

# Dokumentacja końcowa projektu

## PacMan

Przedmiot: Zaawansowane programowanie w C++

Tytuł projektu: Gra „Pac-Man” w C++.

Autorzy projektu: Jakub Wrzosek, Maria Skarbek

### Opis projektu

W ramach projektu została zrealizowana gra PacMan. Jej działanie zostało oparte o bibliotekę SFML (ang. Simple and Fast Multimedia Library). Działanie aplikacji oparte jest na pętli głównej, w której wywoływanych jest 6 głównych metod z klasy Game odpowiedzialnej za integrację wszystkich elementów aplikacji.

Pierwszą z nich jest metoda *handleGhosts()*, która kontroluje zachowanie duszków, kolejna to *handleInput()*, która reaguje na sygnały sterujące pochodzące z klawiatury. W kolejnym kroku wołana jest metoda *update()*. Jej zadaniem jest aktualizowanie stanu rozgrywki poprzez wpływanie na inne obiekty i ich właściwości. Potem występuje metoda *render()* odpowiedzialna za rysowanie obiektów w oknie rozgrywki oraz metoda *restartClock()*, która restartuje zegar gry. Na sam koniec wywoływana jest metoda *sleep()*, która na 1ms odciąża procesor.

Głównymi składowymi klasy Game są obiekty:

- Window.
- Map,
- Player,
- Oraz duchy: ClydeGhost, PinkyGhost oraz BlinkyGhost.

Obiekt	Zadanie
Window	Zawiera składowe (pochodzące głównie z biblioteki SFML) niezbędne do zapewnienie okna rozgrywki
Map	Zawiera elementy składające się na mapę gry. Są to między innymi ściany czy kulki, które zjada Pacman.
Player	Reprezentuje Pacmana. Implementuje metody takie jak poruszanie po labiryncie czy sprawdzanie kolizji z elementami składowymi mapy.

Duchy: ClydeGhost, PinkyGhost, BlinkyGhost	Obiekty ClydeGhost, PinkyGhost oraz BlinkyGhost reprezentują słynne duszki, które przemierzają planszę gry, a jeśli PacMan jest dostatecznie blisko, to go gonią i uśmiercają (jeśli go dogonią)
--------------------------------------------	--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

### Lista zrealizowanych funkcjonalności:

- Okno rozgrywki wraz z wyświetlaniem na nim wszystkich obiektów (Pacmana, kulek, które pożera, duszków, ścian),
- Wyświetlanie żyć PacMana oraz punktów zbieranych przez PacMana,
- Przemieszczanie się PacMana po planszy oraz przemieszczanie się przez Portal,
- Poruszanie się duszków w różnych trybach: trybie startu, trybie rozproszenia, trybie gonienia i tzw. trybie pomiędzy różnymi stanami,
- Gonienie PacMana przez duszki jeśli znajduje się w odpowiedniej odległości,
- Restart PacMana na pozycję startową oraz tracenie przez niego życia,
- Tracenie zainteresowania przez duszki po upływie czasu gonienia i powracanie przez nie do trybu rozproszenia.

### Lista niezrealizowanych funkcjonalności:

- Menu główne,
- Przechowywanie listy najlepszych wyników,
- Pojawianie się owoców w labiryncie (bonusy do punktacji),
- Wyświetlanie komunikatu po przejściu do kolejnego poziomu,
- Sprawdzenie czy uzyskany podczas rozgrywek kwalifikuje się do listy 10 najlepszych wyników (jeśli tak – odpowiednia modyfikacja pliku z wynikami),
- Tryb przerażenia.

### Niepowodzenia oraz wnioski.

Niestety nie udało się zrealizować wszystkich zaplanowanych w dokumentacji wstępnej zadań. Wiele z nich zajęło w rzeczywistości znacznie więcej czasu co wpłynęło na brak realizacji niektórych funkcjonalności. W związku z tym skupiliśmy się głównie na implementacji samej rozgrywki rezygnując z tworzenia menu gry. Implementacja poruszania się duszków zabrała bardzo dużo czasu, bardzo trudnym zadaniem okazała się synchronizacja wszystkich elementów rozgrywki.

Brak doświadczenia w realizacji projektów grupowych spowodował w naszym przypadku wiele problemów podczas pisania aplikacji. Nie zadbaliśmy od samego początku o opracowanie odpowiedniej strategii działania, co przełożyło się później na dużo dodatkowej pracy przy scalaniu programu, w którym kilkakrotnie pojawiała się konieczność gruntownej przebudowy.

Kolejnym popełnionym przez nas błędem było zbytne skupienie się na opracowywaniu jednej funkcjonalności od początku do końca, co skutkowało brakiem

utrzymywania aplikacji w stanie działanie przez cały okres pracy nad nią. Skutkiem tego były sytuacje, w których okazywało się, że sposób w jaki została zaimplementowana jedna funkcjonalność uniemożliwiał zrealizowanie innej. Dodatkowo powodowało to problemy przy podziale obowiązków. Bywały okresy, w których praca jednej osoby była wstrzymywana przez oczekiwanie na inne funkcjonalności.

Błędy popełnione przy organizacji pracy oraz niedoszacowanie czasu potrzebnego na realizację spowodowały pojawienie się ostatecznej wersji projektu dość późno. Ucierpiały przez to testy, których nie byliśmy w stanie zrealizować tak dokładnie jakbyśmy tego chcieli.

Wyżej wspomniane kwestie skłaniają do refleksji, aby w przyszłości większy nacisk położyć w początkowej fazie projektu na planowanie oraz organizację pracy. Pozwoliłoby to na uniknięcie wielu przykrych niespodzianek opóźniających pracę. Ponadto uświadamiają nam jak dużo nieprzewidzianych kwestii pojawia się w trakcie realizacji projektu i jak ważne jest świadome podejście do jego planowania. Dodatkowo dostrzegamy teraz jak ważne jest posiadanie działającej wersji programu przez cały proces jego powstawania.

Tabela zadań

Lp.	Zadanie	Czas przewidziany w dokumentacji wstępnej	Rzeczywisty czas przeznaczony na realizację
<b>Ogólne</b>			
1.	Zapoznanie się z biblioteką SFML.	15h	45h
2.	Projekt architektury.	10h	20h
3.	Wykonanie/znalezienie tekstur.	10h	2h
4.	Implementacja menu głównego.	5h	---
<b>Okno opcji</b>			
5.	Implementacja okna opcji.	5h	---
6.	Zapis ustawień użytkownika do pliku konfiguracyjnego	5h	---
<b>Okno najlepszych wyników</b>			
7.	Implementacja okna najlepszych wyników.	5h	---
	Wczytywanie najlepszych wyników z pliku	5h	---
<b>Okno rozgrywki</b>			
8.	Implementacja okna rozgrywki	20h	50h
9.	Pobieranie nazwy użytkownika	2h	---
10.	Wczytywanie konfiguracji z pliku (plansza, poziom trudności itp.)	2h	8h
11.	Implementacja sterowania i ruchu Pac-Mana	10h	30h
12.	Implementacja algorytmów poruszania się dla każdego ducha (w trybie normalnym oraz w trybie przerażenia)	20h	100h
13.	Implementacja odradzania się duchów.	5h	20h

14.	Implementacja odradzania się Pac-Mana po straconym życiu	3h	5h
15.	Implementacja zliczania punktów oraz żyć.	5h	7h
16.	Wyświetlanie komunikatów po przejściu do kolejnego poziomu.	1h	--
17.	Sprawdzanie czy uzyskany wynik powinien znaleźć się wśród 10 najlepszych (znajdujących się w pliku)	3h	--
<b>Testowanie</b>			
18.	Testowanie aplikacji.	30h	30h
<b>Pisanie dokumentacji</b>			
19.	Pisanie dokumentacji	15h	15h
		<b>Razem</b> =176h	<b>Razem</b> =329h