

Mistrzowie kodowanie - Aktywności "Na start"

Intro	3
Wprowadzenie	3
Czego potrzebujemy?	3
Dlaczego podjąć wyzwanie i wprowadzać dziecko w świat programowania?	3
1. Czujnik czasu	5
Wprowadzenie	5
Opis wyzwania	5
Co to ma wspólnego z nauką programowania?	5
2. Ja, robocik	7
Wprowadzenie	7
Opis wyzwania	7
Co to ma wspólnego z nauką programowania?	8
3. Czekoladożercy	9
Wprowadzenie	9
Opis wyzwania	9
Co to ma wspólnego z nauką programowania?	9
4. Kubeczkowy zawrót głowy	10
Wprowadzenie	10
Opis wyzwania	10
Co to ma wspólnego z nauką programowania?	13
5. Opowiem Ci historię	14
Wprowadzenie	14
Opis wyzwania	14
Co to ma wspólnego z nauką programowania?	15
6. Zaprojektuj deser	16
Wprowadzenie	16
Podejmij wyzwanie	16
Krok pierwszy. Zebranie informacji o potrzebach.	16
Krok drugi. Usystematyzowanie informacji.	16
Krok trzeci. Przygotowanie projektu.	16
Co to ma wspólnego z nauką programowania?	16
7. Sortownia	18
Wprowadzenie	18
Podejmij wyzwanie	18
Co to ma wspólnego z programowaniem?	18
8. Maszyna do sortowania	19

Wprowadzenie.	19
Podejmij wyzwanie	19
Co to ma wspólnego z nauką programowania?	21
9. Rysowanie	21
Wprowadzenie.	21
Podejmij wyzwanie.	21
Co to ma wspólnego z nauką programowania?	24
10. Kamyczkowe zabawy	24
Wprowadzenie	24
Podejmij wyzwanie	24
Co to ma wspólnego z programowaniem?	26
11. Rozgrywki w kratkę (?)	26
Wprowadzenie.	26
Podejmij wyzwanie.	27
Co to ma wspólnego z nauką programowania?	30
12. Papierowe wędrówki	30
Wprowadzenie.	30
Podejmij wyzwanie.	30
Etap 1.	30
Etap 2.	31
Etap 3.	32
Etap 4.	32
Etap 5.	32
Etap 6	33
Co to ma wspólnego z nauką programowania?	34

Intro

Wprowadzenie

Wspólne odkrywanie świata programowania może przynieść wiele dobrych chwil całej rodzinie. Na początku tej interesującej przygody, nie potrzebujemy komputerów, tabletów, ani żadnych urządzeń. Zabawy z dzieckiem dają szansę na wiele ciekawych odkryć i dużo radości. Najprostsze pomysły pozwalają na to, aby dziecko odkrywało uniwersalne zasady świata komputerów.

Do wspólnej pracy możemy wykorzystać sytuacje dnia codziennego, gry i zabawy. Przejście kolejnych etapów ułatwią proponowane przez nas pomysły i aktywności.

Czego potrzebujemy?

Najważniejszy jest czas, który będziemy mogli spędzić razem z dzieckiem. Dzieci uwielbiają odkrywać, eksperymentować, stawiać pytania i poszukiwać odpowiedzi. Jeszcze większą radość sprawia im to, że możemy im w tych odkryciach towarzyszyć. Dlatego bezcenny jest wspólny czas spędzony z dzieckiem.

Dlaczego podjąć wyzwanie i wprowadzać dziecko w świat programowania?

Dzieciństwo to bardzo ważny okres życia, w którym prawidłowo dobrane aktywności mogą wspierać rozwój dziecka w różnych obszarach i umiejętnościach.

Ćwiczenia i zabawy z zakresu programowania mogą dobrze odpowiadać na jego potrzeby. Korzystając z ćwiczeń związanych z tematyką programowania dziecko może rozwijać myślenie w obszarze logiki i matematyki. Rozwijać pamięć, spostrzegawczość, koncentrację uwagi, umiejętność dostrzegania i wyodrębniania wzorów. Dodatkowo dziecko może poszerzać swoją kreatywność i doświadczać na swojej skórze efektów celowego działania.

Na kolejnych etapach rozwoju dziecko zwiększa swoje zrozumienie otaczającego świata, a także tego jak funkcjonują komputery i inne urządzenia cyfrowe, które go otaczają.

Zabawa z programowaniem wymaga także nabywania umiejętności współpracy, posługiwania się informacją, przewidywania oraz stawiania czoła wyzwaniom w postaci błędów, trudności.

Intro

Treści do infografiki

Korzyści z nauki programowania

Rozwój dziecka		
umysłowy	osobisty	społeczny
myślenie logiczne	kreatywność	komunikacja
myślenie matematyczne	sprawczość	umiejętność współpracy
pamięć	podejmowanie decyzji	wiara we własne możliwości
spostrzegawczość	umiejętność uczenia się	umiejętność radzenia w sytuacji trudnej
uwaga	korzystanie z informacji	rozumienie błędów jako informacji zwrotnej

1. Czujnik czasu

Wprowadzenie

Jedni twierdzą, że wystarczy zęby myć rano i wieczorem. Inni uważają, że warto po każdym posiłku. Są domy, w których zęby myje się przed śniadaniem, aby zmyć bakterie zgromadzone nocą. W innych stosuje się zasadę mycia po śniadaniu. A niektórzy myją zarówno przed, jak i po śniadaniu. Można używać szczoteczek klasycznych, elektrycznych, sonicznych. Bogaty jest również wybór dostępnych past do zębów, nici czy taśm dentystycznych. Samych metod mycia zębów znajdziemy kilka np.: Fonesa, Roll, Bassa, Stillmana, Chartersa... Do wyboru do koloru, prawie jak z językami programowania. Tych też jest wiele, wybieramy te, które nam odpowiadają oraz pasują do realizacji danego projektu.

Z całego tematu dotyczącego higieny jamy ustnej wybierzemy teraz jedno hasło, co do którego większość osób jest zgodna. A mianowicie czas, który w przypadku mycia zębów zwykle jest ustalany jako 3 minuty. I to właśnie czas będzie hasłem łączącym mycie zębów z programowaniem.

Opis wyzwania

Zabawa najlepiej wypada, gdy używamy zwykłych szczoteczek (ewentualnie możemy myć elektrycznymi, bez włączania ruchu głowicy). Próbuje myć zęby przez trzy minuty, jednak nie patrząc na zegarek. Zabawa polega na wyczuciu kiedy upłynie 180 sekund. Możemy położyć obok zegarek lub telefon na którym uruchamiamy opcję stoper. W drugim przypadku możemy bawić się wspólnie, ustalając, że osoba która kończy mycie zębów, zaznacza czas swojej decyzji zaznaczając na stoperze opcję 'runda'.

Czy udało się nam myć trzy minuty? Jak blisko/daleko byliśmy w trakcie podejmowania decyzji? Jakie przyjmowaliśmy strategie? Jak wyglądałby świat, gdyby maszyny działały opierając się o ludzką intuicję i poczucie czasu? Gdyby nie miały wbudowanych mechanizmów umożliwiających mierzenie oraz korygowanie odliczanego czasu?

Taką zabawę można powtarzać, sprawdzając czy udaje nam się zwiększyć precyzję naszego odczuwania upływającego czasu.

Co to ma wspólnego z nauką programowania?

Zdobycie nowych umiejętności wymaga czasu. Podobnie jest z nauką programowania. Nie od razu wynik naszych działań będzie taki jakbyśmy chcieli. Z reguły będzie on mniej lub bardziej odbiegał od spodziewanego rezultatu. Najważniejszy jednak

jest regularny trening. Z każdym kolejnym ćwiczeniem nowych umiejętności stajemy się coraz bardziej precyzyjni w tym co robimy.

Treści do infografiki - instrukcja do ćwiczenia

1. Uruchom stoper lub ustaw czasomierz na 3 minuty
2. Odłóż stoper lub czasomierz
3. Rozpocznij mycie zębów bez spoglądania na upływ czasu
4. Zakończ mycie zębów i zatrzymaj odmierzenie czasu
5. Sprawdź i zanotuj swój wynik
6. Kontynuuj swój eksperyment przez kilka dni

2. Ja, robocik

Wprowadzenie

Programy komputerowe są przygotowane w taki sposób, aby rozpocząć działanie po otrzymaniu konkretnego sygnału. Automatyczny włącznik światła może być zaprogramowany tak, aby uruchamiać oświetlenie o zmierzchu. W aplikacjach zainstalowanych na naszych telefonach, sygnałem inicjującym działanie może być dotknięcie palcem ikonki programu. Bywa, że urządzenie nie działa właściwie ponieważ w programie znajduje się błąd, który wymaga naprawy.

A jak to jest z nami? Niektórzy są rannymi ptakami, wraz z pierwszymi promieniami słońca rozpoczynają dzień, nie czekając na dźwięk budzika. Natomiast 'sowy' niekiedy... nierzadko... a nawet dość często... miewają kłopoty z porannym wstawaniem. Dzwoniący budzik daje sygnał do rozpoczęcia dnia. Wstajemy od razu czy przysypiamy ustawiając kilka drzemek? Co robimy najpierw? A co tuż przed wyjściem z domu? Jak długa jest lista wykonywanych przez nas czynności?

Opis wyzwania

Pobawmy się w poranek robotów. Działajmy według instrukcji przygotowanej poprzedniego dnia. Najlepiej żeby był to dzień wolny od pracy. Wybierzcie na przykład sobotę lub niedzielę.

Aby przygotować instrukcję dla robocika, trzeba przeanalizować krok po kroku poranną aktywność, ułożyć działania po kolei. "Po obudzeniu się, umyj się, ubierz i zjedz śniadanie" - brzmi niezłe.

Ale jak takie polecenie zrealizowałby robot? Zapewne zaraz po sygnale budzika, stanąłby przy łóżku i rozpoczął mycie zębów, aby po chwili nakładać sweter na zjadane kanapki... Aby zobaczyć jak mógłby wyglądać poranek robota, można zajrzeć do filmu prezentującego przygotowywanie kanapki według instrukcji:

Exact Instructions Challenge - THIS is why my kids want to kill me
https://youtu.be/cDA3_5982h8 - film w języku angielskim.

Zabawa w programowanie będzie polegała nie tylko na analizowaniu, planowaniu, ale i precyzowaniu komunikatów. Kiedy? Gdy się obudzę... gdy obudzi mnie mama... gdy zadzwoni budzik... Co? Wyłączę budzik... Wstanę... Pójdę... Dokąd? Do łazienki... do kuchni... Co zrobię? Umyję zęby... przygotuję śniadanie... Jak długo? 3 minuty... Tak długo, aż będę najedzona...

Dlatego zanim przejdziecie do działania wspólnie zastanówcie się nad kolejnymi krokami, aby były one precyzyjnie zaplanowane i zapisane.

Jeśli jesteśmy przekonani, że nasza instrukcja jest gotowa, możemy rozpocząć fazę testów. Dokładnie tak, jak ma to miejsce w przypadku realizowania prawdziwych

projektów, kolejnego poranka sprawdzimy, czy będąc robotem sprawnie udałoby się nam wyszykować. Jeśli znajdziemy fragmenty wymagające poprawek, zmodyfikujemy zapiski.

Co zrobić, gdy poranki nie sprzyjają takiej zabawie? Zapiszmy plan wieczornych rytuałów i zrealizujemy zabawę tuż przed snem.

Co to ma wspólnego z nauką programowania?

Umiejętność komunikowania się z komputerami wymaga od nas precyzji. Zaprogramowane przez nas obiekty zrobią dokładnie to, co zostało przez nas zapisane w jednym ze zrozumiałych przez nich języków. Dlatego warto ćwiczyć umiejętność precyzyjnego i spójnego zapisywania przepisów na własne działanie. Będzie to bardzo przydatne kiedy zaczniemy pisać swoje pierwsze programy.

Treści do infografiki

1. Zastanów się i wypisz jakie czynności podejmujesz podczas poranka lub wieczorem
2. Zapisz czynności w sekwencję precyzyjnych i następujących po sobie działań
3. Zrealizuj zapisany przepis na działanie
4. Doprecyzuj i koryguj błędy w swoim przepisie

3. Czekoladożercy

Wprowadzenie

- Co zrobić, aby zjeść czekoladę*?
- Otworzyć opakowanie, odłamać kawałek i włożyć do ust!

Hola, hola! My już wiemy, że jeśli na to pytanie będziemy odpowiadać jak mali programiści, to czeka nas rozbudowanie odpowiedzi o szczegóły. Podobnie jak w przypadku porannego lub wieczornego przepisu na działanie, zapiszcie instrukcję zjedzenia kawałka czekolady, jabłka lub czego chcecie. Po zapisaniu czytajcie swoje polecenia, a osoba z którą pracujecie niech wykonuje tylko to, co zostanie przez Was zapisane.

* *Albo dowolny posiłek :)*

Opis wyzwania

Skoro już wiemy co krok-po-kroku należy zrobić, aby zjeść czekoladę... Możemy sobie uatrakcyjnić zabawę. Przygotowujemy:

- produkt, o który toczy się gra (najlepiej w opakowaniu np.: czekolada),
- kostkę do gry, 2-3
- akcesoria odzieżowe (rękawiczki z pięcioma palcami oraz np.: czapkę, szalik, okulary do pływania),
- opcjonalnie sztuczce (nóż i widelec).

Gracze siadają lub stoją wokół stolika, na którym leży zapakowany produkt. Ustalamy kolejność (np.: zgodnie z ruchem wskazówek zegara) oraz osobę rozpoczynającą (np.: najmłodszy lub osoba która wyrzuciła na kostce sześć oczek). Celem gry jest skosztowanie produktu. Gracze kolejno dokonują rzutu kostką. Gracz, któremu udało się uzyskać na kostce sześć oczek, nakłada przygotowane akcesoria, a następnie próbuje otworzyć opakowanie produktu. W tym czasie pozostali gracze kolejno rzucają kostką. Jeśli któremś z nich uda się wyrzucić na kostce sześć oczek, gracz próbując zjeść deser, musi przerwać swoją aktywność, zdjąć akcesoria i je przekazać tej osobie, która wyrzuciła jako ostatnia sześć oczek.

Próby wygrania deseru trwają tak długo, aż któryś z graczy zdąży skosztować produktu (w wersji zaawansowanej, trzeba dodatkowo kawałek odkroić korzystając ze sztuczków).

Aby wyrównać szanse, w przypadku dużej różnicy wieku u graczy, można ustalić, że najmłodszy nakłada tylko rękawiczki, a rodzice dodatkowo czapkę i szalik.

Co to ma wspólnego z nauką programowania?

W każdej grze zawsze mamy pewne zasady, według których postępujemy i dzięki którym wiemy, jak może przebiegać gra i kiedy jest jej koniec. Od reguł i zasad gry bardzo blisko do warunków, którymi w programowaniu jasno określamy, co ma się wydarzyć, kiedy zaistnieją określone okoliczności. Przykładowo kiedy w czasie popularnego "Chińczyka"

jeden pionek wejdzie na pole na którym stoi pionek przeciwnika, ten drugi musi wrócić na pole startowe. Podobnie kiedy programuje jakiś obiekt w naszej grze, może zapisać, że jeżeli zbierze on określoną liczbę punktów zostanie przeniesiony do kolejnej planszy.

4. Kubeczkowy zawrót głowy

Wprowadzenie

Słodki deser dobrze jest czymś popić. Tu przyda się nam kubek*. A jeśli mamy kilka kubeczków (np.: plastikowych, kartonowych, jednorazowych), możemy wykorzystać je do zabawy w programowanie.

Coraz częściej roboty zastępują ludzi w różnego rodzaju zadaniach. Dzięki coraz bardziej precyzyjnym mechanizmom oraz programom potrafią wykonywać finezyjne i skomplikowane operacje.

Sami przez chwilę możemy też stać się robotem i zrealizować kod napisany specjalnie dla nas. Jednym z zadań może być budowanie konstrukcji z kubków. Przygotowany stosik kubków będziemy układać w wieżę**.

* *Kubeczki możemy zastąpić nakrętkami od butelek, klockami, etc.*

**<http://www.oswajamyprogramowanie.edu.pl/2016/09/zakubkowani.html>

Jaką zapisać instrukcję dla maszyny, aby wybudować taką wieżę/konstrukcję/budowlę? Bywa, że w pozornie prostym zadaniu czeka na nas niespodzianka. Czy instrukcję zapiszemy słowami, czy może symbolami? Czy udało nam się przygotować instrukcję ustawiania kubków bez pominięcia istotnych szczegółów? Czy nasza maszyna odczyta, kiedy chwycić, a kiedy puścić kubeczek? W jaki sposób zapiszemy obrót, a w jaki przesunięcie? Jak poradziliśmy sobie z zapisem ustawiania górnej warstwy?

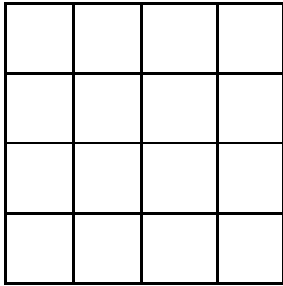
Jak to bywa w programowaniu, tak i w tej zabawie, wiele zależy od ustalonego przez nas kodu. Możemy kodować kolorami, symbolami lub pisać słowa. Ważne, aby ustalić co oznacza dane działanie. Jest to analogia do mnogości języków programowania, do pewnej umowy między nami, w jaki sposób potrzeby ludzi, zostaną przekazane maszynom wykonującym skrypt.

Opis wyzwania

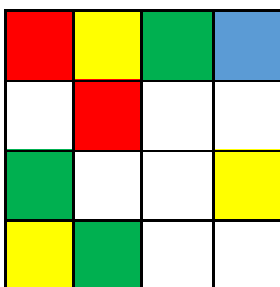
Jeśli dysponujemy większą liczbą plastikowych kubeczków (np.: 16, 25, 36 sztuk), możemy przygotować liczne zabawy. Wystarczy oznaczyć górną część kubka kolorowym znacznikiem, tak aby uzyskać zestaw kubeczków w kilku kolorach (np.: po 4 kubeczków w 4 kolorach).

Jedną z propozycji zabawy może być wówczas układanie sudoku. Gra polega na ustawieniu kubków w taki sposób, aby w kolumnie i wierszu dany kolor pojawił się tylko raz. Jak ustawić kubki by spełnić warunek niepowtarzalności kolorów? Jak szybko uda nam

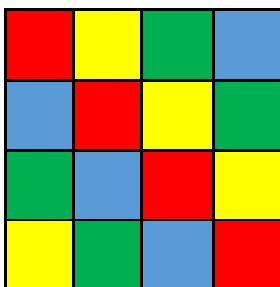
się ustawić właściwy układ? Czy stosujemy przy tym jakiś system czy stosujemy metodę eliminacji, zasadę prób i błędów?



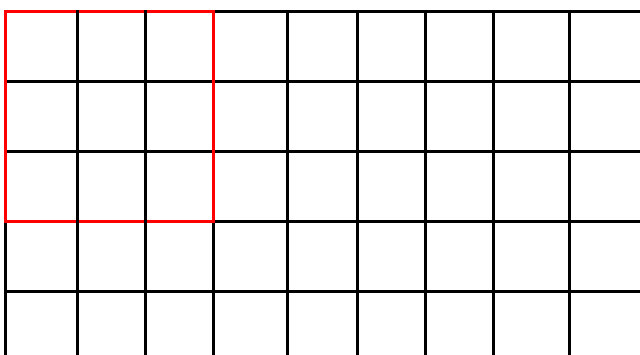
Czy mając cztery kolory możemy dokończyć uzupełnianie poniższego układu?

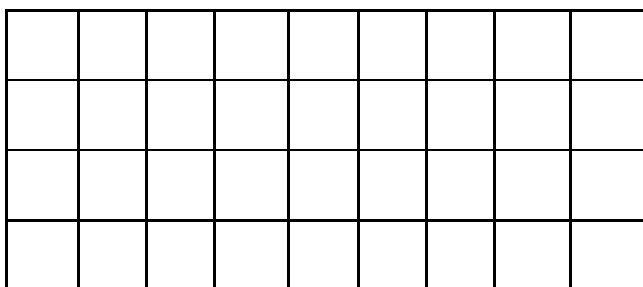


Możliwym rozwiązaniem jest poniższy układ:

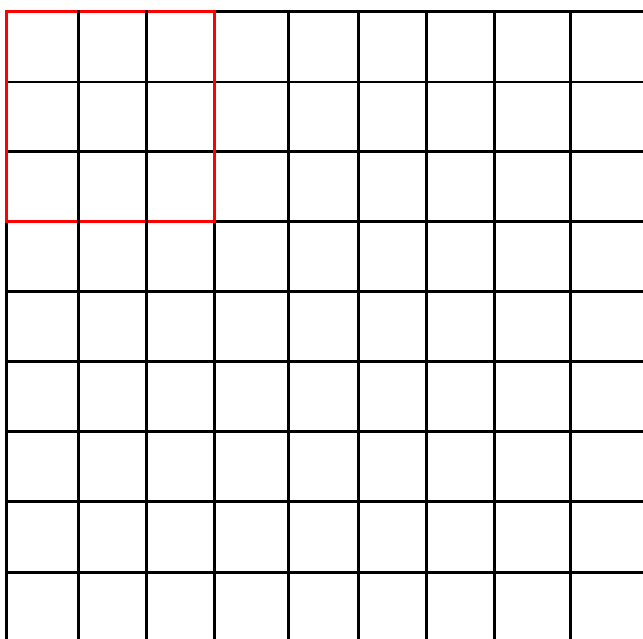


Jeśli dysponujemy wieloma kubeczkami (np.: po 9 kubeczków w 9 kolorach), zabawa w sudoku może przebiegać na tzw.: dużej planszy.

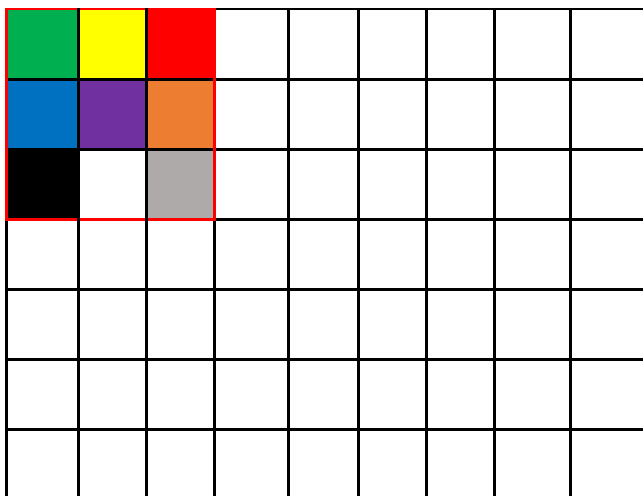




Mając czystą planszę, możemy ułożyć wiele układów, spełniających zadane warunki. Jednym z rozwiązań może być poniższy układ:



Mając do dyspozycji tak dużą planszę możemy dodać dodatkowy warunek. W każdym małym kwadracie 3x3, znajduje się zestaw 9 barw. Kolory nie powtarzają się zatem ani w kolumnie, ani w wierszu, ani w małym kwadracie.



Czy nasz system układania będzie analogiczny jak w przypadku wcześniejszych zabaw? Jak ustawić kubki, by spełnić warunek niepowtarzalności kolorów? Jak szybko uda nam się ustawić właściwy układ? Czy stosujemy tym razem wypracujemy jakiś system, czy stosujemy zasadę prób i błędów?

Co to ma wspólnego z nauką programowania?

Umiejętność dostrzegania powtarzalnych wzorów i dzielenia większych problemów na mniejsze jest ważna, jeśli chcemy radzić sobie z programistycznymi wyzwaniami. Podział rozwiązania na etapy pozwala lepiej panować nad pracą którą chcemy wykonać. Dostrzeżenie, że w złożonym problemie jest kilka mniejszych może być początkiem prowadzącym nas do jego rozwiązania. Zanim przystąpimy do zapisania kodu naszego programu dobrze podzielić zadanie na etapy. Dzięki temu zadanie wyda nam się bardziej przyjazne, a jego rozwiązanie będzie wydawać się bardziej dla nas dostępne.

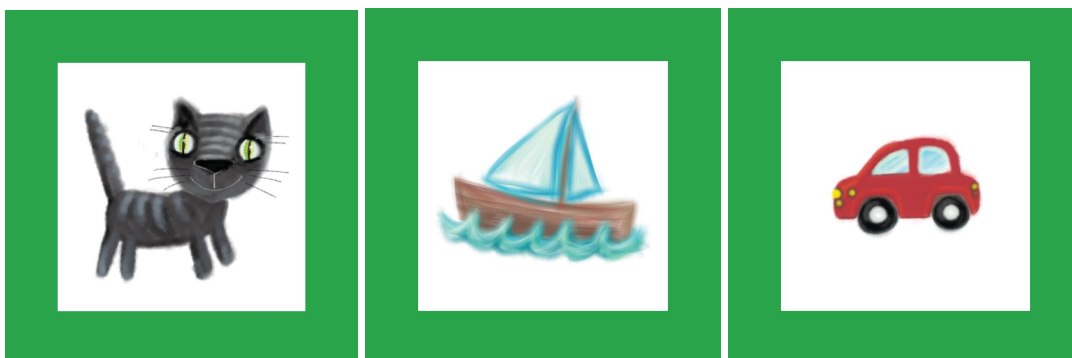
5. Opowiem Ci historię

Wprowadzenie

Programowanie może kojarzyć się z komputerem, robotem, maszynami. Zjrzyjmy dalej. Programowanie to też komunikacja, planowanie oraz związki przyczynowo-skutkowe. Coś jest początkiem, gdzieś czeka koniec. Po drodze snuje się opowieść, którą odczytuje i realizuje program. Jeżeli... Wówczas... Zawsze... Skrypty to szczególny rodzaj historii, przemyślanej, poukładanej tak, by jak najsprawniej omijać błędy, unikać niedopowiedzeń. Opowieść, która od momentu 'start', prowadzi maszynę do końca realizacji programu.

Opis wyzwania

W trakcie zabaw z dziećmi często opowiadamy historyjki. Taką aktywność możemy wykorzystać i w ramach zabaw z podstawami programowania. Do tej zabawy możemy wykorzystać dowolne ilustracje, na przykład zdjęcia z kolorowych czasopism dostępnych w domu. Po wycięciu grafik, możemy ułożyć je w linii. Wówczas opowiadamy historię tak, aby elementy pojawiały się kolejno, tak jak są ułożone.



A co jeśli zastąpimy jedną ilustrację? Czy potrafimy powtórzyć historię nie widząc jakiegoś elementu? A jeśli zamienimy miejscami grafiki? Czy uda nam się opowiedzieć historię, zachowując jej sens? A może będziemy chcieli stworzyć zupełnie nową opowieść? W jaki sposób każdy z nas interpretuje ilustrację? W czym nasze historie się różnią, a w czym są zbieżne?

W trakcie podróży możemy wykorzystywać obrazy mijanych reklam czy plakatów. W czasie wędrowki, mijając kolejno np.: informację o koncercie (data), reklamę produktów spożywczych (nabiał), reklamę firmy telekomunikacyjnej (kolorowe logo), plakat przedstawiający postać kobiety... możemy układać historię:

- Pewnego dnia, w magicznej krainie...
- Pewnego dnia, w magicznej krainie pełnej mleka...
- Pewnego dnia, w magicznej krainie pełnej mleka, pojawił się fioletowy wir...
- Pewnego dnia, w magicznej krainie pełnej mleka, pojawił się fioletowy wir. Oznaczało, to, że do krainy zbliża się wielka królowa...

Inną wersją zabawy może być układanie historii zawierających słowa wylosowane z książek. Każdy może zdecydować o warunkach wyboru słowa. Na przykład przy warunku: „Otwórz książkę na stronie numer X wybierz Y rzeczownik licząc od początku strony”, gdzie X i Y oznaczają liczbę wybraną przez nas, wskazaną w wyniku losowania lub określoną przez warunki (dzisiejsza data, numer domu, wiek dziecka, wiek rodzica, suma cyfr numeru telefonu itp.).

Skompletowane w ten sposób słowa stanowiąc będą bazę naszej opowieści.

Co to ma wspólnego z nauką programowania?

Kody które piszemy, aby powstał program są sekwencjami poleceń. Każda z nich będzie wykonywana zgodnie z kolejnością ich zapisania. Układanie historii, od lewej do prawej strony uczy pewnej struktury i uporządkowania. Poszczególne elementy układanych historii są ze sobą wzajemnie powiązane, tak jak poszczególne części kodu w zapisanym programie. Jeśli zabraknie jednej części historii, pozostałe nie będą już z sobą tak mocno powiązane. Tworząc nawet proste programy musimy pamiętać o tym, żeby uwzględnić w nich wszystkie operacje które są potrzebne do tego, aby dane urządzenie dla nas wykonało określone działanie.

6. Zaprojektuj deser

Wprowadzenie

Przygotowanie deseru zgodnie z przepisem to realizowanie algorytmu. Wprawdzie dość uproszczonego, ale jednak. Pewnie robot miałby kłopot, aby przygotować danie według przepisu ze zwykłej książki kucharskiej, bez dodatkowych informacji... Z drugiej strony my też, zanim nabierzemy wprawy w kuchni, potrzebujemy precyzyjniejszych instrukcji oraz wielu prób. A wszystko po to, aby przygotować deser idealny.

Zabawę można dodatkowo zrealizować bawiąc się w rozpoznawanie potrzeb związanych z deserem. Takie podejście dotyczy również projektowania produktów wirtualnych tak, aby wywoływały u użytkownika pozytywne doświadczenia. I choć deser nie będzie wirtualny, a całkowicie realny... to przecież ma w założeniu wywołać pozytywne wrażenia!

Podejmij wyzwanie

Krok pierwszy. Zebranie informacji o potrzebach.

Np.: Na co mamy ochotę? Jaki deser sprawi nam przyjemność? Co lubimy jeść? Czy są składniki, które nas uczulają, których powinniśmy unikać?

Krok drugi. Usystematyzowanie informacji.

Np.: Jakie są nasze potrzeby, a jakie możliwości? Czy wiemy jak ten deser wykonać? Czy musimy poszukać przepisu? Czy mamy wszystkie składniki? Czy musimy odwiedzić sklep?

Krok trzeci. Przygotowanie projektu.

Np.: Wybór przepisu. Weryfikacja dostępności składników. Jeśli nie mamy wszystkich produktów, przygotowanie listy zakupów. Dokonanie zakupu. gromadzenie składników. Przygotowanie deseru, np.: pieczenie ciasteczek według przepisu.

A co jeśli zdecydowaliśmy się na zbyt skomplikowany deser? Lub wybraliśmy przepis sprawdzony, ale danie nam nie wyszło? Cóż, czasem nawet najlepszy przepis nie uchroni nas od błędu. Od czasu do czasu, mimo naszych starań, coś pójdzie nie tak. Czy jednak wizja przypieczonej blaszki ciastek powstrzyma nas od dalszych prób? Raczej nie powinna, skoro znajdowanie i poprawianie błędów jest elementem programowania.

Co to ma wspólnego z nauką programowania?

Programowanie to nie tylko pisanie kodu. To również projektowanie i uzgadnianie tego, co dany program ma wykonywać i w jaki sposób ma być obsługiwany przez użytkowników. Im lepiej będziemy potrafili określić cele i potrzeby, tym lepiej będziemy w stanie wykorzystać swoje umiejętności. Ważny jest nie tylko kod, ale to dlaczego i dla kogo go tworzymy. Dyscyplina, która zajmuje się obszarem projektowania z uwzględnieniem potrzeb użytkownika to User Experience Design (UX).

7. Sortownia

Wprowadzenie

Programowanie? Przyjemne z pożytecznym! Przyjemnie przecież użytkuje się dobrze zaprogramowane maszyny, sprzęty. Sprawnie działająca aplikacja jest pożądana. A wchodząc z dziećmi podstawami programowania, kładziemy nacisk by na tym etapie nie było to żmudne pisanie kodu, a przyjemność i zabawa.

Co jeśli zabawy w programowanie mogą jednocześnie być pożyteczne?

Co jeśli zabawa splata się z czynnościami zwykle uznawanymi za obowiązki?

Podejmij wyzwanie

Tym razem zajmiemy się ubraniami. Nosimy je każdego dnia. Chcąc utrzymać odzież w czystości - pierzemy, suszymy, prasujemy, układamy w szafkach czy szufladach. Poszukajmy tu związku z programowaniem.

Zanim włożymy ubrania do pralki, sortujemy je. Mając doświadczenie, robimy to niemal automatycznie. Jednak początkujący chętnie korzystają na początku z informacji zapisanej na metce. Pozwólmy dzieciom dekodować umieszczone na metkach znaki i symbole. Co oznaczają te znaki? Które symbole są związane z praniem? Posegregowaną odzież możemy włożyć do pralki. Które symbole odnajdziemy na naszej maszynie? A które powtarzają się na opakowaniach środków piorących? Jak została zaprogramowana pralka, że właściwie odczytuje wytyczne np.: dotyczące temperatury, długości cyklu, odwirowania z sekwencji wybranych przycisków i ustawień pokręteł?

Jeśli nam wciąż mało, możemy przećwiczyć podobne instrukcje i oznaczenia symboli w trakcie prasowania. A później algorytm układania już czystych i wyprasowanych ubrań po praniu ;-)

Co to ma wspólnego z nauką programowania?

W różnych językach programowania różne polecenia mogą służyć do wykonywania podobnych instrukcji. Dlatego podobnie jak przy oznaczeniach związanych z prasowaniem i praniem, dobrze poznać ich przeznaczenie. W przypadku pralki lub ubrań wyjaśnienia dotyczące poszczególnych symboli możemy znaleźć w internecie. W świecie programowaniu mamy dostęp do dokumentacji poszczególnych poleceń, które jasno informują nas jak i kiedy możemy użyć danego kodu.

8. Maszyna do sortowania

Wprowadzenie.

Obecnie niewiele maszyn zajmuje się sortowaniem ubrań, czyli tym w co bawiliśmy się w poprzedniej aktywności. Ale samo hasło 'sortowanie' to jedno z podstawowych zagadnień informatyki, dlatego warto zatrzymać się przy nim na dłużej.

Sortowanie bąbelkowe jest prostą metodą, polegającą na porównywaniu dwóch elementów znajdujących się obok siebie. Taki sposób porządkowania można wykorzystać jeśli chcemy ułożyć elementy od posiadającego najmniejszą wartość, do tego o wartości największej.

Podejmij wyzwanie

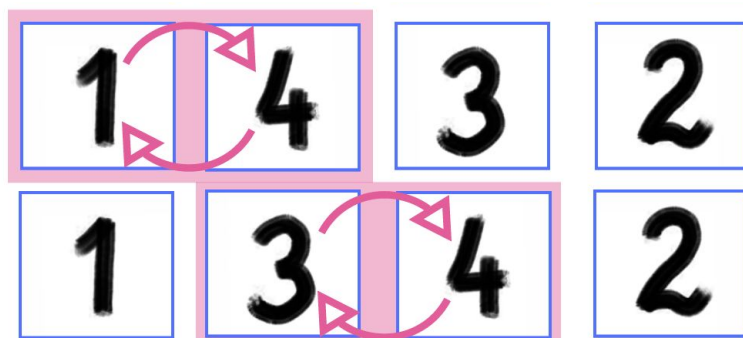
Wystarczy przygotować dowolne przedmioty, które możemy ze sobą porównywać i układać z narastającą wielkością określonego parametru. W warunkach domowych możemy wykorzystać:

- monety o różnych nominałach (1gr, 2gr, 10gr, 20gr, 50gr, 1pln, 2pln, 5pln)
- talię kart w Piotrusia (1 - 13)
- talię kart klasycznych (2 - As)
- zużyte bilety autobusowe z wydrukowanymi numerami serii, kartki z kalendarza itp.

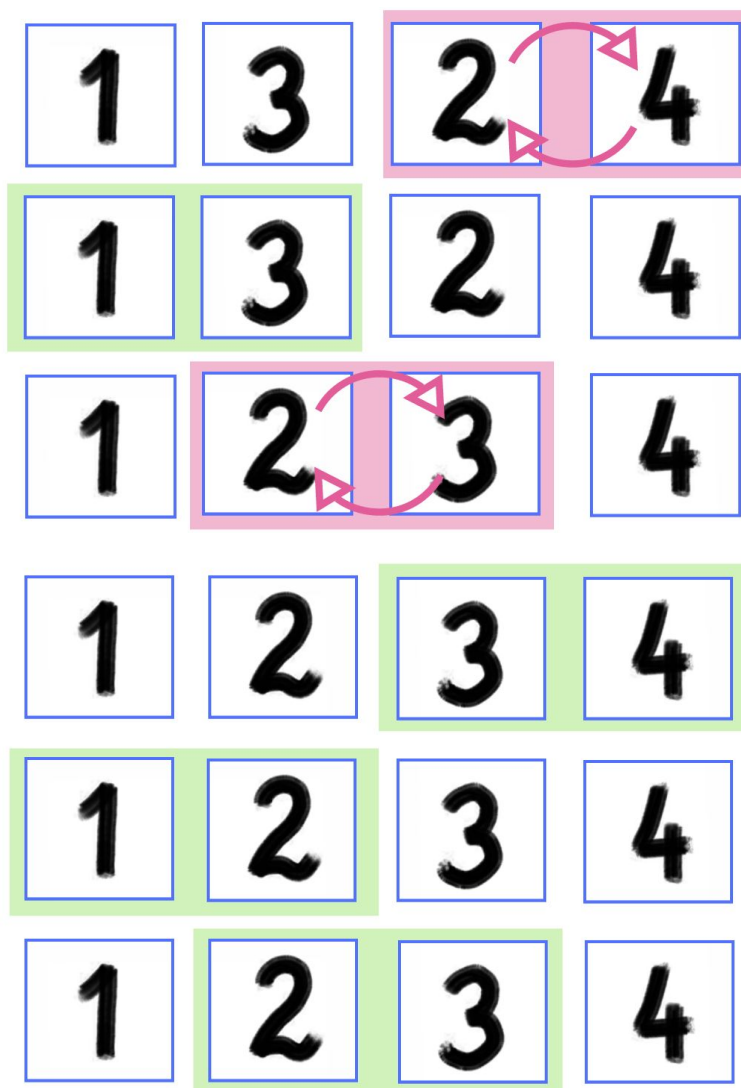
Przedmioty z danej kategorii tasujemy, mieszając ich układ, a następnie układamy na stole w jednym rzędzie.



Teraz należy prześledzić kolejno elementy porównując ich wartość. Nawet jeśli widzimy, że na pierwszym miejscu leży przedmiot o największej wartości (np.: moneta 5pln, karta as), niech nas nie korci droga na skrót... Nie przekładamy tego elementu od razu na koniec rzędu.



Porównujemy po kolei po dwa elementy. Jeśli przedmioty leżą zgodnie z przyjętą zasadą (od mniejszego, do większego), przechodzimy do porównania kolejnej pary. Jeśli przedmioty nie leżą zgodnie z przyjętą zasadą, zamieniamy miejscami tylko te dwa elementy.



Nie cofamy się... Zamieniamy miejscami tylko dwa porównywane przedmioty, kierując się od lewej do prawej strony. Po dotarciu do końca rzędu rozpoczynamy porównywanie par ponownie od lewej strony. Czynność powtarzamy do momentu, gdy uda się nam prześledzić cały rząd bez zamieniania miejscami przedmiotów.

W ilu ruchach udało się sortowanie?

Co to ma wspólnego z nauką programowania?

"Algorytmy sortowania są stosowane w celu uporządkowania danych, umożliwienia stosowania wydajniejszych algorytmów i prezentacji danych w sposób czytelniejszy dla człowieka."*

Gdzie tu jest ułatwienie, skoro nie można układać na skrót? Komputery muszą codziennie radzić sobie z niewyobrażalną liczbą danych. Porządkują i wyszukują dane, które są bardziej złożone i rozbudowane. W takim przypadku tego typu metoda jest ułatwieniem. Szczególnie wtedy kiedy nie jesteśmy w stanie na pierwszy rzut oka ocenić miejsca położenia danego obiektu lub wartości.

*<https://pl.wikipedia.org/wiki/Sortowanie>

9. Rysowanie

Wprowadzenie.

Komputery umożliwiają ludziom komunikację, są pomocne w pracy i dostarczają rozrywki. Czy zastanawialiście się jak to jest, że komputery dokonują szybko trudnych obliczeń, sortują w mgnieniu oka listę plików na kilka sposobów? Ot, na przykład takie zdjęcia z wakacji. Nierzadko folder zawierający kilkaset zdjęć możemy jednym kliknięciem ułożyć według daty wykonania, modyfikacji albo wielkości fotografii. A czy nie jest magicznym fakt, że komputer pokazuje nam sfotografowane kolorowe plaże... Tętniące życiem miasta... Szumiące zielenią lasy... posługując się tylko dwoma cyframi? Zero i jedynka, czyli system binarny. Tylko tyle i aż tyle!

Podejmij wyzwanie.

Na kartce w kratkę przygotowujemy dla każdego gracza planszę. Może mieć wymiary dowolnego prostokąta, ale na początku może to być kwadrat o wymiarach 10x10 kratek (czyli 5x5cm). Cała plansza ma zatem 100 kwadratowych pól. Tak jak w grze w statki, każdą kolumnę oznaczamy na górze literami (A, B, C, D itd), a każdy wiersz oznaczamy liczbą (1, 2, 3, 4 itd). W ten sposób każda ze stu krater naszej planszy ma określone, niepowtarzalne koordynaty (np.: A1, D4). I choć mogłoby się wydawać, że hasło

określające współrzędne jest trudne, to wiele dzieci spotyka się z nawią “koordynaty” w różnych grach. A te, które do tej pory nie miały okazji zetknąć się z takim określeniem, dość szybko, intuicyjnie przyjmują wyjaśnienie.

	A	B	C	D	E	F	G	H	I	J
1										
2										
3										
4										
5										
6										
7										
8										
9										
10										

Teraz jedna osoba zapisuje na kartce litery, druga osoba liczby.

Jeśli plansza gry ma powierzchnię 10x10 krater, możemy umówić się, że gracze wybiorą 10 liter lub 10 liczb. Zarówno litery jak i liczby mogą pojawiać się w dowolnej kolejności. Mogą się powtarzać. Dopuszczalne jest nawet rozwiązanie, w którym gracz zadecyduje, że wybiera dziesięć takich samych liter (np.: C, C, C, C, C, C, C, C, C, C) . Albo po kolei wypisze litery oznaczające kolumny (A, B, C, D, E, F, G, H, I, J). Można stosować dowolne kombinacje (np.: A, E, B, B, I, A, itd). Podobnie z liczbami.

[film]

Po zapisaniu swojego wyboru, gracze łączą swoje propozycje w kolejności. Wybrane przez jednego gracza litery, np.: A, E, B, B, I, A, C, A, H, H, zestawiamy z zapisanymi przez drugiego gracza liczbami np.: 3, 7, 9, 9, 5, 1, 10, 1, 1, 2. Uzyskujemy w ten sposób koordynaty np.: A3, E7, B9, B9, I5, A1, C10, A1, H1, H2.

Teraz każdy z graczy koloruje na swojej planszy określone kratki. Przy takim wyborze liter i liczb powtórzony jest koordynat B9, zatem po pokolorowaniu według instrukcji uzyskujemy planszę z zamalowanymi 9 kratkami:

	A	B	C	D	E	F	G	H	I	J

1	■							■		
2								■		
3	■									
4										
5									■	
6										
7					■					
8										
9		■								
10										

Teraz czas na drugą część zabawy. Do tak przygotowanej planszy gracze dorysowują kolejne kratki, aby stworzyć własny rysunek. Możemy umówić się, że kolorujemy np.: tylko dodatkowych 20, 30, 40 pól, ale tak aby razem z wcześniej pokolorowanymi kratkami stworzyć rysunek... Może to być rysunek dowolnej rzeczy lub wspólnie określonej kategorii Np.: "pojazdy", po domalowaniu czterdziestu ciemnoszarych pól powstaje uproszczony traktor.

	A	B	C	D	E	F	G	H	I	J
1	■	■	■	■				■		
2	■			■				■		
3	■			■				■	■	
4	■	■	■	■				■	■	
5	■	■	■	■	■	■	■	■	■	
6	■	■	■	■	■	■	■	■	■	
7	■	■	■	■	■	■	■	■	■	
8										
9		■						■		
10										

Czy udało się tak wypełnić planszę, aby powstał rysunek? Czy wcześniej pokolorowane pola przeszkadzały czy były pomocne? Czy są takie kratki, które zostały pokolorowane na obu planszach?

Zabawę można kontynuować kodując własny rysunek albo dodając więcej barw.

Jeśli potrzebujecie więcej zabaw na ten temat, możecie odwiedzić stronę [link <http://csunplugged.org/wp-content/uploads/2014/12/A2.pdf>] Znajdują się tu proste zabawy wprowadzające do zagadnień związanych z systemem dwójkowym, zwanym binarnym.

Co to ma wspólnego z nauką programowania?

W tym ćwiczeniu możesz odkryć jedną z największych tajemnic cyfrowego świata. Wszystkie dane, które przetwarzają komputery i inne urządzenia elektroniczne reprezentowane są przez dwie wartości: zero i jeden. Kiedy mamy do czynienia z wartością “jeden” przez małe transformatory, z który zbudowany jest np. procesor komputera przepływa prąd, kiedy “zero” nie ma przepływu prądu. W dużym uproszczeniu możemy sobie wyobrazić, że serce komputerów złożone jest z malutkich żarówek, które albo się zapalają, albo gasną. Komputery potrafią te zmiany odnotowywać bardzo szybko, dzięki temu w błyskawiczny sposób potrafią wykonywać tak wiele różnorodnych operacji związanych z obrazem (grafiką), dźwiękiem, przechowywaniem i wyszukiwaniem informacji, itd.

10. Kamyczkowe zabawy

Wprowadzenie

Podczas spaceru możemy zebrać materiały do kolejnej zabawy w programowaniu. Wystarczy zebrać kilka- kilkanaście kamyczków. Jakich? Dowolnie, nie ma tu limitów... poza kwestią wagi ;-)

Przyniesione do domu kamyczki możemy wykorzystać do działań artystycznych. Przygotujmy farby (np.: akrylowe) a zamiast pędzli - patyczki higieniczne. Zapetnijmy

kamyki 'pikselami'. Malowanie patyczkami ułatwi zaznaczanie punktów. Jest to dobra okazja do rozmowy na temat obrazów cyfrowych, ale i na temat malarstwa impresjonistów.

Podejmij wyzwanie

Możemy tworzyć swobodne malunki albo pobawić się w malowanie według ustalonych zasad, np.: wykorzystując kostkę do gry. wcześniej wspólnie ustalamy jaka liczba oczek odpowiada któremu kolorowi, którego możemy użyć do malowania. Czy uda nam się namalować to, co planujemy, zdając się na wynik losowania?



Do czego możemy wykorzystać nasze kamyki? Np.: do zabawy w układanie wieży. Kamienie różnej wielkości, o nieregularnych kształtach, ustawiamy jeden na drugim, próbując stworzyć najwyższą konstrukcję. Zabawę możemy toczyć indywidualnie, budując obok siebie kilka wież. Możemy ustawiać kamyki dokonując na bieżąco wyboru albo ustalić, że kamyki będą losowane. Możemy również wprowadzić element gry, gdy dokładamy kamyki na przemian, starając się, aby konstrukcja nie rozpadła się w trakcie naszej kolejki. Jest to zadanie wymagające precyzji... ale też umożliwiające ćwiczenie cierpliwości. Jak to jest z maszynami? Jak robot poradziłby sobie z takim zadaniem?



****https://scontent-sit4-1.xx.fbcdn.net/v/t1.0-9/10731086_1112021658823565_6991233303855913824_n.jpg?oh=6d3ee18b1aacdc86ff6dcbb19932ff91&oe=59A5FC59****

Co to ma wspólnego z programowaniem?

Z tych samych elementów możemy stworzyć wiele różnych kompozycji. Przygotowując propozycje rozwiązania problemów programistycznych też mamy do dyspozycji wiele dróg rozwiązań. Możliwość ich sprawdzenia pozwala nam sprawdzić czy nasze wybory prowadzą nas do wybranego przez nas celu.

11. Rozgrywki w kratę

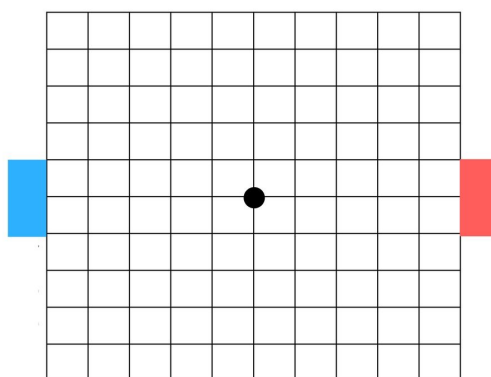
Wprowadzenie.

Zanim siądziemy do programowania z użyciem komputerów (albo w ramach przerwy w takiej aktywności;-) możemy rozwijać myślenie w obszarze logiki i matematyki, korzystając z prostych zabaw i gier wykorzystujących kartkę papieru. Pamięć, spostrzegawczość, koncentracja uwagi, to elementy przydatne nie tylko w pracy programisty, zatem warto ćwiczyć je niezależnie od tego, jakie mamy plany zawodowe.

Podejmij wyzwanie.

Proponowana zabawa ma zapewne tyle lat ile gra w chowanego czy klasy. Jej przygotowanie nie wymaga specjalistycznego sprzętu, a jedynie kartki w kratkę i czegoś do pisania.

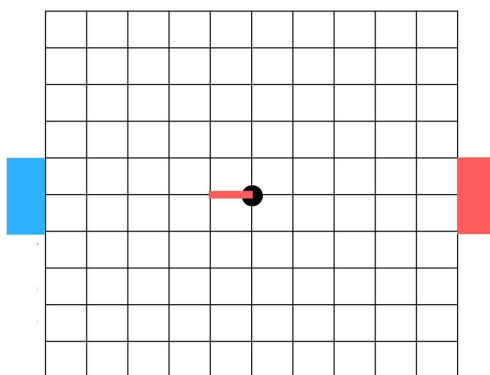
Na kartce w kratkę wytyczamy obszar boiska. Na środku pola, w linii prostej do bramek, znajduje się punkt startowy.



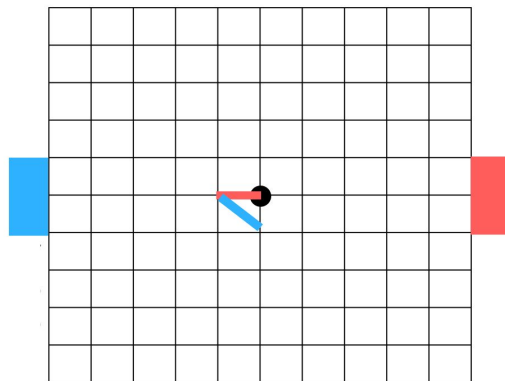
Bramki znajdują się poza głównym prostokątem boiska. Gra polega na tym, aby trafić rysowaną linią do bramki przeciwnika. Gracze ustalają kolejność wykonywania ruchów, a następnie podają "piłkę" oraz oddają "strzały" na papierze. Ważne warunki gry:

- rysujemy kreskę będącą odcinkiem łączącym narożniki kwadratu wyznaczonego przez kratkę na papierze;
- rysowany odcinek może stanowić bok kwadratu lub jego przekątną;
- po raz narysowanej linii strzału nie można prowadzić kolejnej kreski;
- jeśli rysowany odcinek kończy się na narożniku, do którego wcześniej została doprowadzona kreska, gracz może "odbić piłkę" wykonując dodatkowy strzał;
- dodatkowe odbicia wykonujemy tak długo, jak długo piłka trafia na ślad poprzednich strzałów;
- analogicznie, możliwość wykonania "dodatkowego strzału" uzyskuje gracz, którego odcinek kończy się na linii będącej obramowaniem boiska.

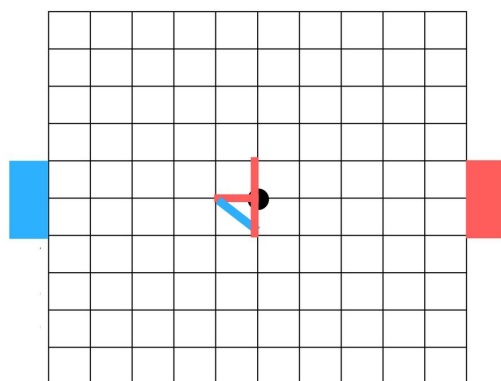
Naszą partię rozpoczyna gracz czerwony, wykonując strzał w stronę niebieskiej bramki.



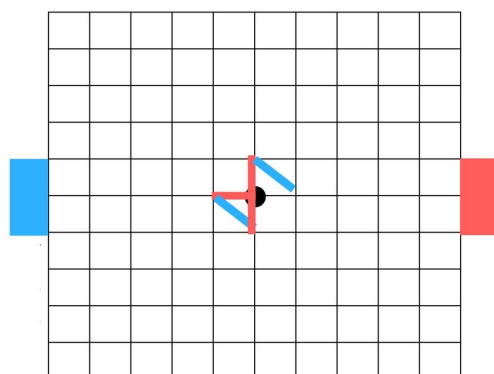
Gracz niebieski wykonał strzał po skosie, w stronę czerwonej bramki.



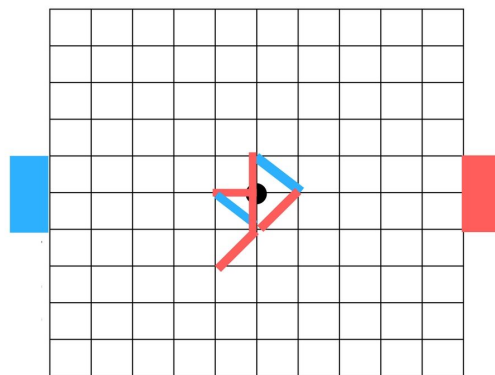
Wówczas gracz czerwony strzelił w górę, trafił na narożnik mający już jedną linię narysowaną... zatem gracz mógł wykonać dodatkowy strzał (wybrał opcję ponownego ruchu w górę).



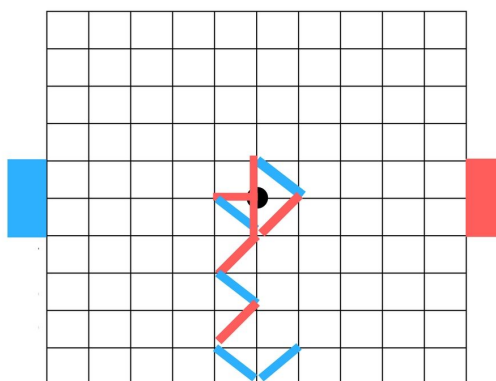
Gracz niebieski postanawia oddać strzał po linii przekątnej kwadratu, w stronę czerwonej bramki.



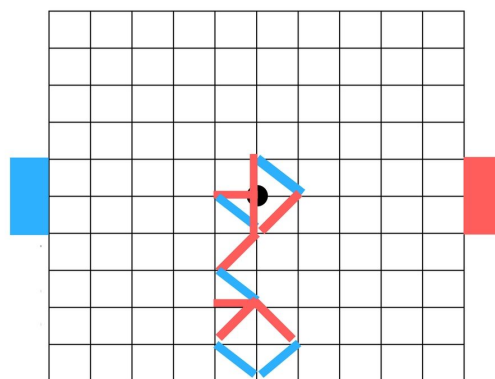
Gracz czerwony również wykonuje ruch po linii przekątnej, trafia na narożnik umożliwiając wykonanie dodatkowego strzału.



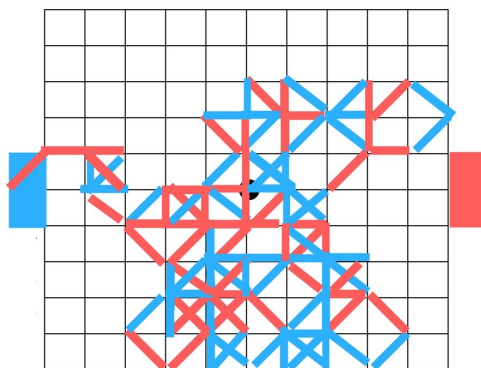
Po kilku ruchach nasi gracze mają następującą planszę. Kolej na ruch niebieskiego koloru. Gracz odbija piłkę o obramowanie boiska i wykorzystując dodatkowy strzał, kieruje piłkę w stronę czerwonej bramki.



Tymczasem, czerwony gracz, wycofuje piłkę ze swojej połowy boiska, wykonując kolejno strzał po linii przekątnej a następnie odbijając się dodatkowym strzałem, w stronę bramki niebieskiej.



Po kilkunastu/kiludziesięciu strzałach plansza jest pełna śladów. Ostatecznie w przykładowej partii, gracz czerwony umieszcza swoją piłkę w niebieskiej bramce.



Bywa też, że sekwencja ruchów doprowadzi do zablokowania planszy. Gracze nie mogą oddać kolejnego strzału muszą zakończyć grę wynikiem bezbramkowym.

Co to ma wspólnego z nauką programowania?

Zanim rozpoczniemy pisać kod programu musimy przygotować sobie strategię rozwiązania problemu. Pisząc kod musimy przewidzieć jak nasza instrukcja wpłynie na zachowanie danego obiektu lub elementu. Jeżeli rozpoczynając przygodę z kodowaniem będziemy zapisywać kod mający spowodować, że nasz obiekt przeniesie z jednego miejsca na drugie, musimy przewidzieć czy poszczególne instrukcje przeniosą nasz obiekt do celu. Jeżeli tak się nie zdarzy to musimy wyszukać miejsca w którym należy wprowadzić zmiany, aby nasz cel został osiągnięty.

<https://youtu.be/aVMCBFFF4Y0>

12. Papierowe wędrówki

Wprowadzenie.

Kartka w kratkę jest świetną przestrzenią do zabaw z podstawami programowania. Poza swobodnymi rysunkami można budować labirynty, zaznaczać ścieżki, tworzyć wzory.

Podejmij wyzwanie.

Etap 1.

Podobnie jak w zabawie (9), na kartce w kratkę przygotowujemy dla każdego gracza planszę. Może ona mieć wymiary dowolnego prostokąta, ale na początku może to być kwadrat o wymiarach 10x10 kratek (czyli 5x5cm). Cała plansza będzie zatem miała 100 kwadratowych pól. Tak jak w grze w statki, każdą kolumnę oznaczamy na górze literami

(A, B, C, D itd), a każdy wiersz oznaczamy liczbą (1, 2, 3, 4 itd). W ten sposób każda ze stu kratek naszej planszy ma określone, niepowtarzalne koordynaty (np.: A1, D4).

	A	B	C	D	E	F	G	H	I	J
1										
2										
3										
4										
5										
6										
7										
8										
9										
10										

Etap 2.

Gracze ustalają, że kratka A1 jest pozycją startową a pole J10 to meta. Każdy z graczy buduje na swojej planszy mury labiryntu, kolorując np.: 30% planszy (czyli 30 kratek). Labirynt ma umożliwić graczom przejście między startem (A1) a metą (J10).

Przykładowa plansza gracza mogłaby wyglądać tak: zielone pole (A1) oznacza start, czerwone pole (J10) oznacza metę. 30 pól w szarym kolorze to mur labiryntu:

	A	B	C	D	E	F	G	H	I	J
1	█			█	█	█	█	█	█	█
2										█
3										█
4						█	█	█		█
5	█									

6	■									■
7	■			■	■	■				■
8	■									■
9	■									
10	■	■	■		■	■	■			■

Etap 3.

- Teraz każdy z graczy zapisuje na kartce zestaw liczb i liter. Gracz pierwszy wypisuje:
- 10 liczb które uzupełnione o litery wybrane przez drugiego gracza, stworzą dodatkowe mury labiryntu;
 - 10 liter, które uzupełnione liczby wybrane przez drugiego gracza, stworzą artefakty, czyli cenne przedmioty które gracz ma zebrać w trakcie marszu po labiryncie.

Etap 4.

Kolejnym etapem jest połączenie liczb i liter w pary koordynatów. Przykładowy układ:

Gracz 1 zapisał:

- 10 liczb labiryntu: 1, 4, 4, 4, 4, 6, 9, 10, 10, 10
- 10 liter artefaktów: A, B, C, D, F, F, G, H, H, H

Gracz 2 zapisał:

- 10 liter labiryntu: A, A, A, C, C, C, G, H, I, I
- 10 liczb artefaktów: 2, 3, 3, 4, 5, 5, 5, 6, 7, 9

Po połączeniu powstały koordynaty:

- ścian labiryntu: A1, A4, A4, C4, C4, C6, G9, H10, I10, I10
- artefaktów: A2, B3, C3, D4, F5, F5, G5, H6, H7, H9

Gracze mogli zapisać jedną liczbę lub literę, powtarzając dziesięć razy. Albo wymienić wszystkie liczby/litery. Mogli również wypisywać bez zachowania kolejności narastającej. To ciekawy przykład by podjąć się wyzwania matematycznego... Ile zatem istnieje możliwych kombinacji liczb i liter?

Etap 5.

W ten sposób otrzymaliśmy informacje o polach, które uzupełnią naszą planszę z narysowaną wcześniej częścią labiryntu. Niektóre koordynaty z przykładowym zestawie, powielają się (A4, C4, I10, F5) oznacza to, że te pola pokolorujemy jeden raz. Dodatkowo, pola 'start' oraz 'meta' są już narysowane, zatem nie możemy wykorzystać koordynatu mającego współrzędne pola 'start' (A1).

W pierwszej kolejności kolorujemy szare pole ścian labiryntu, kolejnym krokiem jest dodanie niebieskich pól artefaktów. Jeśli pola pokrywają się, ponieważ umieściliśmy tam wcześniej szarą barwę ścian labiryntu, pole artefaktu przepada. Gdybyśmy rozpoczęli kolorowanie od artefaktów, mielibyśmy więcej punktów do zdobycia, a jednocześnie mniej zablokowany labirynt.

Nasza przykładowa plansza, z szarymi ścianami labiryntu, zielonym polem 'start', czerwonym polem 'meta' oraz z niebieskimi polami 'artefakt' wygląda teraz następująco:

	A	B	C	D	E	F	G	H	I	J
1	Green			Grey	Grey	Grey	Grey	Grey	Grey	Grey
2	Blue									Grey
3		Blue	Blue							Grey
4	Grey		Grey	Blue		Grey	Grey	Grey		Grey
5	Grey					Blue	Blue			
6	Grey		Grey					Blue		Grey
7	Grey			Grey	Grey	Grey				Grey
8	Grey									Grey
9	Grey						Grey		Blue	
10	Grey	Grey	Grey		Grey	Grey	Grey	Grey	Grey	Red

Szczęśliwie dla gracza, ta trasa umożliwi swobodne przejście od punktu startowego, do końcowego. Dodatkowo gracz ma możliwość zebrania aż 8 punktów z artefaktów.

Etap 6

Komu udało się dotrzeć od startu do mety? Kto zebrał więcej artefaktów? Kto dotarł najkrótszą drogą? Kto miał najmniej skrętów?

Grę można rozwijać ustalając, że zwiększamy liczbę szarych pól, a tym samym utrudniamy przejście między polem startowym a meta. Może się okazać, że trasa będzie zablokowana. Wyboru dodatkowych, losowych pól labiryntu i/lub położenia artefaktów możemy dokonać korzystając np.: z rzutu kostką.

Co to ma wspólnego z nauką programowania?

Przez wszystkie etapy mogłeś doświadczyć wielu wtajemniczeń związanych ze światem programowania. Dlatego teraz, w odniesieniu do powyższego zadania spróbuj zapisać lub podzielić się z drugą osobą jak największą liczbą połączeń pomiędzy powyższą aktywnością, a nauką programowania. Jak myślisz jakie elementy związane z nauką programowania możesz odnaleźć w powyższej zabawie?