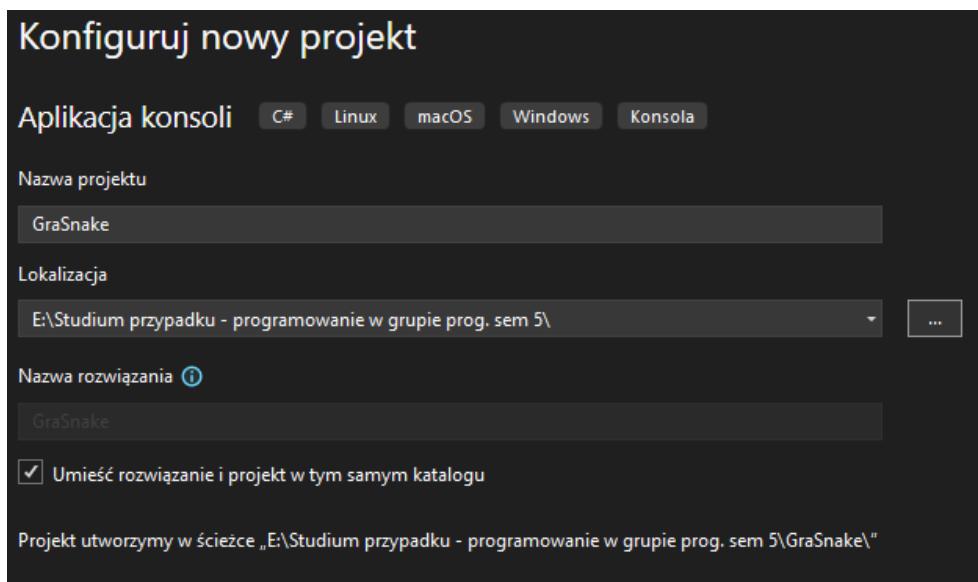
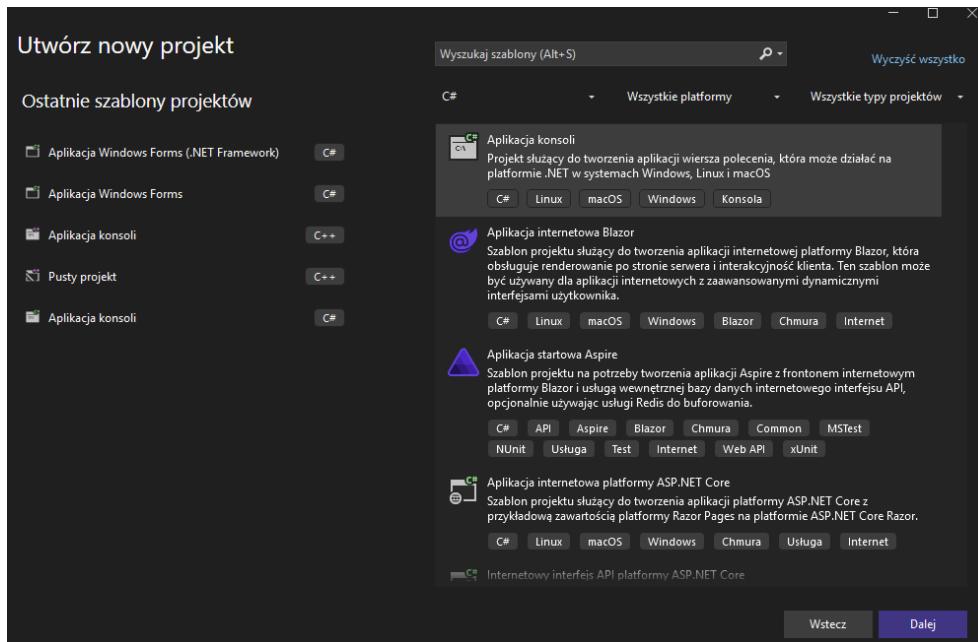


Sprawozdanie i realizacja projektu gry Snake

Krok 1: Tworzenie projektu w Visual Studio

W środowisku Visual Studio utworzono nowy projekt aplikacji konsolowej.

Zawartość dostarczonego pliku Snake.cs skopiowano do głównego pliku projektu Program.cs. Następnie usunięto zbędny plik Snake.cs, a do katalogu projektu przeniesiono wymagane pliki klas: Obstakel.cs oraz Pixel.cs.



The screenshot shows the Visual Studio code editor with the file 'Program.cs' open. The code is a C# program for a snake game. It includes imports for System, System.Collections.Generic, System.Linq, and System.Threading. The Main() method initializes the window size to 16x32 pixels, creates a random number generator, and initializes a 'hoofd' pixel at the center of the screen. The movement direction is set to 'RIGHT'. A list 'telje' is initialized to store the snake's body segments. The score starts at 0.

```
1 using System;
2 using System.Collections.Generic;
3 using System.Linq;
4 using System.Threading;
5
6 class Program
7 {
8     static void Main()
9     {
10        Console.WindowHeight = 16;
11        Console.WindowWidth = 32;
12        int screenwidth = Console.WindowWidth;
13        int screenheight = Console.WindowHeight;
14        Random randomnummer = new Random();
15        pixel hoofd = new pixel();
16        hoofd.xpos = screenwidth / 2;
17        hoofd.ypos = screenheight / 2;
18        hoofd.schermkleur = ConsoleColor.Red;
19        string movement = "RIGHT";
20        List<int> telje = new List<int>();
21        int score = 0;
22    }
23}
```

The screenshot shows a Windows File Explorer window displaying the contents of the 'GraSnake' project folder. The folder structure includes subfolders 'bin' and 'obj', the project file 'GraSnake.csproj', the solution file 'GraSnake.sln', and source files 'Program.cs', 'Obstakel.cs', and 'Pixel.cs'. All files were modified on January 25, 2026, at 22:13, except for 'Pixel.cs' which was modified at 15:51.

Nazwa	Data modyfikacji	Typ	Rozmiar
bin	25.01.2026 22:13	Folder plików	
obj	25.01.2026 22:13	Folder plików	
GraSnake.csproj	25.01.2026 22:13	C# Project File	1 KB
GraSnake.sln	25.01.2026 22:13	Visual Studio Solu...	2 KB
Program.cs	25.01.2026 22:13	Plik CS	1 KB
Obstakel.cs	25.01.2026 15:51	Plik CS	1 KB
Pixel.cs	25.01.2026 15:51	Plik CS	1 KB

Krok 2: Tworzenie repozytorium na GitHub i łączenie

W serwisie GitHub utworzono nowe repozytorium, a następnie zmieniono jego nazwę z GraSnake na SnakeGame.

Create a new repository

Repositories contain a project's files and version history. Have a project elsewhere? [Import a repository](#).
Required fields are marked with an asterisk (*).

1 General

Owner * Repository name *

 kotulski / GraSnake

GraSnake is available.

Great repository names are short and memorable. How about [redesigned-waffle](#)?

Description

Konsolowa gra Snake napisana w języku C#. Projekt edukacyjny mający na celu naukę współpracy przy użyciu :

180 / 350 characters

2 Configuration

Choose visibility *

Choose who can see and commit to this repository

 Public

Add README

READMEs can be used as longer descriptions. [About READMEs](#)

Off

Add .gitignore

.gitignore tells git which files not to track. [About ignoring files](#)

No .gitignore

Add license

Licenses explain how others can use your code. [About licenses](#)

 MIT License

Create repository

GraSnake Public

About

Konsolowa gra Snake napisana w języku C#. Projekt edukacyjny mający na celu naukę współpracy przy użyciu systemu Git (naprawa błędów, refaktoryzacja kodu i obsługa zgłoszeń zmian).

Activity

0 stars
0 watching
0 forks

Releases

No releases published
[Create a new release](#)

SnakeGame Public

Activity

0 stars
0 watching
0 forks

Za pomocą edytora tekstuowego Vim w konsoli edytowano plik .gitignore.
Wprowadzono do niego reguły wykluczające katalogi .vs, bin oraz obj.
Po zapisaniu zmian, polecenie git add . wykonano poprawnie, a następnie przesłano pliki na serwer (z użyciem wymuszenia push -f w celu synchronizacji historii).

```
MINGW64:/e/Studium przypadku - programowanie w grupie prog. sem 5/GraSnake
$ git config --global user.name "Patryk"
$ git config --global user.email "patrykos138@gmail.com"
$ git init
Initialized empty Git repository in E:/Studium przypadku - programowanie w grupie prog. sem 5/GraSnake/.git/
$ git add .
error: open(".vs/GraSnake/FileContentIndex/0beb20be-8528-4db2-9964-20448ab14044.vsidx"): Permission denied
error: unable to index file '.vs/GraSnake/FileContentIndex/0beb20be-8528-4db2-9964-20448ab14044.vsidx'
fatal: adding files failed
$ touch .gitignore
```

```
patryk@Patryk MINGW64 /e/Studium przypadku - programowanie w grupie prog. sem 5/GraSnake (main)
$ git add .
warning: in the working copy of '.gitignore', LF will be replaced by CRLF the next time Git touches it

patryk@Patryk MINGW64 /e/Studium przypadku - programowanie w grupie prog. sem 5/GraSnake (main)
$ git commit -m "Initial commit - start projektu SnakeGame"
[main (root-commit) a7b73b1] Initial commit - start projektu SnakeGame
 6 files changed, 369 insertions(+)
create mode 100644 .gitignore
create mode 100644 GraSnake.csproj
create mode 100644 GraSnake.csln
create mode 100644 Obstakel.cs
create mode 100644 Pixel.cs
create mode 100644 Program.cs

patryk@Patryk MINGW64 /e/Studium przypadku - programowanie w grupie prog. sem 5/GraSnake (main)
$ git branch -M main

patryk@Patryk MINGW64 /e/Studium przypadku - programowanie w grupie prog. sem 5/GraSnake (main)
$ git remote add origin https://github.com/kotulski/SnakeGame

patryk@Patryk MINGW64 /e/Studium przypadku - programowanie w grupie prog. sem 5/GraSnake (main)
$ git push -u origin main
To https://github.com/kotulski/SnakeGame
 ! [rejected]      main -> main (fetch first)
error: failed to push some refs to 'https://github.com/kotulski/SnakeGame'
hint: Updates were rejected because the remote contains work that you do not
hint: have locally. This is usually caused by another repository pushing to
hint: the same ref. If you want to integrate the remote changes, use
hint: 'git pull' before pushing again.
hint: See the 'Note about fast-forwards' in 'git push --help' for details.

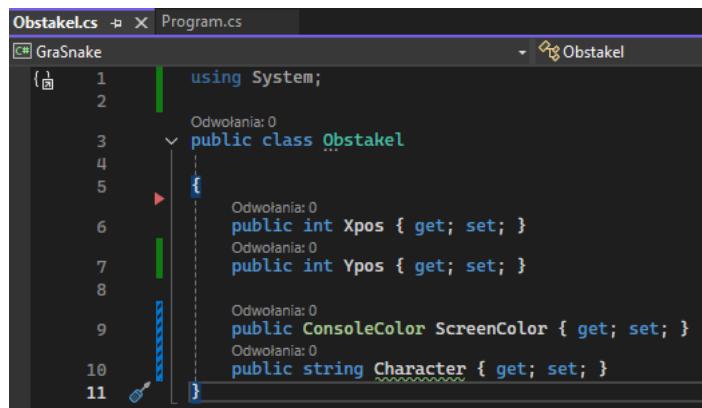
patryk@Patryk MINGW64 /e/Studium przypadku - programowanie w grupie prog. sem 5/GraSnake (main)
$ git push -f origin main
Enumerating objects: 8, done.
Counting objects: 100% (8/8), done.
Delta compression using up to 8 threads
Compressing objects: 100% (7/7), done.
Writing objects: 100% (8/8), 2.25 KiB | 1.13 MiB/s, done.
Total 8 (delta 1), reused 0 (delta 0), pack-reused 0 (from 0)
remote: Resolving deltas: 100% (1/1), done.
To https://github.com/kotulski/SnakeGame
 + 234d761...a7b73b1 main -> main (forced update)
```

Krok 3: Podział zadań

Ze względu na dużą liczbę błędów w dostarczonym kodzie źródłowym, projekt został podzielony na dwa główne etapy naprawcze (Issues), które będą realizowane niezależnie na osobnych gałęziach (branches). Symuluje to pracę zespołu programistycznego.

Krok 4: Tworzenie oddzielnej gałęzi

Realizację czwartego etapu projektu rozpoczęto od utworzenia w lokalnym repozytorium nowej gałęzi roboczej o nazwie fix-classes, co pozwoliło na bezpieczne wprowadzanie zmian bez ingerencji w stabilną wersję kodu. Głównym zadaniem była naprawa błędu w klasie Obstakel, która nie posiadała definicji współrzędnej pionowej, dlatego kod uzupełniono o brakującą właściwość Ypos. Przy okazji edycji pliku przeprowadzono również refaktoryzację nazewnictwa, zastępując błędne i obcojęzyczne nazwy zmiennych (takie jak schermKleur czy karakter) poprawnymi odpowiednikami w języku angielskim, co znacząco poprawiło czytelność kodu. Po zatwierdzeniu zmian i przesłaniu ich na serwer, w serwisie GitHub utworzono oficjalne zgłoszenie zmian (Pull Request), które po weryfikacji zostało scalone z główną gałęzią projektu.



```
patryk@Patryk MINGW64 /e/Studium przypadku - programowanie w grupie prog. sem 5/GraSnake (main)
$ git checkout -b fix-classes
Switched to a new branch 'fix-classes'

patryk@Patryk MINGW64 /e/Studium przypadku - programowanie w grupie prog. sem 5/GraSnake (fix-classes)
$ git status
On branch fix-classes
Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git restore <file>..." to discard changes in working directory)
    modified:   Obstakel.cs
    modified:   Pixel.cs
    modified:   Program.cs

no changes added to commit (use "git add" and/or "git commit -a")

patryk@Patryk MINGW64 /e/Studium przypadku - programowanie w grupie prog. sem 5/GraSnake (fix-classes)
$ git add .

patryk@Patryk MINGW64 /e/Studium przypadku - programowanie w grupie prog. sem 5/GraSnake (fix-classes)
$ git commit -m "Fix: Dodanie brakującej właściwości Ypos do klasy Obstakel, poprawa nazewnictwa zmiennych"
[fix-classes ad5449c] Fix: Dodanie brakującej właściwości Ypos do klasy Obstakel, poprawa nazewnictwa zmiennych
 3 files changed, 10 insertions(+), 12 deletions(-)

patryk@Patryk MINGW64 /e/Studium przypadku - programowanie w grupie prog. sem 5/GraSnake (fix-classes)
$ git push -u origin fix-classes
Enumerating objects: 9, done.
Counting objects: 100% (9/9), done.
Delta compression using up to 8 threads
Compressing objects: 100% (5/5), done.
Writing objects: 100% (5/5), 731 bytes | 731.00 KiB/s, done.
Total 5 (delta 2), reused 0 (delta 0), pack-reused 0 (from 0)
remote: Resolving deltas: 100% (2/2), completed with 2 local objects.
remote:
remote: Create a pull request for 'fix-classes' on GitHub by visiting:
remote:     https://github.com/kotulski/SnakeGame/pull/new/fix-classes
remote:
To https://github.com/kotulski/SnakeGame
 * [new branch]      fix-classes -> fix-classes
branch 'fix-classes' set up to track 'origin/fix-classes'.
```

Krok 5: Kodowanie i wysyłanie zgłoszeń zmian

Po naprawieniu klas przystąpiono do realizacji drugiego kluczowego etapu, jakim była implementacja silnika gry. W tym celu utworzono nową gałąź o nazwie fix-game-logic, aby odseparować prace nad logiką od głównego kodu. W pliku Program.cs całkowicie przepisano główną pętlę programu, zastępując blokującą metodę pobierania znaku mechanizmem sprawdzania dostępności klawisza, co zapewniło płynność rozgrywki. Ujednolicono również nazewnictwo zmiennych, dostosowując je do wcześniejszych zrefaktoryzowanych klas Pixel i Obstakel. Po zakończeniu prac programistycznych i wstępnej weryfikacji działania kodu, zmiany zostały zatwierdzone i przesłane do zdalnego repozytorium na GitHub.

```
patry@Patryk MINGW64 /e/Studium przypadku - programowanie w grupie prog. sem 5/GraSnake (fix-classes)
$ git checkout main
Switched to branch 'main'

patry@Patryk MINGW64 /e/Studium przypadku - programowanie w grupie prog. sem 5/GraSnake (main)
$ git pull origin main
From https://github.com/kotulski/SnakeGame
 * branch            main      -> FETCH_HEAD
Already up to date.

patry@Patryk MINGW64 /e/Studium przypadku - programowanie w grupie prog. sem 5/GraSnake (main)
$ git checkout -b fix-game-logic
Switched to a new branch 'fix-game-logic'

patry@Patryk MINGW64 /e/Studium przypadku - programowanie w grupie prog. sem 5/GraSnake (fix-game-logic)
$ git add .
patry@Patryk MINGW64 /e/Studium przypadku - programowanie w grupie prog. sem 5/GraSnake (fix-game-logic)
$ git commit -m "Fix: Naprawa logiki gry - wąż porusza się, zjada owoce, gra działa!"
[fix-game-logic 18701e1] Fix: Naprawa logiki gry - wąż porusza się, zjada owoce, gra działa!
 3 files changed, 96 insertions(+), 272 deletions(-)

patry@Patryk MINGW64 /e/Studium przypadku - programowanie w grupie prog. sem 5/GraSnake (fix-game-logic)
$ git push -u origin fix-game-logic
Enumerating objects: 9, done.
Counting objects: 100% (9/9), done.
Delta compression using up to 8 threads
Compressing objects: 100% (5/5), done.
Writing objects: 100% (5/5), 1.68 KiB | 1.68 MiB/s, done.
Total 5 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
remote: Resolving deltas: 100% (1/1), done.
remote:
remote: Create a pull request for 'fix-game-logic' on GitHub by visiting:
remote:   https://github.com/kotulski/SnakeGame/pull/new/fix-game-logic
remote:
To https://github.com/kotulski/SnakeGame
 * [new branch]      fix-game-logic -> fix-game-logic
branch 'fix-game-logic' set up to track 'origin/fix-game-logic'.
```

```
Pixel.cs Obstakel.cs Program.cs
GraSnake
1  using System;
2  using System.Collections.Generic;
3  using System.Linq;
4  using System.Threading;
5
6  class Program
7  {
8      static void Main()
9      {
10         Console.WindowWidth = 16;
11         Console.WindowHeight = 32;
12         int screenWidth = Console.WindowWidth;
13         int screenHeight = Console.WindowHeight;
14         Random randomNum = new Random();
15
16         while (true)
17         {
18             int score = 0;
19             string movement = "RIGHT";
20             Pixel head = new Pixel();
21             head.xPos = screenWidth / 2;
22             head.yPos = screenHeight / 2;
23             head.ScreenColor = ConsoleColor.Red;
24
25             List<int> tailPositions = new List<int>();
26             tailPositions.Add(head.xPos);
27             tailPositions.Add(head.yPos);
28
29             Obstakel fruit = new Obstakel();
30             fruit.Character = "#";
31             fruit.ScreenColor = ConsoleColor.Cyan;
32             fruit.xPos = randomNum.Next(1, screenWidth - 2);
33             fruit.yPos = randomNum.Next(1, screenHeight - 2);
34
35             Console.Clear();
36             Console.ForegroundColor = ConsoleColor.White;
37             for (int i = 0; i < screenWidth; i++)
38             {
39                 Console.SetCursorPosition(i, 0); Console.Write("#");
40                 Console.SetCursorPosition(i, screenHeight - 1); Console.Write("#");
41             }
42             for (int i = 0; i < screenHeight; i++)
43             {
44                 Console.SetCursorPosition(0, i); Console.Write("#");
45                 Console.SetCursorPosition(screenWidth - 1, i); Console.Write("#");
46             }
47         }
48     }
49 }
```

Krok 6: Recenzja i zatwierdzenie zmian

W serwisie GitHub utworzono zgłoszenie zmian (Pull Request) dla gałęzi fix-game-logic, opisujące zakres wprowadzonych poprawek w silniku gry. Po zweryfikowaniu, że nowy kod poprawnie obsługuje poruszanie się węża oraz mechanikę zbierania punktów, zgłoszenie zostało zaakceptowane. Nastąpiło scalenie (merge) gałęzi logicznej z główną gałęzią main, dzięki czemu repozytorium zawierało teraz w pełni funkcjonalną wersję bazową gry.

The screenshot shows a GitHub pull request page for a repository named 'SnakeGame'. At the top, there's a yellow banner indicating 'fix-game-logic had recent pushes 43 minutes ago' and a green 'Compare & pull request' button. Below the banner, a green box contains a checkmark icon and the text 'No conflicts with base branch' followed by the subtext 'Merging can be performed automatically.' A green 'Merge pull request' button is visible, along with a note that you can also merge via command line. In the bottom right corner of the main area, there's a link 'Still in progress? Convert to draft'. Below this, another box shows a purple icon and the text 'Pull request successfully merged and closed', with a note that the 'fix-game-logic' branch can be safely deleted. A 'Delete branch' button is located to the right of this message.

Krok 7: Testowanie i poprawki

Po scaleniu zmian przeprowadzono testy manualne na zaktualizowanej gałęzi głównej, które ujawniły problemy z migotaniem ekranu oraz brak możliwości restartu gry po porażce. W ramach prac poprawkowych zoptymalizowano proces renderowania grafiki, zmieniając kolejność czyszczenia i rysowania obiektów oraz ukrywając kursor konsoli, co wyeliminowało błędy wizualne. Dodatkowo cała logika rozgrywki została osadzona w pętli zewnętrznej, co pozwoliło na wprowadzenie mechanizmu ponownego uruchomienia gry po wyświetleniu komunikatu o końcu rozgrywki bez konieczności restartowania całej aplikacji.

```
patry@Patryk MINGW64 /e/Studium przypadku - programowanie w grupie prog. sem 5/GraSnake (fix-game-logic)
$ git checkout main
Switched to branch 'main'

patry@Patryk MINGW64 /e/Studium przypadku - programowanie w grupie prog. sem 5/GraSnake (main)
$ git pull origin main
remote: Enumerating objects: 1, done.
remote: Counting objects: 100% (1/1), done.
remote: Total 1 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
Unpacking objects: 100% (1/1), 965 bytes | 241.00 KiB/s, done.
From https://github.com/kotulski/SnakeGame
 * branch            main      -> FETCH_HEAD
   a7b73b1..c9b1d6a main      -> origin/main
Updating a7b73b1..c9b1d6a
Fast-forward
 Obstakel.cs |  13 +--
 Pixel.cs    |   9 +-
 Program.cs  | 346 ++++++-----+
 3 files changed, 96 insertions(+), 272 deletions(-)
```

Krok 8: Współpraca na konsoli

W celu spełnienia wymagań projektowych dotyczących symulacji współpracy, zaimplementowano moduł interfejsu sieciowego. Na początku działania programu dodano sekwencję komunikatów symulującą nawiązywanie połączenia z drugim graczem w trybie multiplayer. Funkcja ta wizualizuje proces łączenia z serwerem, a po upływie zdefiniowanego czasu i braku odpowiedzi automatycznie przełącza grę w tryb jednoosobowy offline, co stanowi realizację założenia o interakcji środowiskowej na konsoli.

```
// --- MULTIPLAYER
Console.CursorVisible = false;
Console.ForegroundColor = ConsoleColor.Cyan;
Console.WriteLine("\n\n\tSNAKE ONLINE");
Console.WriteLine("\tSzukanie serwera...");
Thread.Sleep(500);
Console.WriteLine("\tŁączenie z graczem 2...");
Thread.Sleep(500);
Console.ForegroundColor = ConsoleColor.Red;
Console.WriteLine("\tBłąd połączenia! Tryb Offline.");
Thread.Sleep(1000);
Console.Clear();
//
```

```
patryk@Patryk MINGW64 /e/Studium przypadku - programowanie w grupie prog. sem 5/GraSnake (main)
$ git pull origin main
From https://github.com/kotulski/SnakeGame
 * branch            main      -> FETCH_HEAD
Already up to date.

patryk@Patryk MINGW64 /e/Studium przypadku - programowanie w grupie prog. sem 5/GraSnake (main)
$ git checkout -b feature-multiplayer
Switched to a new branch 'feature-multiplayer'

patryk@Patryk MINGW64 /e/Studium przypadku - programowanie w grupie prog. sem 5/GraSnake (feature-multiplayer)
$ git add .

patryk@Patryk MINGW64 /e/Studium przypadku - programowanie w grupie prog. sem 5/GraSnake (feature-multiplayer)
$ git commit -m "Dodanie symulacji modułu sieciowego"
[feature-multiplayer cc53081] Dodanie symulacji modułu sieciowego
 1 file changed, 13 insertions(+)

patryk@Patryk MINGW64 /e/Studium przypadku - programowanie w grupie prog. sem 5/GraSnake (feature-multiplayer)
$ git push -u origin feature-multiplayer
Enumerating objects: 5, done.
Counting objects: 100% (5/5), done.
Delta compression using up to 8 threads
Compressing objects: 100% (3/3), done.
Writing objects: 100% (3/3), 539 bytes | 539.00 KiB/s, done.
Total 3 (delta 2), reused 0 (delta 0), pack-reused 0 (from 0)
remote: Resolving deltas: 100% (2/2), completed with 2 local objects.
remote:
remote: Create a pull request for 'feature-multiplayer' on GitHub by visiting:
remote:   https://github.com/kotulski/SnakeGame/pull/new/feature-multiplayer
remote:
To https://github.com/kotulski/SnakeGame
 * [new branch]      feature-multiplayer -> feature-multiplayer
branch 'feature-multiplayer' set up to track 'origin/feature-multiplayer'.
```

Dodanie symulacji modułu sieciowego #2



kotulski wants to merge 1 commit into `main` from `feature-multiplayer`



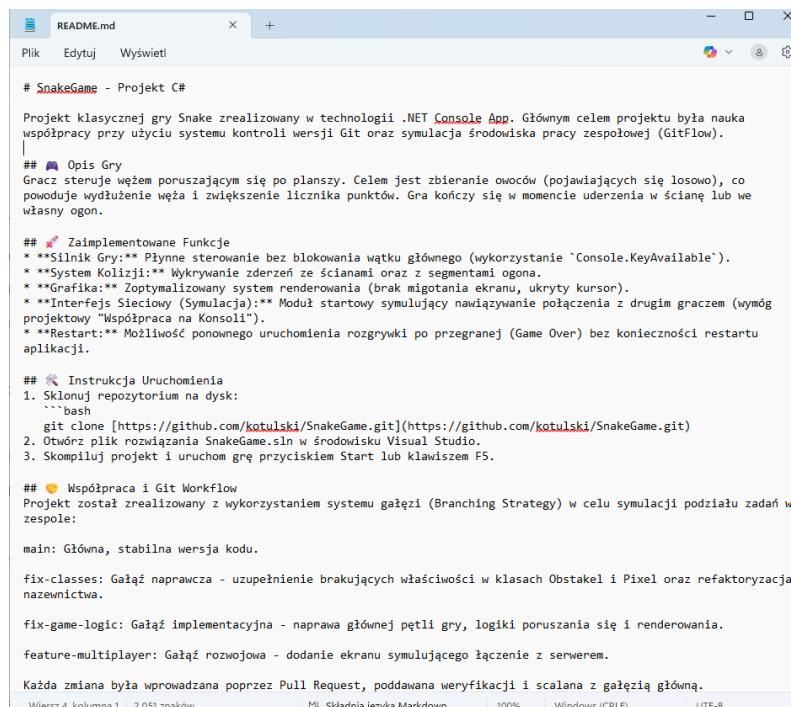
Krok 9: Rozwiązywanie problemów (Issues)

Zarządzanie błędami i nowymi funkcjami odbywało się poprzez systematyczne tworzenie dedykowanych gałęzi dla każdego zagadnienia. Problemy takie jak błędy komplikacji czy brakujące funkcjonalności były traktowane jako osobne zadania, dla których tworzono odpowiednie środowiska pracy w systemie Git. Takie podejście pozwoliło na zachowanie czystości historii zmian oraz ułatwiło śledzenie postępów w naprawianiu kodu gry.

Krok 10: Aktualizacja dokumentacji

Ostatnim etapem prac było opracowanie dokumentacji technicznej projektu.

Utworzono plik README.md, w którym zawarto szczegółowy opis zasad gry, zaimplementowanych funkcji oraz instrukcję uruchomienia projektu w środowisku Visual Studio. Dokumentacja ta została dodana do repozytorium jako osobny commit na gałęzi głównej, co finalnie zamknęło prace nad projektem SnakeGame.



The screenshot shows a Microsoft Word document window titled "README.md". The document contains the following content:

```
# SnakeGame - Projekt C#
Projekt klasycznej gry Snake zrealizowany w technologii .NET Console App. Głównym celem projektu była nauka współpracy przy użyciu systemu kontroli wersji Git oraz symulacja środowiska pracy zespołowej (GitFlow).
## 🎮 Opis Gry
Gracz steruje wężem poruszającym się po planszy. Celem jest zbieranie owoców (pojawiających się losowo), co powoduje wydłużenie węża i zwiększenie licznika punktów. Gra kończy się w momencie uderzenia w ścianę lub we własny ogon.

## 🚀 Zaimplementowane Funkcje
* *Silon*: Wykrywanie zderzeń ze ścianami oraz z segmentami ogona.
* *System Kolizji*: Wykrywanie zderzeń ze ścianami oraz z segmentami ogona.
* *Grafika*: Zoptymalizowany system renderowania (brak migotania ekranu, ukryty kursor).
* *Interfejs Sieciowy (Symulacja)*: Moduł startowy symulujący nawiązywanie połączenia z drugim graczem (wymóg projektowy "Współpraca na Konsoli").
* *Restart*: Możliwość ponownego uruchomienia rozgrywki po przegranej (Game Over) bez konieczności restartu aplikacji.

## 🌐 Instrukcja Uruchomienia
1. Sklonuj repozytorium na dysk:
   ```bash
 git clone https://github.com/kotulski/SnakeGame.git
   ```

2. Otwórz plik rozwiązania SnakeGame.sln w środowisku Visual Studio.

3. Skompiluj projekt i uruchom grę przyciśnięciem Start lub klawiszem F5.

## 💡 Współpraca i Git Workflow
Projekt został zrealizowany z wykorzystaniem systemu gałęzi (Branching Strategy) w celu symulacji podziału zadań w zespole:

main: Główna, stabilna wersja kodu.

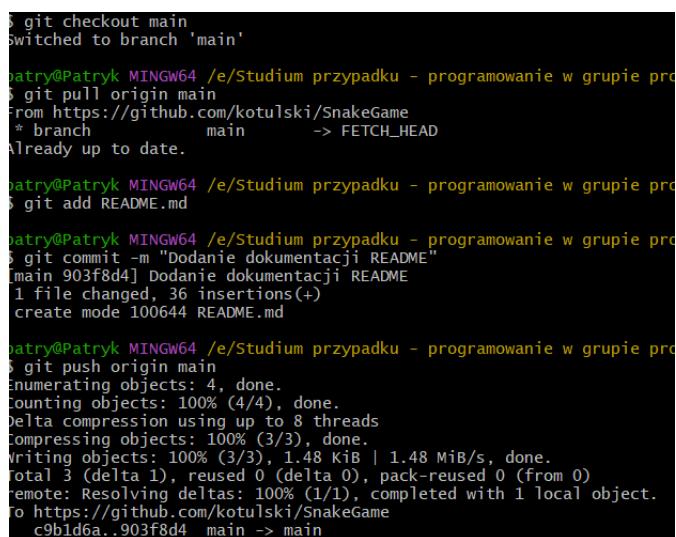
fix-classes: Gałąź naprawcza - uzupełnienie brakujących właściwości w klasach Obstakel i Pixel oraz refaktoryzacja nazewnictwa.

fix-game-logic: Gałąź implementacyjna - naprawa głównej pętli gry, logiki poruszania się i renderowania.

feature-multiplayer: Gałąź rozwojowa - dodanie ekranu symulującego łączenie z serwerem.

Każda zmiana była wprowadzana poprzez Pull Request, poddawana weryfikacji i scalana z gałęzią główną.

Wiersz 4, kolumna 1 | 2 051 znaków | Mi Składnia językowa Markdown | 100% | Windows (CRLF) | UTF-8
```



```
git checkout main
switched to branch 'main'
[Patryk@Patryk MINGW64 /e/Studium przypadku - programowanie w grupie pro]
$ git pull origin main
From https://github.com/kotulski/SnakeGame
 * branch            main      -> FETCH_HEAD
Already up to date.

[Patryk@Patryk MINGW64 /e/Studium przypadku - programowanie w grupie pro]
$ git add README.md

[Patryk@Patryk MINGW64 /e/Studium przypadku - programowanie w grupie pro]
$ git commit -m "Dodanie dokumentacji README"
[main 903f8d4] Dodanie dokumentacji README
 1 file changed, 36 insertions(+)
 create mode 100644 README.md

[Patryk@Patryk MINGW64 /e/Studium przypadku - programowanie w grupie pro]
$ git push origin main
Enumerating objects: 4, done.
Counting objects: 100% (4/4), done.
Delta compression using up to 8 threads
Compressing objects: 100% (3/3), done.
Writing objects: 100% (3/3), 1.48 KiB | 1.48 MiB/s, done.
Total 3 (delta 1), reused 0 (delta 0), pack-reused 0 (from 0)
remote: Resolving deltas: 100% (1/1), completed with 1 local object.
To https://github.com/kotulski/SnakeGame
 c9b1d6a..903f8d4  main -> main
```