

Konspekt-scenariusz Modułu B2.5

 Stanisław Ubermanowicz



Temat: Losowanie i porównywanie



Nazwa implementacji: Gra „Papier-kamień-nożyce”



Opis implementacji: Realizacja gry losowej, polegającej na równoczesnym wyborze dwóch przedmiotów, z których jeden wygrywa według zasady: papier owija kamień, kamień tępi nożyce, a nożyce tną papier. Gra powinna mieć 3 przyciski z miniaturami przedmiotów. Kliknięcie oznacza wybór obiektu przez gracza i powoduje wylosowanie obiektu dla komputera jako przeciwnika. Pojawiają się dwa duże obrazki przedmiotów i zależnie od układu liczone są punkty za wygraną lub remis.



Proponowany czas realizacji: 90 minut



Cele:

a) ogólne (zadanie/przesłanie nauczyciela dla całych zajęć):

- » wdrażanie do pracy w środowisku programowania wizualno-obiektowo-zdarzeniowego;
- » czynnościowe kształtowanie właściwego rozumienia kluczowych pojęć infotechnicznych;
- » motywowanie do interakcji przy implementowaniu od podstaw prostej gry komputerowej;

b) szczegółowe: Uczennica/uczeń...

- » rozpoznaje zintegrowane środowisko programowania wspieranego interfejsem graficznym;
- » zna zasadę tworzenia na ekranie obiektów za pomocą GUI i modyfikowania ich właściwości;
- » zna główne bloki struktury kodu źródłowego oraz elementarne instrukcje języka FreePascal;
- » ma przyswojone i rozumie pojęcia: obiekty (widżety) i ich atrybuty, kod źródłowy, instrukcje;
- » odczuwa satysfakcję z tego, że wykonał implementację i poznał specyfikę gry losowej.

[**opcjonalnie**] Uczeń zaawansowany...

- » umie modyfikować wygląd i rozbudowywać funkcjonalność wytworzonej implementacji.



Materiał nauczania-uczenia się:

- » zintegrowane środowisko programowania: okna edycji, inspekcji obiektów i kompilacji;
- » tworzenie obiektów ekranowych gry: przycisków, obrazków i pól tekstowych na wyniki;

- » programowanie procedur obsługi zdarzeń inicjowanych przez kliknięcia gracza;
- » struktury kodu źródłowego – deklaracje, definicje, nazwy, zmienne, instrukcje;
- » procedury – losowanie, porównywanie, zliczanie, składowanie i transfer obrazków.

UWAGA: Zakres omawiania struktur kodu dobiera trener adekwatnie do możliwości percepcyjnych uczniów.



Metody, działania:

- » zajawka inspirująca – krótki pokaz przykładowej, estetycznie wykonanej gry P-K-N;
- » gra w parach uczniów, za pomocą układów dłoni symbolizujących papier, kamień i nożyce;
- » metoda problemowa – próba określenia, czy istnieje strategia zapewniająca wygraną;
- » metoda projektu – tworzenie na ekranie niezbędnych obiektów, wypełnionych obrazkami;
- » programowanie – procedury obsługi kliknięć, ukazywania obrazków i zliczania punktów;
- » gra z komputerem – sprawdzanie poprawności implementacji; próba różnych strategii.



Wskaźniki osiągnięcia celów (efekty): Uczennica/uczeń...

- » obsługuje zintegrowane środowisko programowania wspieranego interfejsem graficznym;
- » tworzy potrzebne obiekty za pomocą GUI i odpowiednio modyfikuje ich właściwości;
- » nazywa główne bloki struktury kodu źródłowego i elementarne instrukcje języka FreePascal;
- » trafnie operacjonalizuje i objaśnia pojęcia: obiekt, atrybuty, kod źródłowy, instrukcje;
- » prawidłowo wprowadza kod źródłowy i doprowadza do pełnego działania implementacji;
- » próbuje wygrać z implementacją sztucznej inteligencji – nielosowej strategii predykcyjnej.

[opcjonalnie] Uczeń zaawansowany...

- » prawidłowo modyfikuje wygląd i rozbudowuje funkcjonalności wytworzonej implementacji.

Czynności uczniów	Działania nauczyciela	Materiał
Oglądają krótką prezentację gry. Poznają zasadę gry.	Pokazuje grę i zachęca uczniów do przyjrzenia się zasadzie działania przykładowej implementacji.	Gra wyświetlana z projektora lub on-line: www.supergry24.pl/zagraj,w,gre,196,2.html
Grają parami z użyciem dłoni do wizualizacji wyboru. Analizują, czy istnieje strategia wygranej.	Zwraca uwagę na brak strategii zapewniającej wygraną w grze czysto losowej.	Pięść zaciśnięta jako <i>kamień</i> ; palec wskazujący i środkowy jako <i>nożyce</i> ; dłoń otwarta jako <i>papier</i> .

Rozpoznają elementy interfejsu zintegrowanego środowiska Lazarus & FreePascal.	Objaśnia istotę i funkcjonalności środowiska oraz sposób edycji i przełączania między oknami.	Środowisko programowania wizualno-obiektowo-zdarzeniowego: okna edycji, inspekcji i kompilacji.
Ćwiczą sposób umieszczania na <i>Oknie</i> obiektów-kontenerów obrazkowych i tekstowych.	Wspiera uczniów w tworzeniu na ekranie niezbędnych obiektów. Wskazuje źródła obrazków.	Pasek narzędziowy Lazarusa z <i>Paletą komponentów</i> – widżetów: TImage, TImageList i TLabel.
Ćwiczą nadawanie obiektom odpowiednich atrybutów: nazw, wielkości, położenia na ekranie.	Objaśnia sposób określania właściwości. Uzgadnia z uczniami nazwy i optymalne rozmiary.	Okno Inspektora obiektów Lazarusa z zakładką <i>Właściwości</i> .
Deklarują procedury i sprzęgają je z obiektami klikanymi. Definiują typy zmiennych.	Omawia struktury kodu źródłowego. Uzgadnia z uczniami nazwy procedur i zmiennych.	Okno <i>Edytora źródeł</i> w Lazarusie oraz Inspektor obiektów z zakładką <i>Zdarzenia</i> . Język FreePascal.
Piszą instrukcje realizujące losowanie i porównywanie liczb, wyświetlanie obrazków i napisów.	Objaśnia pojęcia <i>randomizacji</i> oraz <i>procedury obsługującej kliknięcie</i> . Omawia funkcję <i>GetBitmap(...)</i> .	Okno <i>Edytora źródeł</i> w Lazarusie. Język FreePascal. Okno dialogowe <i>Edytor ImageList</i> .
Uruchamiają implementację i sprawdzają jej poprawność.	Wyjaśnia potrzebę kompilacji kodu i wspiera w korygowaniu błędów.	Oferty poleceń w menu <i>Uruchom</i> i Okno <i>Komunikaty</i> w Lazarusie.
Grają z komputerem.	Omawia czysto losowy wynik gry.	Własna implementacja gry P-K-N.
Umieszczają swój projekt gry w e-Repozytorium. Opisują swe dokonania w e-Portfolio.	Formułuje i sprawdza zadania obligatoryjne dokumentowania wytworów i osiągnięć.	Internet, przeglądarka. Funkcje Serwisu e-Swoi
Poznają grę <i>online</i> , zawierającą elementy sztucznego intelektu.	Zwraca uwagę na implementację „uczącą się od ludzi grających”.	Gra z prognozowaniem ruchów: www.nytimes.com/interactive/science/rock-paper-scissors.html
Poznają inne wersje gry w sieci.	W razie trudności wspiera poprzez pytania naprowadzające.	Zaawansowane odmiany online: www.gry.jeja.pl/3813,papier-kamien-nozyce-25.html

**Zadania rozszerzające:**

- » Zmodyfikuj implementację tak, aby po wylosowaniu obiektów wyświetlane były obrazki ilustrujące efekt wygranej, tj.: owijanie kamienia, tępienie nożyc, przecinanie papieru.
- lub:
- » Zaprojektuj i wykonaj bardziej złożoną grę z większą liczbą obiektów (zob. np. w Wikipedii zasady gry: „Papier-kamień-nożyce-jaszczurka-Spock”).

Uwaga: Zadania rozszerzające przeznaczone są do realizacji poza zajęciami, lecz można podjąć je z grupą zaawansowaną.

Konspekt-scenariusz Modułu B2.6

 Stanisław Ubermanowicz



Temat: Rozrzucanie i porządkowanie



Nazwa implementacji: Układanka alfabetyczna



Opis implementacji: Realizacja gry logicznej typu puzzle, polegającej na porządkowaniu 24 liter alfabetu (A+X) ułożonych w tablicy 5x5, poprzez przemieszczanie liter z pól stykających do pola pustego. Gra powinna mieć Menu z trzema poziomami „Rozrzucenia” (łatwe, średnie, trudne) i z możliwością automatycznego „Uporządkowania” wszystkich liter jednym kliknięciem. Założeniem jest to, że pola z literami nie mają być przesuwane na ekranie, a efekt animacji uzyskuje się przez samą zmianę wyświetlanych liter na nieruchomych polach tekstowych.



Proponowany czas realizacji: 90 minut



Cele:

a) ogólne (zadanie/przesłanie nauczyciela dla całych zajęć):

- » wdrażanie do pracy w środowisku programowania wizualno-objektowo-zdarzeniowego;
- » wzbudzenie zainteresowania poznawaniem podstaw tworzenia prostej gry komputerowej;
- » czynnościowe kształtowanie właściwego rozumienia kluczowych pojęć infotechnicznych;
- » motywowanie do poszukiwania strategii wygranej, prowadzącej do rozwiązania układanki;

b) szczegółowe: Uczennica/uczeń...

- » ma przyswojone i rozumie pojęcia: kody liter, tablica, menu, obsługa zdarzeń;
- » zna zasadę kreowania kolekcji obiektów TPanel i nadawania im atrybutów z poziomu kodu;
- » umie uzupełnić fragment kodu źródłowego, wzorując się na fragmencie podobnym;
- » poszukuje, odkrywa i stosuje w praktyce strategię wygranej w układance alfabetycznej;
- » odczuwa satysfakcję z tego, że znalazł strategię wygranej i potrafi ułożyć wszystkie litery.

[**opcjonalnie**] Uczeń zaawansowany...

- » zna sposób definiowania i wiązania wywołań Menu z procedurami obsługi zdarzeń.

**Materiał nauczania-uczenia się:**

- » quasi-losowe rozrzucanie (układy rozwiązywalne); poszukiwanie strategii porządkowania;
- » programowanie obiektowe z obsługą zdarzeń sterowanych przez gracza za pomocą myszy;
- » struktury języka – iteracja w dostępie do elementów tablicy (pętla w pętli); indeksy (Tag); tworzenie obiektów TPanel.Create(); właściwości obiektów: Caption, Visible ...; kodowanie liter Char(); obsługa kliknięć: procedura ClickAction(); tworzenie opcji Menu i przypisywanie ich do procedur realizujących wybraną opcję.

**Metody, działania:**

- » zajawka inspirująca – krótki pokaz z wirtualną grą w puzzle przesuwane (np. z Internetu);
- » gra dydaktyczna – rozwiązywanie układanki 3x3 z użyciem 8 kartek z literami A+H;
- » metoda problemowa – sposób ustawiania ostatniego rzędu: układy nierozwiązywalne;
- » metoda projektu – dobór właściwości widżetów TPanel imitujących elementy układanki 5x5;
- » metoda ćwiczebna – uzupełnianie kodu: przenoszenie właściwości między obiektami;
- » gry logiczne – samodzielne rozwiązywanie układanek o różnych poziomach trudności.

**Wskaźniki osiągnięcia celów (efekty):** Uczennica/uczeń...

- » trafnie operacjonalizuje i objaśnia pojęcia: kody liter, tablica, menu, obsługa zdarzeń;
- » tworzy foremną macierz widżetów TPanel i modyfikuje ich atrybuty z poziomu kodu;
- » prawidłowo uzupełnia fragmenty kodu źródłowego i doprowadza do działania implementacji;
- » samodzielnie stosuje w praktyce strategię porządkowania w układance alfabetycznej;
- » chętnie rozwiązuje zadanie uporządkowania całej macierzy o rozmiarach 5x5.

[opcjonalnie] Uczeń zaawansowany...

- » sam prawidłowo tworzy komponenty Menu oraz wiąże je z procedurami obsługi zdarzeń.

Czynności uczniów	Działania nauczyciela	Materiał
Oglądają krótką prezentację. Poznają zasadę gry.	Pokazuje przykład gry i zachęca do przyjrzenia się zasadzie działania.	Pokaz z projektora lub <i>online</i> : gry.pl/graj/Jungle-Squares.html
Próbują rozwiązać układankę 3x3, przesuując kartki na wolne pola. Rozpoznają potrzebę rozrzucania, zamiast losowania.	Inspiruje uczniów do samodzielnego ułożenia wszystkich liter. Ukazuje brak rozwiązania po zamianie miejsc 2 ostatnich liter.	Po 8 małych kwadratowych kartek z literami od A do H (po jednej literze na kartce).

Analizują struktury Menu oraz poznają powiązania opcji Menu z procedurami ich obsługi.	Objasnia sposób tworzenia Menu jako komponenty i właściwości widżetu TMainMenu oraz obsługę zdarzeń.	Środowisko Lazarus & FreePascal. Okno przykładowej implementacji, okno dialogowe <i>Edytor Menu</i> oraz <i>drzewo Inspektora obiektów</i>
Poznają alternatywną metodę generowania z poziomu kodu wielu obiektów w macierzy 5x5. Poznają funkcję Char()	Omawia struktury kodu, zwłaszcza programowy sposób kreowania i rozmieszczania obiektów TPanel oraz sposób kodowania liter.	Kod przykładowej implementacji i okno Edytora źródeł Lazarusa. Język FreePascal.
Poznają sposób indeksowania obiektów przez atrybut Tag.	Wyjaśnia indeksowe sterownie procedurą obsługi wielu obiektów.	Okno <i>Edytora źródeł</i> w Lazarusie. Język FreePascal.
Uzupełniają luki w procedurach obsługi quasi-animacji obiektów.	W razie trudności wspiera w pisaniu brakującego kodu.	Okno <i>Edytora źródeł</i> w Lazarusie. Język FreePascal.
Uruchamiają implementację i sprawdzają jej poprawność.	Weryfikuje prawidłowość funkcjonowania gry.	Oferty poleceń w menu <i>Uruchom</i> i Okno <i>Komunikaty</i> w Lazarusie.
Umieszczają swój projekt w e-Repozytorium. Opisują swe dokonania w e-Portfolio.	Formułuje i sprawdza zadania obligatoryjne – dokumentowania wytworów i osiągnięć.	Internet, przeglądarka. Funkcje Serwisu e-Swoi
Ćwiczą układanie liter.	Naprowadza na strategię wygranej.	Własna implementacja gry.
Samodzielnie rozwiązują różne odmiany gier układanek.	Zachęca do ćwiczenia innych strategii. W razie trudności wspiera poprzez pytania naprowadzające.	Przykłady układanek <i>online</i> , np.: gry.pl/gry/przesuwane-ukladanki/przesuwane-ukladanki.html

UWAGA: Zakres omawiania struktur języka dobiera trener adekwatnie do możliwości percepcyjnych uczniów.

**Zadania rozszerzające:**

- » Rozbuduj układankę do wymiaru 6x6, zawierającą wszystkie 35 liter polskiego alfabetu, tj.: A, Ą, B, C, Ć, D, E, Ę ... Z, Ź, Ż (trudność wskutek nieciągłości kodów polskich liter).
- lub: » Zaimplementuj puzzle działające na zasadzie faktycznego przesuwania widżetów TImage, (inaczej niż tutaj, tj. nie na zasadzie przepisywania zawartości i odkrywania /ukrywania pól).

Uwaga: Te trudne zadania są przeznaczone do realizacji poza zajęciami, lecz można podjąć je z grupą zaawansowaną.

Konspekt-scenariusz Modułu B2.7

 Stanisław Ubermanowicz



Temat: Odkrywanie i stosowanie algorytmu



Nazwa implementacji: Wieże Hanoi



Opis implementacji: Wizualizacja strategii wygranej w grze decyzyjnej, polegającej na przenoszeniu obiektów o różnej wielkości między 3 cokołami tak, aby obiektu większego nie stawiać na mniejszym. Początkowo na pierwszym cokole znajduje się 5 obiektów, ustawionych jeden na drugim. Najmniejszy obiekt przemieszcza się w co drugim, nieparzystym ruchu, a podczas ruchów parzystych pozostają tylko jedyne możliwości ułożenia obiektu mniejszego na większym.



Proponowany czas realizacji: 90 minut



Cele:

a) ogólne (zadanie/przesłanie nauczyciela dla całych zajęć):

- » kształtowanie u uczniów umiejętności programowania wizualno-obiektowo-zdarzeniowego;
- » zapoznanie ze strategią wygranej, prowadzącą najkrótszą drogą do rozwiązania łamigłówki;
- » czynnościowe kształtowanie właściwego rozumienia kluczowych pojęć infotechnicznych;
- » wzbudzenie motywacji do przyswojenia skutecznego sposobu optymalnego działania.

b) szczegółowe: Uczennica/uczeń...

- » ma przyswojone i rozumie pojęcia: strategia, algorytm, animacja, wizualizacja;
- » zna zasadę ustalania położenia obiektów na ekranie w układzie współrzędnych;
- » umie uzupełnić fragmenty kodu źródłowego, wzorując się na strukturach podobnych;
- » dostrzega i stosuje w praktyce strategię wygranej w łamigłówce „Wieże Hanoi”;
- » odczuwa satysfakcję z tego, że zrozumiał algorytm i potrafi zrealizować zadanie.

[**opcjonalnie**] Uczeń zaawansowany...

- » umie zaprojektować implementację ilustrującą strategię gry „Wieże Hanoi”.



Materiał nauczania-uczenia się:

- » zintegrowane środowisko programowania: okna edycji, inspekcji obiektów i kompilacji;

- » poszukiwanie strategii wygranej poprzez obserwację, odkrywanie optymalnego algorytmu;
- » programowanie obiektowe, z procedurą animacji sekwencyjnej, sterowanej Timerem;
- » struktury języka – instrukcje: warunkowa [if... else...]; procedura Timer1Timer();
- » rozdzielczość (piksele), wymiary i współrzędne obiektów [Width, High, Left, Top].

Uwaga: Zakres omawiania struktur kodu dobiera trener adekwatnie do możliwości percepcyjnych uczniów.



Metody, działania:

- » zajawka inspirująca – przekaz legendy o niewyobrażalnie długotrwałej pracy mnichów;
- » gra dydaktyczna – próba rozwiązania łamigłówki (np. z użyciem 3 monet różnej średnicy);
- » metoda problemowa – próba odkrycia i opisu prawidłowości tworzącej strategię wygranej;
- » metoda projektu – tworzenie na ekranie obiektów imitujących 3 cokoły i 5 elementów wieży;
- » badanie – analiza struktur kodu źródłowego animacji obiektów z algorytmem iteracyjnym;
- » programowanie – uzupełnianie części kodu źródłowego w miejscach celowo zostawionych luk;
- » zabawa *online* – samodzielne rozwiązanie łamigłówki z pozycji pośredniej nieregularnej.



Wskaźniki osiągnięcia celów (efekty): Uczennica/uczeń...

- » trafnie operacjonalizuje i objaśnia pojęcia: strategia, algorytm, animacja, wizualizacja;
- » wpisuje właściwe parametry umiejscawiające obiekty na ekranie w odpowiednim miejscu;
- » prawidłowo uzupełnia fragmenty kodu źródłowego i doprowadza do działania implementacji;
- » opisuje werbalnie i optymalnie realizuje strategię wygranej w łamigłówce „Wieże Hanoi”;
- » chętnie rozwiązuje utrudnione zadanie uporządkowania z pozycji pośredniej nieregularnej.

[**opcjonalnie**] Uczeń zaawansowany...

- » projektuje prawidłowo od podstaw implementację ilustrującą strategię gry „Wieże Hanoi”.

Czynności uczniów	Działania nauczyciela	Materiał
Wysłuchują legendę o mnichach. Oglądają krótki pokaz układanki.	Opowiada pseudo-legendę i szybko przestawia klocki zabawki <i>Wieży</i> zgodnie z zasadą.	Opis legendy i strategii: lordya314159.livejournal.com/ 29140.html Zabawka <i>Wieża</i>

Próbują rozwiązać łamigłówkę np. z trzema różnymi monetami.	Inspiruje uczniów do samodzielnego ułożenia złotych krążków.	Po 3 obiekty o różnej średnicy (np. monety, krążki, tacki, klocki).
Oglądają pełną wizualizację gry i poszukują strategii wygranej.	Podczas animacji zachęca do prób odkrycia strategii. Naprowadza na zasadę ruchu najmniejszego krążka.	Wzorcowa implementacja z animacją układania krążków, wyświetlana z projektora.
Poznają optymalny algorytm najkrótszego rozwiązania.	Wyjaśnia kluczowe dwa kroki optymalnego algorytmu.	Zabawka-układanka <i>Wieża</i> lub inne 5 obiektów różnej średnicy.
Ćwiczą określanie wielkości obiektów i ich usytuowanie na współrzędnych ekranu.	Prosi o modyfikowanie projektu. Uzgadnia z uczniami optymalne rozmiary obiektów i ich położenie.	Środowisko Lazarus & FreePascal. Projekt wzorcowej implementacji. Okno <i>Inspektora obiektów</i> : atrybuty [Width, High, Left, Top]
Poznają mechanizm okresowego wywoływania procedur i obsługę zdarzeń cyklicznych.	Omawia struktury implementacji, a szczególnie rolę <i>widżetu Timer</i> i procedurę <code>Timer1Timer()</code> .	Projekt wzorcowej implementacji i okno <i>Edytora źródeł</i> Lazarusa. Język FreePascal.
Uzupełniają luki w procedurach obsługi animacji obiektów.	W razie trudności wspiera w pisaniu brakującego kodu.	Okno <i>Edytora źródeł</i> w Lazarusie. Język FreePascal.
Uruchamiają implementację i sprawdzają jej poprawność.	Weryfikuje prawidłowość funkcjonowania wizualizacji.	Oferty poleceń w menu <i>Uruchom</i> i Okno <i>Komunikaty</i> w Lazarusie.
Umieszczają swój projekt w e-Repozytorium. Opisują swe dokonania w e-Portfolio.	Formułuje i sprawdza zadania obligatoryjne – dokumentowania wytworów i osiągnięć.	Internet, przeglądarka. Funkcje Serwisu e-Swoi
Samodzielnie rozwiązują układy z pozycji pośredniej nieregularnej.	Zachęca do ćwiczenia strategii. W razie trudności wspiera poprzez pytania naprowadzające.	Implementacja gry <i>online</i> , np.: wipos.p.lodz.pl/zylla/games/hanoi5p.html

UWAGA: Początkujący uczniowie powinni tylko uzupełniać celowo usunięte obiekty i fragmenty kodu źródłowego.

**Zadania rozszerzające:**

- » Rozszerz kod źródłowy implementacji Wieże Hanoi w taki sposób, aby obiektów przenoszonych (tj. bloków tworzących wieżę) było więcej niż w pierwotnym projekcie.
- lub: » Zaprojektuj i wykonaj bardziej realistyczne obiekty graficzne bloków tworzących wieżę, a następnie podstaw je do implementacji w miejsce obiektów o uproszczonej grafice.

Formułujemy treści poleceń możliwych do dalszego wykonania przez ucznia samodzielnie bądź ze wsparciem; np. rozbudowa implementacji, inny sposób wykonania lub odmienny projekt o podobnej funkcjonalności. Te zadania są zasadniczo przeznaczone do realizacji poza zajęciami, lecz można podjąć je z grupą zaawansowaną.

Konspekt-scenariusz Modułu B2.8

 Stanisław Ubermanowicz



Temat: Namiastka sztucznej inteligencji



Nazwa implementacji: Gra logiczna NIM



Opis implementacji: Realizacja gry logicznej, w której chodzi o to, aby podczas naprzemiennego pobierania obiektów z jednego rzędu na planszy uniknąć konieczności zabrania obiektu ostatniego. Losowane są różne układy do 10 obiektów w każdym z trzech rzędów. Komputer ma zaprogramowaną strategię wygranej, dlatego rozpoczyna gracz, aby mieć szansę zwycięstwa.



Proponowany czas realizacji: 90 minut



Cele:

a) ogólne (zadanie/przesłanie nauczyciela dla całych zajęć):

- » kształtowanie u uczniów umiejętności programowania wizualno-obiektowo-zdarzeniowego;
- » zapoznanie z implementacją zawierającą logikę działań imitujących sztuczną inteligencję;
- » czynnościowe kształtowanie właściwego rozumienia kluczowych pojęć infotechnicznych;
- » wzbudzenie motywacji do działań twórczych, inferencyjnych i konstruktywistycznych.

b) szczegółowe: Uczennica/uczeń...

- » ma przyswojone i rozumie pojęcia: *warunki, operatory logiczne, sterowanie, indeksy*;
- » zna zasadę kreowania kolekcji obiektów TImage i nadawania im atrybutów z poziomu kodu;
- » umie uzupełnić fragmenty kodu źródłowego, wzorując się na strukturach podobnych;
- » wie, jak stosować w praktyce strategię wygranej poprzez analizę parzystości grup binarnych;
- » odczuwa satysfakcję z tego, że zrozumiał strategię i wygrywa z grającymi bezbłędnie.

[**opcjonalnie**] Uczeń zaawansowany...

- » umie zaimplementować grę NIM w wersji dla dwóch osób, bez kodu sztucznej inteligencji.



Materiał nauczania-uczenia się:

- » poszukiwanie strategii wygranej, prowadzącej do układów końcowych: 1-1-1, 2-2, 3-2-1;

- » programowanie obiektowe z obsługą zdarzeń sterowanych przez gracza za pomocą myszy;
- » instrukcje ukazywania /ukrywania obiektów oraz blokowania repetycji szybkich kliknięć;
- » struktury języka – indeksowanie obiektów TImage; sterowanie właściwością Visible; funkcje Random(), GetBitmap(); instrukcje if... then...; operatory: AND, XOR.

Uwaga: Zakres omawiania struktur języka dobiera trener adekwatnie do możliwości percepcyjnych uczniów.



Metody, działania:

- » zajawka inspirująca – pokaz zasad gry za pomocą rekwizytów lub prezentacji z projektora;
- » gra dydaktyczna – uczniowie grają parami, analizując końcówki z max. 6 rekwizytami;
- » metoda problemowa – próba odkrycia strategii wygranej w końcowej fazie gry;
- » metoda projektu – analiza procedury tworzenia obiektów z obrazkami w trzech rzędach;
- » metoda ćwiczebna – analiza procedur obsługi gry: losowanie, ukrywanie obiektów;
- » operacjonalizacja – zobrazowanie klucza do wygranej, liczenie w systemie dwójkowym;
- » gra logiczna – gra z komputerem, z zastosowaniem strategii parzystości grup binarnych.



Wskaźniki osiągnięcia celów (efekty): Uczennica/uczeń...

- » trafnie operacjonalizuje i objaśnia pojęcia: *warunki, operatory logiczne, sterowanie, indeksy*;
- » z poziomu kodu prawidłowo tworzy obiekty TImage lub modyfikuje ich atrybuty;
- » prawidłowo uzupełnia fragmenty kodu źródłowego i doprowadza do działania implementacji;
- » opisuje werbalnie i optymalnie realizuje strategię wygranej w końcowej fazie gry NIM;
- » chętnie stosuje w praktyce strategię wygranej poprzez analizę parzystości grup binarnych.

[opcjonalnie] Uczeń zaawansowany...

- » tworzy od podstaw prawidłowo działającą implementację gry NIM dla dwóch osób.

Czynności uczniów	Działania nauczyciela	Materiał
Oglądają prezentację. Poznają grę o strategii „ostatni przegrywa”.	Pokazuje przykład gry i zachęca do przyjęcia się zasadzie działania.	Wzorcowa implementacja gry NIM wyświetlana z projektora.
Grają parami analizując taktykę wygrania w układach końcowych. Odkrywają strategię wygranej przy małej liczbie bierek.	Zachęca do analizy możliwych przypadków pod koniec gry. Prosi o słowne opisywanie strategii podczas 2 i 3 ostatnich ruchów.	Po 6 małych przedmiotów (np. patyczki, bierki, kamyczki).

Rozpoznają obiekty na oknie gry: przycisk Start i napisy. Analizują kolekcję obrazków w TImageList.	Wspiera w analizie roli widżetów zastosowanych we wzorcowej implementacji gry.	Środowisko Lazarus & FreePascal. Okno przykładowej implementacji; okno dialogowe <i>Edytor ImageList</i> .
Utrwalają metodę generowania z poziomu kodu do 10 obiektów w macierzy 3 rzędów.	Omawia struktury kodu, zwłaszcza programowy sposób kreowania i rozmieszczania obiektów TImage.	Kod przykładowej implementacji i okno <i>Edytora źródeł</i> Lazarusa. Język FreePascal.
Poznają sposób sterowania liczbą obiektów widocznych na ekranie.	Wyjaśnia procedury obsługujące ukazywanie /ukrywanie obiektów.	Okno <i>Edytora źródeł</i> w Lazarusie. Język FreePascal.
Uzupełniają luki w procedurach kreowania i ukrywania obiektów.	W razie trudności wspiera w pisaniu brakującego kodu.	Okno <i>Edytora źródeł</i> w Lazarusie. Język FreePascal.
Uruchamiają implementację i sprawdzają jej poprawność.	Weryfikuje prawidłowość funkcjonowania gry.	Oferty poleceń w menu <i>Uruchom</i> i Okno <i>Komunikaty</i> w Lazarusie.
Umieszczają swój projekt w e-Repozytorium. Opisują swe dokonania w e-Portfolio.	Formułuje i sprawdza zadania obligatoryjne – dokumentowania wytworów i osiągnięć.	Internet, przeglądarka. Funkcje Serwisu e-Swoi
Starają się zrozumieć strategię prowadzącą do wygranej. Głośno proponują optymalny ruch dla gracza.	Wyjaśnia strategię: rysuje po kilka kresek w 3 rzędach, zaznacza pary grup binarnych i wymazuje obiekty niepasujące do par (różne układy).	Szkolna tablica, pisak lub kreda i gąbka do zmywania.
Ćwiczą w grze z komputerem zastosowanie strategii wygranej.	Wspiera w przypadkach trudności ze zrozumieniem strategii wygranej.	Własna implementacja gry.
Grają online w grę NIM w wersji „ostatni wygrywa”.	Zachęca do poznania implementacji z odwrotną strategią w Scratchu.	Gra online: scratch.mit.edu/projects/jeh-som/50329

**Zadania rozszerzające:**

- » Zmień wystrój graficzny opracowanej gry NIM według całkowicie własnego projektu.
- lub: » Rozbuduj grę NIM tak, aby losowane było w każdym z trzech rzędów do 16 obiektów.
- lub: » Zmień grę NIM tak, aby generowane były cztery rzędy obiektów i aby dla takich układów prawidłowo funkcjonowała procedura „sztucznej inteligencji” ze strategią wygranej.

Uwaga: Zadania te przeznaczone są do realizacji poza zajęciami, lecz można podjąć je z grupą zaawansowaną.