Politechnika Świętokrzyska
Wydział Elektrotechniki, Automatyki i Informatyki
Kierunek Informatyka
Rok 2021 Semestr 4

Programowanie w języku Java (Projekt)

Temat: Gra Arkanoid

Grupa 2ID13B
Rok 2.
Zespół projektowy:
Wiktor Simlat
Jakub Stawiarz

Opis Projektu:

Gra Arkanoid - Gracz porusza paletką, odbija piłkę, która ma zniszczyć wszystkie bloczki znajdujące się na ekranie. Otrzymuje za to punkty a zniszczenie wszystkich bloczków równa się wygranej i przejściem do następnego poziomu. Wypadnięcie piłeczki z obszaru gry czyli przez podłogę kończy się utraceniem żyć oraz przegraną.

Do poprawnego działania gry stworzyliśmy pełny silnik gry Arkanoid. Poruszanie paletki realizowane jest przy użyciu strzałek czyli zmniejszanie lub zwiększanie wartości współrzędnych, tak samo dla piłeczki. Dodaliśmy warunek wygranej i przegranej czyli życia jak i również punkty za zbijanie bloczków i ukończenie poziomu. Kolizje wszystkich bytów gry są sprawdzane przy pomocy porównywania współrzędnych oraz wykonania odpowiednich działań, jeśli taka kolizja wystąpiła. Obecny jest też interfejs graficzny dla gracza do obsługi gry.

Zastosowane Środowiska:

Do wykonania projektu wykorzystaliśmy środowisko programistyczne w języku Java o nazwie IntelliJ IDEA 2020.3.4 wraz z biblioteką graficzną libGDX do tworzenia grafiki na ekranie oraz z Gradle do pomocy w budowie projektu. Do wykonania grafiki wykorzystaliśmy MS Paint oraz do efektów dźwiękowych różne miksery dzwięków 8-bitowych.

Uruchomienie projektu:

Aby uruchomić projekt należy zaimportować cały folder jako projekt do IDE IntelliJ IDEA lub otworzyć plik build.gradle z głównego folderu projektu przy pomocy IDE. Następnie uruchomić klasę DesktopLauncher zawartą w folderze \desktop\src\com\ark\game\desktop

Klasy i wybrane ważne metody projektu:

DesktopLauncher – Klasa uruchamiająca grę z biblioteki libGDX
Arkanoid – Główna Klasa gry
create() - Metoda z klasy Arkanoid wywoływana na początku do inicjalizacji silnika gry.
render() - Metoda wywoływana w każdym kroku gry do generowania obrazu/sterowania
BackgroundFX – Klasa z animowanym tłem do gry
draw() - Metoda rysująca tło animowane
BallObject – Klasa z definicją piłki z gry z poruszaniem, rysowaniem i kolizją.
resetBall() - Ustawianie początkowych wartości piłki.
draw() - Rysowanie piłki na ekranie
movement() - Poruszanie piłki przez dodawanie wartości prędkości do współrzędnych

colPaddle() - Kolizja z paletką
gluedBall() - Przyklejanie piłki do paletki

BlockObject – Klasa z danymi pojedynczego bloczku

BlockGroup – Klasa z grupą wszystkich bloczków
draw() - Rysuje wszystkie bloczki
colBall() - Sprawdzenie kolizji piłki z każdym bloczkiem
levelGen() - Tworzenie nowego poziomu przez ustawienie bloczków

GameEngine – Główna klasa silnika gry łącząca wszystkie klasy do wyświetlania i inputu
drawScreen() - Determinuje, czy ma być rysowana gra czy menu
handleInput() - Determinuje, skąd jest przyjmowane sterowanie (Menu lub gra)
Events() - Determinuje, czy wydarzenia są brane z menu czy gry
drawGameScreen() - Rysuje ekran gry z elementami
handleGameInput() - Obsługuje sterowanie w grze
gameEvents() - Obsługa wydarzeń w grze
drawMenuScreen() - Rysuje ekran menu
handleMenuInput() - Obsługuje sterowanie w menu
menuEvents() - Obsługa wydarzeń w menu takich jak wybór opcji

Hud – Klasa zawierająca interfejs graficzny w grze
draw() - Rysuje interfejs w grze

Menu – Klasa zawierająca menu gry
draw() - Rysuje menu
setSound() - Zmienia ustawienie głośności
input() - Obsługuje sterowanie → Wybiera odpowiednią opcję przez sterowanie

PaddleObject – Klasa zawierająca paletkę i jej parametry
draw() - Rysuje paletkę na ekranie
resetPaddle() - Przywraca paletkę do jej początkowego położenia
getInput() - Steruje paletką

Player – Klasa zawierająca informacje o graczu takie jak wynik, życia
newGame() - ustawia początkowe parametry gracza
checkLevelComplete() - Sprawdza, czy poziom został ukończony
incScore() - Dodaje do wyniku określoną w parametrze wartość
decLive() - Zmniejsza życia w wypadku porażki oraz przerywa grę
levelManagement() - Przechodzenie do następnego poziomu w wypadku ukończenia
highScoreManagement() - Nadaje nową wartość w wypadku osiągnięcia wysokiego wyniku

SoundFX – Klasa do obsługi efektów dźwiękowych
sound_set_on_off() - Ustawienie dźwięku i włączenie muzyki
playSound() - Odtworzenie wybranego efektu dźwiękowego określonego w parametrze

Projekt został wykonany we współpracy i ciągłej komunikacji przy użyciu komunikatora Discord oraz środowiska udostępniającego efekty pracy nad projektem GitHub przez wszystkich członków grupy projektowej.

Sterowanie:

Po menu poruszamy się przy pomocy Strzałek góra, dół oraz potwierdzamy przyciskiem ENTER. W grze poruszamy się strzałkami lewo, prawo. Piłeczkę początkowo odbijamy przyciskiem SPACJA. Aby przejść do menu naciskamy przycisk ESC.

Zrzuty Ekranu:


LEVEL: 2
SCORE: 7100
HIGH SCORE: 7100