

**Nazwa implementacji:**

Pong

Autor:

Jarosław Żok

**Opis implementacji:** Zmiany w skryptach gry Pong, dodające kolejny element zmieniający jej zasady.

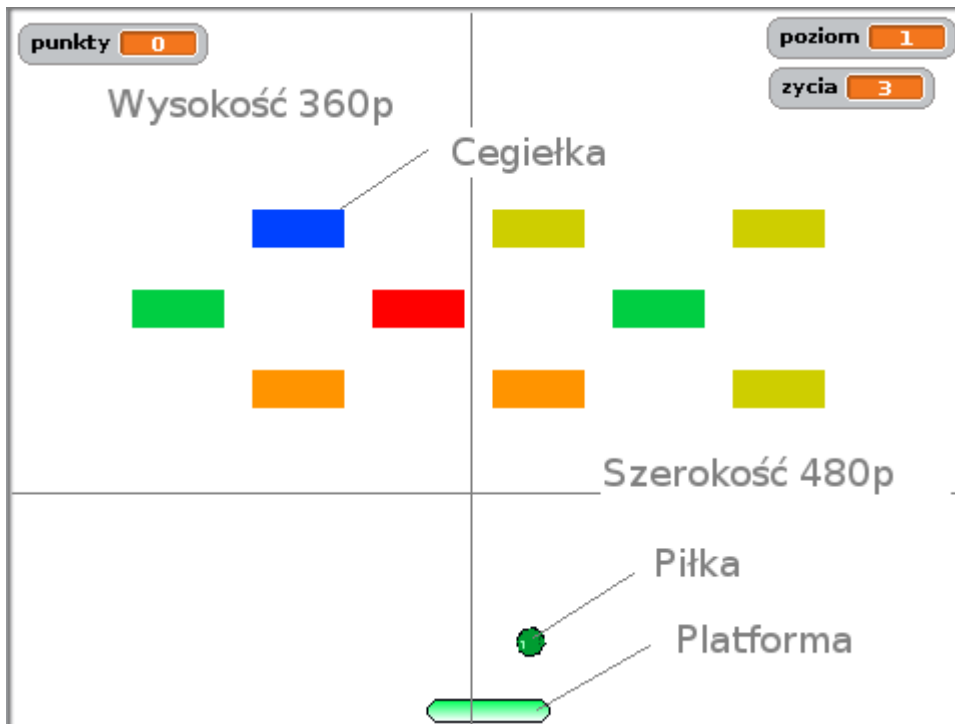
Gra została zaimplementowana z wykorzystaniem programu BYOB. Jest to modyfikacja Scratcha, która rozszerza go o wiele przydanych funkcji. Ma możliwość tworzenia list w listach, skryptów w postaci własnych bloczków, które są prawdziwymi podprogramami.

Pong znana pod wieloma nazwami i modyfikacjami gra polegająca na odbijaniu piłki za pomocą platformy znajdującej się w dolnej części planszy. Platforma początkowo znajduje się na spodzie planszy w środku jej szerokości. Piłka znajduje się na wysokości platformy i po naciśnięciu spacji uruchamiany jest ruch piłki. Gdy piłka doleci do którejś z trzech krawędzi (lewej, prawej lub górnej), zmienia kierunek, odbijając się od niej. Gdy doleci na wysokość platformy i jej dotknie, piłka zmienia kierunek podobnie jak w przy odbiciu od pozostałych krawędzi planszy. Jeżeli jednak w momencie, kiedy piłka znajduje się na wysokości platformy, a platformy w danym miejscu nie ma, gracz przegrywa. Zadaniem gracza jest, takie sterowanie platformą za pomocą strzałek w lewo i w prawo na klawiaturze, żeby platforma zawsze znajdowała się pod piłką.

Aby grę nieco urozmaicić, wprowadzono pewne modyfikacje do jej zasad.

- Pierwsza modyfikacja gry polega na wprowadzeniu do gry poziomów i ilości piłek - „żyć” gracza. Domyślnie są trzy życia, ilość żyć zmniejsza się, jeżeli piłka doleci do poziomu platformy, ale się z nią nie zderzy. Po osiągnięciu 0 żyć, gracz przegrywa całą grę.
- Druga modyfikacja polega na wprowadzeniu punktacji. Gracz startuje z 0 ilością punktów. Za każde odbicie piłki od platformy gracz nagradzany jest punktem, które się sumują.
- Trzecia modyfikacja jest taka, że na planszy oprócz piłki i platformy znajdują się cegiełki, które piłka w trakcie swojego lotu strąca. Piłka dolatując do cegiełki, uderza w nią i zmienia kierunek. Ruch piłki staje się przez to mniej przewidywalny i gra staje się większym wyzwaniem.
- Czwarta modyfikacja zasad gry polega na przypisaniu do cegiełki jakiegoś przedmiotu zmieniającego chwilowo zasady gry. Przedmiot wypada z cegiełki, kiedy piłka w nią uderzy. Przedmiot spada w dół aż do poziomu platformy i jeżeli jej dotknie, zasady gry zmieniają się, zgodnie z tym, co ukrywa przedmiot (na przykład zwiększa się szybkość piłki, zmniejsza się rozmiar platformy lub piłka przestaje odbijać się od cegiełek, po prostu je rozbijając). Zmiany zasad są czasowe i wracają do przyjętych po 10 sekundach.
- Piąta zmiana to wprowadzenie poziomów gry. Poziomy gry, to kolejne konfiguracje cegiełek, które pojawiają się przy przejściu na kolejny poziom. Konfiguracje cegiełek w kolejnych poziomach są ustalane na samym początku gry. Gracz, zbijając wszystkie cegiełki, przechodzi do kolejnego poziomu.
- Organizacja planszy gry:





Szerokość planszy wynosi 480 pikseli, jej wysokość 360. Należy zwrócić uwagę, że początek układu współrzędnych znajduje się w środku planszy, czyli środek planszy ma współrzędne (0,0). Idąc w lewo od środka współrzędnych, wartości X są ujemne, w prawo od środka są dodatnie. Podobnie ze współrzędną Y, w górę od środka są dodatnie, w dół są ujemne.

Gra jest podzielona na kilka skryptów, najbardziej ogólne nie dotyczą żadnego z duszków znajdują się na scenie. Są tam dwa skrypty, pierwszy, uruchamiany po kliknięciu w zieloną flagę, ustawia wszystkie zmienne globalne na wartości domyślne. Dodaje także do listy ułożenia cegiełek konfigurację cegiełek dla kolejnych poziomów gry. Organizacja tej listy zostanie opisana poniżej. Drugi skrypt znajdujący się na scenie, uruchamiany jest, gdy nadane zostanie zdarzenie „kolejny poziom”. Zdarzenie to pobiera konfigurację cegiełek dla kolejnego poziomu gry, tworzy listę z położeniami cegiełek, inicjuje licznik cegiełek na planszy i wysyła zdarzenie informujące każdą z cegiełek, że ma ułożyć się zgodnie z zapisaną listą położenia na planszy.

Jak jest zorganizowana lista położenia cegiełek na planszy?

Istnieje globalna lista poziomów nazywająca się „ulozenie cegielek”. Lista ta ma tyle elementów, ile jest poziomów w grze. Każdy z elementów tej listy to konfiguracja aktywnych i nieaktywnych cegiełek zorganizowana jako kolejna lista trzech elementów. Każdy element tej listy jest ciągiem znaków i reprezentuje konfigurację aktywnych lub nieaktywnych cegiełek w wierszu, który opisuje. 1 w danym miejscu oznacza, że cegielka znajdująca się na danym poziomie w danym wierszu i kolumnie ma być włączona, 0 że ma być wyłączona. Piłka zderza się tylko z aktywnymi cegiełkami.



Przypisanie położenia poszczególnych cegiełek do wartości w tej liście dla jednego poziomu:

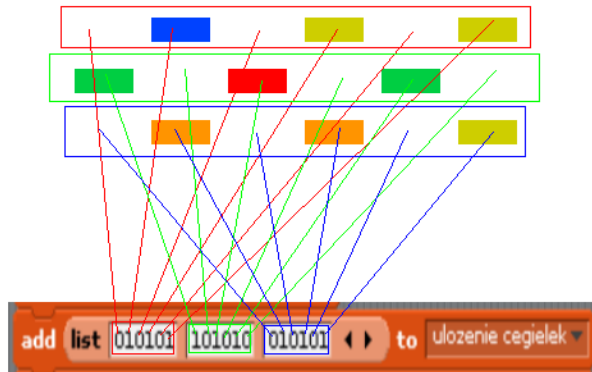




punkty 0

poziom 1

zycia 3



Z listy tej w zdarzeniu „kolejny poziom” wyliczana jest kolejna lista nazwana „wolne miejsca”, przechowująca 18 wartości współrzędnych, na których mają znaleźć się cegiełki oraz informacja czy cegiełka ma być widoczna czy ukryta. Następnie po wyliczeniu pozycji wszystkich 18 cegiełek wysyłany jest komunikat „stwórz cegiełkę” odbierany przez każdego z 18 duszków reprezentujących cegiełkę. Każdy taki duszek w odpowiedzi na komunikat pobiera ostatni element z listy współrzędnych oraz usuwa go z listy współrzędnych, aby inny duszek nie wykorzystał go po raz drugi i nie ustawił się w tym samym miejscu. Taki sposób organizacji danych nazywany jest często stosem albo kolejką. W tym przypadku lista „wolne miejsca” jest kolejką typu LIFO (od angielskiego terminu Last In First Out – ostatni wchodzi, pierwszy wychodzi). Czyli ostatni element, który dodaliśmy do tej listy, jest pierwszym, który z tej listy zostanie „zdjęty” przez duszka i wykorzystany do ustalenia pozycji w planszy. Cegiełki mają różne kolory zdefiniowane jako kolejne kostiumy, kiedy cegiełka otrzyma komunikat „stwórz cegiełkę” losuje najpierw kolor, poprzez numer kostiumu. Numer ten może przyjąć wartości od 1 do 6

```
switch to costume pick random 1 to 6
```

Dodatkowo pod każdą z cegiełek może potencjalnie ukrywać się przedmiot zmieniający czasowo zasady gry. To czy pod cegiełką znajduje się taki przedmiot, jest losowane w trakcie ustawiania cegiełki w tym samym skrypcie. Takich przedmiotów może być obecnie 5, jeżeli chcemy, żeby średnio pod co trzecią cegiełką ukryty był przedmiot, losujemy  $5 * 3 = 15$  wartości i jeżeli wylosowana wartość jest mniejsza niż 6, to pod cegiełką znajduje się wylosowany przedmiot. Dzięki temu, że każdy skrypt w BYOB może mieć własną lokalną listę zmiennych, każda cegiełka pamięta tylko swoją konfigurację i swoją wylosowaną „właściwość”, którą jest brak przedmiotu albo numer przedmiotu, który ma spaść na platformę po uderzeniu cegiełki przez kulkę.

Po wylosowaniu kostiumu, właściwości cegiełki, ustawieniu się w odpowiedniej pozycji na planszy, cegiełka czeka aż zostanie dotknięta przez kulkę:

```
wait until touching kulka ?
```

Po czym zmniejsza licznik cegiełek na planszy i jeżeli wylosowana właściwość określa, że pod cegiełką znajduje się przedmiot, przechodzi zmiany kostiumu na reprezentujący go przedmiot i do animacji jego spadania oraz sprawdza czy przedmiot dotyka platformy. Jeżeli dotyka, nadaje odpowiednie zdarzenie, które czasowo modyfikuje zasady gry.

### Animacja kulki

Kulka uruchamiana jest po naciśnięciu klawisza spacji odbieranego w skrypcie duszka platformy, który wysyła zdarzenie „uruchom kulke”. Skrypt obsługi tego zdarzenia przełącza kostium duszka kulki na podstawowy, ustala domyślną prędkość w pikselach na krok na 3 piksele w





osi X i Y, kierunki dla obu tych osi na rosnące oraz uruchamia pętlę, w której kulka jest animowana. Za każdym obrotem tej pętli wyliczane jest położenie kulki dla obu osi osobno. Każda współrzędna kulki modyfikowana jest o wartość równą aktualnej szybkości pomnożonej razy kierunek. Dzięki temu zmiana kierunku sprowadza się do zmiany znaku kierunku, który przyjmuje wartość 1 dla rosnących współrzędnych i -1 dla malejących. Następnie sprawdzane są warunki dla osi X, czyli czy piłka dotyka lewej lub prawej krawędzi. Jeżeli tak to kierunek zmieniany jest na przeciwny. Dla osi Y, warunki są nieco inne. O ile sprawdzenie, czy piłka odbija się od górnej krawędzi planszy, jest takie samo jak dla osi X, to gdy piłka doleci do dolnej krawędzi, musimy sprawdzić czy dotyka platformy. Jeżeli nie, to zmniejszamy ilość „żyć” - piłek o 1 i jeżeli wynosi ona 0, to gracz przegrywa grę. Jeżeli piłka dotyka platformy, to dodajemy punkt do zmiennej przechowującej punktację i sprawdzamy, czy ilość punktów jest wielokrotnością 15. Możemy to sprawdzić za pomocą funkcji modulo – zwraca ona resztę z dzielenia przez liczbę, w tym przypadku 15. Jeżeli wartość modulo 15 jest równa 0 (dla wartości  $> 1$ ), to znaczy, że wartość jest wielokrotnością liczby 15. Po co nam ta informacja? Po to, żeby przejść do kolejnego poziomu, gdy użytkownik zbierze kolejne 15 punktów. Przechodząc do kolejnego poziomu, zwiększamy zmienną „poziom” o 1 i jeżeli jest ona większa niż długość listy konfiguracji kłoców „ulożenie cegiełek” to znaczy, że gracz przeszedł wszystkie poziomy.

Kulka reaguje także na kilka dodatkowych zdarzeń, wysyłane są one przez cegiełkę, gdy kulka straci cegiełkę pod którą znajduje się przedmiot spadający z góry, który dotyka platformy a który modyfikuje właściwości kulki. Są to:- „super szybkość” - kulka przyspiesza po otrzymaniu tego zdarzenia. Przyspieszony ruch trwa 10 sekund - „super kulka” - normalnie kulka po uderzeniu w cegiełkę, zmienia kierunek, odbijając się od niej, obsługa tego zdarzenia zmienia na 10 sekund wartość zmiennej „super kulka” z 0 na 1. Gdy zmienna „super kulka” jest ustawiona na 1, zderzenie z cegiełką nie powoduje odbicia piłki.

#### Ruch platformy

Ruch platformy jest chyba najłatwiejszy do opisania. Platforma rusza się w lewo i w prawo wzdłuż osi X, w osi Y jest nieruchoma. Do obsługi ruchu platformy przeznaczone są dwa zdarzenia „strzałka w prawo” i „strzałka w lewo”. Zgodnie ze strzałkami współrzędne X platformy zmieniają się o 10 pikseli (+10 dla ruchu w prawo i -10 dla ruchu w lewo). Platforma obsługuje także parę dodatkowych zdarzeń:

- „kiedy naciśnięto spację” - wysyła zdarzenie „start” ustawiając kulkę na domyślne wartości, wysyła „uruchom kulke”, który startuje jej ruch.
- „kiedy kliknięto zieloną flagę” - uruchamia cały program, ustawia platformę na wartości domyślne.
- „mała platforma” - gdy przedmiot z cegiełki ma zmniejszyć platformę i spadając, dotknie platformy, platforma zmniejsza swoje rozmiary na 10 sekund.
- „duża platforma” - gdy przedmiot z cegiełki ma zwiększyć platformę i spadając, dotknie platformy, platforma zwiększa swoje rozmiary na 10 sekund.

#### Zadanie

Naszym zadaniem będzie rozbudowanie gry o dodatkowy przedmiot spadający z cegiełki, który jeżeli dotknie platformy, na 10 sekund zamieni kierunki ruchu platformy na przeciwny (czyli strzałka w lewo spowoduje ruch w prawo, a strzałka w prawo spowoduje ruch w lewo).

- Pierwsze co musimy zrobić to zdefiniować nową zmienną duszka platformy, zmienna niech nazywa się „odwrotny ruch”. Zmienna ta po naciśnięciu spacji musi być ustawiona na 0. W momencie, kiedy otrzymamy nowe zdarzenie „zamien strzałki” zostaje na 10 sekund ustawiona na 1.
- W zdarzeniach „kiedy wciśnięto strzałka w lewo” i „kiedy wciśnięto strzałka w prawo”, musimy dodać dwa warunki z alternatywą. W warunkach sprawdzamy czy zmienna „odwrotny ruch” jest ustawiona na 0, wtedy dla strzałki w lewo odejmujemy 10 pikseli od współrzędnej X, a dla strzałki w prawo dodajemy 10 pikseli do współrzędnej X. W przeciwnym wypadku, odejmujemy 10 pikseli dla ruchu w prawo i dodajemy dla ruchu w lewo. Mamy rozbudowany ruch platformy w reakcji na zdarzenie „zamien strzałki”.
- Zdarzenie będzie wysyłane z cegiełki po uderzeniu w nią przez kulkę i gdy przedmiot opadnie na platformę. Przejdźmy zatem do duszka pierwszej cegiełki. Musimy zwiększyć zakres losowanych właściwości dla duszka. Dodajmy kolejną właściwość, a chcemy mieć przynajmniej jedną trzecią cegiełek z wylosowanymi przedmiotami. Obecnie jest 5 takich przedmiotów, będzie 6, gdy dodamy kolejny.  $6 * 3 = 18$ , zatem wpiszmy 18 w górną granicę wartości losowanych właściwości.





- Kolejną zmianą w duszku cegiełki jest dodanie warunku, który wylosowana właściwość jest równa 6. Zmianę wprowadzamy, dodając warunek, jeżeli właściwość = 6 i w warunku dodajemy nadanie zdarzenia „zamien strzałki”. Musimy też zmienić warunek sprawdzający czy właściwość jest mniejsza lub równa 5, ustawmy sprawdzenie czy właściwość jest mniejsza lub równa 6 (czyli mniejsza od 7).
- Przejdźmy teraz do edycji kostiumów cegiełki i zróbmy kopię ostatniego kostiumu reprezentującego cegiełkę z dodatkowym życiem „+1”. Edytujmy kostium i w prostokącie reprezentującym przedmiot dodajmy jakieś symbole oznaczające zamianę strzałek.
- Ostatnią zmianą jest usunięcie wszystkich duszków cegiełek oprócz pierwszego, którego skrypt modyfikowaliśmy i dodanie kolejnych 17 cegiełek poprzez kopiowanie duszka pierwszej cegiełki. W ten sposób otrzymamy 18 kopii cegiełki z naszym zmodyfikowanym skryptem.
- Po uruchomieniu gry, zależnie czy została wylosowana cegiełka z naszym dodatkowym przedmiotem, gdy piłka dotknie takiej cegiełki, powinien pojawić się duszek z kostiumem, który właśnie przygotowaliśmy i po dotknięciu platformy działanie strzałek powinno zostać zmienione na 10 sekund.

#### Podsumowanie

Jest to przykład implementacji zasad gry. Jednego ze sposobów mapowania konfiguracji elementów planszy za pomocą mapy. Nauczyłeś się wykorzystywania jednej ze struktur danych stosowanych w informatyce, czyli kolejki LIFO. Wiesz czym są zmienne globalne i lokalne. Poznałeś elementy programowania obiektowego, poprzez implementację skryptów odpowiedzialnych za animację, interakcję poszczególnych duszków oraz sposobów rozbudowywania gry o własne dynamiczne zasady.

