

# Ομαδικό Προγραμματιστικό Project σε Python: Παιχνίδι Μνήμης με Τράπουλα

---

ΣΥΜΜΕΤΕΧΟΝΤΕΣ: ΘΕΟΛΌΓΟΥ ΣΠΥΡΪΔΩΝ, ΜΙΧΑΛΌΠΟΥΛΟΣ ΛΕΩΝΪΔΑΣ

PROJECT ID: 05

ΚΩΔΙΚΌΣ ΤΜΉΜΑΤΟΣ: ΠΛΗΠΡΟ-ΗΛΕ62

ΑΡΙΘΜΌΣ ΟΜΆΔΑΣ: ΟΜΆΔΑ 3

ΚΑΘΗΓΗΤΉΣ ΣΥΜΒΟΥΛΟΣ: ΖΑΦΕΪΡΗΣ ΒΑΣΪΛΕΙΟΣ

ΑΚΑΔΗΜΑΪΚΌ ΈΤΟΣ: 2021-2022

# Προδιαγραφές/Κανόνες Παιχνιδιού

---

- Ο παίκτης επιλέγει 2 κάρτες οι οποίες είναι γυρισμένες ανάποδα. Αυτές αποκαλύπτονται και αν ταιριάζουν, παραμένουν ανοικτές, αλλιώς γυρίζουν και πάλι ανάποδα.
- Επίπεδα Δυσκολίας(Εύκολο-16 Κάρτες/ Μέτριο-40 Κάρτες/ Δύσκολο-52 Κάρτες)
- Υλοποίηση για 1-4 Παίκτες. Όταν είναι 1 παίκτης, παίζει με αντίπαλο τον υπολογιστή
- Κάθε κάρτα έχει αξία, η οποία προσάπτεται ως πόντοι στον εκάστοτε παίκτη που θα ανοίξει ζεύγος καρτών με το ίδιο σύμβολο(αριθμός ή φιγούρα). Ο παίκτης με τους περισσότερους πόντους στο τέλος κερδίζει. Το παιχνίδι λήγει όταν ανοίξουν όλες οι κάρτες
- Ειδικές Κάρτες(Βαλέδες – Ρηγάδες – Ζεύγος Ντάμα – Ρήγα)
- Αποθήκευση του παιχνιδιού και δυνατότητα συνέχειας από το σημείο που έμεινε στην τελευταία συνεδρία
- Η γραφική διεπαφή είναι υλοποιημένη μέσω της βιβλιοθήκης tkinter.

# Κλάσεις:

---

- Menu → Υλοποιεί το αρχικό μενού
- NewGame → Υλοποιεί το παιχνίδι
- ContinueGame(NewGame) → Ανακατασκευάζει και υλοποιεί το αποθηκευμένο παιχνίδι
- Player → Ο παίκτης
- Computer(Player) → Ο αντίπαλος στο single player
- Tile(Button) → Η κάρτα
- GameState → Περιέχει πληροφορίες για την ανακατασκευή του παιχνιδιού

# Κλάση Menu:

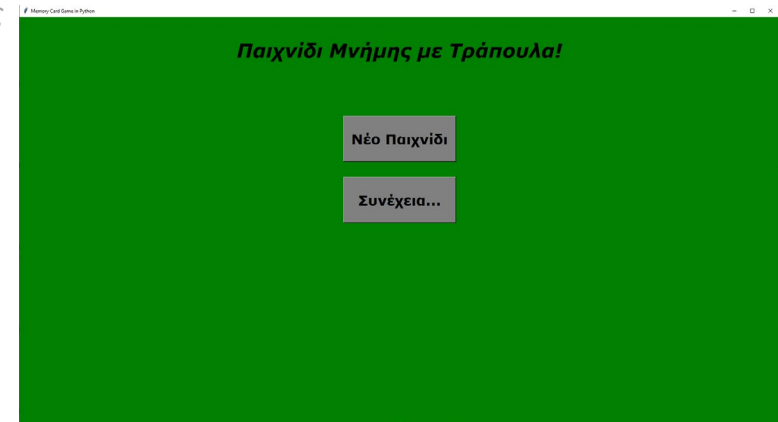
## Γνωρίσματα:

- master → Tk() αντικείμενο, το κύριο παράθυρο της εφαρμογής
- frame → frame που τοποθετούνται τα Buttons
- welcome\_message → Label με τον τίτλο του παιχνιδιού
- mode → νέο παιχνίδι ή συνέχεια προηγούμενου
- players → ο αριθμός των παικτών που επιλέχθηκε
- difficulty → το επίπεδο δυσκολίας που επιλέχθηκε

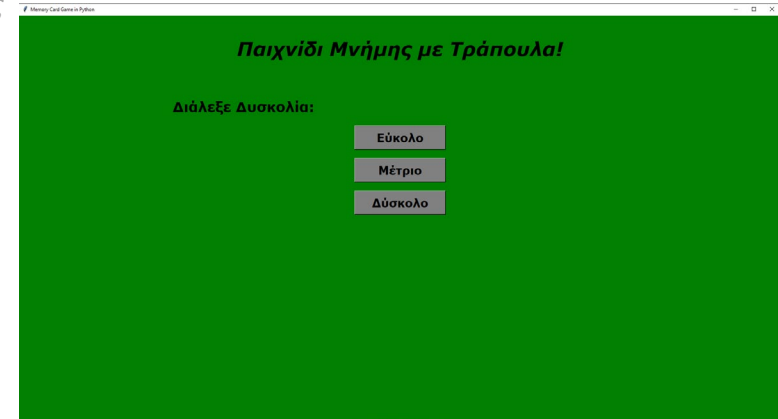
## Μέθοδοι:

- new\_game → καταχωρεί ότι επιλέχθηκε νέο παιχνίδι και εμφανίζει τα κουμπιά επιλογής δυσκολίας
- num\_of\_players → εμφανίζει τα κουμπιά επιλογής αριθμού παικτών
- clear\_screen → καταστρέφει widgets που δέχεται ως ορίσματα
- set\_difficulty → καταχωρεί την επιλεγμένη δυσκολία
- start\_game → καλεί την NewGame ή την ContinueGame ανάλογα με τις επιλογές του χρήστη

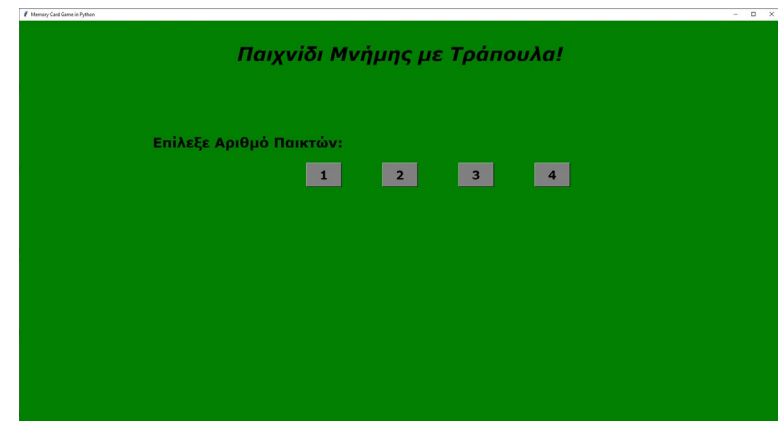
## 1. Έναρξη εφαρμογής



## 2. Επιλογή Δυσκολίας



## 3. Επιλογή Αριθμού Παικτών



# Κλάση Tile(Button):

*Πρόσθετα Γνωρίσματα:*

- rank → η αξία του φύλλου (“ace”, 2, 3, ... , “king”)
- suit → το σύμβολο της κάρτας (“spades”, “diamonds”, “hearts”, “clubs”)
- back\_image → η εικόνα της πίσω όψης
- front\_image → η εικόνα της μπροστινής όψης
- is\_flipped → True αν η κάρτα είναι face up, αλλιώς false

*Μέθοδοι:*

- is\_figure → επιστρέφει True αν η κάρτα είναι φιγούρα
- flip → αλλάζει το is\_flipped.
- value → επιστρέφει την αριθμητική αξία της κάρτας



1. Η πίσω όψη μιας κάρτας



2. Η μπροστινή όψη μιας κάρτας

# Κλάση Player:

## *Γνωρίσματα:*

- player\_number → κάθε παίκτης έχει το δικό του μοναδικό αριθμό
- name → το όνομα του παίκτη(f"Player {player\_number}")
- score → για κάθε παίκτη κρατάμε ένα σύνολο πόντων

## *Μέθοδοι:*

- player\_number → κάθε παίκτης έχει το δικό του μοναδικό αριθμό
- name → το όνομα του παίκτη(f"Player {player\_number}")
- score → για κάθε παίκτη κρατάμε ένα σύνολο πόντων

# Κλάση Computer(Player):

## *Γνωρίσματα:*

- name → “Computer”
- history → λίστα με το ιστορικό καρτών που απομνημονεύει ο υπολογιστής
- score → οι πόντοι που έχει συλλέξει ο υπολογιστής

## *Μέθοδοι:*

- memorize\_tile → προστίθεται μια κάρτα στο ιστορικό καρτών(max = 5)
- remove\_tiles(tile\_list) → αφαιρεί κάρτες από το ιστορικό

# Κλάση GameState:

*Γνωρίσματα:*

- `player_index` → το index του τρέχοντος παίκτη, στη λίστα με τους παίκτες
- `tiles_info` → λίστα με λίστες στοιχείων που χρειάζεται κάθε κάρτα για να οριστεί( `[tile.rank, tile.suit, tile.is_flipped]` )
- `players_list` → λίστα που περιέχει τους παίκτες(πλην του `computer`, αν υπάρχει)
- `open_tiles` → ο αριθμός των ανοικτών καρτών
- `total_tiles` → ο συνολικός αριθμός των καρτών
- `difficulty` → η δυσκολία του παιχνιδιού
- `computer_score` → το σκορ του υπολογιστή(αν υπάρχει)
- `computer_history` → λίστα με το index της κάθε κάρτας που βρίσκεται στο ιστορικό καρτών του υπολογιστή(αν υπάρχει)



# Κλάση NewGame:

*Γνωρίσματα:*

- master
- difficulty
- total\_players → συνολικός αριθμός παικτών
- board\_frame → το frame στο οποίο τοποθετούνται οι κάρτες
- tiles → λίστα με τις κάρτες
- players → λίστα με τους παίκτες
- scoreboard → Label με τους πόντους κάθε παίκτη
- score\_frame → το frame του scoreboard
- player\_scores → λεξικό με key τον παίκτη και value το Label που δείχνει το σκορ του.
- message → λεζάντα με τα μηνύματα του παιχνιδιού
- message\_frame → το frame των μηνυμάτων παιχνιδιού
- current\_player\_index → το index του τρέχοντος παίκτη στην λίστα players
- current\_player → ο τρέχων παίκτης
- click\_count → ο αριθμός των επιλεγμένων καρτών ανά γύρο
- clicked\_tiles → οι επιλεγμένες κάρτες ανά γύρο
- open\_tiles → ο αριθμός των ανοικτών καρτών
- total\_tiles → ο συνολικός αριθμός των καρτών
- current\_state → Αντικείμενο τύπου GameState με την τρέχουσα κατάσταση του παιχνιδιού
- computer → ο AI αντίπαλος(αν υπάρχει)

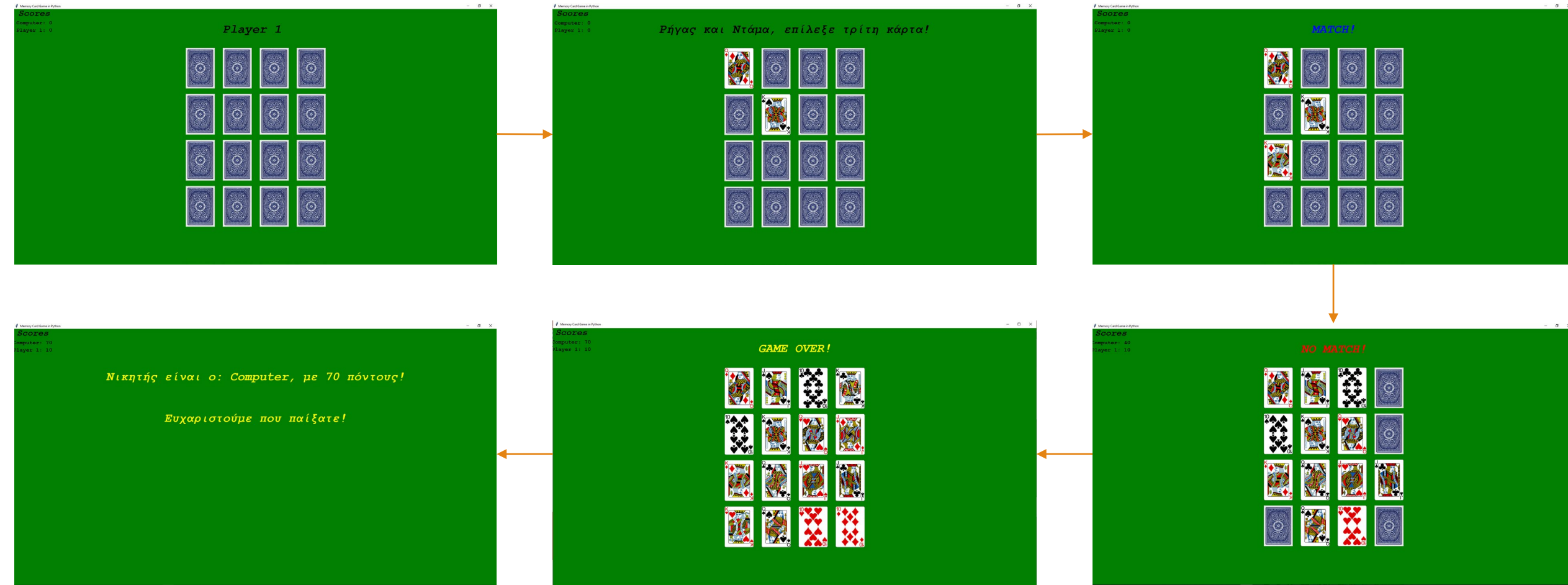
# Κλάση NewGame:

## Μέθοδοι:

- `init_tiles` → δημιουργεί τη λίστα με τις κάρτες
- `create_board` → τοποθετεί τις κάρτες στο «τραπέζι»
- `init_players` → δημιουργεί τη λίστα με τους παίκτες
- `create_scoreboard` → δημιουργεί και τοποθετεί το scoreboard
- `update_scoreboard` →
- `create_message_board` → δημιουργεί και τοποθετεί το message
- `flip` → «γυρίζει» μια κάρτα
- `compare_tiles` → συγκρίνει τις κάρτες της `clicked_tiles`, αλλάζει το μήνυμα και το scoreboard ανάλογα με το αποτέλεσμα της σύγκρισης
- `check_king_queen` → ελέγχει αν οι 2 πρώτες κάρτες είναι Ρήγας-Ντάμα, αλλάζει το μήνυμα και μειώνει το `click_count` ώστε να επιλεγεί τρίτη κάρτα
- `change_player` → αλλάζει τον τρέχων παίκτη λαμβάνοντας υπόψη τις κάρτες που επιλέχθηκαν
- `save_game` → ενημερώνει το `current_state` και αποθηκεύει την τρέχουσα κατάσταση του παιχνιδιού
- `computer_turn` → υλοποιεί τον γύρο του computer (αν υπάρχει)
- `choose_third_card` → επιλέγει 3<sup>η</sup> κάρτα για τον computer στην περίπτωση Ρήγα-Ντάμας
- `check_game_end` → ελέγχει αν έχει ολοκληρωθεί το παιχνίδι, επιστρέφει `True` ή `False`
- `button_click` → καλείται με το κλικάρισμα μιας κάρτας, εμπλέκει τις `flip`, `check_king_queen`, `compare_tiles`, `change_player`, `save_game`, `computer_turn` και `check_game_end`

# Κλάση NewGame:

Παράδειγμα Εκτέλεσης(1 Παίκτης – Εύκολο):



# Κλάση ContinueGame(NewGame):

*Μέθοδοι:*

- `extract_data` → εξάγει τις πληροφορίες για τη συνέχεια του παιχνιδιού από το αποθηκευμένο GameState στιγμιότυπο
- `tile_reconstruction` → κατασκευάζει τη λίστα tiles και τη λίστα με το ιστορικό του computer(αν υπάρχει) με βάση τα δεδομένα
- `board_reconstruction` → τοποθετεί τις κάρτες στο ταμπλό

# Προκλήσεις/Δυσκολίες

---

- Η ανάθεση καθηκόντων σε ένα ομαδικό project, διαχείριση χρόνου.
- Η αναζήτηση παραδειγμάτων, documentation και tutorials για το tkinter και τη δημιουργία ενός προγράμματος με GUI
- Η εξοικείωση με το Git και το GitHub
- Ο τρόπος απεικόνισης και αποθήκευσης των οντοτήτων του παιχνιδιού(Λίστες; Λεξικά; Μπορούμε να αποθηκεύσουμε ένα Tk αντικείμενο με pickling;)
- Προγραμματισμός(Ποια η ροή του προγράμματος; Πώς θα πρέπει να γίνει η υλοποίηση;)

# Πηγές/Παραπομπές:

Γενικές πληροφορίες/tutorials για το tkinter:

- <https://www.geeksforgeeks.org/flipping-tiles-memory-game-using-python3/>
- <https://stackoverflow.com/questions/tagged/tkinter>
- <https://www.geeksforgeeks.org/python-gui-tkinter/>
- <https://www.activestate.com/resources/quick-reads/how-to-position-widgets-in-tkinter/>
- <https://docs.python.org/3/library/tkinter.html>
- <https://www.tutorialspoint.com/how-to-use-an-image-as-a-button-in-tkinter>

Tile Matching Game - Python Tkinter GUI Tutorial:

- <https://www.youtube.com/watch?v=tIMPVGSEEDw>

Let's Code Python: Memory Tile Game with GUI:

- <https://www.youtube.com/watch?v=-jlfRMbjCMc>

Create Graphical User Interfaces With Python And Tkinter:

- <https://www.youtube.com/watch?v=yQSEXcf6s2I>

Python Object Oriented Programming (OOP) - For Beginners:

- [https://www.youtube.com/watch?v=JeznW\\_7DIB0](https://www.youtube.com/watch?v=JeznW_7DIB0)

Serialize Python Objects With Pickle:

- <https://www.youtube.com/watch?v=qt15PnF8x-M>

Πηγή των .png εικόνων των καρτών της τράπουλας:

- <https://opengameart.org/content/playing-cards-vector-png>

Ευχαριστούμε για την προσοχή σας!

---