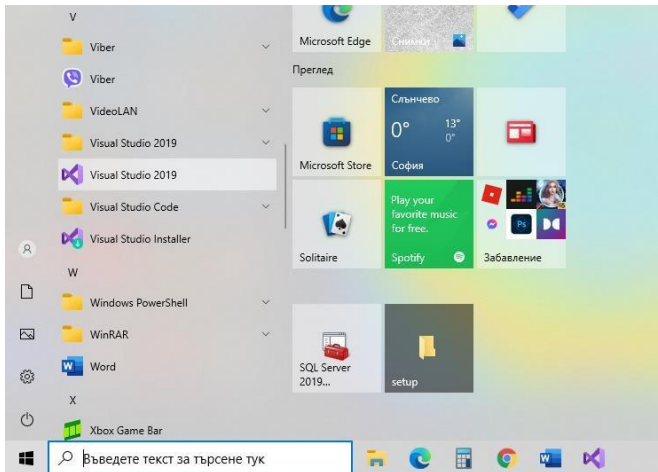


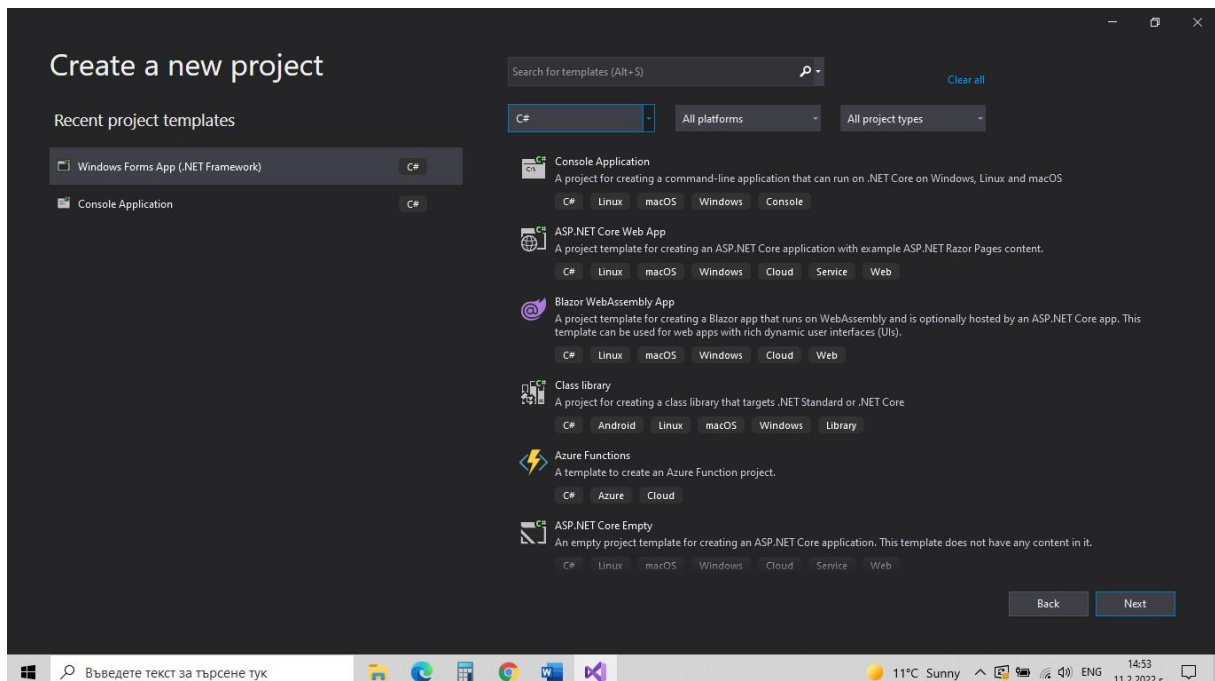
СЪЗДАВАНЕ НА ИГРА FLAPPY BIRD GAME С WINDOWS FORMS APP И C#

Стъпка 1: създаване на нов проект

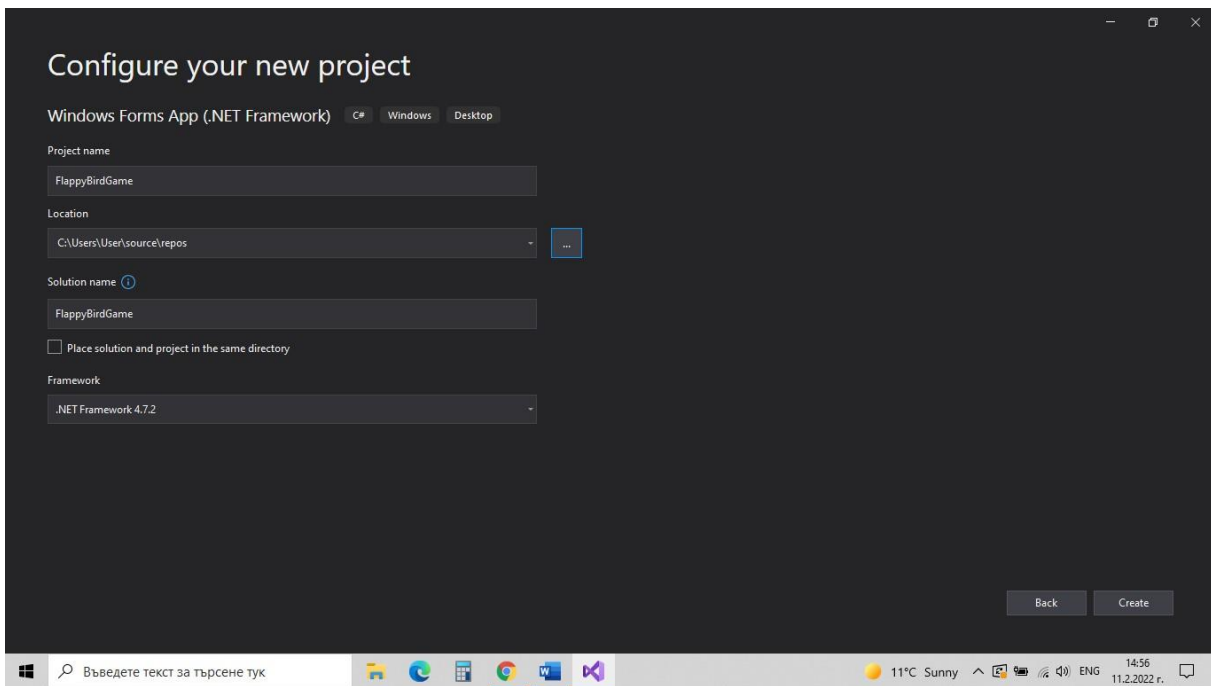
Стартираме **Visual Studio 2019**.



Избираме **Create a new project, Windows Forms App (.NET Framework) и C#**

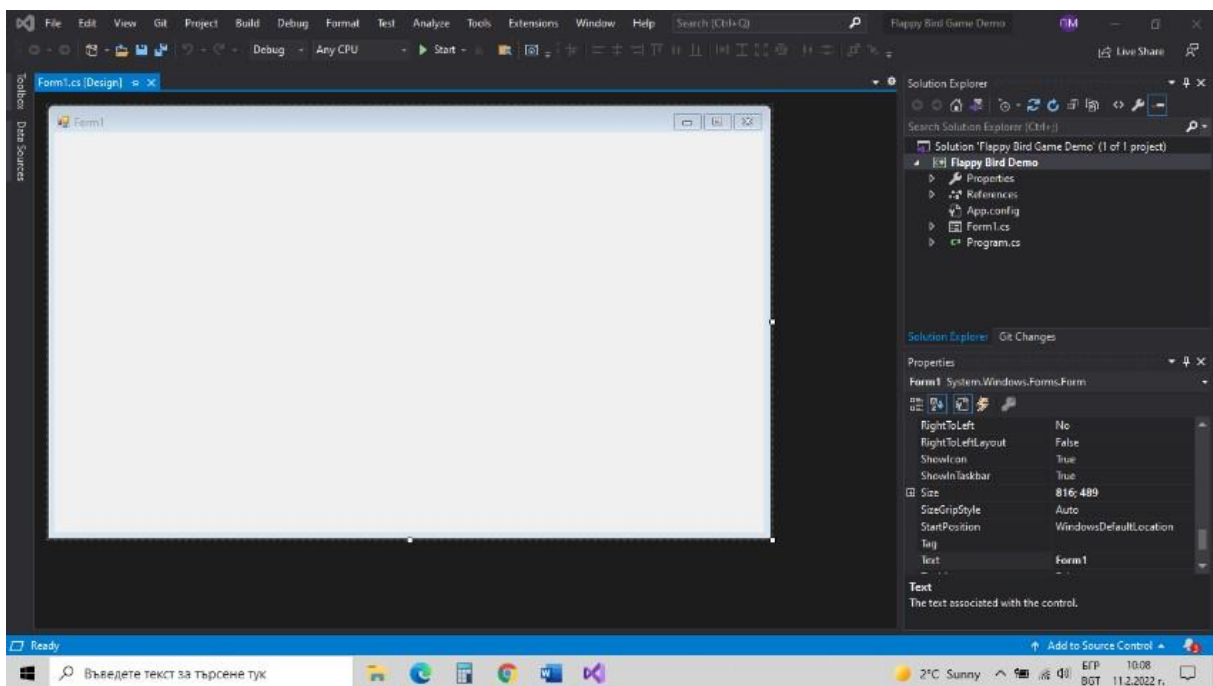


Пишем име на проекта в **Project name** и **Solution name**. Променяме мястото на съхранение на проекта, ако желаем.



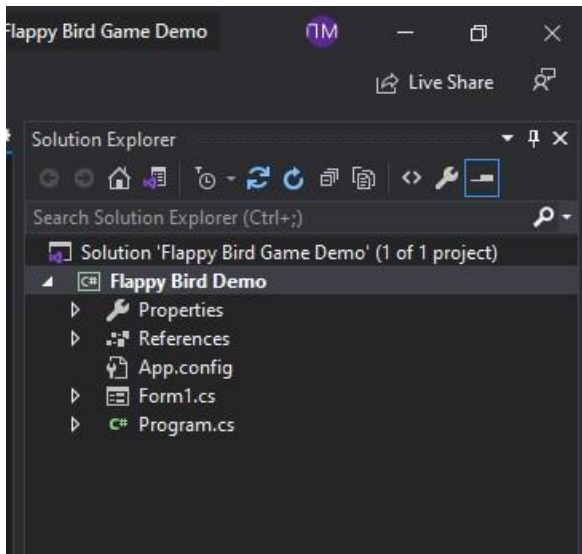
Какво имаме на основния екран?

- **Form1**



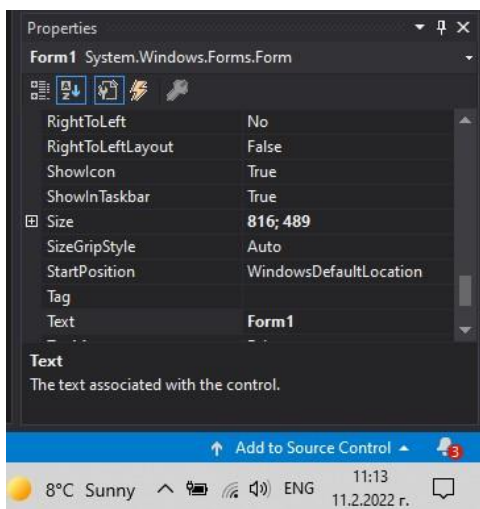
Form1 ще представлява полето на нашата игра. Form1 е вече създаден обект в нашия проект и тя има определени свойства, които можем да настроим и действия, които могат да се изпълнят.

- **Solution Explorer**



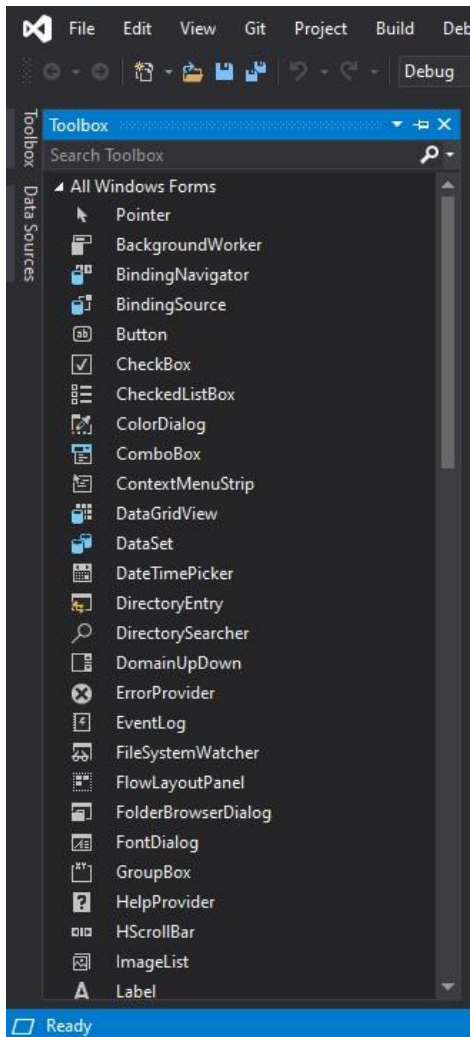
В дясната част имаме всички файлове в проекта, които са събрани в един общ сълюшън.

- **Git Changes**



Имаме два важни таба тук – properties и events. В първия таб можем да видим всички свойства, които са приложими за избрания обект и съответно да им зададем стойности. Във втория таб създаваме събития, при извикването на които ще се изпълняват дадени действия.

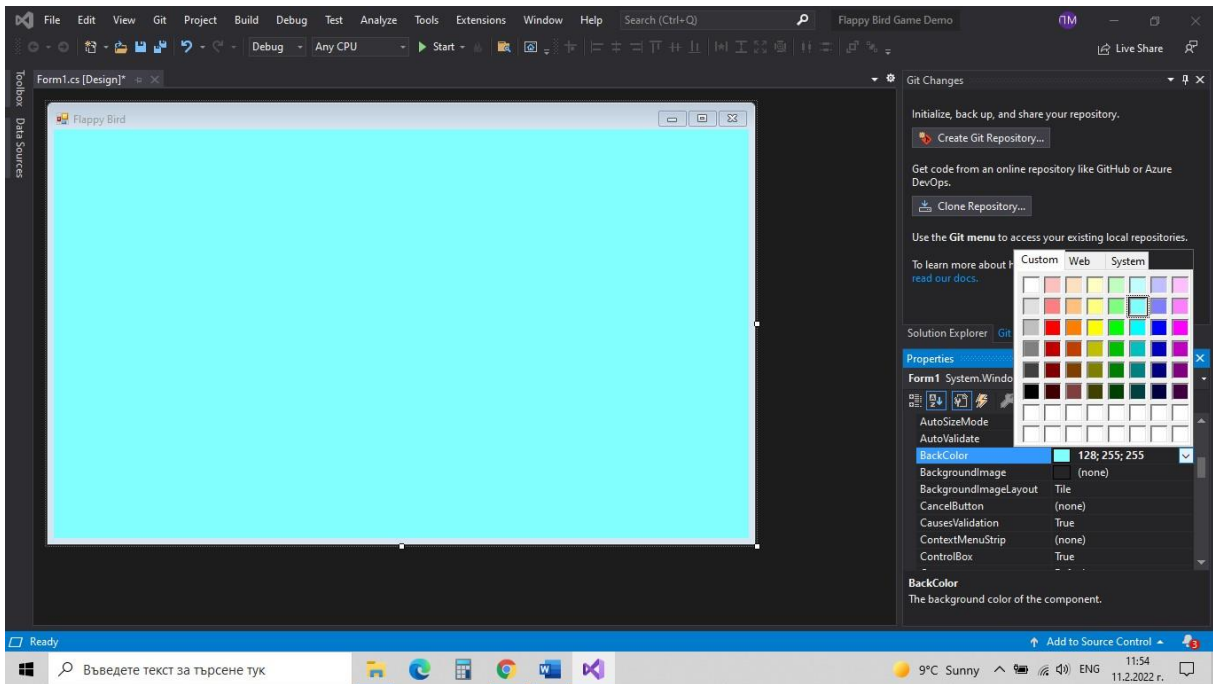
- **Toolbox**



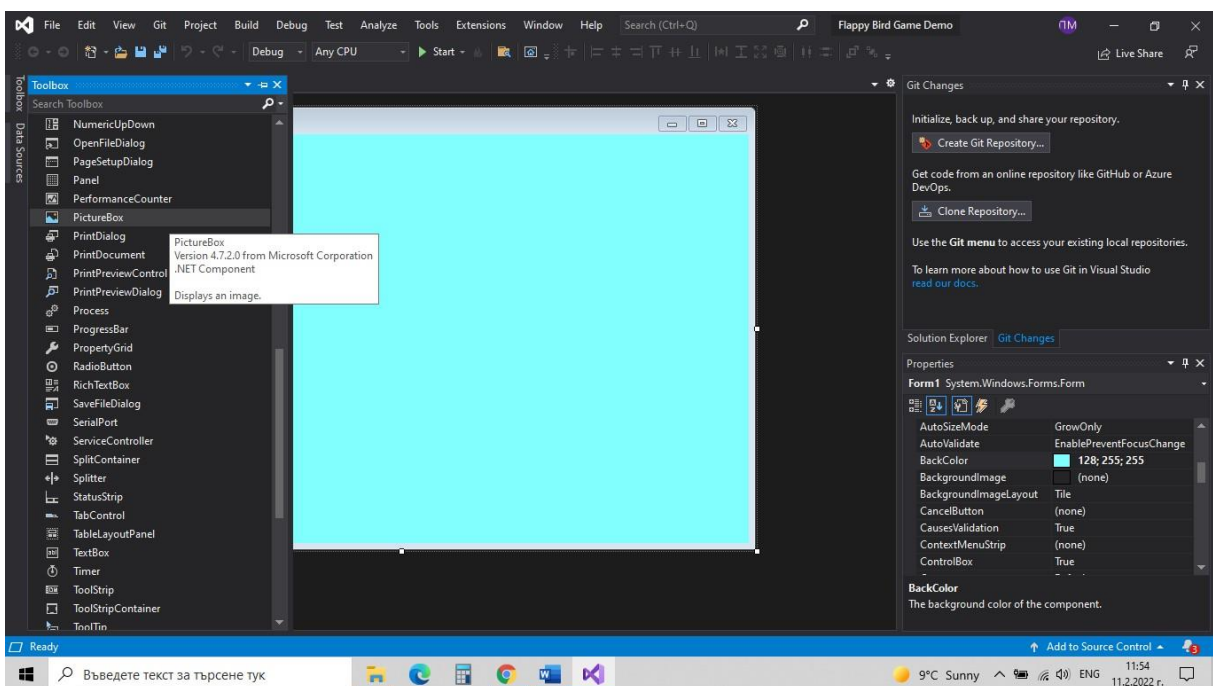
От тук избираме обекта, от който се нуждаем в нашия проект. Чрез влачене го прибавяме към работната област.

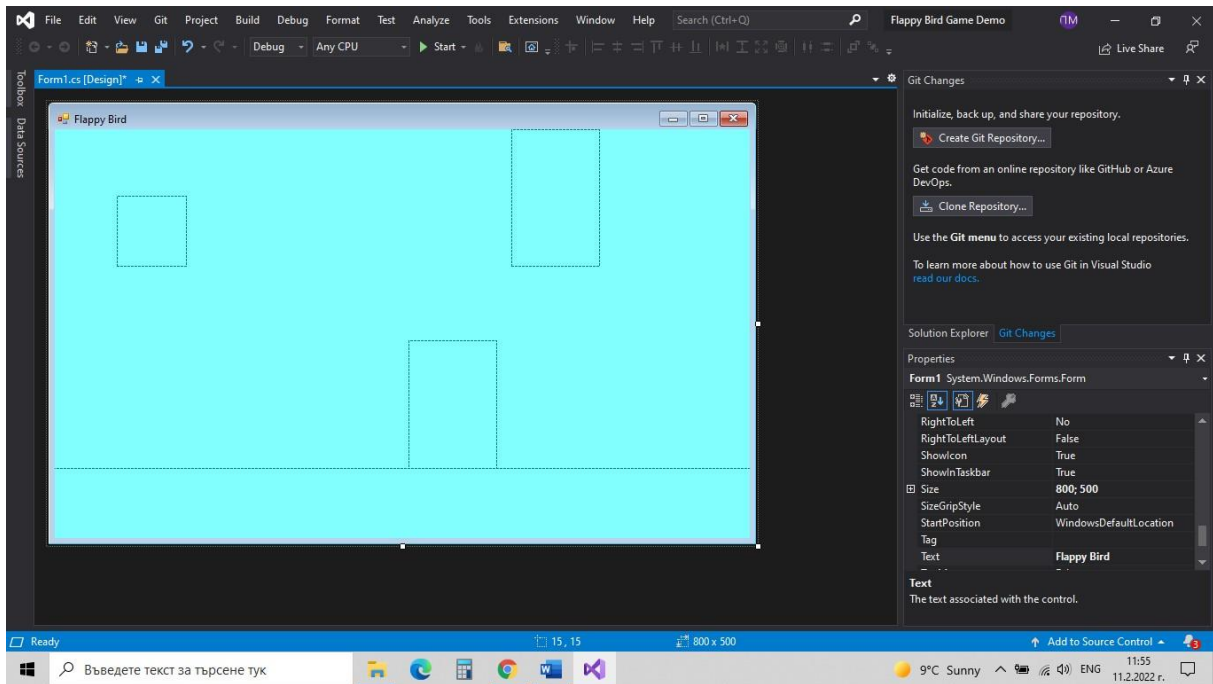
Стъпка 2: Задаване на свойства на Form1:

- (Name): gameField
- Size: 800;500
- Text: Flappy Bird
- BackColor: some kind of blue



Стъпка 3: Добавяне и настройване на четири обекта картинки - pictureBox

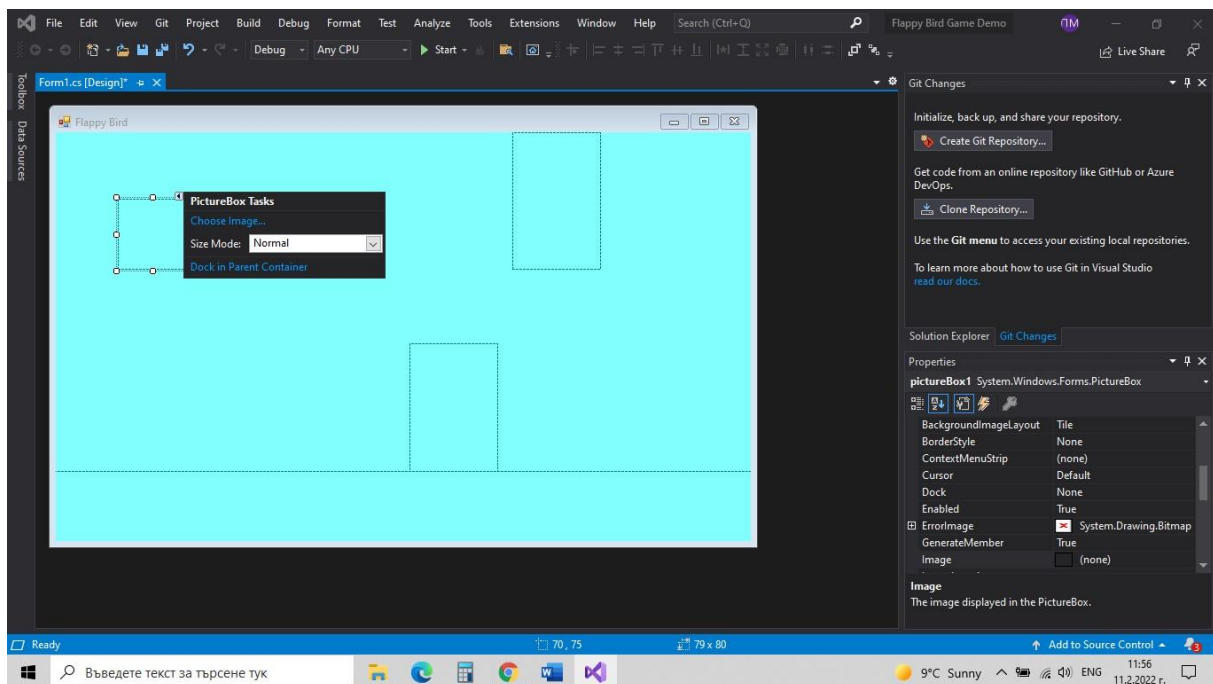


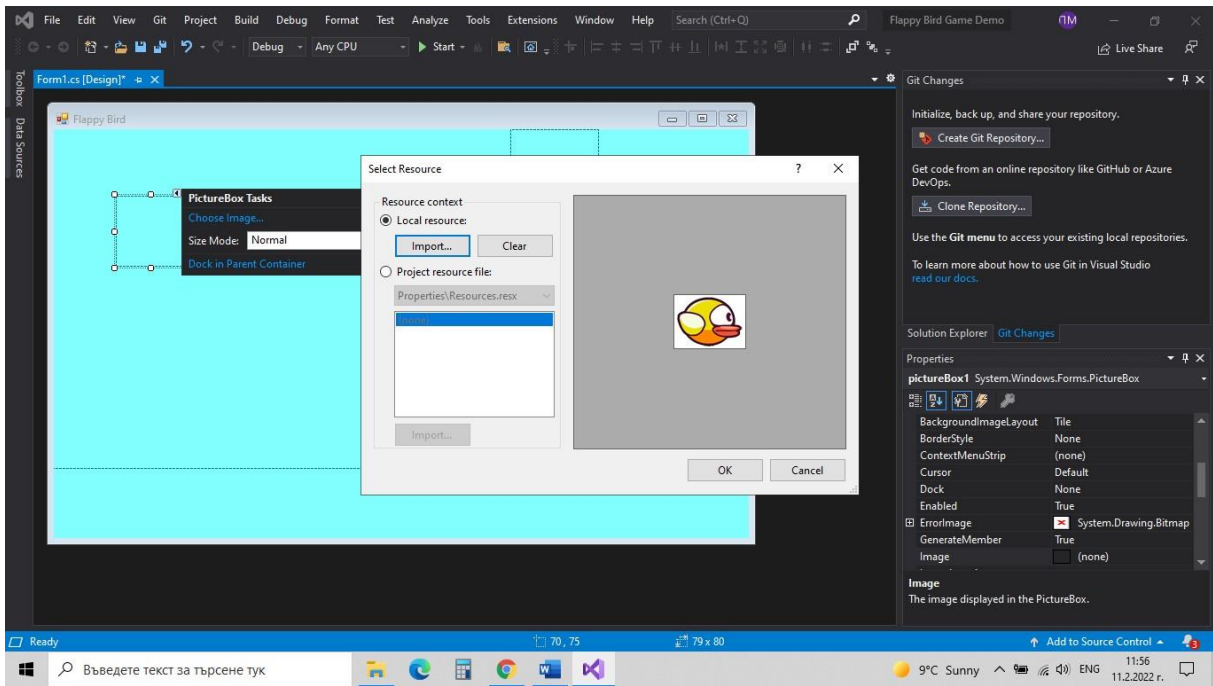


pictureBox1 ще бъде птицата, pictureBox2 ще бъде една тръба, pictureBox3 ще бъде другата тръба и pictureBox4 ще бъде земята.

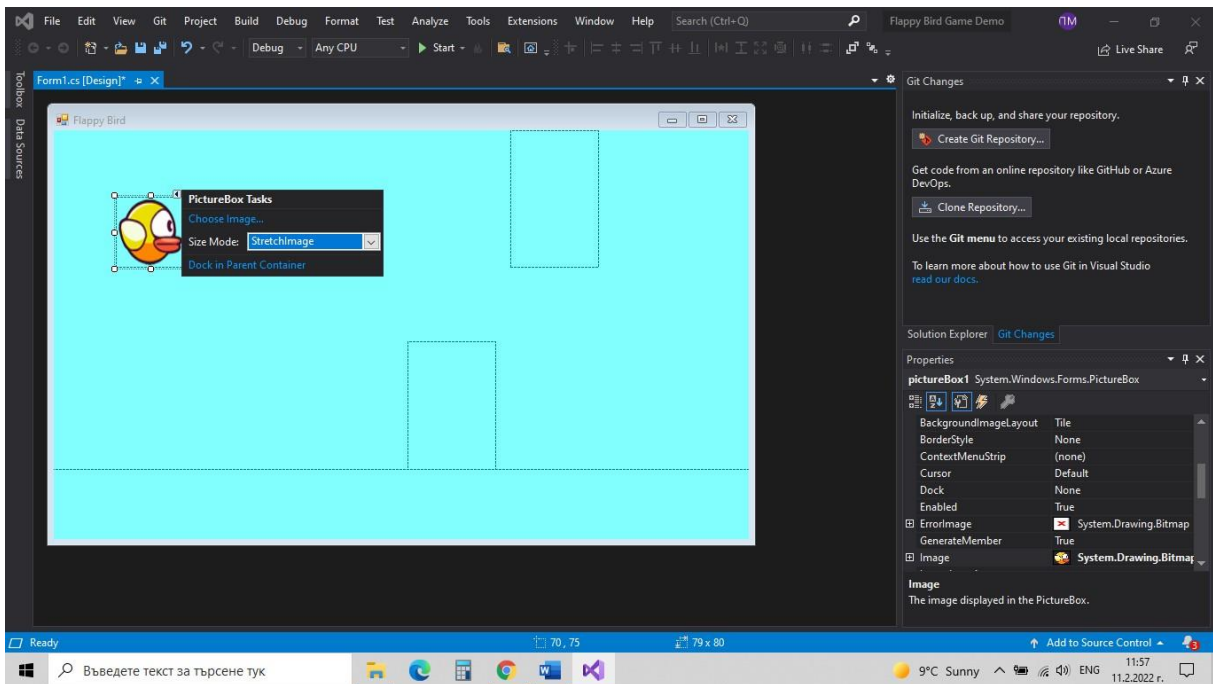
Настройки на pictureBox1:

- Choose image -> Local Resource -> Import...

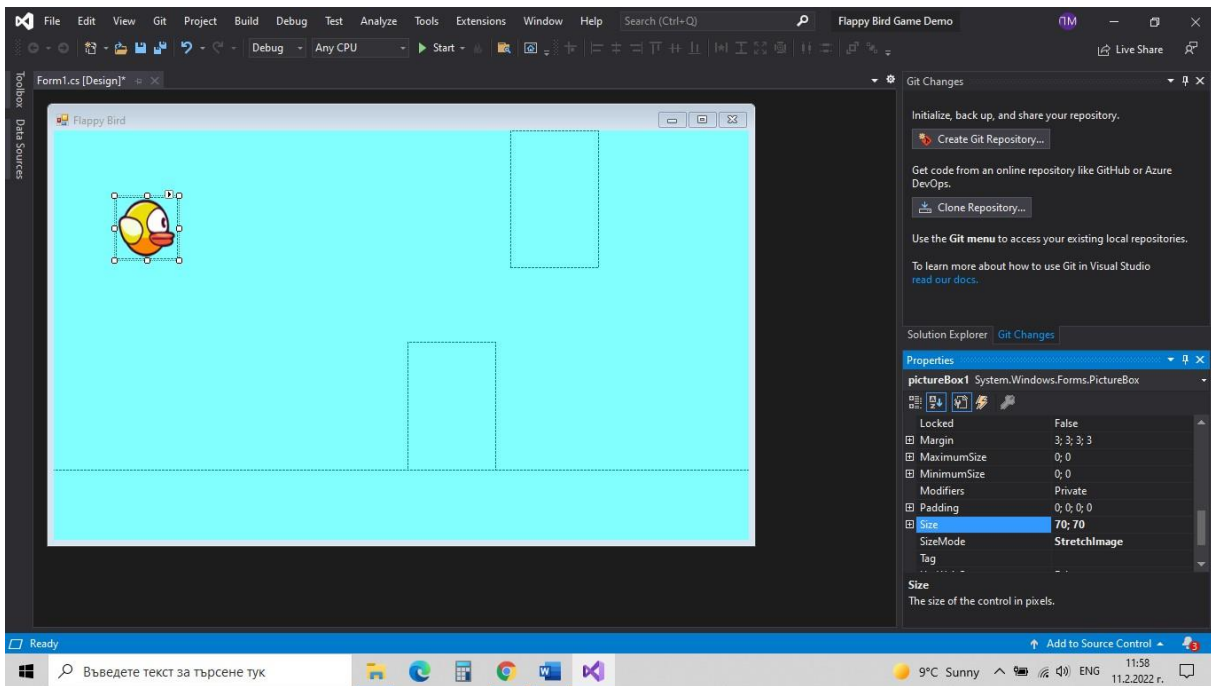




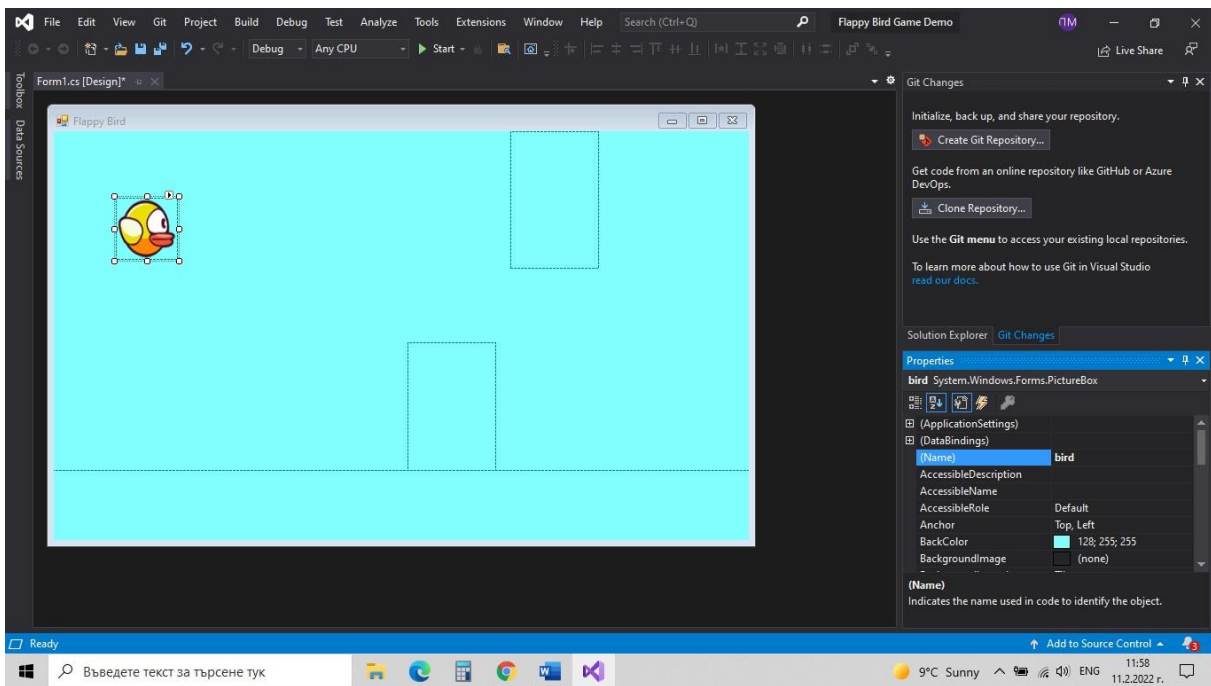
- Size Mode: StretchImage



- Size: 70;70



- (Name): bird



Правим настройки и на останалите картинки.

Настройки на pictureBox2:

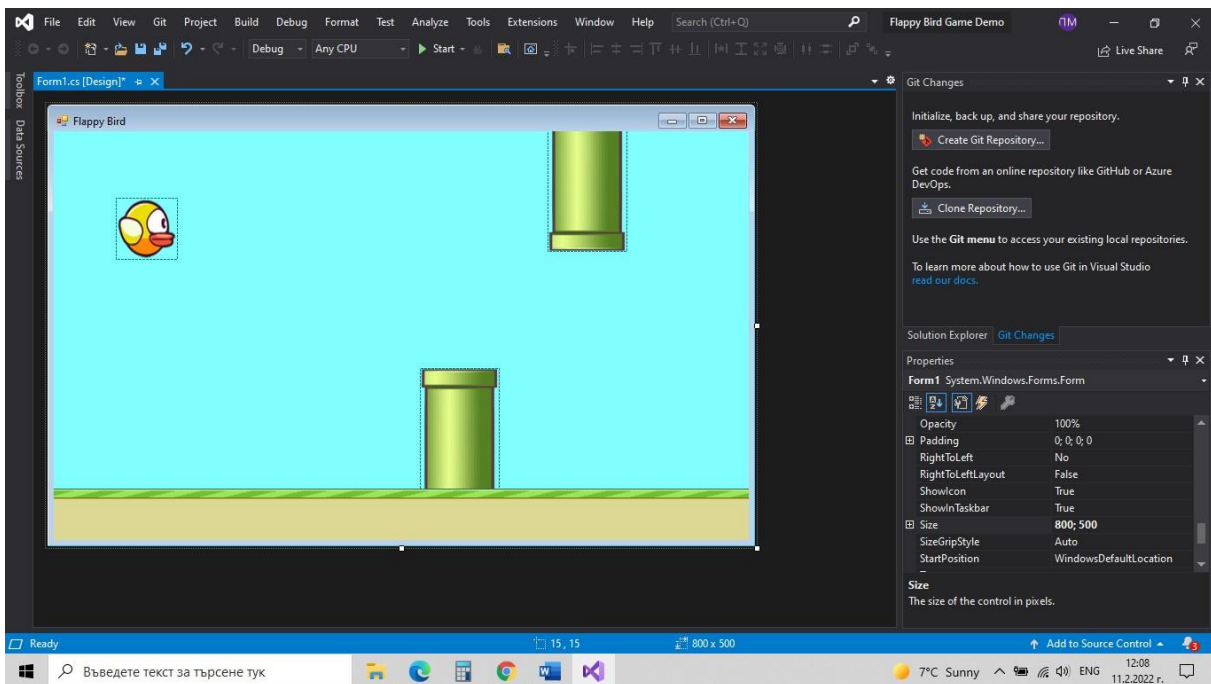
- (Name): pipeUp
- Size Mode: StretchImage
- Size: 90;140

Настройки на pictureBox3:

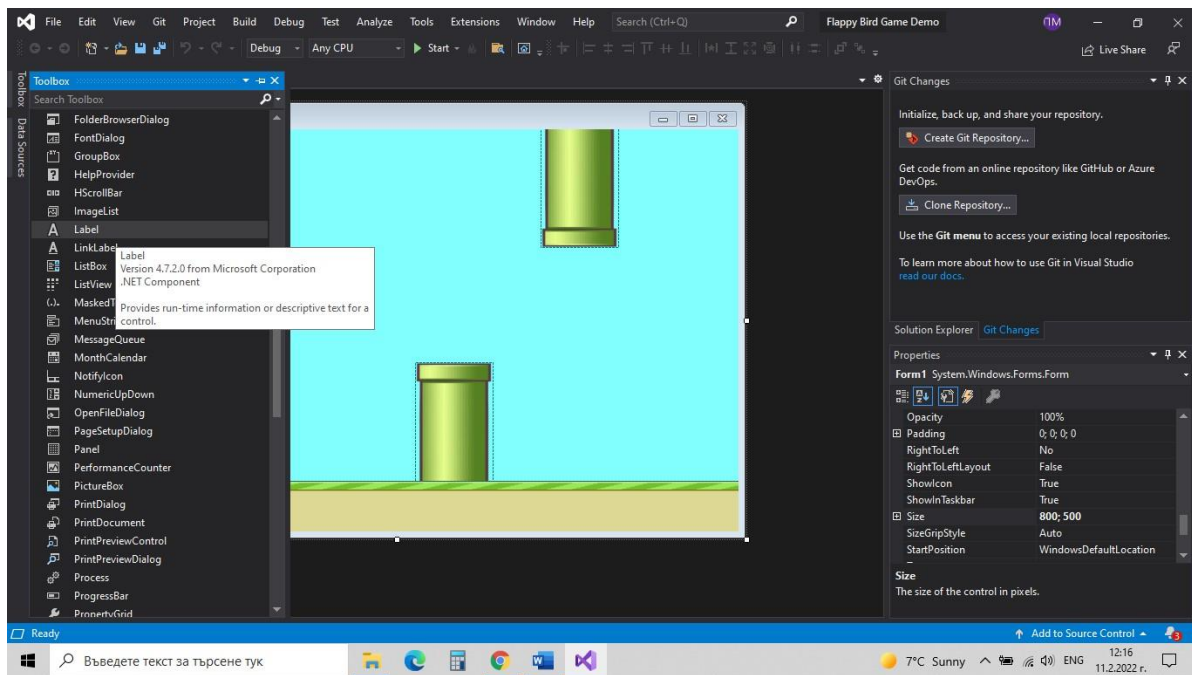
- (Name): pipeDown
- Size Mode: StretchImage
- Size: 90;140

Настройки на pictureBox4:

- (Name): ground
- Size Mode: StretchImage
- Size: 800;60

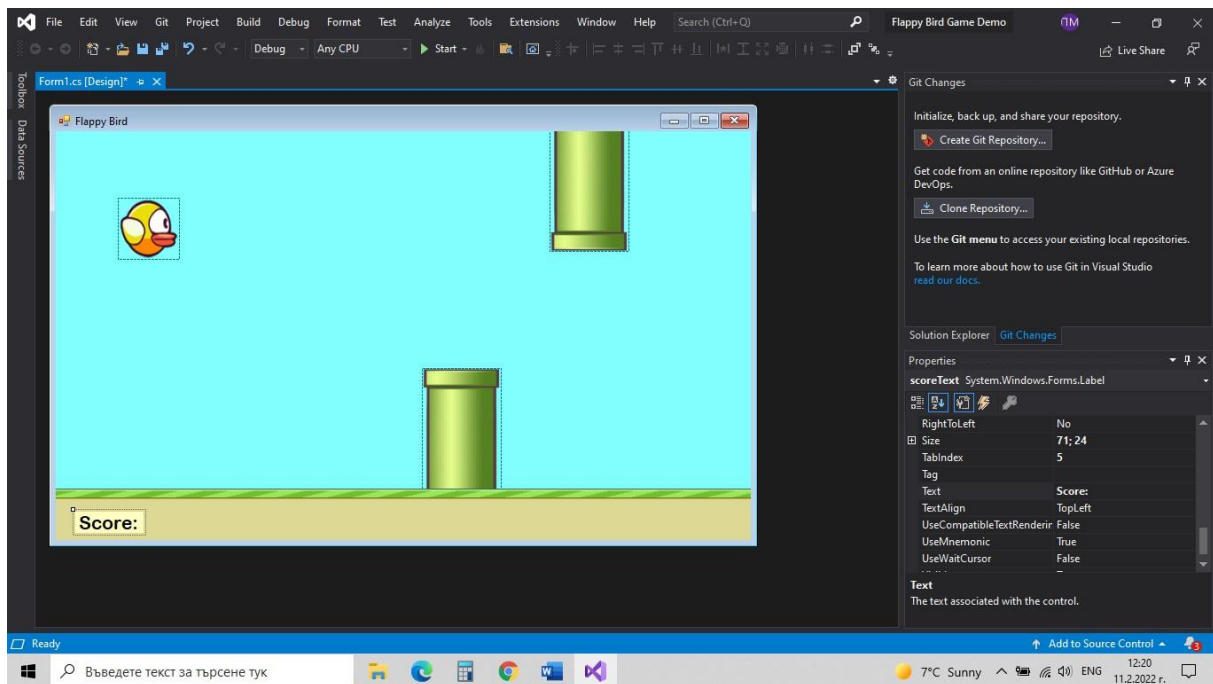


Стъпка 4: Добавяне и настройване на обект етикет - Label

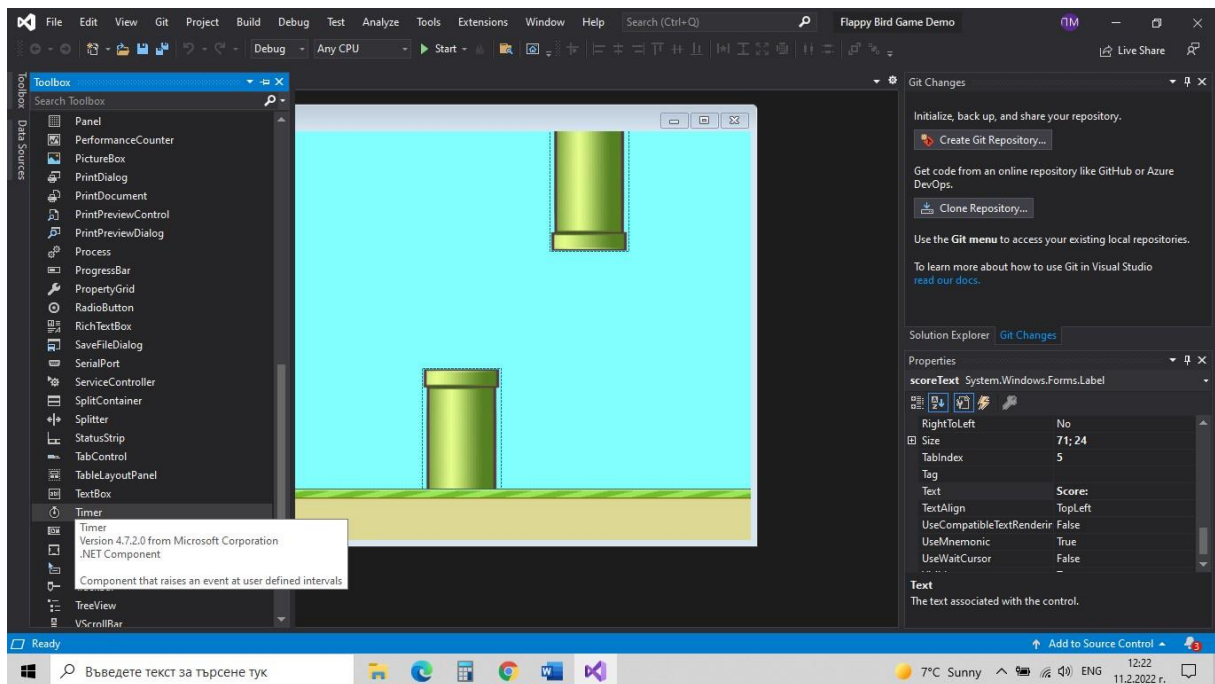


Свойства на label1:

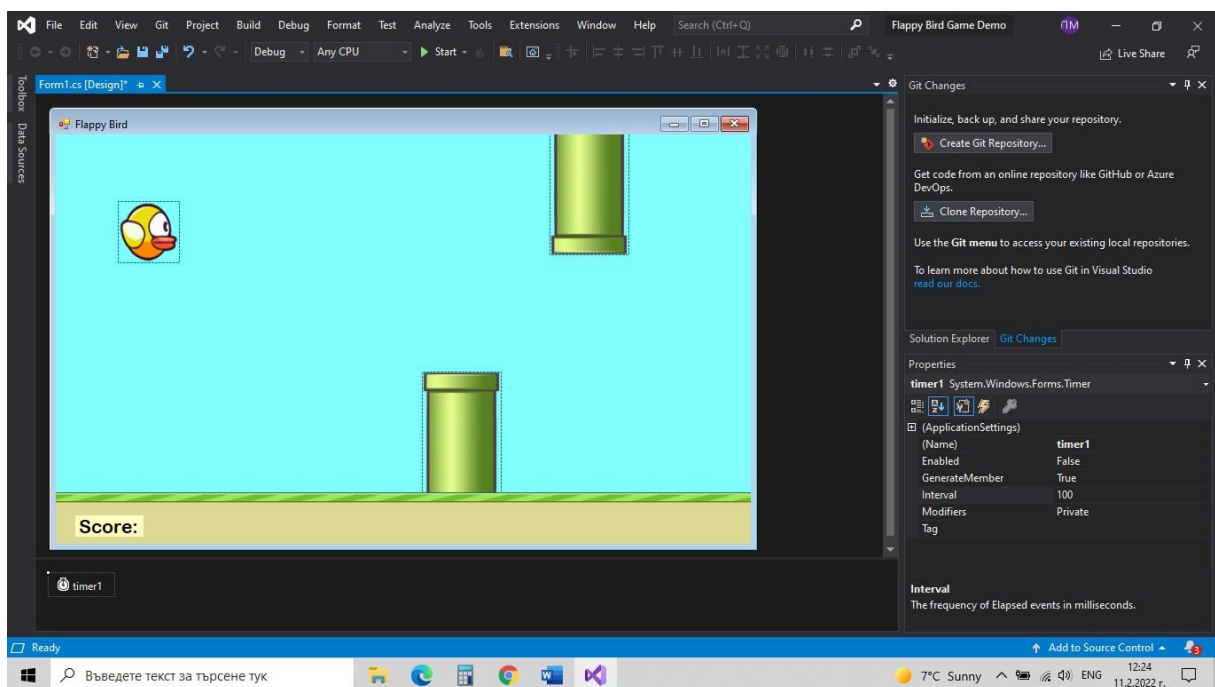
- (Name): scoreText
- BackColor: similar to the ground
- Font: Arial; 16
- Text: Score



Стъпка 5: Добавяне на елемент таймер - timer



Този елемент е невидим на игралното поле.



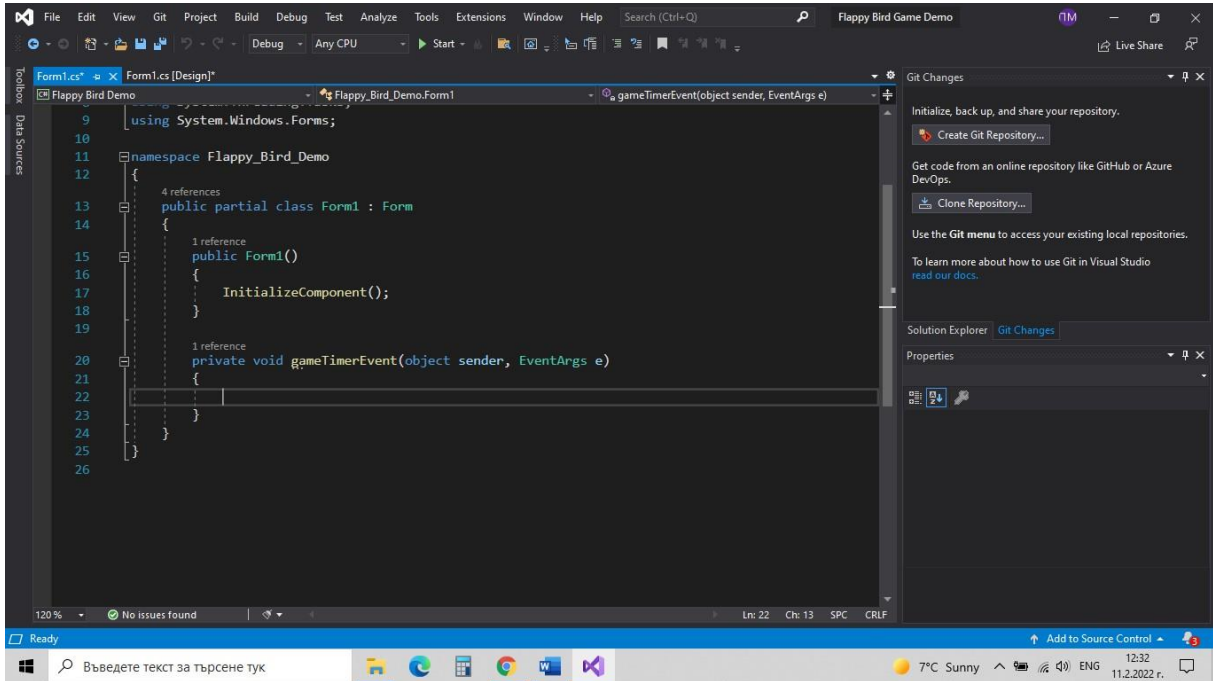
Свойства на timer1:

- (Name): gameTimer
- Enabled: true
- Interval: 20

И създаваме събитие на този елемент:

- Tick: gameTimerEvent

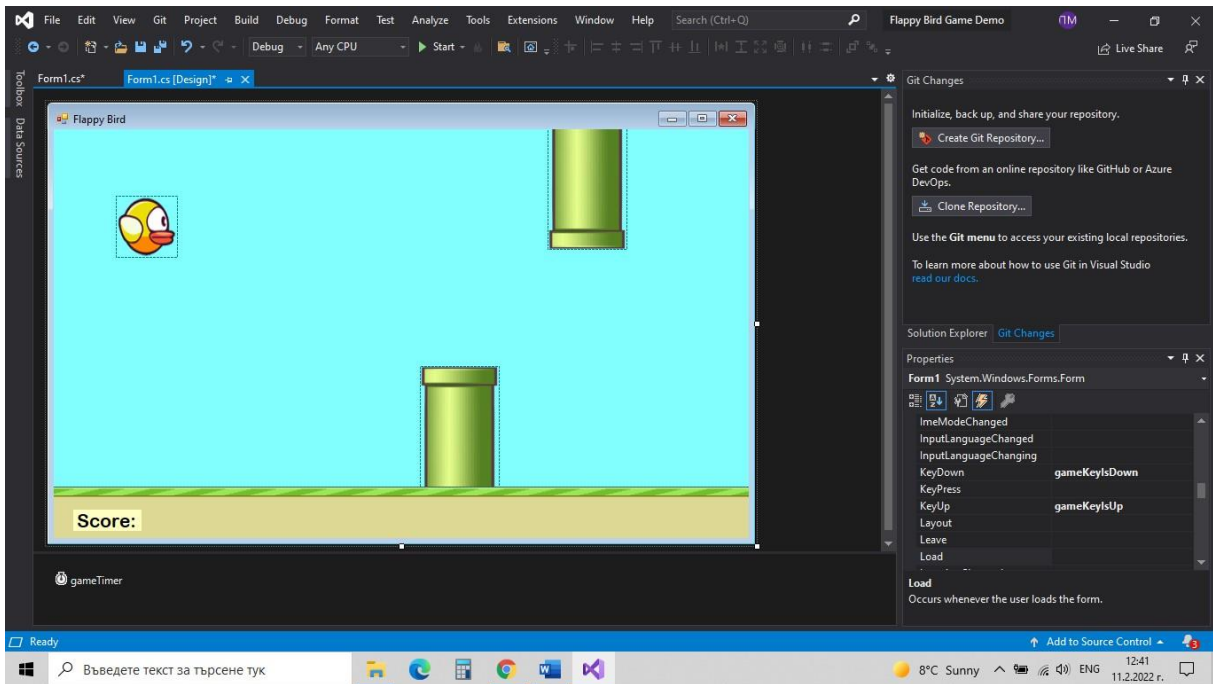
Кликването на създаденото събитие отваря кода на проекта, където вече има създаден метод за действие при извикване на събитието. Тук се програмират действия, които се случват непрекъснато.



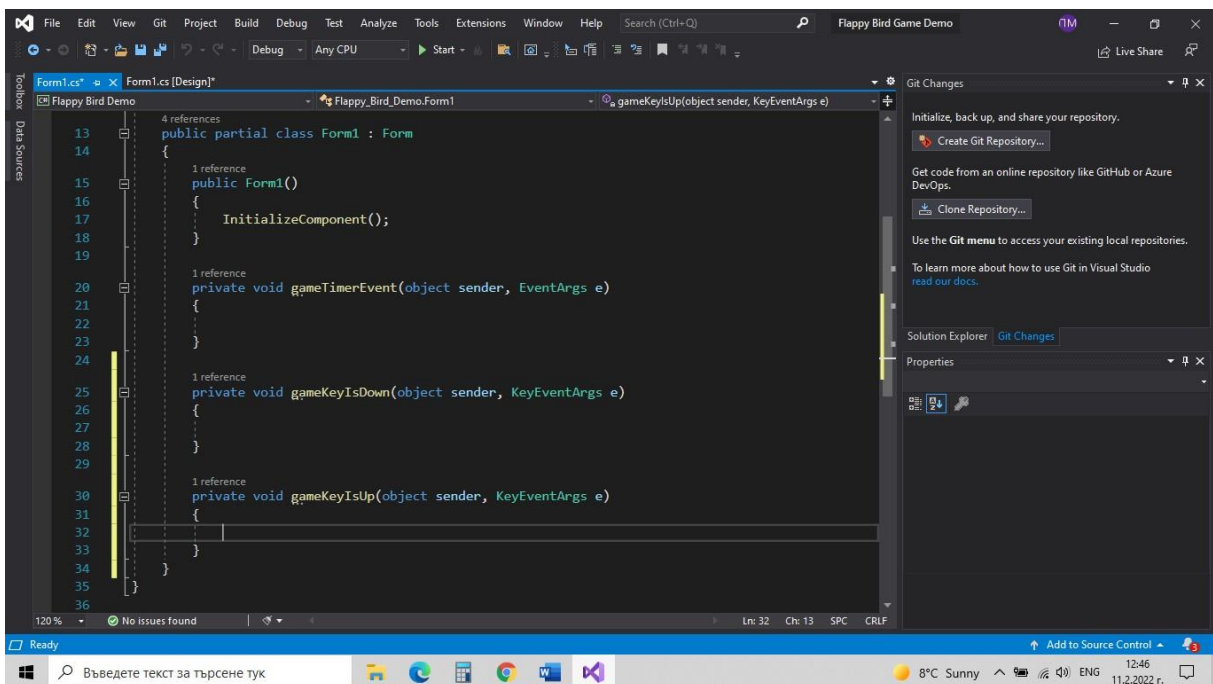
Стъпка 6: създаване на две събитие на основния елемент Form1:

Играта ще се играе с клавишните стрелки. При натискане на клавиш, птицата ще се издига нагоре. Ако клавиш не е натиснат, птицата ще пада. Затова се нуждаем от две събития:

- KeyDown: gameKeysIsDown
- KeyUp: gameKeysIsUp



Във файла Form1.cs започваме да програмираме нашата игра.



Стъпка 7: Създаване на променливи

В началото се нуждаем от три променливи – една за скоростта на тръбите, една за гравитацията, с която птицата ще пада надолу и една за резултата. И на трите променливи даваме начални стойности.

```
int pipeSpeed = 8;
int gravity = 5;
int score = 0;
```

Стъпка 8: движение на птицата надолу

За преместване на птицата надолу ще използваме нейното свойство `Top`. Увеличаването на стойността на това свойство ще увеличи разстоянието от горния край на обекта (картинката) до горния край на игралното поле. Т.е. ако непрекъснато увеличаваме това свойство със стойността на променливата за гравитация, птицата ще пада надолу. Тъй като това е действие, което се извършва непрекъснато, ще го опишем в метода `GameTimerEvent()`.

```
1 reference
private void gameTimerEvent(object sender, EventArgs e)
{
    bird.Top += gravity;
}
```

Стъпка 9: преместване на птицата нагоре при натиснат клавиш и надолу, когато не е натиснат клавиш

Ако намаляваме свойството `Top`, птицата ще се издига нагоре. Това може да стане като променим гравитацията да бъде отрицателно число. Това ще стане в метода `gameKeyIsDown()` и съответно в метода `gameKeyIsUp()` отново ще направим гравитацията положително число.

```
1 reference
private void gameKeyIsDown(object sender, KeyEventArgs e)
{
    gravity = -5;
}

1 reference
private void gameKeyIsUp(object sender, KeyEventArgs e)
{
    gravity = 5;
}
```

Стъпка 10: движение на тръбите

Тръбите са обекти с имена `pipeUp` и `pipeDown` и те също имат свойства. За да ги задвижим наляво ще използваме тяхното свойство `Left`. Ако намаляваме стойността на свойството, тръбите ще се движат наляво. Това действие трябва да бъде описано в метода `gameTimerEvent()`, защото то се случва непрекъснато.

```
1 reference
private void gameTimerEvent(object sender, EventArgs e)
{
    bird.Top += gravity;
    pipeDown.Left -= pipeSpeed;
    pipeUp.Left -= pipeSpeed;
}
```

Стъпка 11: появяване на тръбите отново в дясно

Движейки се наляво тръбите излизат от игралното поле. Трябва да направим проверка и когато тяхната лява страна започне да става по-малка от 0, трябва да и зададем висока стойност, за да отиде отново в дясно. Това действие се описва в метода `gameTimerEvent()`, защото то е повтарящо се.

```
if (pipeUp.Left < 0)
    pipeUp.Left = 900;
if (pipeDown.Left < 0)
    pipeDown.Left = 1300;
```

Стъпка 12: Край на играта ако птицата се блъсне

Сега ще създадем метод `endGame()`, където използвайки вградения метод `Stop()` ще спрем таймера.

```

0 references
private void endGame()
{
    gameTimer.Stop();
}

```

Нищо не се случва в момента, защото методът endGame() не е извикан. Трябва да го извикаме, ако има сблъсък между птицата и една от двете тръби или земята. За проверка за сблъсък ще използваме вградения метод Bounds.IntersectsWith(), който връща булев резултат.

```

if (bird.Bounds.IntersectsWith(pipeDown.Bounds))
    endGame();

if (bird.Bounds.IntersectsWith(pipeUp.Bounds))
    endGame();

if (bird.Bounds.IntersectsWith(ground.Bounds))
    endGame();

```

Step 13: отброяване на точки

Ще отброяваме точка всеки път, когато тръба премине игралното поле и излезе в ляво.

```

if (pipeUp.Left < 0)
{
    pipeUp.Left = 900;
    score++;
}

if (pipeDown.Left < 0)
{
    pipeDown.Left = 1300;
    score++;
}

```

И нека да направим точките да се виждат:

```

scoreText.Text = "Score: " + score.ToString();

```