

# Propozycje tematów zadań

## 1. WARCABY

Opracować program do gry w warcaby dla dwu graczy. Program ma umożliwiać przesuwanie kursora na zmianę po polach białych lub czarnych, wskazywanie początku końca ruchu. Uwzględnić bicia obowiązkowe oraz pion damkę. Po podjęciu przez gracza decyzji program powinien sprawdzać wykonalność ruchu i realizować przemieszczanie pionów.

**Ocena dostateczna:** Dopuszczalne są drobne błędy merytoryczne w działaniu programu oraz brak obsługi damki. Program nie musi być podzielony na moduły i odporny na błędy wprowadzone przez użytkownika.

**Ocena dobra:** Nie są dopuszczalne błędy merytoryczne w działaniu programu. Program powinien być też odporny na błędy: wprowadzone przez złośliwego użytkownika oraz mogące pojawić się w trakcie wykonywania instrukcji programu. Odporność na błędy oznacza, że program nie zawiesza się, informuje użytkownika o wystąpieniu błędu, jeśli pozwala na to charakter błędu program prosi użytkownika o wprowadzenie poprawek oraz wystąpienie błędu nie powoduje otrzymania niewłaściwych wyników.

**Ocena bardzo dobra:** Program powinien oferować rozgrywkę gracza z komputerem lub wykonywać operacje na plikach (zapis/odczyt stanu gry do/z pliku) Program powinien zostać podzielony w logiczny sposób na moduły ([http://www.cs.put.poznan.pl/wcomplak/BFILES/C\\_W\\_5.PDF](http://www.cs.put.poznan.pl/wcomplak/BFILES/C_W_5.PDF) „Programowanie w dużej skali”) – dopuszczalne jest jednorazowe poprawienie programu i przesłanie w przypadku niewłaściwego podziału na moduły.

## 2. REVERSI

Opracować program do gry w Reversi dla dwu graczy. Program ma umożliwiać przesuwanie kursora wyłącznie po polach niezajętych, po podjęciu przez gracza decyzji powinien sprawdzać jej dopuszczalność i realizować odpowiednie zmiany pól.

**Ocena dostateczna:** Dopuszczalne są drobne błędy merytoryczne w działaniu programu. Program nie musi być podzielony na moduły i odporny na błędy wprowadzone przez użytkownika. Wystarczy przyjąć standardowe zasady gry tzn.

plansza ma wymiary 8x8 i zaczynamy grę od standardowego ustawienia początkowego.

**Ocena dobra:** Nie są dopuszczalne błędy merytoryczne w działaniu programu. Program powinien być też odporny na błędy: wprowadzone przez złośliwego użytkownika oraz mogące pojawić się w trakcie wykonywania instrukcji programu. Odporność na błędy oznacza, że program nie zawiesza się, informuje użytkownika o wystąpieniu błędu, jeśli pozwala na to charakter błędu program prosi użytkownika o wprowadzenie poprawek oraz wystąpienie błędu nie powoduje otrzymania niewłaściwych wyników. Należy również umożliwić użytkownikowi wybór wielkości planszy (nie mniejsza niż 4x4 i nie większa niż 20x20).

**Ocena bardzo dobra:** Program powinien wykonywać operacje na plikach (zapis/odczyt stanu gry do/z pliku) Program powinien zostać podzielony w logiczny sposób na moduły ([http://www.cs.put.poznan.pl/wcomplak/BFILES/C\\_W\\_5.PDF](http://www.cs.put.poznan.pl/wcomplak/BFILES/C_W_5.PDF) „Programowanie w dużej skali”) – dopuszczalne jest jednorazowe poprawienie programu i przesłanie w przypadku niewłaściwego podziału na moduły.

### 3. MONOPOL

Opracować program do gry dla dwóch graczy w monopol. Opis gry i jej zasady można znaleźć na stronie internetowej: <http://www.kurnik.pl/netopol/zasady.phtml>.

**Ocena dostateczna:** Program powinien zawierać kilka posesji, z których każda składa się z kilku pól oraz pole startowe, po minięciu którego dany gracz powinien otrzymywać pewną sumę pieniędzy. Nie trzeba implementować żadnych „specjalnych” pól na planszy. Program nie musi być podzielony na moduły i odporny na błędy wprowadzone przez użytkownika..

**Ocena dobra:** Nie są dopuszczalne błędy merytoryczne w działaniu programu. Program powinien być też odporny na błędy: wprowadzone przez złośliwego użytkownika oraz mogące pojawić się w trakcie wykonywania instrukcji programu. Odporność na błędy oznacza, że program nie zawiesza się, informuje użytkownika o wystąpieniu błędu, jeśli pozwala na to charakter błędu program prosi użytkownika o wprowadzenie poprawek oraz wystąpienie błędu nie

powoduje otrzymania niewłaściwych wyników. Należy również dodać możliwość budowy domków/hoteli i przynajmniej jedno pole z losową szansą.

**Ocena bardzo dobra:** Program powinien wykonywać operacje na plikach (zapis/odczyt stanu gry do/z pliku) Program powinien zostać podzielony w logiczny sposób na moduły ([http://www.cs.put.poznan.pl/wcomplak/BFILES/C\\_W\\_5.PDF](http://www.cs.put.poznan.pl/wcomplak/BFILES/C_W_5.PDF) „Programowanie w dużej skali”) – dopuszczalne jest jednorazowe poprawienie programu i przesłanie w przypadku niewłaściwego podziału na moduły. Należy dodać kilka pól „specjalnych” na planszy (podatek, parking, więzienie, pomysł własny).

#### 4. CZWÓRKI

Opracować program do gry dla dwóch graczy w czwórki (czyli „cztery w rzędzie”). Opis gry i jej zasady można znaleźć na stronie internetowej: [http://en.wikipedia.org/wiki/Connect\\_Four](http://en.wikipedia.org/wiki/Connect_Four).

**Ocena dostateczna:** Dopuszczalne są drobne błędy merytoryczne w działaniu programu. Program nie musi być podzielony na moduły i odporny na błędy wprowadzone przez użytkownika. Wystarczy przyjąć, że plansza ma „klasyczne” wymiary 7x6.

**Ocena dobra:** Nie są dopuszczalne błędy merytoryczne w działaniu programu. Program powinien być też odporny na błędy: wprowadzone przez złośliwego użytkownika oraz mogące pojawić się w trakcie wykonywania instrukcji programu. Odporność na błędy oznacza, że program nie zawiesza się, informuje użytkownika o wystąpieniu błędu, jeśli pozwala na to charakter błędu program prosi użytkownika o wprowadzenie poprawek oraz wystąpienie błędu nie powoduje otrzymania niewłaściwych wyników. Należy również umożliwić użytkownikowi wybór wielkości planszy (nie mniejsza niż 5x5 i nie większa niż 15x15).

**Ocena bardzo dobra:** Program powinien wykonywać operacje na plikach (zapis/odczyt stanu gry do/z pliku) Program powinien zostać podzielony w logiczny sposób na moduły ([http://www.cs.put.poznan.pl/wcomplak/BFILES/C\\_W\\_5.PDF](http://www.cs.put.poznan.pl/wcomplak/BFILES/C_W_5.PDF) „Programowanie w dużej skali”) – dopuszczalne jest jednorazowe poprawienie programu i przesłanie w przypadku niewłaściwego podziału na moduły. Należy zaimplementować

dodatkowo możliwość gry z komputerem (opis wielu algorytmów można znaleźć w linkach na angielskiej stronie Wikipedii pod hasłem „Connect Four”).

## 5. SAPER

Opracować program gry Saper. Po wyświetleniu planszy i wylosowaniu położenia min, program umożliwi odkrywanie kolejnych pól i informuje o liczbie min znajdujących się w otoczeniu.

**Ocena dostateczna:** można przyjąć, że plansza ma określone wymiary i zawiera określoną liczbę min (np. 8x8 i zawiera 10 min), których położenie musi być wygenerowane losowo przez program. Program powinien pytać użytkownika o współrzędne (numer rzędu i kolumny) pola, które ma odsłonić. W przypadku trafienia na minę program powinien wyświetlić planszę z lokalizacją wszystkich min i komunikat o przegranej. W przypadku, gdy gracz odsłoni wszystkie wolne, niezaminowane pola powinna się wyświetlić plansza i komunikat o przegranej. Program nie musi być podzielony na moduły.

**Ocena dobra:** należy dodać menu, w którym użytkownik ustala wymiary planszy i ilość min (wraz z kontrolą – liczba min musi być mniejsza niż liczba pól planszy). Należy też dodać opcję zaznaczania min. Jeśli mina została źle zaznaczona, a gracz wpadnie na minę przegrywając to pole ze źle zaznaczoną miną powinno być oznaczone specjalnym symbolem. Program powinien być też odporny na błędy: wprowadzone przez złośliwego użytkownika oraz mogące pojawić się w trakcie wykonywania instrukcji programu. Odporność na błędy oznacza, że program nie zawiesza się, informuje użytkownika o wystąpieniu błędu, jeśli pozwala na to charakter błędu program prosi użytkownika o wprowadzenie poprawek oraz wystąpienie błędu nie powoduje otrzymania niewłaściwych wyników.

**Ocena bardzo dobra:** należy zaprogramować rekurencyjne odkrywanie pól (które wiadomo, że są wolne). Należy zaimplementować posługiwanie się kursorem w przechodzeniu pomiędzy polami zamiast podawania za każdym razem współrzędnych ruchu gracza. Program powinien wykonywać operacje na plikach (zapis/odczyt stanu gry do/z pliku) Program powinien zostać podzielony w logiczny sposób na moduły ([http://www.cs.put.poznan.pl/wcomplak/BFILES/C\\_W\\_5.PDF](http://www.cs.put.poznan.pl/wcomplak/BFILES/C_W_5.PDF) „Programowanie w

dużej skali”) – dopuszczalne jest jednorazowe poprawienie programu i przesłanie w przypadku niewłaściwego podziału na moduły.

## 6. OKRETY

Opracować program umożliwiający grę w okręty z komputerem. Gra powinna składać się z *fazy rozstawienia* – gracz ustawia okręty na planszy i *fazy gry* – gracz i komputer na przemian strzelają i próbują zatopić okręty przeciwnika.

**Ocena dostateczna:** Wystarczy przyjąć, że plansza ma wymiary 10x10, a okręty każdego gracza to 8 „jedynek”. Rozstawiając okręty program powinien zabronić ustawienia okrętu na polu graniczącym (także na skos) z innym, już wcześniej ustawionym okrętem. Komputer powinien ustawić okręty losowo, z zachowaniem reguły, że okręty nie mogą się stykać. W fazie strzelania komputer powinien strzelać losowo, omijając pola w które już wcześniej strzelał.

**Ocena dobra:** Plansza powinna mieć rozmiar 10x10 i każdy gracz ma następujące okręty: jedna „czwórka”, dwie „trójki”, trzy „dwójki”, cztery „jedyneki”. Przy rozstawianiu (zarówno gracza jaki i komputera) program powinien zabraniać ustawiania okrętów w niedozwolonym miejscu tj. graniczącym (także na skos) z już rozstawionym masztem. Przy rozstawianiu okrętów powinna być opcja wycofania całego aktualnie rozstawianego okrętu. Należy zaimplementować też ustawianie okrętów przez komputer losowo z zachowaniem reguł rozstawiania. W fazie strzelania komputer powinien strzelać losowo, omijając pola w które już wcześniej strzelał oraz graniczące (także na skos) z zatopionym wcześniej okrętem. Jeśli komputer trafił nie zatopiony jeszcze okręt gracza, to powinien strzelić w jedno (losowo wybrane) z możliwych pól, gdzie może być następny maszt tego okrętu.

**Ocena bardzo dobra:** Program powinien wykonywać operacje na plikach (zapis/odczyt stanu gry do/z pliku). Należy również umożliwić użytkownikowi wybór wielkości planszy - mała 10x10 lub duża 15x15. Należy umożliwić dodatkowo opcję gry dla dwóch graczy tj. przy rozstawieniu każdy gracz widzi tylko swoją planszę, a w fazie gry gracz widzi na swojej planszy rozstawione okręty (i to, w co trafił jego przeciwnik) oraz na planszy przeciwnika to, w co sam ten gracz trafił i/lub spudłował. Program powinien zostać podzielony w logiczny sposób na moduły ([http://www.cs.put.poznan.pl/wcomplak/BFILES/C\\_W\\_5.PDF](http://www.cs.put.poznan.pl/wcomplak/BFILES/C_W_5.PDF))

„Programowanie w dużej skali”) – dopuszczalne jest jednorazowe poprawienie programu i przesłanie w przypadku niewłaściwego podziału na moduły.

## 7. SZACHY

Opracować program do gry w szachy dla dwu graczy. Program ma umożliwiać przesuwanie kursora po polach planszy, wskazywanie początku i końca ruchu. Po podjęciu przez gracza decyzji program powinien sprawdzać wykonalność ruchu i realizować przemieszczanie figur.

**Ocena dostateczna:** Dopuszczalne są drobne błędy merytoryczne w działaniu programu. Program nie musi być podzielony na moduły i odporny na błędy wprowadzone przez użytkownika.

**Ocena dobra:** Nie są dopuszczalne błędy merytoryczne w działaniu programu. Program powinien być też odporny na błędy: wprowadzone przez złośliwego użytkownika oraz mogące pojawić się w trakcie wykonywania instrukcji programu. Odporność na błędy oznacza, że program nie zawiesza się, informuje użytkownika o wystąpieniu błędu, jeśli pozwala na to charakter błędu program prosi użytkownika o wprowadzenie poprawek oraz wystąpienie błędu nie powoduje otrzymania niewłaściwych wyników. Program powinien mieć zaimplementowane specjalne posunięcia: przemiana piona, roszada, remis przez trzykrotne powtórzenie pozycji, bicie w przelocie.

**Ocena bardzo dobra:** Program powinien wykonywać operacje na plikach (zapis/odczyt stanu gry do/z pliku – zapisując aktualną pozycję z uwzględnieniem danych określających, które pionki mogą w danej chwili dokonać bicia w przelocie i które królowie/wieże już się w tej partii poruszyły (od tego zależy czy można wykonać roszadę) i informację do kogo należy następne posunięcie). Program powinien zostać podzielony w logiczny sposób na moduły ([http://www.cs.put.poznan.pl/wcomplak/BFILES/C\\_W\\_5.PDF](http://www.cs.put.poznan.pl/wcomplak/BFILES/C_W_5.PDF) „Programowanie w dużej skali”) – dopuszczalne jest jednorazowe poprawienie programu i przesłanie w przypadku niewłaściwego podziału na moduły.