

# Instrukcja projektowa; projekt numer 2

---

*Podstawy Programowania 2014/15, kierunek Informatyka*

autor: Dariusz Dereniowski<sup>1</sup>

## Projekt ARKANOID

---

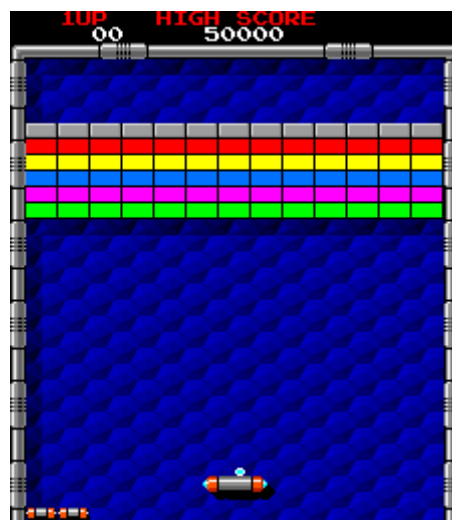
### Cel

Celem projektu jest implementacja gry „Arkanoid”.

Gra polega na odbijaniu *piłeczki* za pomocą *paletki* znajdującej się u dołu ekranu. Piłeczka odbija się od przeszkód zgodnie z „prawami fizyki”, za wyjątkiem odbicia od paletki: im bliżej brzegu paletki, tym bardziej skracany (jeśli piłeczka nadlatuje z lewej (odpowiednio prawej) strony i odbija się od prawej (odpowiednio lewej) paletki) lub powiększany (w przeciwnym przypadku) jest kąt odbicia względem konta padania. Celem gry jest trafienie piłeczką w każdą *cegielekę* znajdującą się na planszy, przy czym cegielełka po trafieniu jest usuwana z planszy. W przypadku trafienia w cegielekę, możliwe jest pojawienie się *nagrody*. Nagroda pojawia się początkowo w miejscu cegielełki i z ustaloną prędkością opada ku dołowi planszy. Kontakt paletką z nagrodą owocuje odpowiednią modyfikacją mechaniki gry (opisane poniżej). Obiekt nagrody poruszający się w dół planszy nie podlega żadnym kolizjom z otoczeniem, za wyjątkiem kontaktu z paletką. Oprócz powyższych cegiełek, na planszy znajdują się *blokady*, które mają ten sam rozmiar, co cegielełki, lecz piłeczka odbija się od nich nie powodując ich usunięcia z planszy.

*Rozgrywka* rozpoczyna się od sytuacji, w której piłeczka jest „przyklejona” do paletki – kliknięcie przycisku s „uwolnienie” piłeczki, tzn. rozpoczyna ona ruch ku górze ze z góry ustaloną prędkością oraz z góry ustalonym kontem.

Gra jest podzielona na *etapy*. Każdy etap jest z góry ustaloną planszą (układ cegiełek i blokad). W ramach jednej gry sytuacja, nazywana dalej *przegraniem rozgrywki*, w której piłeczka mija paletkę i trafia w dolny brzeg ekranu może nastąpić 3-krotnie.



Ilustracja 1: (Źródło: <http://en.wikipedia.org/wiki/Arkanoid>)

---

<sup>1</sup>Uwaga: W razie niejasności lub niejednoznaczności w poniższym opisie proszę kontaktować się z autorem instrukcji; pokój EA 209; konsultacje odbywają się w poniedziałki, w godz. 15:15-17:00

Powyższe zasady gry mogą być modyfikowane przez wspomniane wcześniej nagrody.

Poniżej znajduje się przykład, na którym można się wzorować:

<http://www.youtube.com/watch?v=z9WP1OIgBNo>

## Środowisko programistyczne

Do instrukcji dołączony jest program startowy w którym zaimplementowano:

- obliczanie przyrostu czasu, co pozwala śledzić jego wpływ
- wyświetlanie na ekranie plików graficznych w formacie BMP
- rysowanie piksela, linii, prostokąta
- wyświetlanie tekstu

Program działa w oparciu o bibliotekę SDL2 (2.0.3) – <http://www.libsdl.org/>. Jest ona dołączona do projektu startowego i nie trzeba pobierać jej źródeł.

Kompilacja pod systemem linux wykonujemy za pomocą komendy (w systemie 32-bitowym):

```
g++ -O2 -I./sdl/include -L. -o main main.cpp -lm -lSDL2 -lpthread -ldl -lrt
```

oraz (w systemie 64-bitowym)

```
g++ -O2 -I./sdl/include -L. -o main main.cpp -lm -lSDL2-64 -lpthread -ldl -lrt
```

W celu pomyślnej kompilacji projektu startowego, w katalogu, w którym znajduje się plik main.cpp powinny znajdować się

- Bitmapy z wymaganymi rysunkami (cs8x8.bmp, eti.bmp). Uwaga na wielkość liter w nazwach plików!
- Plik libSDL2.a (libSDL2-64.a przy kompilacji 64 bitowej).
- Katalog sdl dołączony do projektu.

Do projektu dołączone zostały skrypty, które mogą być użyte do kompilacji (comp w środowisku 32-bitowym oraz comp64 w środowisku 64-bitowym).

Prezentacja programu (zaliczenie tej części projektu) odbywać się będzie w wybranym przez studenta środowisku spośród dwóch poniższych opcji:

- w systemie linux. Student jest zobowiązany sprawdzić przed przybyciem na zaliczenie czy program poprawnie się kompiluje i uruchamia pod dystrybucją dostępną w laboratorium,
- w systemie windows, w środowisku MS Visual C++ w wersji zgodnej z tą dostępną w laboratorium.

Uruchomienie programu podczas zaliczenia jest warunkiem koniecznym uzyskania punktów z projektu nr 2.

W programie nie należy używać biblioteki C++ **stl**.

## Wymagania obowiązkowe (5 pkt.)

Wszystkie wymienione tutaj elementy należy zaimplementować. Brak któregośkolwiek z poniższych elementów skutkuje otrzymaniem 0 pkt. z tego projektu. Wymagane elementy:

- przygotowanie jednego etapu gry (**1 pkt.**) – plansza powinna być wczytywana z pliku tekstowego. Należy samodzielnie opracować tekstowy format kodowania planszy w pliku. Format ten powinien być na tyle czytelny, aby użytkownik mógł w łatwy i wygodny sposób dokonywać jego modyfikacji. Opis powinien umożliwiać dowolne (z dokładnością do zdefiniowanych "komórek") rozmieszczenie różnokolorowych klocków w górnej połowie planszy, na przykład plansza z ilustracji może być opisana przez:

```
.....  
.....  
.....  
.....  
AAAAAAAAAAAAA  
BBBBBBBBBBBBB  
CCCCCCCCCCCCC  
DDDDDDDDDDDDD  
EEEEEEEEEEEEEE  
FFFFFFFFFFFFFFF  
.....  
.....  
.....  
.....
```

Brzegi planszy, jej szczyt i pole z wynikiem (o ile liczenie punktów poniżej jest zaimplementowane) powinny być odpowiednio wyróżnione (np., jak na ilustracji, brzegi wyróżnione są ramką, a pole z punktami innym tłem). Ruch wszystkich obiektów powinien uwzględniać wygląd planszy, tzn. piłeczka ma się odbijać od brzegu ramki, a nie przenikać przez nią i odbijać się od krawędzi ekranu. Podobnie, paletka ma zatrzymywać się po dodaniu do brzegu ramki.

- sterowanie paletką oraz start piłeczki („odklejenie” od paletki na początku rozgrywki). Paletką sterujemy używając klawiszy „strzałka w lewo” oraz „strzałka w prawo”. Piłeczka rozpoczyna ruch po wciśnięciu przycisku s. Paletka przesuwa się tak długo, jak długo wciśnięty jest odpowiedni klawisz; dojście do brzegu planszy powoduje zatrzymanie paletki. (**1 pkt.**)
- oprogramowanie ruchu piłeczki i jej odbijanie od przeszkód (od brzegów planszy, od cegiełek, blokad i paletki). Odbijanie od paletki w ramach wymagań minimalnych odbywa się bez modyfikacji kąta odbicia (tzn. kąt padania jest równy kątowi odbicia, bez względu na miejsce odbicia piłeczki od paletki). (**1 pkt.**)
- Implementacja usuwania cegiełek, od których odbiła się piłeczka (w przypadku trafienia piłeczki w cegiełkę, najpierw następuje odbicie piłeczki, a bezpośrednio po nim usunięcie cegiełki). (**1 pkt.**)
- Rozpoznanie faktu przegranej rozgrywki i zakończenie gry. Program powinien dać użytkownikowi możliwość ponownej gry lub zakończenia programu. (**1 pkt.**)

## Wymagania nieobowiązkowe (10 pkt.)

- *Liczenie liczby przegranych rozgrywek.* W szczególności, na planszy powinno być oznaczone i stale widoczne, ile przegranych rozgrywek pozostało do końca gry (początkowo 3). Po każdej przegranej rozgrywce, nowa rozgrywka wznawia się w tej samej konfiguracji na planszy, w której zakończyła się poprzednia. Po ostatniej przegranej rozgrywce powinna się pojawić informacja o zakończeniu gry. **(0.5 pkt.)**
- *Liczenie punktów.* Gracz otrzymuje pewną liczbę punktów za trafienie w każdą cegielkę. Punkty te powinny być widoczne w trakcie rozgrywki. **(0.5 pkt.)**
- *Kąt odbicia piłeczki.* Piłeczka odbija się pod różnymi kontami od paletki – im bliżej brzegu paletki następuje odbicie, tym większa zmiana kąta odbicia. Na przykład, piłeczka nadlatująca z lewej strony: odbicie z lewego brzegu paletki zmniejsza kąt odbicia względem konta padania, natomiast odbicie z prawego brzegu paletki zwiększa ten kąt. **(1 pkt.)**
- *Implementacja dowolnej z opisanych poniżej nagród* skutkuje otrzymaniem (oprócz liczby punktów podanych przy danej nagrodzie) dodatkowo **1.5 pkt.**
- *Nagroda: zmiana tempa gry.* Nagroda pojawia się losowo w efekcie trafienia w cegielkę. Efekt tej nagrody jest taki, że przez 20 sekund piłeczka porusza się dwukrotnie wolniej niż wynosi jej normalna prędkość. Jeśli zaimplementowano liczenie punktów, to uzyskanie nagrody powoduje również uzyskanie pewnej liczby punktów. **(0.5 pkt.)**
- *Nagroda: kleista paletka.* Nagroda pojawia się losowo w efekcie trafienia w cegielkę. Przy każdym kontakcie piłeczki z paletką, nie następuje odbicie, lecz piłeczka pozostaje w niezmięnionej pozycji względem paletki, aż do momentu, w którym gracz kliknie przycisk **s**. Wówczas, kąt pod którym piłeczka rozpoczyna swój ruch względem paletki jest taki sam, jakby odbicie nastąpiło natychmiast (bez fazy „przyklejenia”). Jeśli zaimplementowano liczenie punktów, to uzyskanie nagrody powoduje również uzyskanie pewnej liczby punktów. **(0.5 pkt.)**
- *Nagroda: dodatkowa rozgrywka.* Nagroda pojawia się losowo w efekcie trafienia w cegielkę. W jej wyniku liczba możliwych rozgrywek pozostałych do zakończenia gry wzrasta o jeden. Jeśli zaimplementowano liczenie punktów, to uzyskanie nagrody powoduje również uzyskanie pewnej liczby punktów. **(0.5 pkt.)**
- *Nagroda: szeroka paletka.* Nagroda pojawia się losowo w efekcie trafienia w cegielkę. Efektem jest dwukrotne poszerzenie paletki na okres 20 sekund. Jeśli zaimplementowano liczenie punktów, to uzyskanie nagrody powoduje również uzyskanie pewnej liczby punktów. **(1 pkt.)**
- *Nagroda: trzy piłeczki.* Nagroda pojawia się losowo w efekcie trafienia w cegielkę. Efektem jest zastąpienie piłeczki trzema piłeczkami, poruszającymi się pod nieco różnymi kontami (efekt „roztrojenia” piłeczki). Jeśli zaimplementowano liczenie punktów, to uzyskanie nagrody powoduje również uzyskanie pewnej liczby punktów. **(1 pkt.)**
- *Przyspieszanie piłeczki.* Z biegiem czasu rozgrywki, piłeczka powinna stopniowo przyspieszać swój ruch. Przyspieszenie należy dobrać wg uznania tak, aby było ono możliwe do zaobserwowania podczas 30-sekundowej rozgrywki. Po osiągnięciu górnego limitu, piłeczka nie powinna już dalej przyspieszać. Jeśli zaimplementowano nagrodę „zmiana tempa gry”, to nagroda ta przestawia piłeczkę do 50% wyjściowej prędkości, a gdy czas funkcjonowania nagrody minie, piłeczka powinna poruszać się z prędkością ustawianą na początku rozgrywki. Wówczas, proces przyspieszania piłeczki powinien rozpocząć się od początku. **(1 pkt.)**
- *Historia najlepszych wyników.* Program powinien przechowywać 5 najlepszych wyników punktowych uzyskanych dotychczas. Wyniki powinny być zachowywane pomiędzy

poszczególnymi uruchomieniami programu w pliku tekstowym. Jeśli gracz uzyska wynik kwalifikujący się do grona najlepszych, to program powinien poprosić o podanie imienia, dopisać imię do listy w pliku, wraz z wynikiem punktowym. Następnie, lista najlepszych wyników powinna zostać wyświetlona na ekranie. Lista ta powinna być wyświetlona na zakończenie każdej gry. (1 pkt.)

- *Wieloetapowość.* Program powinien wczytywać z pliku tekstowego wiele etapów. Liczba etapów nie powinna być ograniczona z góry w kodzie programu, tzn. program powinien obsługiwać taką ilość etapów jaką dodamy do pliku. Format kodowania należy opracować samodzielnie. Dodanie nowego etapu lub usunięcie istniejącego nie powinno wymagać żadnej zmiany w kodzie. Kolejność wystąpienia etapów podczas gry powinna być zgodna z kolejnością w pliku – zamiana kolejności etapów w pliku powinna skutkować zamianą ich kolejności podczas gry bez żadnych dodatkowych zmian w kodzie. Plik powinien być tekstowy i intuicyjny w obsłudze. Podczas gry, ukończenie kolejnego etapu powinno skutkować pewną dodatkową liczbą punktów. (1 pkt.)

## Wymagania „z gwiazdką” (3 pkt.)

Tą część można zaimplementować tylko wówczas, gdy zaimplementowano wszystkie wcześniejsze elementy.

- *Nagroda: strzelająca paletka.* Nagroda pojawia się losowo w efekcie trafienia w cegielkę. Paletka ma możliwość wypuszczania pocisków w kierunku pionowym, z pewną ustaloną częstotliwością przez czas 20 sekund. Wypuszczenie pocisku następuje po przyciśnięciu przycisku **f**. Kolizja pocisku z cegielką powoduje usunięcie obu obiektów z planszy. Uzyskanie nagrody powoduje również uzyskanie pewnej liczby punktów. (1 pkt.)
- *Ruchome obiekty.* W sposób losowy (w losowych ostępach czasu), w wybranym punkcie na brzegu planszy generowane są niewielkie ruchome obiekty. Obiekty te podlegają kolizjom z otoczeniem: nie mogą przenikać sobie nawzajem, brzegu planszy, cegiełek i blokad. Ich ruch jest losowy i powinien być opracowany w taki sposób, aby sprawiały wrażenie eksploracji całej planszy (należy unikać efektu kumulacji ruchomych obiektów w jednym obszarze – brak „rozproszenia”). Obiekt ruchomy w kontakcie z paletką jest usuwany z planszy. W przypadku kontaktu z piłeczką, ruchomy obiekt również jest usuwany z planszy, a piłeczka zmienia losowo swój kierunek ruchu (każdy nowy kąt spośród 360 stopni jest możliwy). (2 pkt.)

## Uwagi końcowe

- Wymagania dotyczące szaty graficznej: wystarczające jest użycie dowolnych bitmap (dobranych właściwie pod względem rozmiaru).
- Wszystkie nagrody powinny być graficznie rozróżnialne: ikonka nagrody poruszająca się ku dołowi planszy powinna być inna dla każdej z nagród.
- Konfiguracja programu powinna umożliwiać łatwą zmianę wszelkich parametrów (prędkości, liczby punktów za poszczególne zdarzenia, rozmiar listy najlepszych wyników, domyślna liczba przegranych rozgrywek, prawdopodobieństwa wystąpienia poszczególnych zdarzeń, itd.). Przez łatwą zmianę rozumiemy modyfikację stałej w programie.