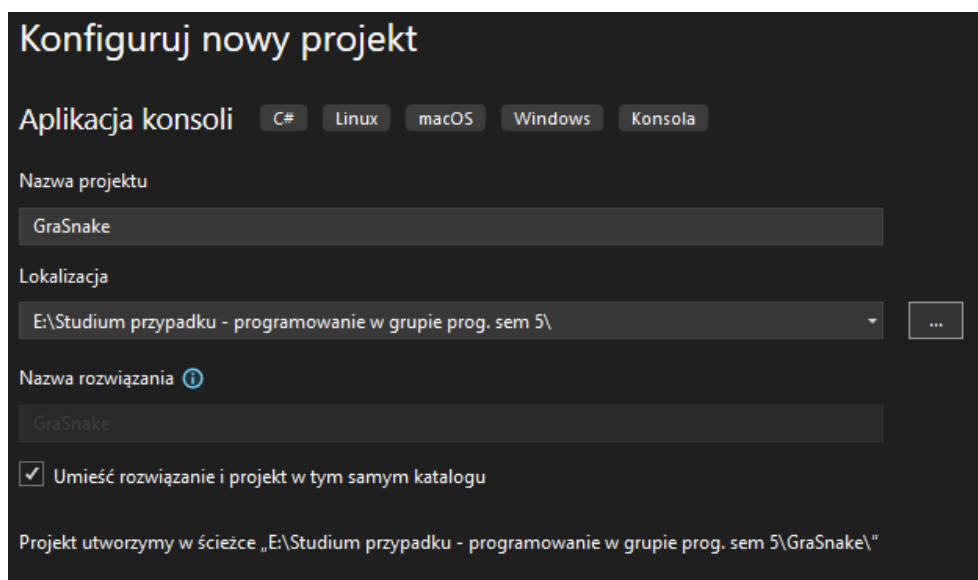
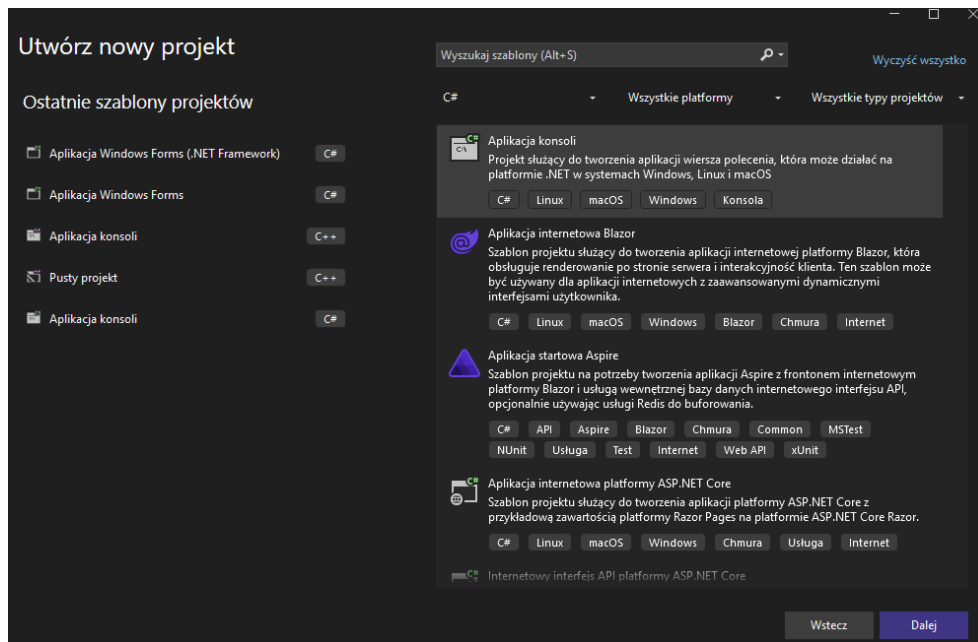


# Sprawozdanie i realizacja projektu gry Snake

## 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: Obstacle.cs oraz Pixel.cs.



```
Program.cs x
GraSnake Program
1 using System;
2 using System.Collections.Generic;
3 using System.Linq;
4 using System.Threading;
5
6 1 odwołanie
7 class Program
8 {
9
10 Odwołania: 0
11 static void Main()
12 {
13
14     Console.WindowHeight = 16;
15
16     Console.WindowWidth = 32;
17
18     int screenwidth = Console.WindowWidth;
19
20     int screenheight = Console.WindowHeight;
21
22     Random randomnummer = new Random();
23
24     pixel hoofd = new pixel();
25
26     hoofd.xpos = screenwidth / 2;
27
28     hoofd.ypos = screenheight / 2;
29
30     hoofd.schermkleur = ConsoleColor.Red;
31
32     string movement = "RIGHT";
33
34     List<int> telje = new List<int>();
35
36     int score = 0;
37
```

GraSnake

Dysk (E:) > Studium przypadku - programowanie w grupie prog. sem 5 > GraSnake >

Nowy

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 \*

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](#)

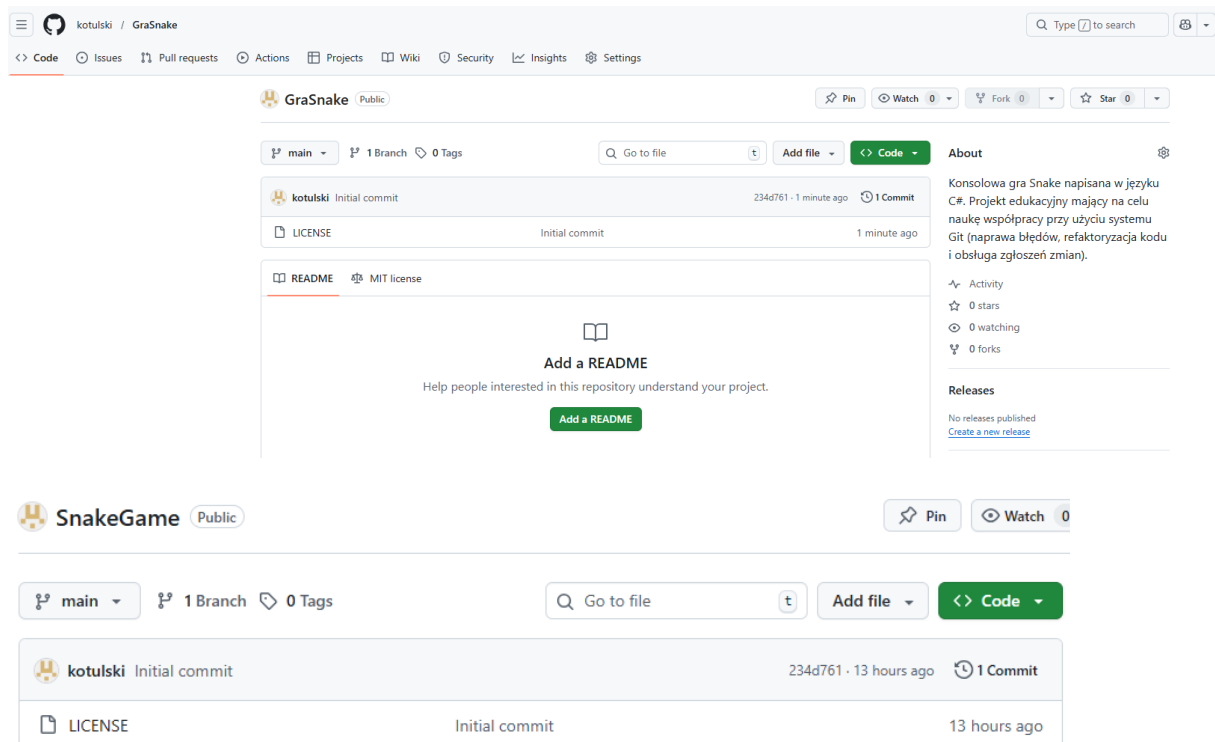
No .gitignore ▾

Add license

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

 MIT License ▾

Create repository



Za pomocą edytora tekstowego 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

patry@Patryk MINGW64 /e:/Studium przypadku - programowanie w grupie prog. sem 5/GraSnake
$ git config --global user.name "Patryk"

patry@Patryk MINGW64 /e:/Studium przypadku - programowanie w grupie prog. sem 5/GraSnake
$ git config --global user.email "patrykos138@gmail.com"

patry@Patryk MINGW64 /e:/Studium przypadku - programowanie w grupie prog. sem 5/GraSnake
$ git init
Initialized empty Git repository in E:/Studium przypadku - programowanie w grupie prog. sem 5/GraSnake/.git/

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

patry@Patryk MINGW64 /e:/Studium przypadku - programowanie w grupie prog. sem 5/GraSnake (main)
$ 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

patry@Patryk MINGW64 /e:/Studium przypadku - programowanie w grupie prog. sem 5/GraSnake (main)
$ touch .gitignore
```

MINGW64:/e/Studium przypadku - programowanie w grupie prog. sem 5/GraSnake

```
.vs/  
bin/  
obj/  
*.user
```

.gitignore [unix] (12:31 26/01/2026)

".gitignore" [unix] 4L, 22B written

```
patry@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
```

```
patry@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.sln  
create mode 100644 Obstacle.cs  
create mode 100644 Pixel.cs  
create mode 100644 Program.cs
```

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

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

```
patry@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.
```

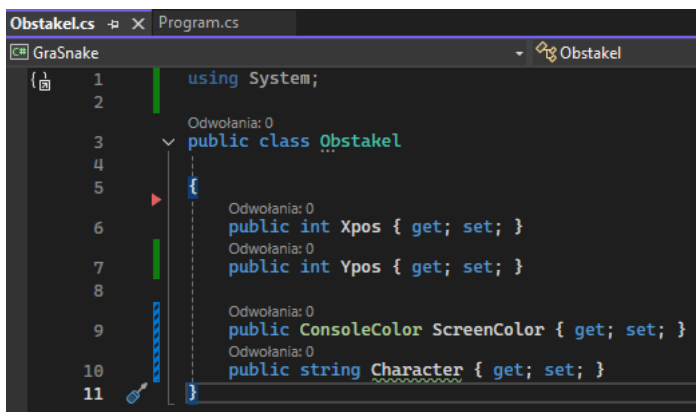
```
patry@Patryk MINGW64 /e/Studium przypadku - programowanie w grupie prog. sem 5/GraSnake (main)  
$ git push -f origin main  
Enumerating objects: 8, done.  
Counting objects: 100% (8/8), done.  
Delta compression using up to 8 threads  
Compressing objects: 100% (7/7), done.  
Writing objects: 100% (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 Obstacle, 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.



```
using System;

Odwołania: 0
public class Obstacle
{
    Odwołania: 0
    public int Xpos { get; set; }
    Odwołania: 0
    public int Ypos { get; set; }

    Odwołania: 0
    public ConsoleColor ScreenColor { get; set; }
    Odwołania: 0
    public string Character { get; set; }
}
```

```
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:   Obstacle.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 Obstacle, poprawa nazewnictwa zmiennych"
[fix-classes ad5449c] Fix: Dodanie brakującej właściwości Ypos do klasy Obstacle, 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 wcześniej 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'

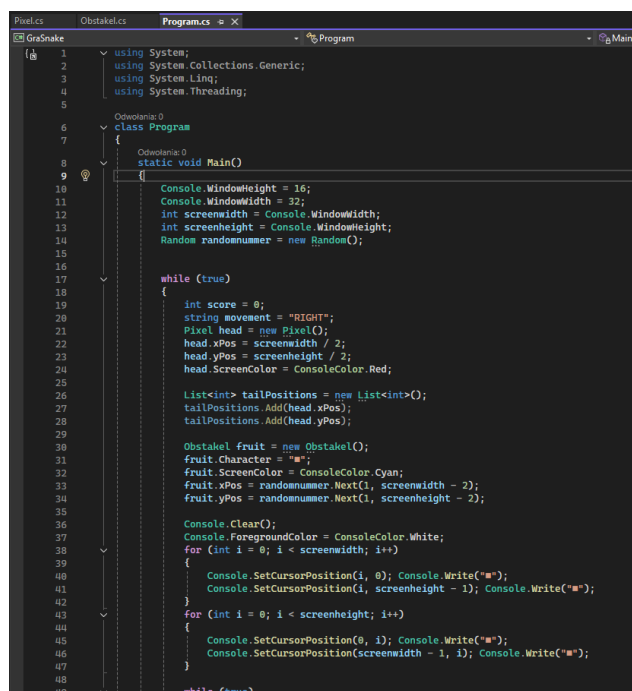
patry@Patryk MINGW64 /e/Studium przypadku - programowanie w grupie prog. sem 5/GraSnake (main)
$ git pull origin main
From https://github.com/kotulski/SnakeGame
* branch      main      -> FETCH_HEAD
Already up to date.

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(-)

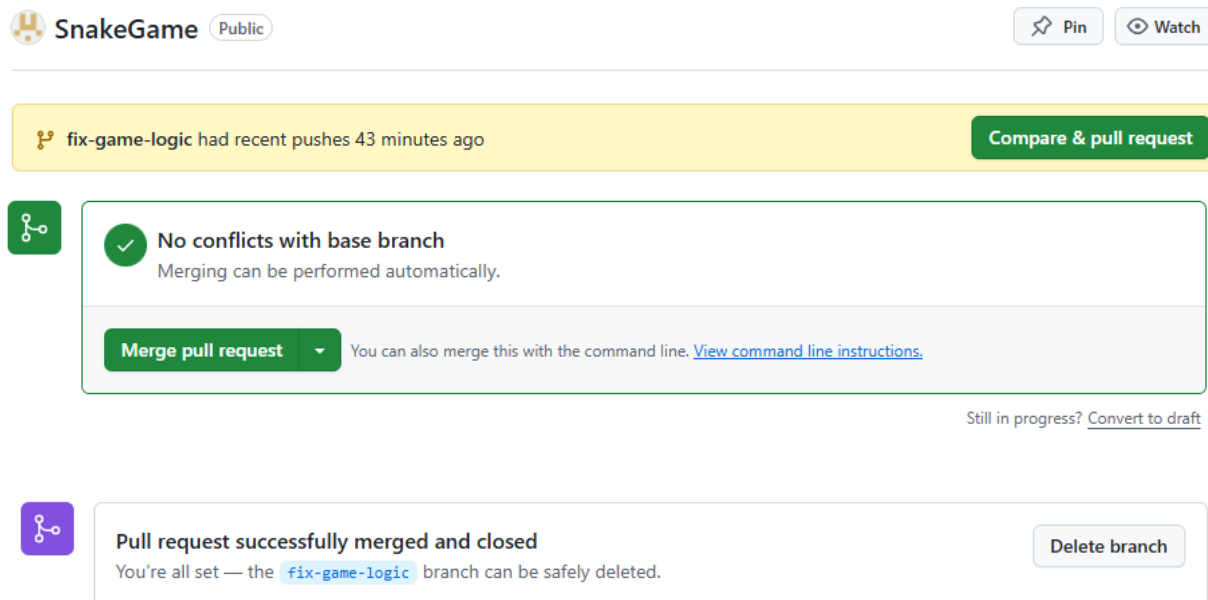
patry@Patryk MINGW64 /e/Studium przypadku - programowanie w grupie prog. sem 5/GraSnake (fix-game-logic)
$ git push -u origin fix-game-logic
Enumerating objects: 9, done.
Counting objects: 100% (9/9), done.
Delta compression using up to 8 threads
Compressing objects: 100% (5/5), done.
Writing objects: 100% (5/5), 1.68 KiB | 1.68 MiB/s, done.
Total 5 (delta 1), 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'.
```



```
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 randomnumber = new Random();
15
16
17         while (true)
18         {
19             int score = 0;
20             string movement = "RIGHT";
21             Pixel head = new Pixel();
22             head.xPos = screenWidth / 2;
23             head.yPos = screenHeight / 2;
24             head.ScreenColor = ConsoleColor.Red;
25
26             List<int> tailPositions = new List<int>();
27             tailPositions.Add(head.xPos);
28             tailPositions.Add(head.yPos);
29
30             Obstakel fruit = new Obstakel();
31             fruit.Character = "M";
32             fruit.ScreenColor = ConsoleColor.Cyan;
33             fruit.xPos = randomnumber.Next(1, screenWidth - 2);
34             fruit.yPos = randomnumber.Next(1, screenHeight - 2);
35
36             Console.Clear();
37             Console.ForegroundColor = ConsoleColor.White;
38             for (int i = 0; i < screenWidth; i++)
39             {
40                 Console.SetCursorPosition(i, 0); Console.Write("■");
41                 Console.SetCursorPosition(i, screenHeight - 1); Console.Write("■");
42             }
43             for (int i = 0; i < screenHeight; i++)
44             {
45                 Console.SetCursorPosition(0, i); Console.Write("■");
46                 Console.SetCursorPosition(screenWidth - 1, i); Console.Write("■");
47             }
48
49             while (true)
```



## 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.




**SnakeGame** Public

fix-game-logic had recent pushes 43 minutes ago [Compare & pull request](#)

  **No conflicts with base branch**  
Merging can be performed automatically.

[Merge pull request](#) You can also merge this with the command line. [View command line instructions.](#)

Still in progress? [Convert to draft](#)

 **Pull request successfully merged and closed** [Delete branch](#)  
You're all set — the `fix-game-logic` branch can be safely deleted.

## 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
 Obstacle.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();
//
```

```
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.

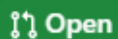
patry@Patryk MINGW64 /e/Studium przypadku - programowanie w grupie prog. sem 5/GraSnake (main)
$ git checkout -b feature-multiplayer
Switched to a new branch 'feature-multiplayer'

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

patry@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(+)

patry@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



Open

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



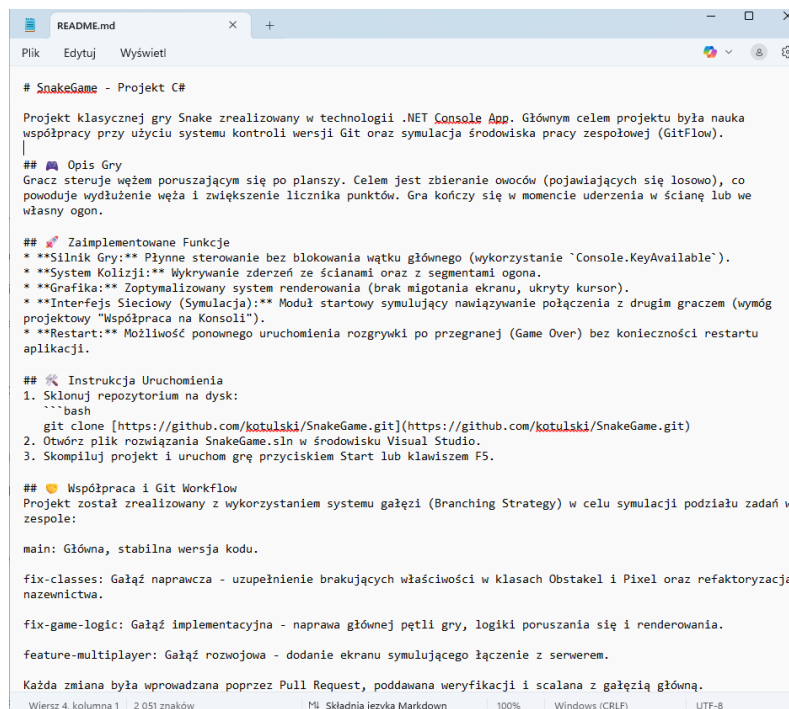
## Krok 9: Rozwiązanie problemów (Issues)

Zarządzanie błędami i nowymi funkcjami odbywało się poprzez systematyczne tworzenie dedykowanych gałęzi dla każdego zagadnienia. Problemy takie jak błędy kompilacji czy brakujące funkcjonalności były traktowane jako osobne zadania, dla których tworzone 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.



```
# 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
* **Silnik Gry:** Płynne sterowanie bez blokowania wątku głównego (wykorzystanie `Console.KeyAvailable`).
* **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](https://github.com/kotulski/SnakeGame.git)
2. Otwórz plik rozwiązania SnakeGame.sln w środowisku Visual Studio.
3. Skompiluj projekt i uruchom grę przyciskiem 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.

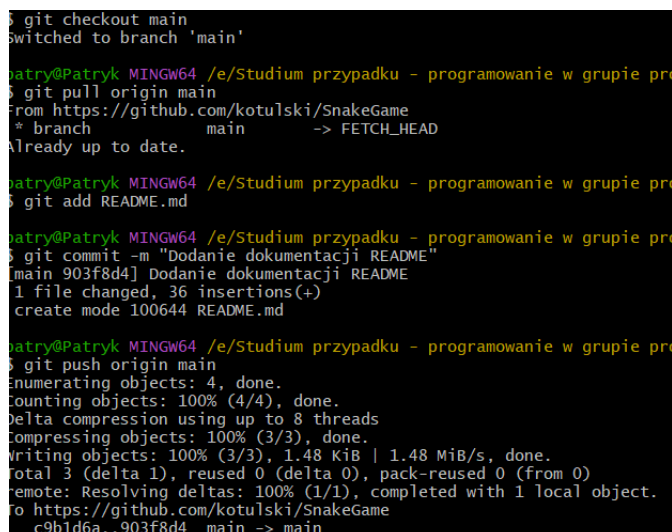
fix-classes: Gałąź naprawcza - uzupełnienie brakujących właściwości w klasach Obstacle i Pixel oraz refaktoryzacja nazewnictwa.

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

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

Každą zmianę była wprowadzana poprzez Pull Request, poddawana weryfikacji i scalana z gałęzią główną.
```


```



```
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
```