Manual to our game SET

June 27^{th}, 2025

Welcome to the game SET! In this manual, we will guide the user through the game and what special features we implemented.

1 Prerequisites

First off, make sure there is access to these prerequisites:

a. We will only instruct how the game works in VScode, so make sure this is installed;

b. the coding language Python;

c. the following libraries:

   (i) itertools;

   (ii) pygame;

   (iii) random;

   (iv) time;

   (v) sys; and

   (vi) threading.

d. The GitHub repository, which can be accessed through the following link:
   https://github.com/MifiLoiz/Sett/branches
   and make sure you the files can be opened in VScode.

2 First round

2.1 Selecting cards

Now, we are ready to start the game. Open the file:

`set_game.py`

and hit the run-button in the top right corner of the screen. Now, a window will be opened, primarily showing a table with 12 numbered cards. An example would be:

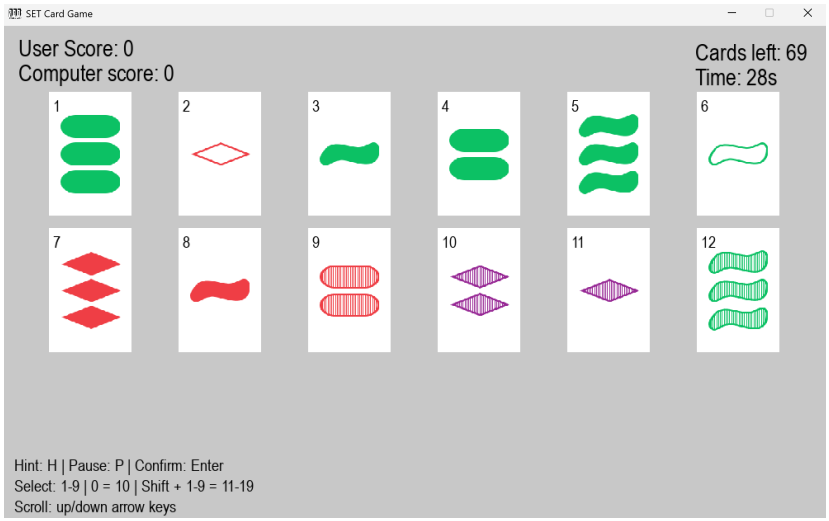

Figure 1: Example of what a screen could look like when a game is first started

In addition, in the top left corner we can check for current scores and in the top right corner the remaining time and amount of cards left in deck of cards. On the bottom, in the left corner we have a quick manual of how the keys work.
Let us start with checking out our first SET and let's suppose this consists of the second, tenth and twelfth cards. When a card is successfully selected, this cards get a red border. There are two ways of giving the input, namely via keyboard keys and clicking. We will discuss both step-by-step.

Keys:
It comes down to typing the correct numbers, three in total in order to create a SET. The moment the user types the third number, this SET is automatically entered and checked. Later on we will discuss what happens next.
What keys correspond to what card number? For the numbers 1-9, simply type the corresponding key, which will turn the border of the card red. If the tenth must be selected, press 0. For the numbers 11-19, press shift + the second digit of the number you need. So, for the second, tenth and twelfth cards we enter (bold indicates that shift is pressed):

2**0****2**

Notes:
The twentieth and twenty-first do not have corresponding keys, so the next method should be used in order to select these cards.
Seperation by space is not needed, the program actually ignores this.

Clicking:
In order to select a card by clicking, stand on the desired card with the mouse and click on it. The card should get the red border immediately.

Becoming more advanced
The user's gaming can be enhanced by making use of the following abilities built in this game. The game is designed to be dynamic and both ways of clicking can be used interchangeably! Not happy with the selected card? Type the corresponding key or click on the card again in order to deselect it!

2.2 Submitting a SET

Once you have selected three cards, these are automatically checked for validity.
Was the SET a valid one? Congrats! Then, this message will appear on the screen:

Figure 2: Message received when users submitted a valid SET

As the message says, the user's score will increase by one and the timer is reset to 30 seconds *.
Was it not a valid SET? On the screen, this message appear:

Figure 3: Message when users enters an invalid SET

Do not worry yet! There is no consequence to this invalid SET. The user can continue looking for one that does make a valid SET by following the same steps as described before. Once they do make a valid SET, they will get to see Figure 2 and score one point.

2.3 No SET found

When the user was unable to find a SET, two possibilities may have caused this:

No valid SETs on table:
If no valid SETs were present on the table, of course no valid SET can be found. If this is the case, a message will pop up and announce this. Simultaneously, three cards will be added to the current cards. This is an example of what the screen could look like after three cards were added:

*Unless the default setting was changed.

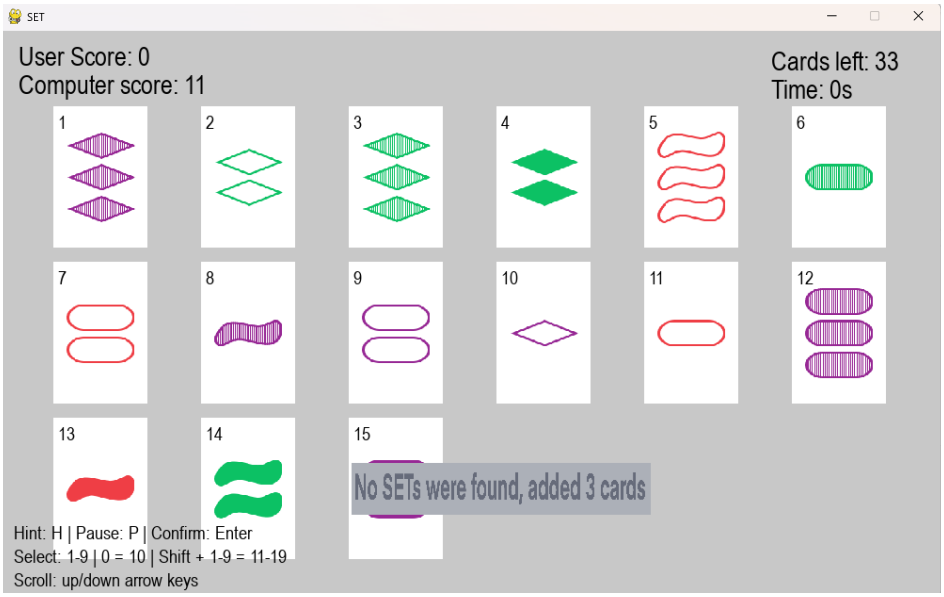

Figure 4: What happens when no SETs were present?

What happens next? A new round will be started, but now the table holds fifteen instead of twelve cards. The timer will be reset and no one gets an extra point. The game continues as it did before.

Valid SET present, but none were found:
Was there a valid SET on the table, but did the user not succeed in finding it? Too bad! This message will be displayed on the screen:

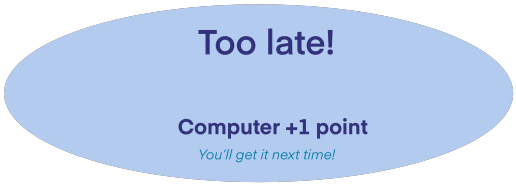

Figure 5: Message when user found no valid SET and computer gets a point

Now, the computer takes a valid SET and replaces the corresponding cards on the table with ones from the deck. This match will be highlighted by rimming the cards blue, for example like this:

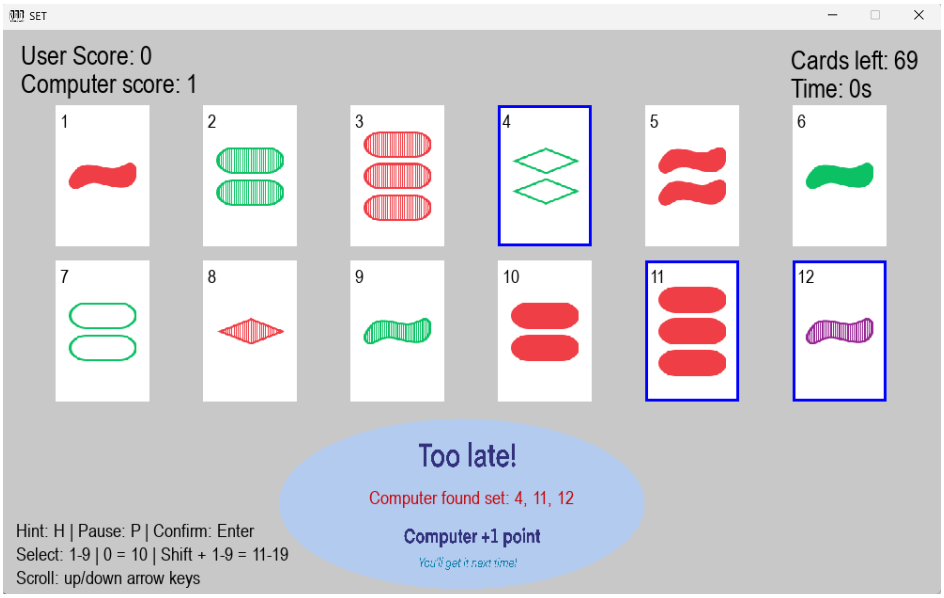

Figure 6: User was too late and the computer found a valid SET, these cards are highlighted

Note that the picked cards are also displayed in red in the blue message.
As a consequence of the user not finding a SET, a point will be rewarded to the computer, the timer gets reset and a new round begins. Good luck!

3 End of the game

The game continues until the deck has run out of cards and no other valid SETs can be made. Then, depending on the winner, a window will occur and the final scores are displayed. The three possible scenarios are:

User wins!

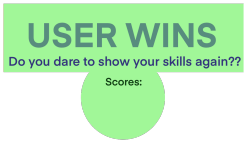

Figure 7: Message when user wins the game

When this screen appears, congratulations, user won!

Tie

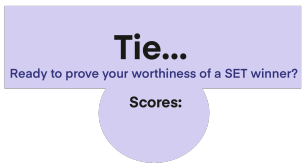

Figure 8: Message when the game ends in a tie

Ah no, victory was close!

User loses...
Ah, unfortunately user lost if this appears on the screen:

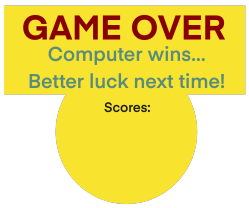

Figure 9: Message when the user loses

Would the user like to play another game? Just click on the following button and a new game will be started:

Figure 10: Play again-button

Enjoy!

4 Extra features

In order to make the game even more intriguing, we built in some extras, namely a pause and a hint button. In this section, we will walk through the implementations of these.

Pause:
Does the user have to catch some breath, but do they not want the computer to take advantage of their lacking stamina? Pause the game by pressing the key 'P'. Now, this image will appear over the open cards:

Figure 11: Message when game is paused

Exiting pause mode is done by pressing the key "P" again.
This pause mode comes with a few consequences. Most importantly, this prevents the player from being able to see the cards and from proposing a valid SET [†]. It has no consequence to the score counts.

Hint:
Is the user determined not to let the computer get a point, but are they unable to find a valid SET? The user can do this by pressing the key "H", which rims two cards of a valid SET green, so all they have to do is to find the third one. For example, take this table:

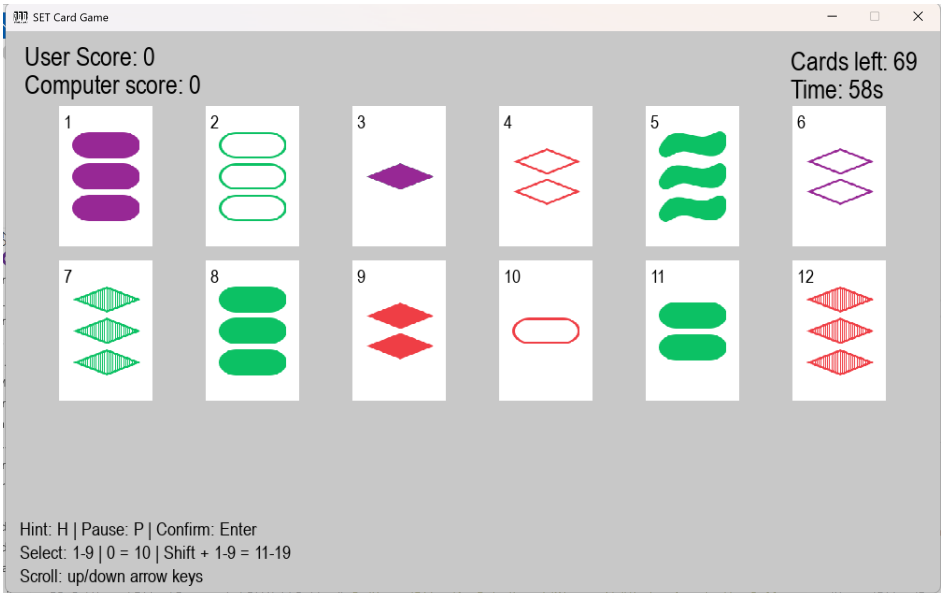

Figure 12: Example use of hint: a table

[†]The player can still cheat by taking a screenshot first, something we have not thought of yet.

Now, we want a hint, so we press "H", which highlights the fifth and seventh card:

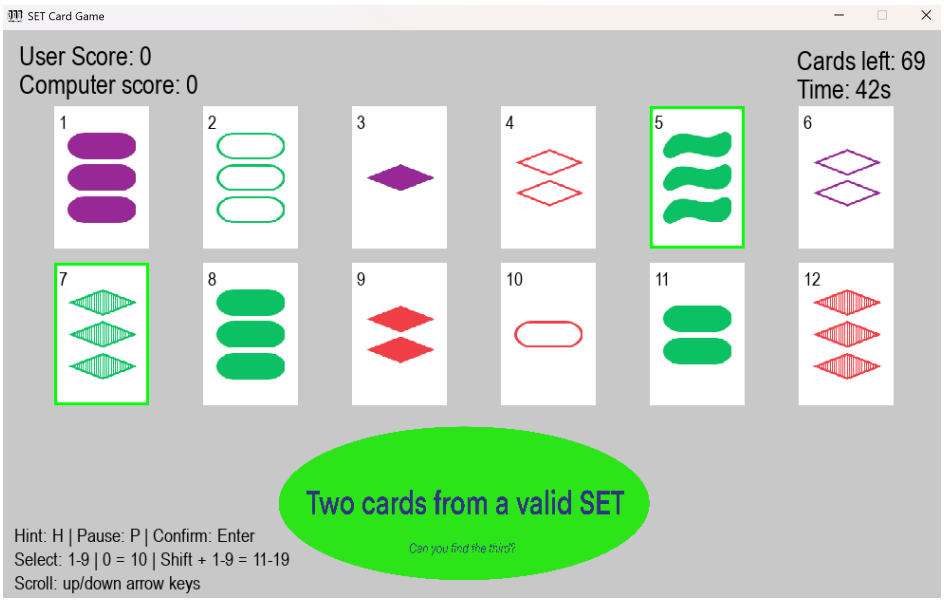

Figure 13: Example use of hint: two green rimmed cards

Given these two cards, we see that the SET is completed by the second card. So, now we select these three cards. Since this is a valid SET, the following message will appear:

Figure 14: Message when valid SET is when a hint was used during that round

Since we did use a hint, we do not get a point. But, at least the computer does not either.
Notes
Do note that it should not be forgotten to select the cards suggested by hint. If the user has run out of time after all, the computer still gets a point.
Laslty. one is free to submit another SET, but they will not get rewarded for this.

5 Optional settings

Would the user prefer not to have the pressure of the timer, or would they like to challenge themselves by granting less playtime? For the game we have set some default constants, which could be changed to the user's wishes. Constants that could be changed are: allowed play time; and duration of the messages. The latter might be especially interesting for accelerating games.
These constants have to be changed in the following file:

```
constants_pygame.py
```

When the user has opened this file, they see a list of constants. In order to change the duration of the game, first check out:

```
TIMER_DURATION = 30
```

So, by default, the playtime is set to 30 seconds. In order to change this, set this constant any desired number of seconds. This number must be an integer of at least 1.
For the message duration, check out:

```
MESSAGE_DURATION = 3
COMPUTER_PAUSE = 3
```

The message duration is changed by setting both constants to the desired number of seconds. The default number is 3.
Again, both should be integers of at least 1.