

Card Finder

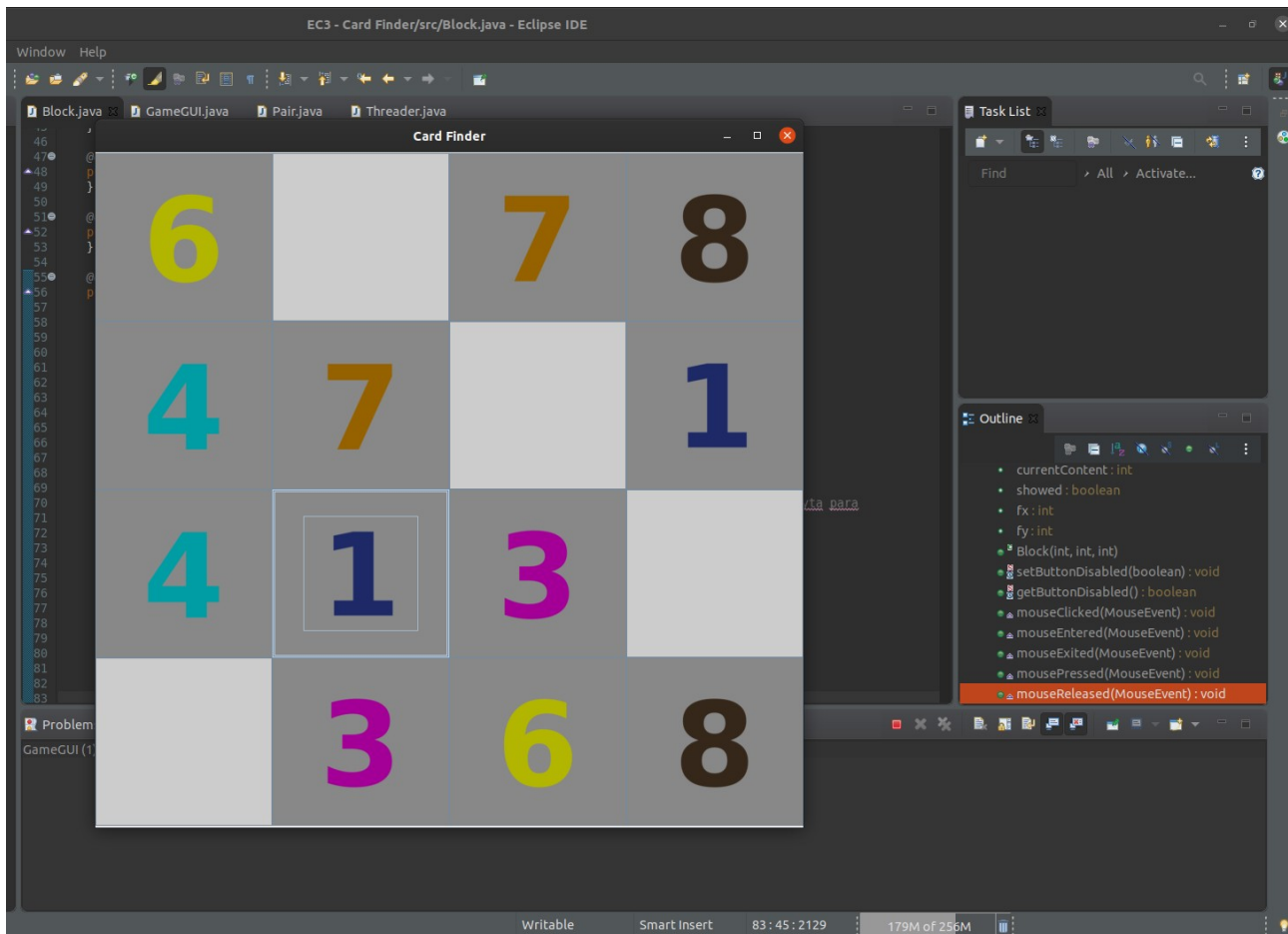
Projekt indywidualny z przedmiotu

Projektowanie w Języku Java

Wykonał Rafał Mikrut

1. Opis Gry

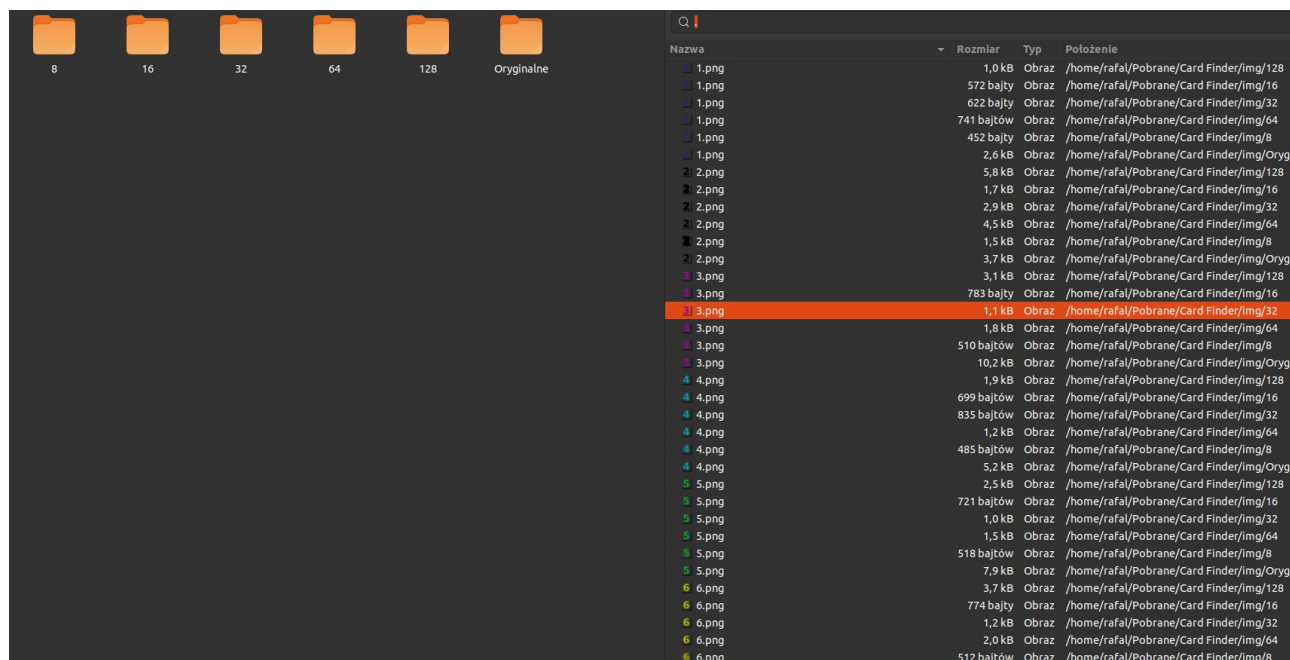
Card Finder to prosta gra w której musimy zapamiętać gdzie są położone dane liczby i połączyć je w pary.



2. Folder z plikami

Gra w systemie plików jest podzielona na dwa foldery:

- img – folder z obrazami
- src – folder z kodem źródłowym



W folderze img znajdują się różne obrazy liter(1-8) w różnych rozdzielczościach, gdyby akurat było potrzeba użyć mniejszych/większych obrazów.

3. Kod źródłowy

Kod jest podzielony na 4 pliki:

- Block.java – klasa odpowiedzialna za wykrywanie kliknięć i reagowanie na nie
- GameGUI.java – klasa tworząca i wyświetlająca GUI gry
- Pair.java – prosta klasa pokazująca aktualne pary, które są naciśnięte
- Threader.java – klasa umożliwiająca czekanie określonej ilości czasu bez potrzeby blokowania głównego wątku aplikacji.

Uruchomienie projektu powinno sprowadzać się jedynie do importu folderu/projektu do Eclipse.

W razie problemów, należy przenieść te dwa foldery w inne miejsce i spróbować ponownie.

Usunąłem pliki .classpath oraz .project, aby nie było problemów z kompatybilnością różnych wersji Javy

Niestety nie dałem rady spakować projektu jako .jar, ponieważ po spakowaniu nie wyświetlały się obrazy z folderu img.

4. Działanie gry

Na początku w klasie GameGUI cały interfejs jest tworzony - okno o wymiarach 800x800 oraz siatka 4x4 przycisków.

Następnie za pomocą specjalnej funkcji przyporządkowującej numery, dla danych przycisków przypisywana jest początkowa wartość, tak aby na mapie znajdowały się pary liczb np. [1,2,2,1]. Kolejno są tworzone przyciski i nadawana jest im wartość, lecz nie jest jeszcze wyświetlana.

Nieco wcześniej ma miejsce stworzenie pustej pary(zarówno x oraz y mają wartość -1, które nie jest prawidłowymi współrzędnymi).

Teraz oczekuje się na działanie gracza, który to musi nacisnąć na jakikolwiek przycisk. Przyciski obsługiwane są przez MouseListener zaimplementowany w tej klasie Button i wszystkie operacje z nimi związane są realizowane w funkcji mouseReleased.

Po kliknięciu na dowolny przycisk i jego odcisnięciu w funkcji mouseReleased sprawdzane jest czy nie występuje globalna blokada naciskania(o tym za chwilę), lub czy dany przycisk nie jest już widoczny.

Następnie wczytywana i wyświetlana jest ikona zawierająca obraz danej liczby, którą przycisk przechowuje.

Pierwsza liczba w parze jest zawsze poprawna dlatego jest ona wpisywana do statycznych zmiennych x1 oraz y1 bez żadnego sprawdzania.

Jeśli jednak została wprowadzona druga liczba w parze, to wtedy są możliwe dwa rozwiązania:

- Para jest prawidłowa:

W tym przypadku, zmienne showed obydwu przycisków są zmienione na true, co oznacza, że dane wartości są pokazane na mapie, przez co nie można ich kliknąć ponownie.

Dodatkowo tło zmienia się na nieco bardziej szare, przez co jest widoczne, które elementy zostały już sprawdzone.

Jako ostatnia rzecz, jest resetowana para – skoro jest poprawna to po wykonaniu powyższych metod nie jest ona nam już potrzebna.

- Para jest nieprawidłowa

Od razu na obu przyciskach pojawia się czerwony kolor, świadczący o błędzie.

Dodatkowo do pary dopisywana jest wartość bieżącego przycisku, ponieważ za chwilę będzie to potrzebne w Threaderze.

Następnie tworzymy nowy obiekt klasy Threader, ustawiamy jego czas na 0.5 sekundy, ustawiamy tryb na CLEAR_PAIR i go uruchamiamy

W klasie Threader – dziedziczącej po klasie Thread, znajduje się kilka stałych, które mogą sterować jego działaniem

Po uruchomieniu możliwe jest ustawienie flagi `DISABLE_BUTTONS`, które blokuje przyciski, czeka określony czas a potem je odblokowuje. To była wstępna koncepcja blokowania przycisków po błędnym kliknięciu, lecz trochę ją zmieniłem w późniejszym etapie.

Drugą flagą jest `ENABLE_BUTTONS`, które natychmiast odblokowuje te przyciski.

Ostatnią i najważniejszą jest flaga `CLEAR_PAIR`, która używana jest w przypadku, gdy użytkownik wybierze złą parę.

W tym przypadku wszelkie interakcje z przyciskami są blokowane na określony z góry czas.

Po upływie tego czasu przyciski są odblokowywane, jest im zmieniane tło na domyślne oraz ikony liczb znikają.

Na koniec pary są resetowane.

