

REGULAMIN KONKURSU

na

implementacje programistyczne

Konkurs organizowany jest w ramach Projektu "Strategia Wolnych i Otwartych Implementacji jako innowacyjny model zainteresowania kierunkami informatyczno-technicznymi oraz wspierania uczniów i uczennic w kształtowaniu kompetencji kluczowych"

Organizator Konkursu

Fundacja Wolnego i Otwartego Oprogramowanie

Adresaci Konkursu

Konkurs kierowany jest do nauczycieli szkół podstawowych, gimnazjalnych i ponadgimnazjalnych oraz do studentów i pracowników naukowych i dydaktycznych z kierunków informatycznych i powiązanych, takich jak: elektronika, mechatronika, automatyka, robotyka, kierunki pedagogiczne związane z informatyką, elektrotechnika, itp.

Cel konkursu

Wzbogacenie procesu nauczania informatyki o treści zwiększające aktywność edukacyjną uczniów i pobudzające ich zainteresowanie naukami informatyczno-technicznymi.

Informacja o projekcie

Działania projektu skierowane są na wypracowanie innowacyjnych scenariuszy zajęć z informatyki opartych na wolnym i otwartym oprogramowaniu oraz programu nauczania-uczenia się informatyki (produkt finalny), w ramach projektu zostanie wydana publikacja „Strategia Wolnych i Otwartych Implementacji” (tytuł roboczy).

Wypracowanie Programu Nauczania-Uczenia się przebiega w trakcie trzykrotnego testowania. Wykonawca będzie realizował zadania po etapie wstępnego testowania, biorąc udział w przygotowaniu prototypu produktu, oraz po testowaniu prototypu, biorąc udział w przygotowaniu produktu finalnego.

W ramach testowania projekt zakłada realizację zajęć praktycznych w formie kół zainteresowań w 50 szkołach gimnazjalnych i 60 ponadgimnazjalnych. W kole weźmie udział średnio 10 uczniów. Zajęcia prowadzone będą przez profesjonalnych trenerów wspieranych przez szkolnych opiekunów kół zainteresowań.

Zadanie konkursowe

Opracowanie minimum dwóch implementacji programistycznych o różnym stopniu trudności. Implementacje realizowane będą podczas zajęć dla szkół podstawowych, gimnazjalnych i ponadgimnazjalnych. Obowiązujące języki programowania to: Lazarus, Scratch C, C++, Phyton, PHP i Java. Prostsze implementacje nie muszą zawierać kodów źródłowych, mogą stanowić wytwór programu (np. Inkscape).

Tematy implementacji muszą bazować na obowiązującej podstawie programowej dla typów szkół obejmowanych projektem, natomiast wprowadzane treści powinny mieć charakter innowacyjny i użytkowy, dostosowany do wieku odbiorców.

Pojedyncza implementacja ma służyć realizacji zajęć w wymiarze 30-60 minut i musi zawierać informację, dla jakiego typu szkoły jest przeznaczona. Każda z opracowanych implementacji musi być uzupełniona konspektem, którego wzór stanowi załącznik nr 1 do ogłoszenia o konkursie.

Nagrody w konkursie

Nagrodzona zostanie 16 najlepszych prac. Z autorami zwycięskich opracowań zostaną podpisane umowy o dzieło, w ramach których dopracują oni pod kątem metodycznym nadesłane prace zgodnie z wytycznymi zespołu projektu. Każda z umów opiewać będzie na 500 złotych brutto.

Nagrodzone prace (z podaną informacją o autorze) zostaną włączone w wypracowywany Program Nauczania-Uczenia się, który będzie upowszechniany na terenie całego kraju.

Zasady dostarczania i oceny prac konkursowych

1. Opracowane implementacje z dołączonym CV autora nadsyłać można w formie elektronicznej na adres: info@fwioo.pl do dnia 3 stycznia 2012 roku. Odbiór każdej z prac potwierdzony zostanie zwrotnym mailem. Jego nieotrzymanie oznacza, że praca nie dotarła do organizatora konkursu.
2. Do dnia 10 stycznia 2012 komisja w składzie: koordynator projektu, główny programista, specjalista ds. dydaktyki, specjalista ds. badań i ewaluacji wybierze zwycięzców, których lista zostanie ogłoszona na stronie www.e-swoi.pl. Wybrane prace wejdą w skład zestawów zajęć testowanych przez Organizatora Konkursu w szkołach.
3. Do dnia 13 stycznia 2012 z wybranymi autorami zostaną podpisane umowy o dzieło.

4. Do dnia 25 stycznia 2012 roku autorzy, pod kierunkiem specjalisty ds. dydaktyki dopracują implementacje i konspekty.
5. Wynagrodzenie dla autorów zostanie wypłacone po odebraniu przez zespół projektu dopracowanych implementacji. Odebranie pracy zostanie potwierdzone w formie e-maila.
6. Wszystkie nadesłane prace muszą być udostępnione na wolnej licencji Non commercial (a'la Uznanie autorstwa-Użycie niekomercyjne-Na tych samych warunkach 3.0 Polska). W oparciu o tę licencję opracowano autorski model udzielenia upoważnienia do wykorzystywania projektów lub elementów projektów programu nauczania, obejmujący zezwolenie na rozpowszechnianie, przedstawianie i wykonywanie utworu jedynie w celach niekomercyjnych oraz tak długo, jak utwory zależne będą również obejmowane tą samą licencją (więcej informacji o licencjach w projekcie można znaleźć pod adresem <http://www.e-swoi.pl/wiki/article/licencja/>).
7. Nienagrodzone implementacje (wybrane przez specjalistę ds. dydaktyki) umieszczone zostaną w serwisie e-Swoi.pl.

Postanowienia końcowe

1. Każde zgłoszenie musi zawierać minimum 2 implementacje o różnym stopniu trudności.
2. Organizator zastrzega sobie prawo do zmiany terminu rozstrzygnięcia konkursu, o czym niezwłocznie poinformuje uczestników, publikując stosowną informację na stronie www.e-Swoi.pl.
3. Zgłaszając pracę do konkursu, autor tym samym oświadcza, że praca została wykonana osobiście i nie jest obciążona żadnymi roszczeniami i innymi prawami osób trzecich.
4. Zgłoszenie udziału w konkursie jest jednoznaczne z akceptowaniem jego warunków.
5. Osoba kontaktowa: Małgorzata Kikiewicz, malgorzata.kikiewicz@fwioo.pl,
tel. 61 624 34 74

Załącznik nr 1**Struktura konspektu**

na konkretnym przykładzie, wraz z objaśnieniami co do oczekiwanej zawartości pól

Temat: Dostrzec i zrozumieć Algorytm

wskazujemy motyw przewodni zajęć, zagadnienie lub wiodące hasło, np. kluczowe pojęcie informatyczne;

Nazwa implementacji: Wieże Hanoi (A)

tytuł danego wytworu z oznaczeniem poziomu trudności (A lub B)

Opis implementacji: Wizualizacja strategii w grze decyzyjnej polegającej na przenoszeniu obiektów o różnej wielkości między trzema cokołami tak, aby obiektu większego nie stawiać na mniejszym.

syntetyczne ujęcie istoty użytkowego funkcjonowania wytworu, tj.: co implementacja ma wykonywać i czemu służyć; np. w tym przykładzie nie chodzi o możliwość grania, lecz o obejrzenie i zrozumienie optymalnej strategii wygranej;

Cele:

- zrozumienie i przyswojenie pojęć: *strategia, algorytm, animacja, wizualizacja;*
- zapoznanie ze strategią prowadzącą najkrótszą drogą do rozwiązania ciekawej łamigłówki;
- zaimplementowanie programu ilustrującego efektywny sposób realizacji zadania z wieżami;
- wzbudzenie satysfakcji z tego, że to komputer wykonuje żmudne czynności przestawiania figur;

wytyczamy planowane osiągnięcia z danej jednostki zajęciowej

Treści:

- poszukiwanie strategii wygranej, wypracowanie optymalnego algorytmu i jego wizualizacja;
- programowanie obiektowe, z procedurą animacji sekwencyjnej, sterowanej Timerem;
- struktury języka – instrukcje: iteracyjna [For...], warunkowa [If...], ruchu [Move...];
- rozdzielczość (piksele), wymiary i współrzędne obiektów na ekranie [Width, High, Left, Top];

lista zagadnień informatycznych lub mechatronicznych, wyznaczających merytoryczny zakres danej jednostki zajęć;

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 wyrażenia prawidłowości tworzącej strategię wygranej;
- metoda projektu – tworzenie na ekranie obiektów imitujących 3 cokoły i 5 elementów wieży;
- metoda ćwiczebna – pisanie kodu źródłowego animacji obiektów z algorytmem iteracyjnym;
- zabawa on-line – samodzielne rozwiązanie łamigłówki z pozycji pośredniej nieregularnej;

opisujemy fazy zajęć, metody ich realizacji oraz główne czynności niezbędne do wykonania implementacji;

Wymagania, zalecenia: Trener zachęca uczniów do prób samodzielnego rozwiązania łamigłówki z trzema obiektami. Na zwolnionej animacji z 5 obiektami inspirowanie do postrzegania i opisywania prawidłowości ruchu, a w razie trudności z odkryciem strategii, trener prosi o werbalny opis ruchu najmniejszego obiektu. Trener akcentuje znamiennej cykliczność i prostą zasadę dwóch kluczowych ruchów w strategii i na przykładzie tym objaśnia kluczowy problem – istotę tworzenia skutecznego algorytmu. Podczas programowania najważniejsze jest objaśnianie fragmentów procedury realizującej animację, dlatego w grupach początkujących warto wkleić część kodu źródłowego, zamiast wpisywania.

wskazujemy ew. założenia co do wstępnego przygotowania ucznia, nauczyciela lub pomocy dydaktycznych do zajęć; akcentujemy kluczowe problemy oraz podajemy trenerom wytyczne co do wskazówek i pytań naprowadzających;

Media i materiały: Każdy uczeń ma po 3 płaskie obiekty o różnej wielkości (klocki, monety lub plastikowe talerzyki). Warto, aby trener miał oryginalną Zabawkę-układankę Wieżę, składającą się z min. 6 krążków; język programowania obiektowo-zdarzeniowego

Legenda i objaśnienie algorytmu np.: lordya314159.livejournal.com/29140.html

Implementacja gry on-line np.: wipos.p.lodz.pl/zylla/games/hanoi5p.html

media rozumiane szeroko jako wszelkie wykorzystywane środki dydaktyczne (także proste, nieelektroniczne) oraz materiały źródłowe i pomocnicze (drukowane, audio, wideo, komputerowe, internetowe);

Efekty: Uczennica/Uczeń

- zna sposób kreowania prostych figur geometrycznych i definiowania ich właściwości;
- zna metodę ustalania położenia obiektów na ekranie w układzie współrzędnych;
- umie napisać procedurę dwuwymiarowej animacji obiektów graficznych na ekranie;
- potrafi zastosować w praktyce optymalną strategię zabawy z łamigłówką „Wieże Hanoi”.

ujmujemy praktyczne skutki zajęć w kategoriach realnych osiągnięć uczniów, jako rezultaty zoperacjonalizowane;

Zadania: Rozszerz kod źródłowy implementacji Wieże Hanoi b001 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;