego.

Autor podanego wyżej filmu zasugerował stworzenie portu na „nowoczesne” platformy, co wydaje mi się nieprzesadnie skomplikowanym zadaniem, ale jednocześnie wciąż ciekawym.

Port gry

Jak to będzie wyglądać?

Samego portu gry nie pokażę Wam w jednym wpisie z prostej przyczyny: nie jest gotowy, jednakże chciałbym się z Wami dzielić tym co udało mi się osiągnąć, a gdy nowości będzie nieco więcej (tzn po osiągnięciu każdego milestone’a) to utworzę nowy wpis.

Postęp prac nad portem można będzie śledzić na moim GitHubie: /sheadovas/oel-port (https://github.com/sheadovas/oel-port)

Część 1 – przygotowanie

Jak to w każdym projekcie warto na początku odpowiednio się przygotować. Pierwszym krokiem to oczywiście zaznajomienie się z tym co chcemy portować, a więc warto zaopatrzyć się w [obraz gry (http://www.oldgamesfinder.com/?m=7&q=Oel%20Pompowacze)] oraz jakiś emulator (np. [VICE (http://vice-emu.sourceforge.net/)]).

Dalszym krokiem jest wstępny głębszy research: jak wspomniałem gra powstała w BASIC’u, co jest o tyle ważne że dostępne są jej [źródła (https://www.c64-wiki.de/wiki/Oel/Listing)].

Skoro mamy źródła, a nie mieliśmy wcześniej do czynienia z tym językiem to warto zaopatrzyć się w jakiś [instruction set (https://en.wikipedia.org/wiki/BASIC#Typical_BASIC_keywords)], aby móc względnie komfortowo dokonywać translacji na bardziej nowoczesny język.

Przykładowa funkcja

```
1 | 260 gosub60000
2 | ...
3 | 60000 forb=1to6:print"{down}":nextb
4 | 60050 return
```

Powyższy listing skojarzył mi się pierwotnie z kodem assemblera i z góry założyłem że *gosub* to nic innego jak wywołanie funkcji, z kolei dalszy fragment to etykieta miejsca gdzie znajduje się funkcja lub jak w tym przypadku oznacza miejsce pod określonym adresem / offsetem. Skojarzenie okazało się trafne i zgodne z tym co mówi dokumentacja.

Dalszy fragment przypomina tradycyjną pętlę for, dla lepszej czytelności zmodyfikujmy ją nieco:

Przepisanie powyższego listingu w czytelniejszej formie

```
1 | for b=1 to 6:
2 |   print "{down}":
3 |   next b
4 | return
```

Co mniej więcej oznacza tyle: „wykonaj 6 razy funkcję print”, zauważamy też że print nie jest zwyczajny („format” string), a jego finalny efekt to wykonanie komendy „cursor down”. O specjalnych komendach możemy przeczytać [tutaj (http://www.atariarchives.org/2bml/appendix_e.php)].

Po takim wstępnym rozpoznaniu jesteśmy w stanie stwierdzić, że składnia jest dość łatwa, ale może mieć kilka niuansów, które warto będzie rozpatrzeć w przyszłości.

Po takim rozpoznaniu języka, warto ustalić sobie kilka „kamieni milowych”, które wyznaczą także tematykę dalszych wpisów:

- 1. PoC, kod „1:1” w C++ i SFML.
- 2. Dodatkowe optymalizacje, modyfikacje pod nowoczesny styl programowania.

Jak można zauważyć, wybrałem C++ i SFML. Powód jest dość prosty: dzisiaj do programowania gier korzysta się z odpowiednich narzędzi, SFML już jest *overkill’em* dla takiej gry, ale je znam dość dobrze i pisze mi się w nim komfortowo więc świetnie się nadaje do kodu w stylu *Working Demo*.

Podsumowanie

To tyle jeżeli chodzi o pierwszą część „dziennika” dotyczącego portu tej gry, w dalszych częściach przedstawię znacznie ciekawsze rzeczy. Zapraszam także na git’a projektu oraz do dzielenia się swoimi przemyśleniami np. poprzez system komentarzy.

Code ON!