

**KOMUNIKACJA CZŁOWIEK KOMPUTER**

**DOKUMENTACJA PROJEKT NR 1**

***Gra Tamagotchi***

Wykonujący:

**Adam Bajguz**

**Magdalena Kalisz**

**Paweł Gryc**

Studia dzienne

Kierunek: Informatyka

Semestr: V

Prowadzący: mgr inż. Łukasz Gadomer

...........................

OCENA

1. OPIS PROJEKTU

Celem projektu było stworzenie gry lub przynajmniej jej działającej wersji demonstracyjnej, zainspirowanej oryginalną elektroniczną zabawką Tamagotchi. Gra polega na opiece nad wirtualnym zwierzątkiem – należy je karmić, sprzątać po nim itp.

1. OPIS FUNKCJONALNOŚCI

* Możliwość ustawienia rozmiaru tekstu konsoli;
* Dwa tryby gry: demonstracyjny i normalny;
* 16 dynamicznie wybieranych kolorów z palety RGB;
* 5 różnych etapów ewolucji zwierzątka (od jajka do dorosłego);
* 2 różne postaci dziecka, 4 postaci brzdąca i nastolatka oraz aż 11 postaci dorosłego;
* 30 różnych animowanych zachowań w każdej postaci;
* Umożliwienie użytkownikowi zmiany grafiki w grze poprzez edycję bitmap lub czcionki zewnętrznym programem graficznym;
* Kupowanie jedzenia za punkty i karmienie zwierzątka;
* Sprzątanie po zwierzątku;
* Ewolucja zwierzątka
* Zdobywanie punktów za rozgrywkę w minigrze – Skakance;
* Wyswietlanie statystyk;
* Nadawanie imienia zwierzątku (maksymalnie 8 liter);
* Zapisywanie ustawień i stanu gry do pliku;
* Możliwość robienia dwóch rodzajów zrzutów ekranu – o rozmiarze 225 × 160 pikseli lub   
  (225 \* *szerokość czcionki*) × (160 \* *wysokość czcionki* /2) piksel.

1. OPIS WYKORZYSTANYCH NARZĘDZI

Cały projekt został wykonany przy pomocy języka C# oraz biblioteki ColorfulConsole   
i Windows API. Jako środowisko programistyczne wykorzystano Microsoft Visual Studio 2017 Enterprise. Przy tworzeniu i edycji grafik wykorzystano darmowe programy takie jak GIMP lub Paint.NET.

1. INSTRUKCJA INSTALACJI I KONFIGURACJI

Instalacja i uruchomienie gry polegają na:

* wypakowaniu archiwum zip z grą,
* uruchomieniu pliku wykonywalnego gry (w przypadku błędu należy upewnić się, że ma   
  się zainstalowany Microsoft .NET Framework w wersji 4.7.2 - starsze wersje mogą nie funkcjonować poprawnie).

**Gra nie wymaga innych czynności instalacyjnych ani konfiguracji.**

1. INSTRUKCJA UŻYTKOWNIKA

**UWAGA:**

Gra działa wyłącznie na systemach Microsoft Windows w wersji 7 lub wyższej. Ponadto wymaga zainstalowanego Microsoft .NET Framework oraz około 120MB wolnej pamięci RAM.

* Interfejs został przystosowany do poruszania się strzałkami.
* Zatwierdzanie akcji odbywa się klawiszem Enter.
* Wyjście z poszczególnych ekranów odbywa się poprzez naciśniecie odpowiedniego przycisku w interfejsie lub klawisza Esc.
* W mini grze Skakanka dodatkowo używany jest klawisz spacji do skoku.
* Klawisz F10 służy do robienia zrzutów ekranu o rozmiarze 225 × 160 pikseli, a F12 do robienia zrzutów ekranu o rozmiarze (225 \* *szerokość czcionki*) × (160 \* *wysokość czcionki* /2) pikseli. Zrzuty ekranu zapisują się w katalogu *screenshots* w katalogu z grą.

**Opis przycisków menu w grze:**



**Mini gra - Skakanka**

**wybrany element**

**Zmiana wieku**

**Wyjście**

**Zmiana zachowania**

**Toaleta**

**Sklep z jedzeniem**

**Statystyki**

1. WNIOSKI I SAMOOCENA

* Nieszablonowe podejście do wykorzystania konsoli i inspiracja grafiką High ANSII Art oraz biblioteką AAlib, a także znudzenie rysowaniem standardowych interfejsów TUI pozwoliły na stworzenie oryginalnego konceptu gry tekstowej, wyposażonej w interfejs symulujący tryb graficzny.
* Projekt umożliwił stworzenie ciekawego konceptu warstwowego silnika graficznego   
  z buforowaniem wyświetlanych elementów na konsoli i symulacją dwóch pikseli na każdy znak konsoli. Silnik ten opiera się w głównej mierze na bitmapach i posiada rozbudowane mechanizmy zarządzające nimi (animacje, przyciski, menu, tekst).
* Wykorzystanie części klas i metod z biblioteki ConsoleColor pozwoliło na jej integrację   
  z silnikiem graficznym oraz dynamiczne remapowanie standardowych 16 kolorów na dowolne   
  z 24-bitowej palety RGB.
* Dużym ułatwieniem w projektowaniu gry okazało się wykorzystanie bitmap, które pomimo konieczności pilnowania ograniczenia w postaci 16 wyświetlanych jednocześnie na ekranie kolorów, pozwoliły na szybkie tworzenie grafiki i interfejsu w grze.
* Chęć optymalizacji wyświetlania grafiki umożliwiła zapoznanie się z częścią Windows API odpowiedzialnej za obsługę konsoli systemu Windows, a zwłaszcza buforowania jej okna   
  i błyskawicznego przełączania bufora.
* Projekt pozwolił również na wykorzystanie wskaźników w języku C# (unsafe code) do szybszego iterowania i manipulacji bitmap.
* Wykorzystanie serializacji umożliwiło łatwe i szybkie zapisywanie oraz wczytywanie ustawień   
  i zapisu gry.