

# Project Report - ***Tile Duo***

Group 6 - Team 5

Shreeya Venneti (IMT2020535)

Yash Mogal (IMT2020537)

Paras Vekariya (IMT2020547)

Tejas Sharma (IMT2020548)

# Abstract

\_\_\_\_\_This document presents the implementation of a collaborative game that has been taken as a project by the Group-6 Team-5 members. This is a memory Tile Matching game called “Tile Duo” that is solely written in Python Language. This game has introduced and taught us to admire the power, popularity, and versatility of Python's Tkinter Library.

This project has taught us some important qualities like team-work, leadership, responsibility, and listening & respecting others' thoughts and opinions.

We have learnt and used various modules and functions of the Tkinter library. Moreover, out of curiosity, we explored some features of the Pygame library to implement some features to our Project. This project strengthened our theoretical concepts of Object-Oriented Programming in Python to write code efficiently and improve practicality of scaling it. .

## Introduction

The project takes input from the user using the mouse. It uses functions from pygame and tkinter libraries to perform the required functions. Pygame is used for background music whereas the bulk of the code requires the use of the tkinter module.

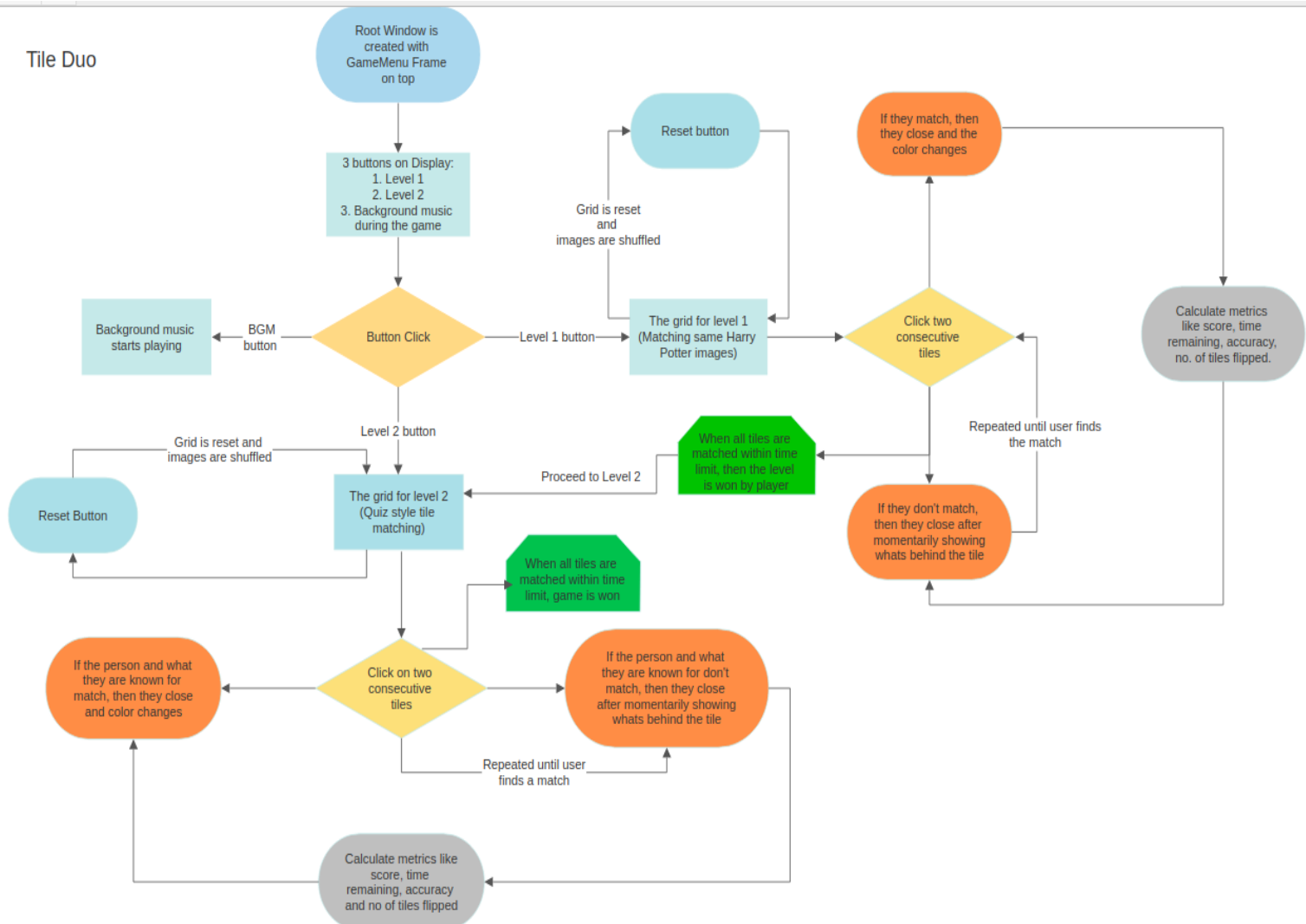
The game requires the user to memorize the tiles as they are shown and try to match them with another tile which has a relation to it (either the same tile in case of level 1 or a tile related to it in case of level 2). There is a scoreboard, timer, accuracy and number of tiles opened label that shows game stats.

There is also a reset option which is used to close all the tiles again. There are penalty and bonus tiles to make the game interesting!

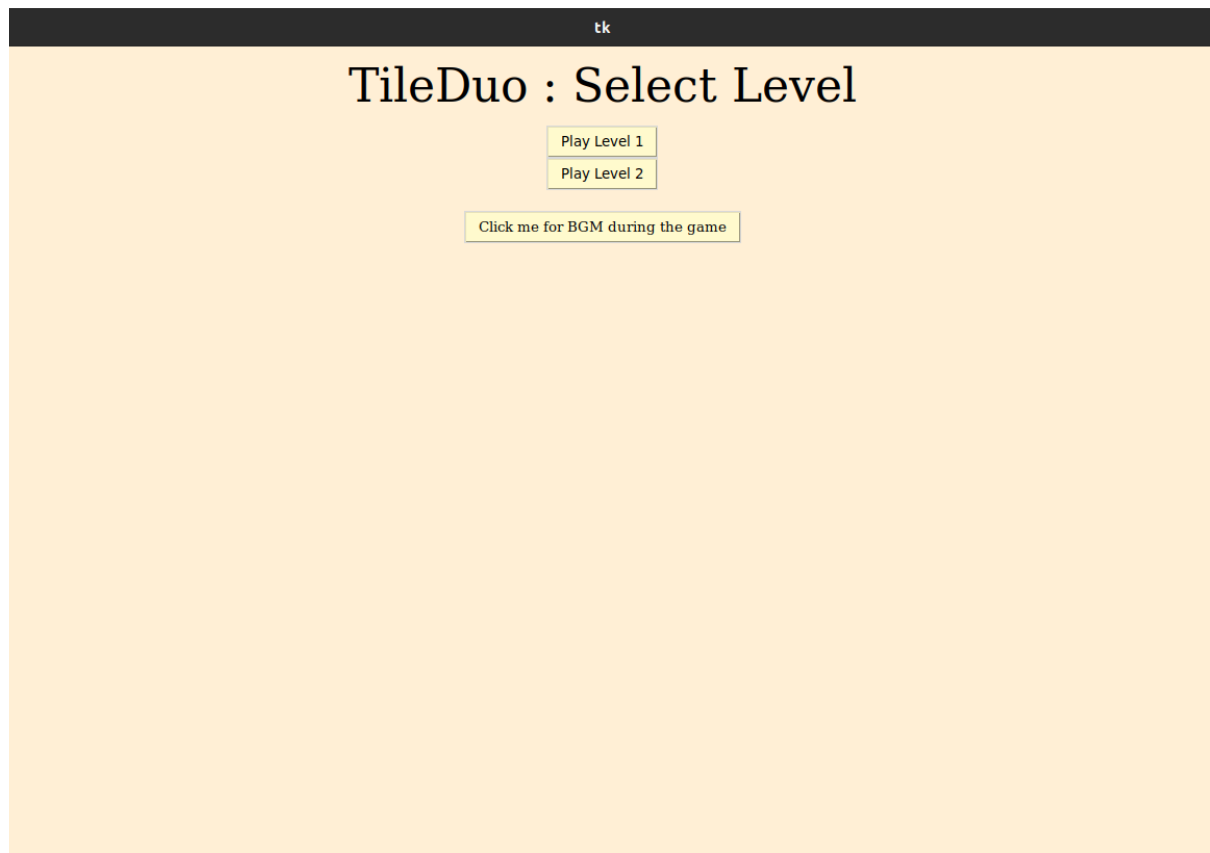
# Flowchart

The game consists of two levels, each level has a 4x4 grid of closed tiles (face down). The player must click on a tile to open it, revealing an image behind it. This tile stays open until another tile is clicked. If the image under the second tile matches with the first, then both tiles are matched and they are disabled. Otherwise, both tiles are closed. Thus it is a memory game, and to win the game the player must open all the tiles within the stipulated time. There are two special tiles : a bomb tile and a bonus tile hidden within the grid to make the game more interesting. The player's score gets reduced and increases upon clicking each of them respectively. Please follow the arrows in the flowchart to trace all possible scenarios in the game:

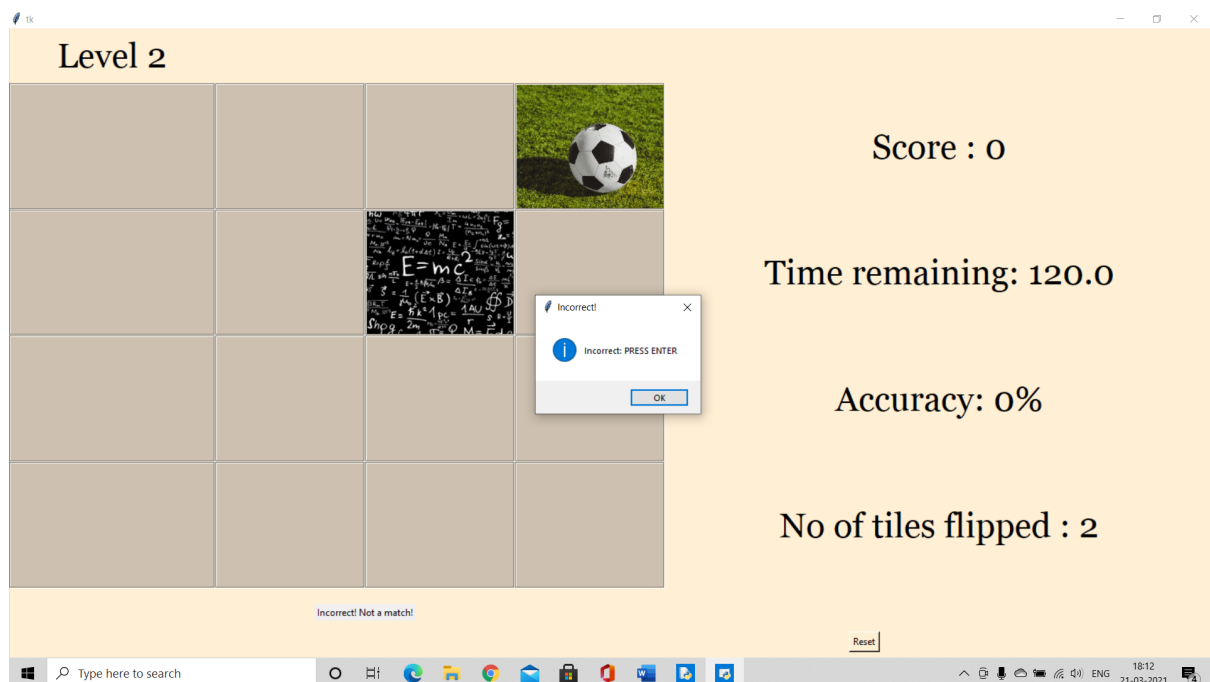
## Tile Duo



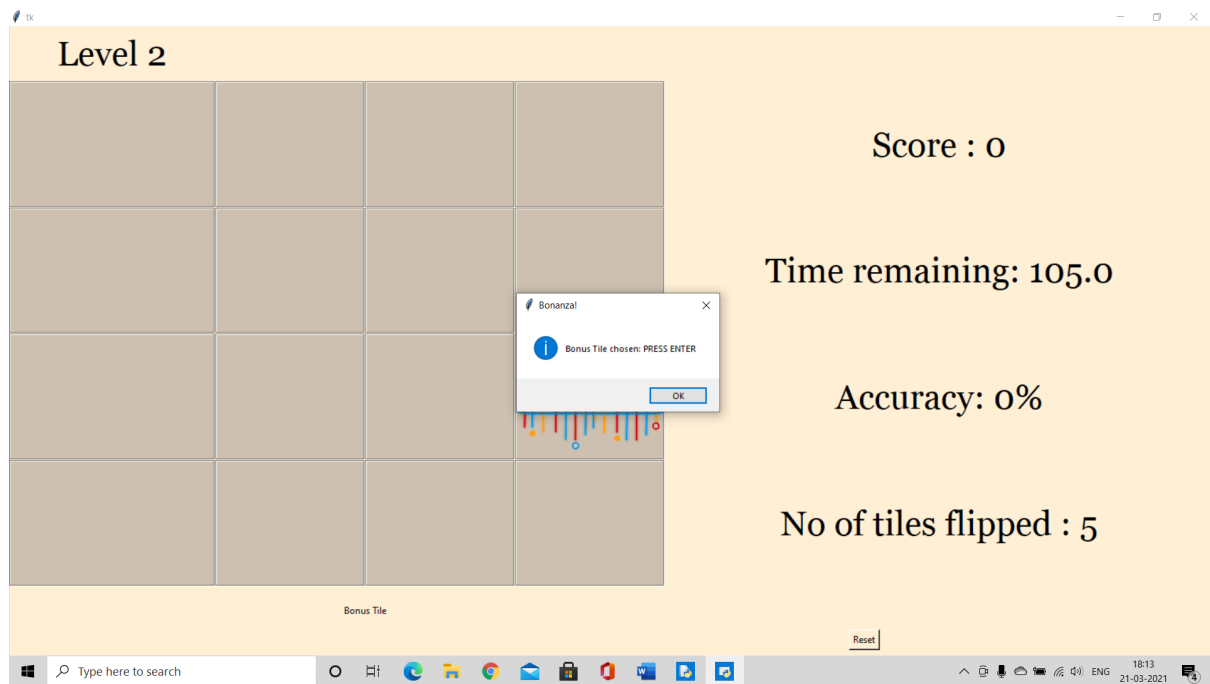
# Demo



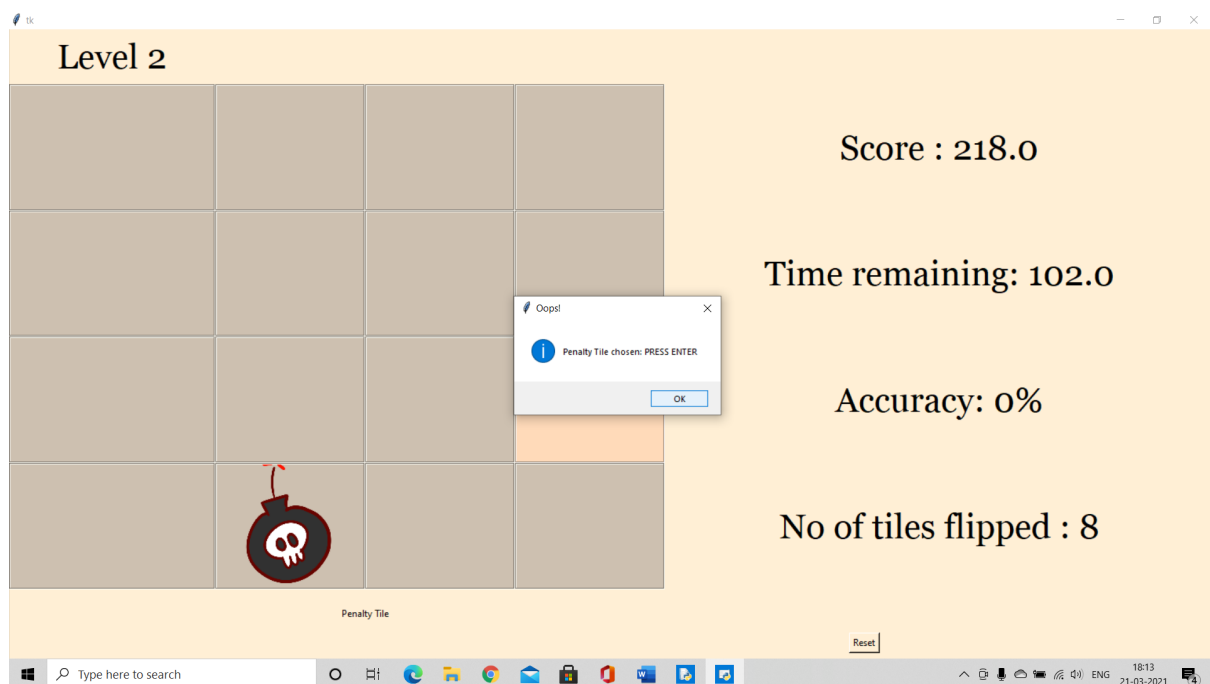
This is the game menu.



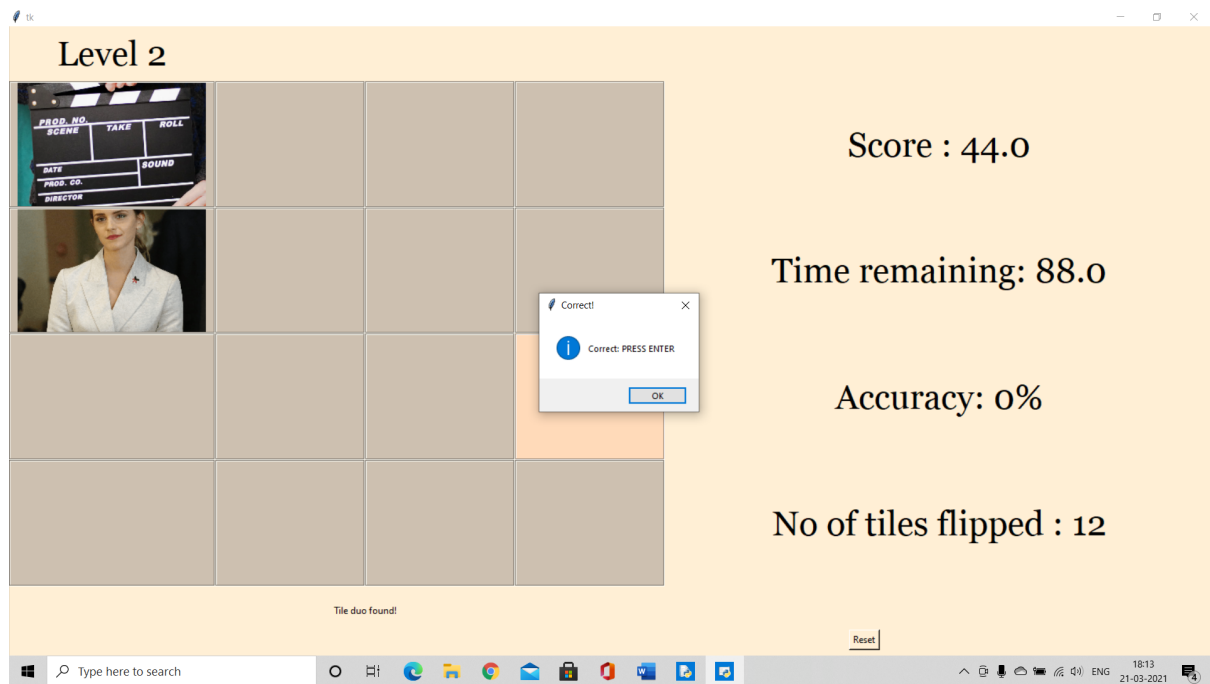
This is the message that pops up when 2 tiles don't match.



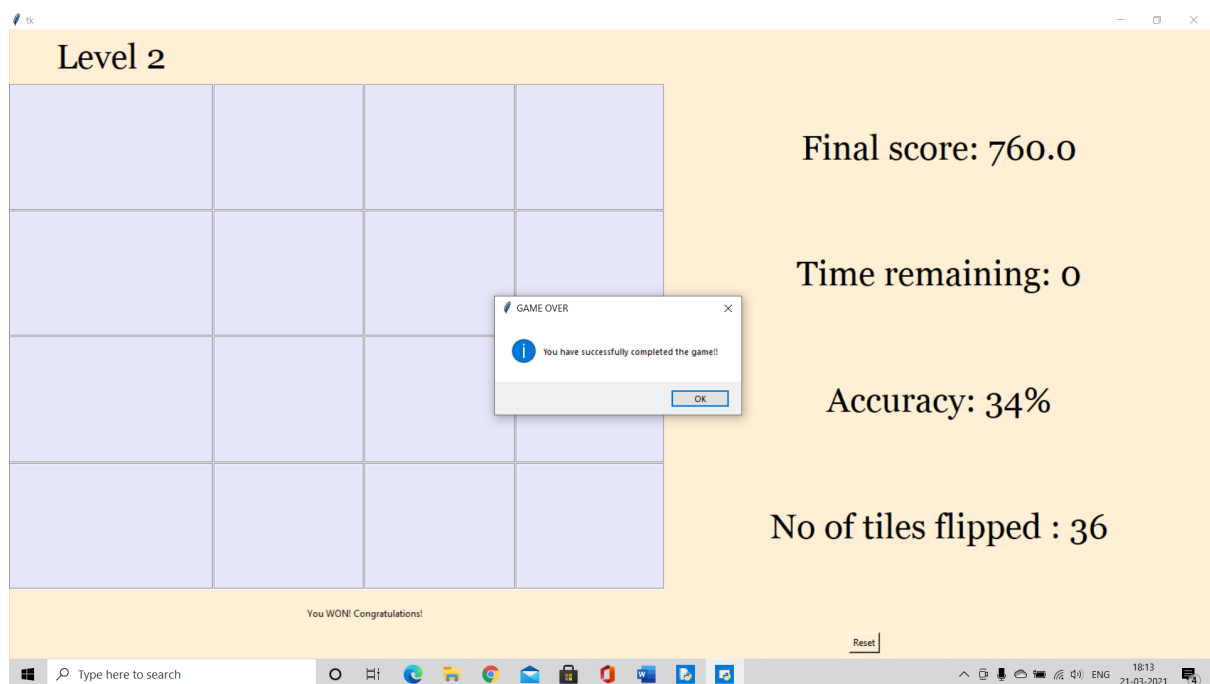
This message is popped when a bonus tile is clicked.



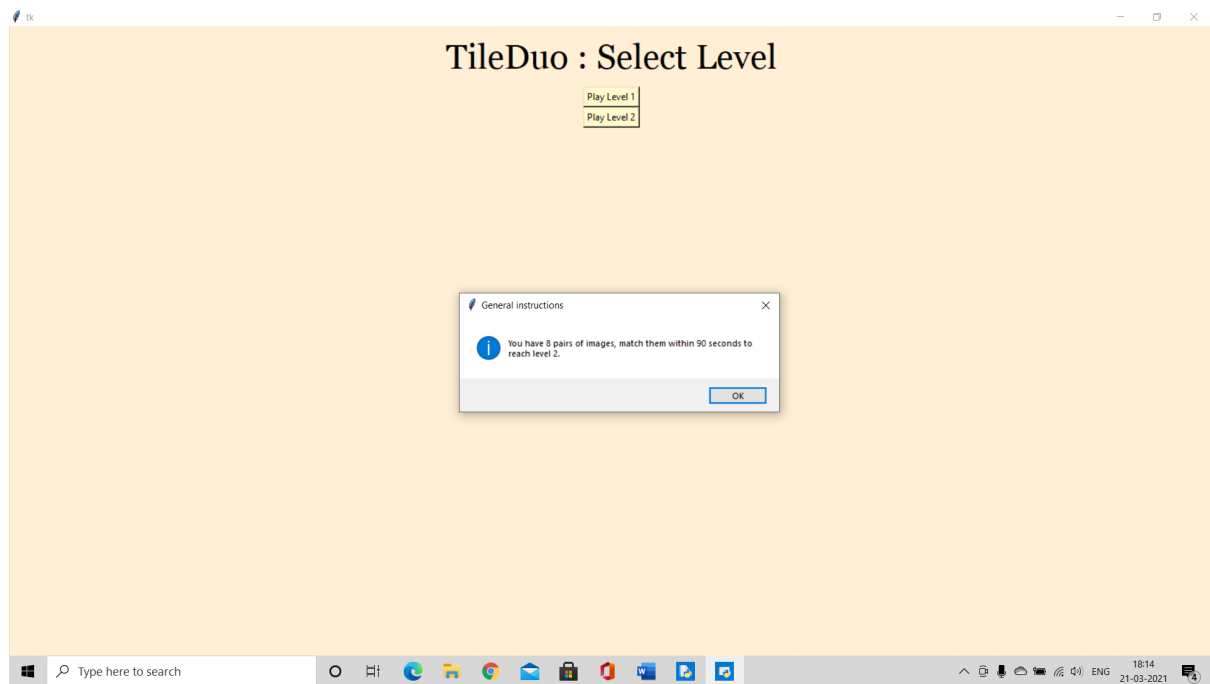
This message is popped when a penalty tile is clicked.



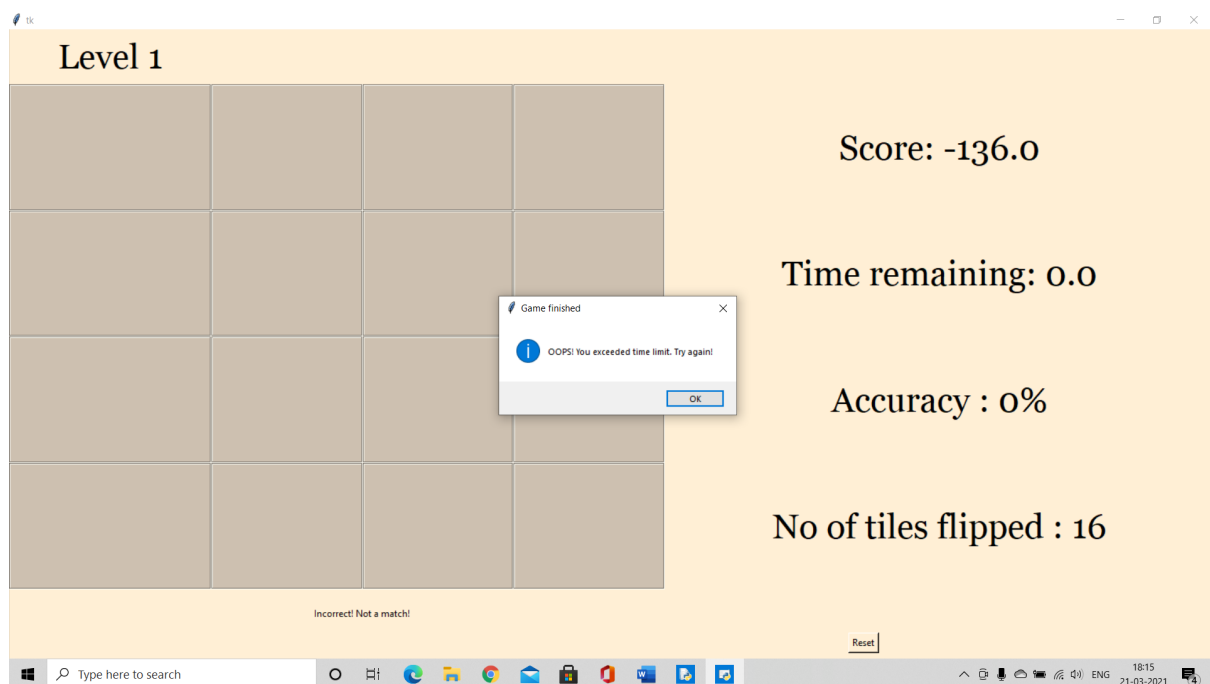
This is the message that pops up when 2 matching tiles are



This message is popped when the game is finished.



This message comes at the beginning of each level.



This message pops up when the time limit is exceeded.

# System Requirements

This game is a cross-platform piece of software, i.e, it is a platform-independent game. It is compatible with almost every OS, like windows, Linux, MacOS.

- 1.) Tkinter is the de facto way in Python to create Graphical User interfaces (GUIs). Installing Tkinter on Windows is as simple as installing Python 3.x since Tkinter is included in the Python 3 core.
- 2.) Tkinter is also included in Python 2.7. The only difference is that it is being imported using 'Tkinter' in Python 2.7 whereas it is imported as 'tkinter' in Python core 3.
- 3.) For enabling Background sound feature, one has to make sure that his/her system has installed Pygame.(Since, pygame is not a builtin library of python, one has to download it from the pip package installer.)
- 4.) The game requires a platform which supports GUI(Graphical User Interface) applications like VS code editor, Pycharm, Python 3.9 IDLE,etc on OS where there is no terminal to run it.

# Contributions

(1) Shreeya Venneti (IMT2020535)

- I. Classes for Screens, buttons and navigation between them
  - A. Created the 3 game screens - GameMenu screen, Level-1 screen, Level-2 screen and stacking of



frames for displaying each screen. Wrote this in an object oriented programming fashion using a total of 4 classes. Classes named LevelOne and LevelTwo for both the levels, another class called Initialize Game for initializing the game by creating the root window and specifying it's geometry, and one more class named GameMenu for the GameMenu screen.

B. Created 3 buttons for the GameMenu screen on clicking they navigate the user to the next screen, and the BGM button which plays background music upon clicking it. Used the pygame mixer to include an audio .ogg file, defined a method that will be called when the BGM button is clicked to play the audio file.

II. Logic for both the 4x4 grids present in level-1 and level-2

A. Wrote the logic for the 4x4 grid of level-1, under the class named LevelOne. Created a dictionary to store the images, sent them into a list to shuffle the keys (using the random module) in order to shuffle the images indirectly, maintained another dictionary to store these shuffled images. For level-2, added an extra dictionary to maintain the matching for distinct images(matching the famous person with what they are known for). Wrote a method named "matched" that will be called whenever the player matches all the tiles and configures a label. Wrote a method named "tile\_select" which is called whenever the player clicks on a tile button, which checks whether both tiles selected have matched or not with the help of a list and a dictionary. Both of them are cleared every time two tiles are selected. When there is a

match, the tiles are disabled, otherwise they simply close.

B. Maintained a counter displayed by a dynamic label for the number of tiles opened by the player before winning the game to check the efficiency of the play. Created another dynamic label which tells the status of the two tiles selected whether they have matched or not at the bottom of the screen.

III. Suggested name "TileDuo", and ideas and features to the game for the team members to implement like special tiles (penalty/bonus), timer, reset which were implemented by team members, and analog clock, high score, sounds for button clicks, background image for the game etc for future versions of the game. Worked along with the team in solving bugs.

## (2) Yash Mogal (IMT2020537)

Worked on:

- A. Adding feature tiles to the game like-
- B. Penalty- It reduces both score and time of the player. This feature tests memory of the player, since it doesn't get disabled once open until the end of game.
- C. Bonus Tile-It increases score and time of the player. However, this feature tile, unlike Penalty tile, can be used only once in a Level.

## (3) Paras Vekariya (IMT2020547):

Worked on:

- A. Reset feature on both game frames which reset current lvl only.
- B. Quit option- which helps the user/player to close and quit the game.

#### (4) Tejas Sharma (IMT2020548)

I have worked on the features of the game, mainly focusing on the player stats. These features include: timer, scoreboard, accuracy.

For the timer I have used the time module, which is stored in a variable whenever the first button of the level is clicked and used the current time to calculate the time elapsed.

For the scoreboard I have tried to relate it with the time taken to solve one tile.

For the accuracy I used the number of tiles flipped and the number of correct tiles flipped.

I have also contributed little bit towards the flow of the program. I have tried to bring as much correlation between timer and scoreboard as possible. I have helped my teammates in understanding the logic of the program and also gave them hints on how they can implement their work.

### Future Aspects

- Addition of analogue clock to the game in place of the timer object so that the player can see the clock ticking.
- Addition of a blinking timer which blinks when the timer is less than or equal to 15 sec between red and white. Currently we have a feature which turns all tiles red when the timer is less than equal to 15 secs.
- Addition of sounds for button clicks
- Background image for the game to make it more appealing.

## References

1. Youtube - Introduction to tkinter
2. Tkinter documentation