

1. **Gra logiczna „Kółko i krzyżyk” dla dwóch graczy**
 - Program powinien umożliwiać przesuwanie kursora po dostępnych polach planszy. Po zatwierdzeniu ruchu gracza aplikacja musi weryfikować poprawność ruchu oraz kontrolować warunki zwycięstwa i remisu. Stan gry powinien być zapisywany i wczytywany z plików.
2. **Gra „Statki” (Battleship) z obsługą planszy graficznej**
 - Gracz powinien ustawiać własne statki za pomocą kursora. Program powinien zarządzać ukrytymi planszami obu graczy oraz realizować ataki z uwzględnieniem pełnej obsługi błędów i zapisu stanu gry.
3. **Gra planszowa „Młyneczek” dla dwóch graczy**
 - Program powinien umożliwiać ustawianie pionów na planszy, przemieszczanie kursora po dozwolonych pozycjach oraz realizację zasad gry wraz z graficznym oznaczaniem tworzonych „młynków”. Dane o przebiegu rozgrywki powinny być zapisywane w plikach logów.
4. **Gra logiczna „2048” z interfejsem graficznym**
 - Program powinien umożliwiać sterowanie grą poprzez klawisze kursora, realizując odpowiednie przesunięcia i scalanie liczb według reguł gry. Stan gry powinien być zapisywany do plików, umożliwiając kontynuację przerwanej rozgrywki.
5. **Gra „Snake” z dynamicznie zarządzaną długością węża**
 - Sterowanie wężem za pomocą kursora, dynamiczna alokacja pamięci na segmenty węża, zapis punktacji oraz poziomów do pliku konfiguracyjnego, pełna obsługa błędów graficznych oraz mechaniki gry.
6. **Program do gry „Memory” z interfejsem graficznym**
 - Aplikacja powinna umożliwiać odsłanianie pól z obrazkami za pomocą kursora. Realizacja logiki dobierania par, zapis wyników oraz postępu gry w plikach konfiguracyjnych, pełna kontrola błędów oraz wyraźne oznaczanie trafionych par.
7. **Symulator układania puzzli graficznych**
 - Program powinien umożliwiać manipulowanie fragmentami obrazka za pomocą kursora. Implementacja dynamicznych struktur danych dla przechowywania fragmentów puzzli, operacje zapisu i odczytu stanu gry oraz pełna obsługa interfejsu graficznego.
8. **Gra logiczna „Labirynt” z edytorem plansz**
 - Gracz porusza się kursorem po labiryncie próbując znaleźć wyjście. Program powinien umożliwiać ładowanie różnych labiryntów z plików oraz zapisywanie tras pokonanych przez graczy wraz z edytorem graficznym plansz.
9. **Gra „Pasjans” (Klondike Solitaire) z graficzną obsługą kart**
 - Pełna interakcja graficzna przy użyciu kursora myszy, dynamiczne zarządzanie taliami kart oraz operacje na plikach (zapisywanie postępu gry,

konfiguracji, statystyk gracza). Weryfikacja poprawności ruchów zgodnie z regułami gry.

10. Gra strategiczna „Connect Four” dla dwóch graczy

- Program powinien umożliwiać opuszczanie żetonów do wybranych kolumn planszy przy użyciu kursora. Dynamiczne zarządzanie strukturą planszy, zapis stanów gry oraz wyników w plikach tekstowych, graficzna reprezentacja przebiegu rozgrywki oraz automatyczna weryfikacja zwycięstwa lub remisu.

11. Graficzny menedżer plików

- Program powinien umożliwiać graficzną nawigację po katalogach, kopiowanie, przenoszenie, usuwanie oraz przeglądanie plików przy pomocy kursora. Wszystkie operacje powinny posiadać zaawansowaną obsługę błędów i zapis logów do pliku.

12. Program do zarządzania notatkami tekstowymi

- Graficzny edytor pozwalający na tworzenie, edytowanie, usuwanie i organizowanie notatek. Dynamiczna alokacja pamięci na notatki oraz zarządzanie nimi za pomocą kursora, zapis notatek w plikach konfiguracyjnych.

13. Prosty edytor graficzny

- Program umożliwiający podstawową edycję obrazów (rysowanie kształtów, wypełnianie kolorami, kadrowanie). Obsługa dynamicznych struktur danych do przechowywania obiektów graficznych, zapis wyników pracy do pliku graficznego, obsługa błędów interakcji użytkownika.

14. Graficzny monitor zasobów systemowych

- Wyświetlanie aktualnego zużycia pamięci RAM, CPU oraz przestrzeni dyskowej w czasie rzeczywistym. Program powinien zapisywać logi wykorzystania zasobów do plików oraz umożliwiać graficzną interakcję z użytkownikiem (np. wybór interwałów pomiarowych kursorem).

15. Prosty system alarmowy na mikrokontrolerze (embedded)

- System wykorzystujący sensory (ruchu, temperatury, światła) do wykrywania zdarzeń. Graficzny panel sterowania do konfiguracji progów alarmowych, zapisywanie zdarzeń w logach na kartę SD lub do plików, pełna obsługa błędów (np. brak czujnika, problem z pamięcią).

16. Program do wizualizacji danych pomiarowych z czujników (embedded)

- Pobieranie danych z czujników (np. temperatury, wilgotności, przyspieszenia), wizualizacja na ekranie graficznym LCD, dynamiczna alokacja buforów danych oraz ich zapis do pliku w pamięci EEPROM lub na karcie pamięci.

17. Program do zarządzania kalendarzem i harmonogramem wydarzeń

- Graficzny interfejs umożliwiający dodawanie, edytowanie, usuwanie zdarzeń za pomocą kursora. Dynamiczne zarządzanie strukturami danych

wydarzeń oraz przechowywanie harmonogramu w plikach tekstowych lub binarnych.

18. Graficzny odtwarzacz muzyczny

- Program umożliwiający przeglądanie katalogów z muzyką, odtwarzanie utworów, pauzowanie, przewijanie, dynamiczne zarządzanie playlistami. Zapamiętywanie stanu odtwarzania oraz konfiguracji w plikach tekstowych lub binarnych, interfejs graficzny z obsługą kursora.

19. Panel sterujący inteligentnym oświetleniem (embedded)

- Aplikacja umożliwiająca sterowanie systemem oświetlenia (włączanie, wyłączanie, regulacja jasności, kolorów) z poziomu graficznego interfejsu użytkownika. Program powinien dynamicznie zarządzać konfiguracją urządzeń świetlnych oraz zapisywać ustawienia do plików.

20. Interaktywny kalkulator graficzny

- Aplikacja do przeprowadzania podstawowych i zaawansowanych obliczeń matematycznych oraz wizualizacji wyników na wykresach. Dynamiczne zarządzanie pamięcią na dane liczbowe oraz wyniki obliczeń, graficzna reprezentacja danych wejściowych i wyników, zapis historii obliczeń do plików tekstowych lub binarnych.