

## **Opis struktury programu**

### **Moduły**

Kod jest podzielony na moduły `files.c`, `game.c`, `render.c`, `buttons.c`, `editor.c`, `run.c` i `main.c`. Odpowiadają one odpowiednio za: zapis i odczyt plików, obsługę rozgrywki, wyświetlanie, obsługę przycisków, obsługę trybu edycji, obsługę wszystkich elementów programu oraz uruchomienie programu.

### **Struktury danych**

Najważniejsze struktury danych to `button`, `slot` (przechowujące dane o przyciskach), `board` (przechowująca dane planszy), `tile` (przechowująca dane jednego pola na planszy) i `tile_type` (przechowująca dane typu elementu).

### **Najważniejsze funkcje**

- 1) `runBoard` – jest odpowiedzialna za obsługę rozgrywki, korzysta z większości funkcji pomocniczych w module `game.c`.
- 2) `initBoard` – tworzy pustą planszę o określonych rozmiarach.
- 3) `boardCopy` – kopiuje planszę.
- 4) `run` – najważniejsza funkcja obsługująca wszystkie funkcjonalności programu.
- 5) `renderBoard` – wyświetla planszę w trybie gry.
- 6) `renderBoardE` – wyświetla planszę w trybie edycji.
- 7) `renderGame`, `renderMainMenu`, `renderEditor`, `renderEditorSubmenu`, `renderEditorSubmenu2`, `renderLevelMenu` – odpowiadają za wyświetlanie w poszczególnych stanach programu.
- 8) `readFromFileE`, `readFromFileG`, `saveToFile` – odpowiadają za wczytywanie plansz z plików i zapisywanie plansz do plików.
- 9) `runEditor` – jest odpowiedzialna za obsługę trybu edycji.
- 10) `mouseOnButton` – sprawdza, czy kursor myszy wskazuje dany przycisk, analogicznie `mouseOnSlot`.
- 11) `showButton` – wyświetla przycisk, analogicznie `showSlot`.
- 12) `createButton` – tworzy nowy przycisk, analogicznie `createSlot`.