اسم المشروع :

**Flappy bird**

كود الفريق :

GEN\_160

الاسماء :

* محمد سامي عبد الكريم احمد حسان
  + 20201700691
* يوسف عبدالحميد فرج عبدالحميد
  + 20201701158 (old, written in the form)
  + 20201701249 (new)
* ندى محمد حامد فضل
  + 20201700922
* يارا زيدان رمضان زيدان
  + 20201700976
* مصطفى جمال محمد حسن
  + 20201700826

**Special thanks for TA: Esraa Hamdi, For supervising and rating our project.**

**FLAPPY BIRD PROJECT**

**Team leader**

Game physics and pause function and assets editing and second level

by: Mohamed Samy

Menus GUI & menu assets

by: Youssef Abdul-Hamid

Pipes and background, parallax scrolling and score calculation.

by: Nada Mohamed

Collision detection and game over screen

by: Yara Zidan

Music and Sound effect

by: Mai Adel

We did our best to make our project as much cross platform as possible, so we did not use any System specific code such as: system("pause") or WinMain which makes the game theoretically compliable to Linux Systems using GCC compiler instead of MSVC

**Assets :**

**File contains sprites, music ,sound effects and all assets used in the game.**

**Source.cpp**

**Contains the driver code ( main() ) of the game and GUI.**

* int counterForText:

for moving between menu options.

**Engine.h :**

* A file contains main headers for running the project. link functions definitions with headers.
* #Pragma once: a macro that increases the compilation speed by compiling the file once even if it is called many times
* Called in Source.cpp to run the game functions

**Engine.cpp:**

* int updateBrd(int& x): responsible for bird animation.
* bool playGame():main game function.
* bool playGameHard():game function for hard level.
* bool isFisrtpress:

### a Boolean that is used to pause the game when escape is pressed or at the beginning of the game.

//physics variables:

* Gravity: added continuously to velocity to simulate acceleration due to gravity.
* Velocity: initial velocity of the bird at the beginning of the game.
* velocity1:a reference for initial velocity in case it got changed during gameplay
* velocityMax: max limit for velocity of the bird when its velocity on y-axis reaches it due to gravity.
* Clock Vclk: a constant time that is compared to “Time Vvalue” to control rate of acceleration with time.
* Time Vvalue: measuring time the bird travelling by a certain velocity & if it exceeds “Clock Vclk” it will be rested.
* Time basicDelay: makes input lag to prevent repetitive game pausing and resuming.

Time of the game & if exceeded “constanttime” it will be reseted.

* bool jumping : for detecting if the bird jumped or not.

//rotation controls

* initialRot: initial value for rotation.
* maxRot: max value of rotation bird cannot exceed.
* accRot: rate of change of rotation.
* Clock Rclk: clock of rotation.
* Time Rvalue: time the bird moving with a certain rotation & if it exceeds “Clock Rclk” it will be rested.

//////////////OTHERS/////////////

* Collision::CreateTextureAndBitmask :

It detects pixelperfect colloision for a sprite.

* Unreal pipe (for score):
* Transparent texture used calculate score.
* It is located in the middle of real pipes.
* int counterForText :
* for moving between menu options.

//////////////SCORE/////////////

* int hs :

for saving heighst score.

//////GAMEOVER//////

* bool drawing :

For detecting collision with pipes or ground so game endend.

**Collosion.h**

**A library that contains some necessary functions to make pixel perfect collosions.**

**Collosion.cpp**

* bool CreateTextureAndBitmask:

### Replaces Texture::loadFromFile

### This is much faster than creating the bitmask for a texture on the first run of "PixelPerfectTest"

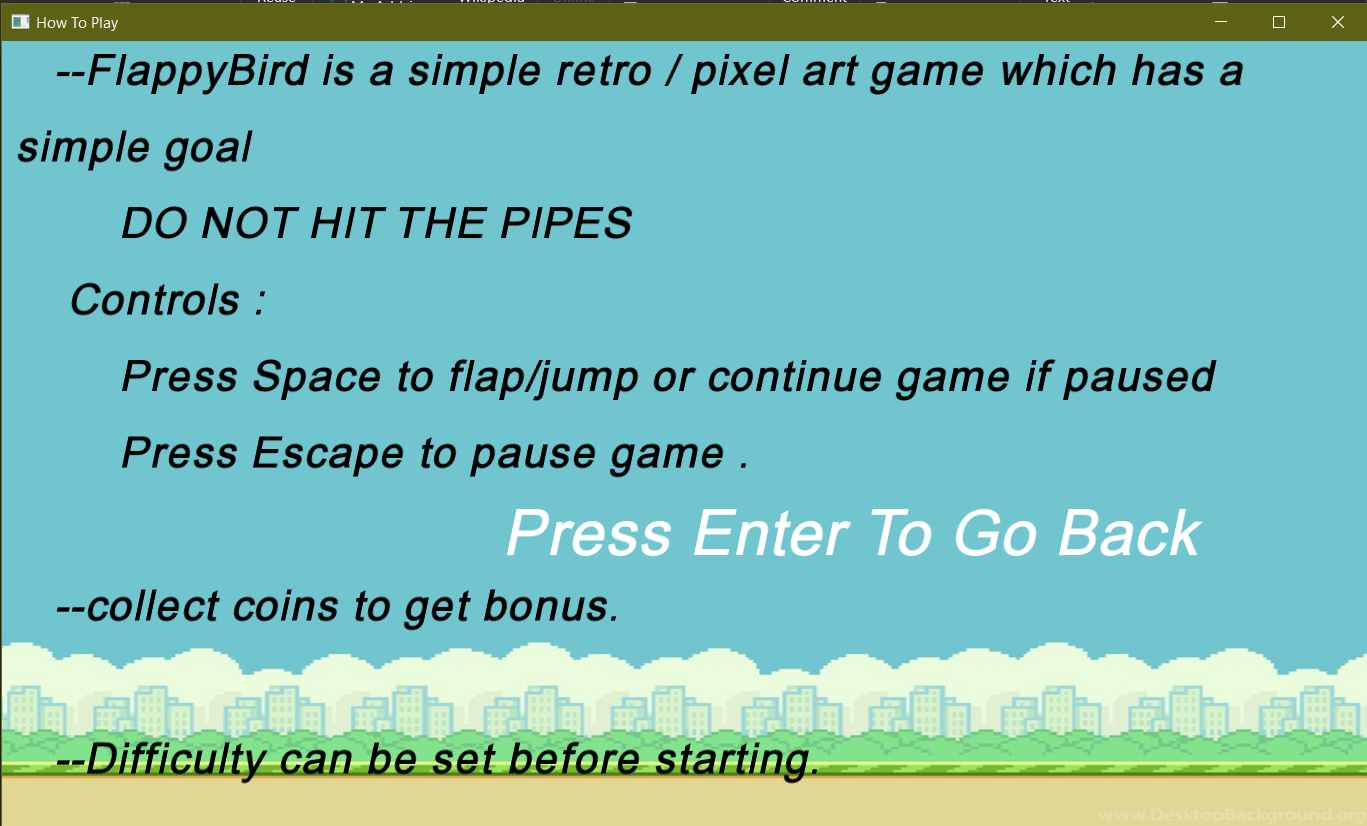
\* bool PixelPerfectTest:

### Test for collision using circle collision detection.

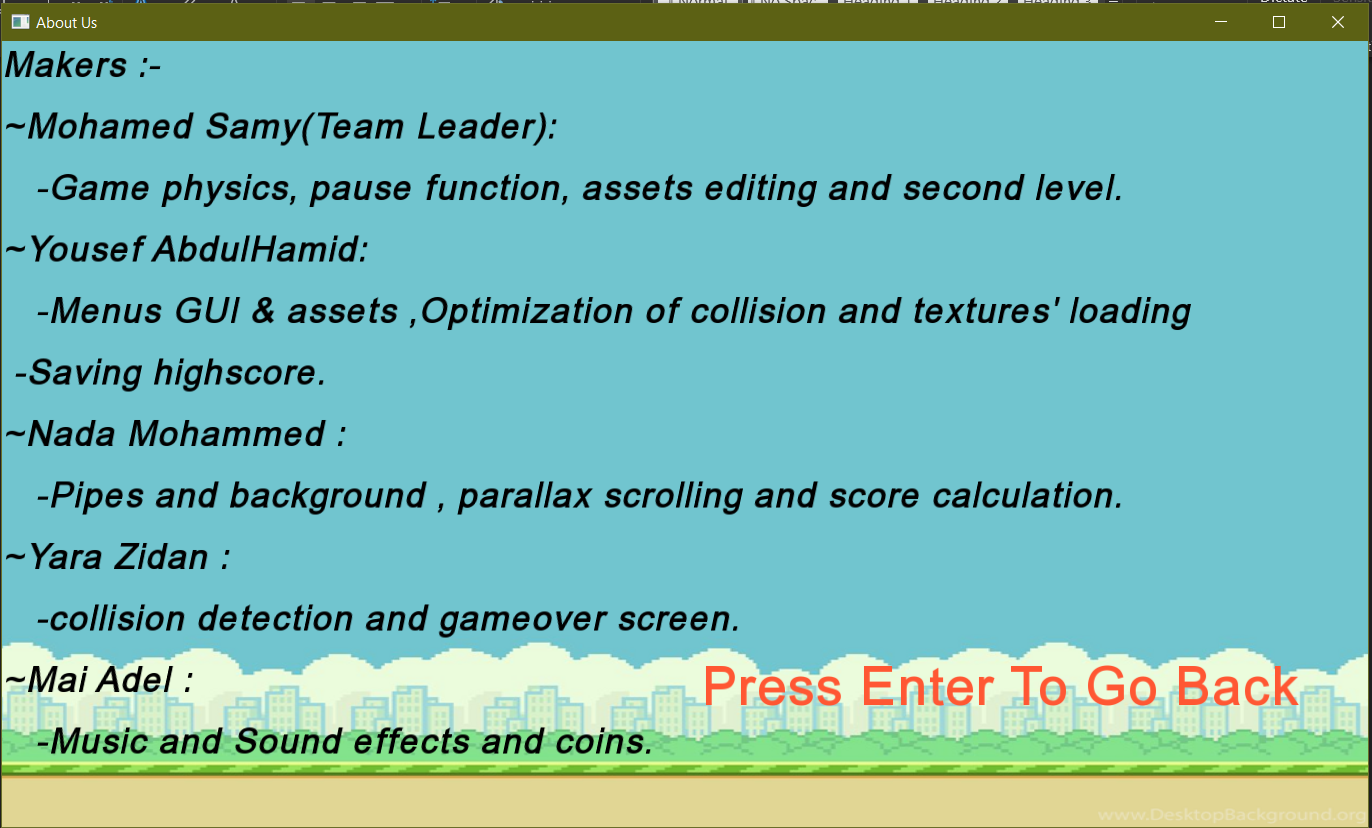
### Radius is averaged from the dimensions of the sprite so roughly circular objects will be much more accurate.

Main game features:

Advanced game GUI

Simple game tutorial

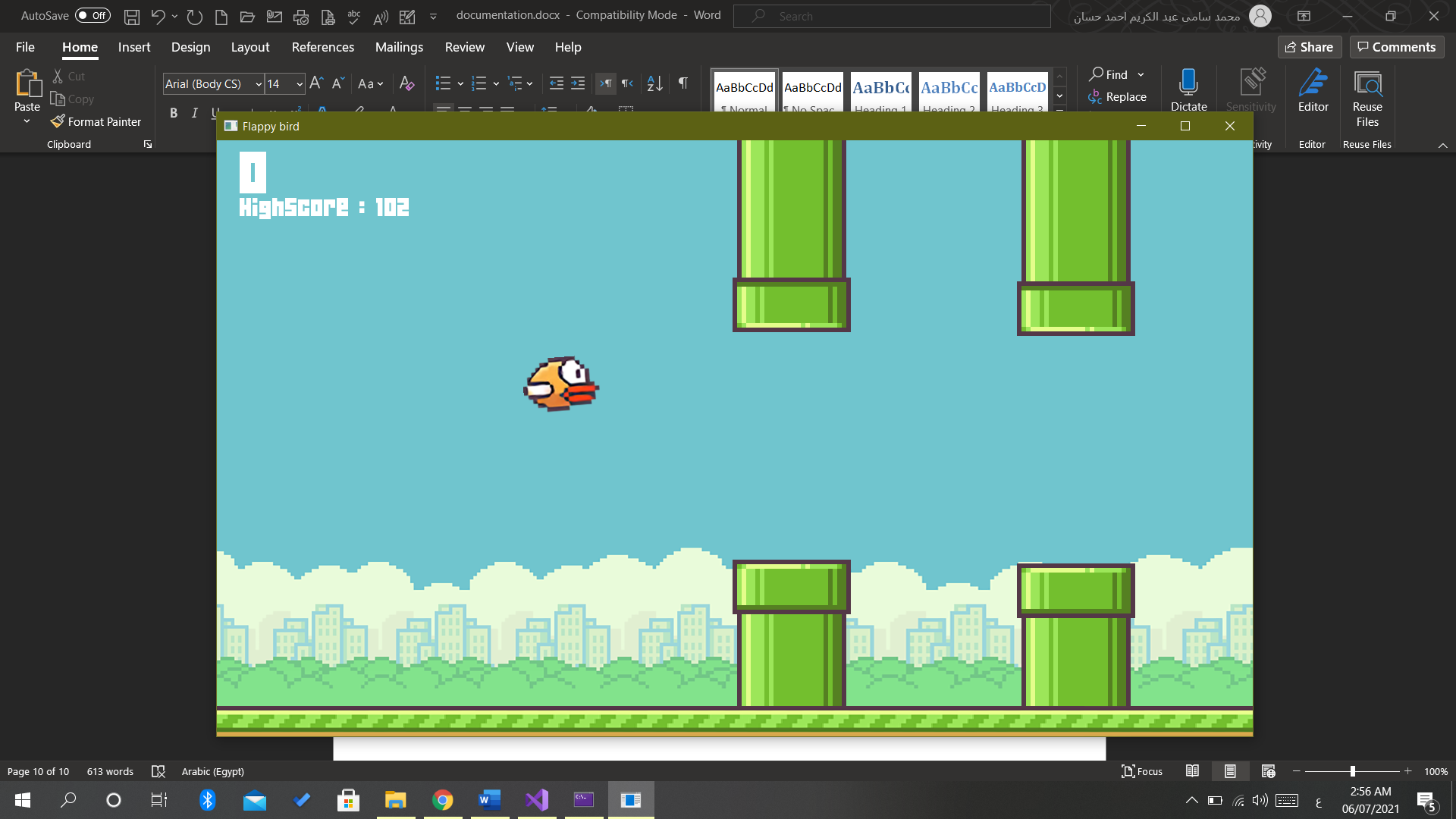
Some information about game creators



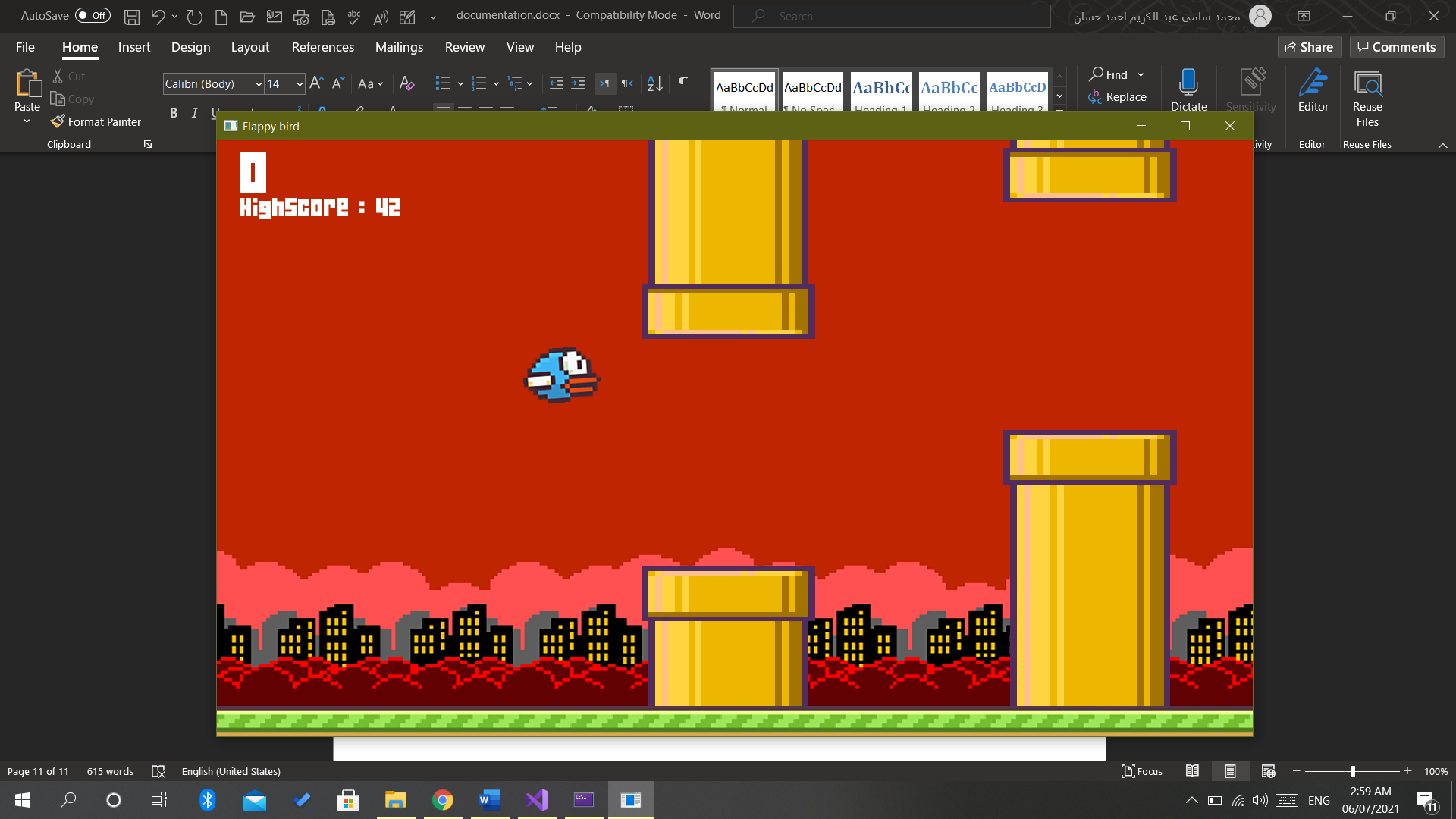
Ability to choose between two levels with different difficulties



Level 1:



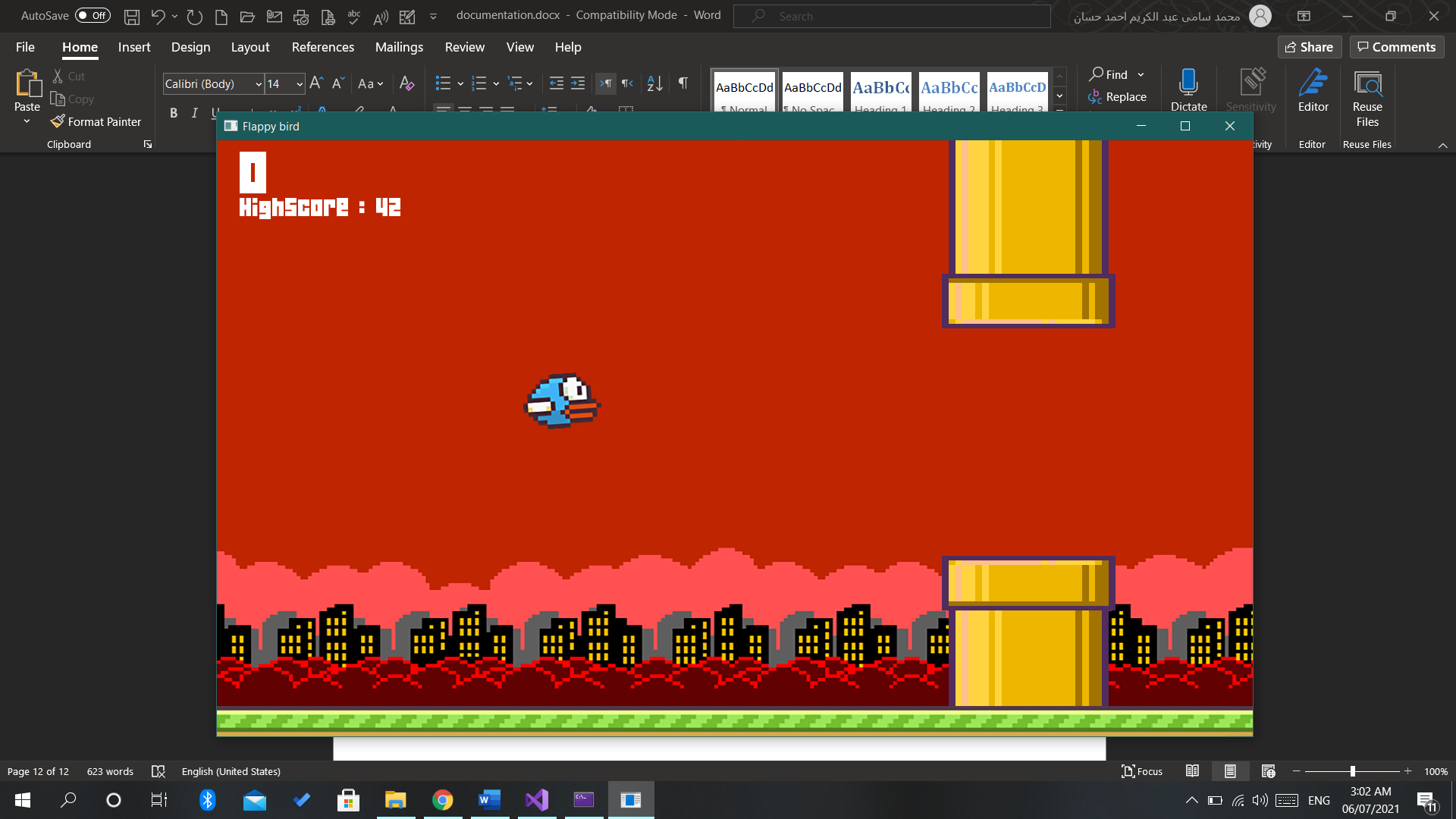
Level 2:



Game over menu with ability to play again:



View current score and save Highest score to view it:



Ability to pause during gameplay:



**\***All levels have music and sound effects.

**\***All menus have sound effects.