Filmik udostępniowy w repo – żadne z nas nie posiada (nie chciało posiadać) konta na YT i mamy nadzieję, że udostępnienie go tutaj nie będzie nas dyskwalifikowało.

Zespół:

* Karolina Skiba (lider)
* Tomasz Skiba

Mini gra – EatME

Zbierz losowo generowane obiekty w jak najkrótszym czasie.

+ technologia: Console

1. IntroduceYourself Tomasz

Wszelkie zmienne – private, listy – readonly.

Konstruktor – dwie listy. Jedna odpowiadająca za wybór koloru (który będzie towarzyszyć graczowi), druga wzoru pionka (public readonly List<ConsoleColor> WhichColor = new List<ConsoleColor>(); public readonly List<char> WhichPawn = new List<char>()).

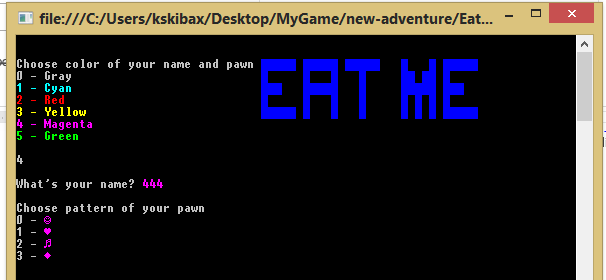
Wybór koloru – foreach wyświetlanie możliwych opcji (skorzystano również z enum), try – catch w pętli while wpisywanie (public void Color())

Pobranie koloru – pobranie i przypisanie wybranego indeksu listy (public void GetColor())

Wybór wzoru pionka – for wyświetlanie możliwych opcji, try – catch w pętli while wpisywanie wyboru (public void Sign())

Pozostałe funkcje – niewymagające dokładniejszego komentarza, czyli – używanie Console.WriteLine() oraz Console.ReadLine() z ewentualnym uwzględniem powyższych, bądź pobieranie prywatnych zmiennych potrzebnych w dalszej części programu (return)

* public void Name(),public void GetName() i public void GetNameInColor();
* public char GetPawnPattern() i public void GetPawnPatternInColor();



1. BoardGame Karolina

Wszelkie zmienne – private, tablica – protected.

Konstruktor – alokowanie pamięci na planszę gry (tablica statyczna dwuwymiarowa), utworzenie ramki.

Region dla wnętrz poszczególnych plansz – plansze tworzone for + if.

Wybór planszy – random + switch (private void WhichBoard())

Losowe generownia obiektów do zjedzenia - random + for z while dopóki dopóty nie znajdzie odpowiedniego miejsca (private void RandomObjects())

Czyszczenie wnętrza plansz – by uniknąć nakładania się plansz na siebie (private void RandomObjects())

Clu tej klasy.

Wyświetlenie wylosowanej planszy wraz z obiektami do zjedzenia (public void DisplayField()) oraz bool dla zjedzonych elementów będący podstawą prawidłowego przejścia i zakończenia gry (public bool HasEverythingEaten()).

1. In-GameControl dziedziczenie po klasie BoardGame Tomasz

Wszelkie zmienne – private, utworzono również obiekty klas IntroduceYourself i Messages

Konstruktor – ustawienie pozycji gracza wcześniej wybranym wzorem pionka.

Pobranie pozycji gracza – płynne przemieszczanie gracza, reset pozycji (public int GetPlayerPositionY(), public int GetPlayerPositionX(), public Tuple<int, int> ResetPosition())

Kierunki – sterownia pionkiem z uwzględnieniem nieprzechodzenia przez ściany opatrzone w region (public void Up(),public void Down(),public void Right(),public void Left()).

1. Messages Karolina

Czyli wszelkie komunikaty towarzyszące graczowi przez całą grę. Wykorzystano podstawowe komendy, jak Console.WriteLine(), Console.SetCursorPosition(),Console.ForegroundColor = ConsoleColor, Console.ResetColor().

(public void Legend(), public void Attention(), public void Logo()).

1. Scores dziedziczenie po interfejsie IComparable<Scores> Tomasz

Klasa, w której wszelkie dane są serializowane. Zgodnie z wymogiem serializacji utworzony został pusty konstruktor, a zmienne są publiczne. Wykorzystano tutaj właściwości get – set oraz stoworzono własną metodę sortującą.

(public Scores(string name, string time), public Scores(),public int CompareTo(Scores other)).

1. BestPlayersList Tomasz

Wszelkie zmienne – private, lista – public (zgodnie z wymaganiami serializacji).

Reset zmiennych – pozwala na wyświetlanie zawsze w tym samym miejscu i w tej samej ilości rankingu najlepszych 10 wyniików posortowanych po czasie (public Tuple<int, int, int, int> ResetValues()).

Serializcja: Czytanie z i zapisywanie do pliku \*.xml (public void ReadTheFile(), public void WriteTheFile())

Uzupełnianie listy kolejnym rekordem (public void FillTheList(string name, string time))

Wyświetlanie pozyskanych podczas gry wyników z zabezpieczeniem dla pustej listy i/lub braku pliku z wynikami (public void DisplayList()).

1. PlayTheGame Karolina

Wszelkie zmienne – private, utworzono również obiekty klas In\_GameControl, IntroduceYourself, BestPlayersList i Messages

Połączenie wszystkich powyższych metod w działającą grę.

public void PrepareToPlay().

Metodę rozpoczyna kilka funkcji z klas opisanych powyżej IntroduceYourself, BestPlayersList i Messages. public void ReadTheFile(), public void Logo(), Console.SetCursorPosition(), public void GetNameInColor(). W tym też momencie startuje zegar odliczający czas przejścia gry (Stopwatch stopwatch = new Stopwatch(); stopwatch.Start()) oraz pobrane zostają współrzędne pionka (public int GetPlayerPositionY(), public int GetPlayerPositionX()). Następny etap to „wieczna pętla” (while(true)), która w zależności od pobranego znaku (Console.ReadKey(true).Key), wykona akcję (switch). By uniknąć odświeżania kosoli po każdym wciśnietym przycisku wykorzystano nadpisywanie poszczególnych pól (control.GetBoard()[prevY, prevX] = ' ';). Całość „wiecznej pętli” kończy if + bool HasEverythingEaten() wraz z komunikatem uwzględniającym czas przejścia i zapytanie „Wanna play the game?”.

Akcje switch ():

* Opisane już kierunki: public void Up(), public void Down(), public void Right(), public void Left().
* public void NewPlayer() – z akcji skorzystano dwukrotnie. W zależności od wartości bool (bool firstPlay) funkcja służy do wywołania metod personalizujących gracza (bool firstPlay = true) – kolor, nazwa i wzór pionka (metody kolejno: Console.Clear(), public void Logo(), public void Color(), public void Name(), public void Sign(), public Tuple<int, int> ResetPosition() i public void PrepareToPlay()) lub zmianę planszy (bool firstPlay = false) (wywołanie Console.Clear() oraz public void PrepareToPlay()).
* public void DisplayList() – opisane w klasie BestPlayersList.
* public void Exit() – pożeganie się z graczem – wyświetlenie w kolorze napisu „The End” uśpionego na 1s. (Thread.Sleep(1000)).

public string GetTime() – zwrócenie czasu przejścia gry jako główne kryterium sortowania rankingu njalepszych graczy

public void WantPlay().

Metodę rozpoczyna proste zapytanie „Wanna play the game?”, które w zależności od wybranej odpowiedzi albo kontynuuje działanie programu public void NewPlayer() albo je kończy public void Exit().

1. Program Karolina

Main – uruchomienie gry za pomocą wywołanej metody public void WantPlay().