

## Вовед

Овој проект претставува игра наречена **Color Fill Puzzle**, во која корисникот треба да ја пополни таблата со бои според одредени правила. Играта има мени, логика за победа и пораз, звучни ефекти и интуитивна визуелизација користејќи ја библиотеката **Pygame**.

## Главни глобални променливи и иницијализација

На самиот почеток:

- Ја иницијализираме библиотеката **pygame** со **pygame.init()**.
- Ги дефинираме глобалните променливи, како што се:
  - **Димензии на екранот:** **SCREEN\_WIDTH** и **SCREEN\_HEIGHT**.
  - **Табла:** **GRID\_SIZE** (5x5 клетки).
  - **Димензии на клетките:** **CELL\_SIZE** и маргините **XMARGIN** и **YMARGIN**.
  - **Бои:** Листа од бои за клетките (**COLORS**), позадинската боја (**BG\_COLOR**), боја за текстови и копчиња (**BUTTON\_COLOR, TEXT**).
- Се создава прозорец за играта со **pygame.display.set\_mode()** и му се дава наслов.
- Звучни ефекти (**pygame.mixer.Sound**) и позадинска музика (**pygame.mixer.music**) се иницијализираат.

## Табла и логика за цртање

- **Мрежа (grid):**
  - Таблата е претставена како 2D листа (**grid**), каде секоја клетка може да биде **None** (празна) или да има боја.
  - Пример: **grid[row][col]** го означува редот и колоната.
- **Функција за цртање на таблата (draw\_grid):**
  - Се пресметува позицијата на секоја клетка врз основа на нејзините координати.
  - Ако клетката е празна, се прикажува сива боја; инаку, се прикажува нејзината боја.
  - Се користи **pygame.draw.rect()** за прикажување на клетките и нивните граници.

## Обработка на кликови

- **Функција handle\_click:**
  - Се определува која клетка е кликната врз основа на позицијата на глумчето.
  - Ако клетката е веќе обоена, корисникот не може да ја промени бојата.
  - Проверува дали корисникот може да ја обоји одредената клетка.
  - Се користат соседите на клетката (горе, долу, лево, десно) за да се спречи две соседни клетки да имаат иста боја.
  - Ако сосед има иста боја како предложената, потегот е невалиден.
  - Ако клетката е празна и потегот е валиден, клетката се пополнува со случајна боја од **COLORS**.

## Логика за победа и пораз

- **Функција `check_win`:**
  - Проверува дали сите клетки се пополнети. Ако сите клетки се пополнети и ги почитуваат правилата за соседите, играчот победува.
- **Game loss:**
  - Проверува дали скорот е нула, и има 3 пати шанси да згреши. Ако згреши повеќе од 3 пати, играчот губи.

## Прикажување пораки

- **`display_win_message` и `display_lose_message`:**
  - Печати порака ("You Win!" или "You Lose!") на средината на екранот.
  - Се пуштаат соодветни звуци (**`win_sound`**, **`lose_sound`**).

## Главно мени

- **Функција `pre_game_menu`:**
  - Прикажува мени со две копчиња: „Start Game“ и „Exit“.
  - Ако корисникот го избере „Start Game“, играта започнува.
  - Ако го избере „Exit“, играта завршува.

## Ресетирање на играта

- **Функција `reset_game`:**
  - Ја враќа мрежата во празна состојба (2D листа со **`None`** вредности).
  - Ова се користи при изборот на „New Game“ од менито по победа или пораз.

## Главната функција (`main`)

- **Потек на играта:**
  1. Се повикува **`pre_game_menu`** за да го прикаже главното мени.
  2. Играта продолжува во главната јамка:
    - Се црта таблата.
    - Се проверува дали играта завршила (победа или пораз).
    - Ако играчот победи или загуби, се појавуваат копчиња за „New Game“ и „Exit“.
  3. Секој клик на глумчето се обработува со **`handle_click`**.
- **Излез од играта:**
  1. Ако корисникот го затвори прозорецот или избере „Exit“, **`pygame.quit()`** го прекинува програмот.