Sprawozdanie

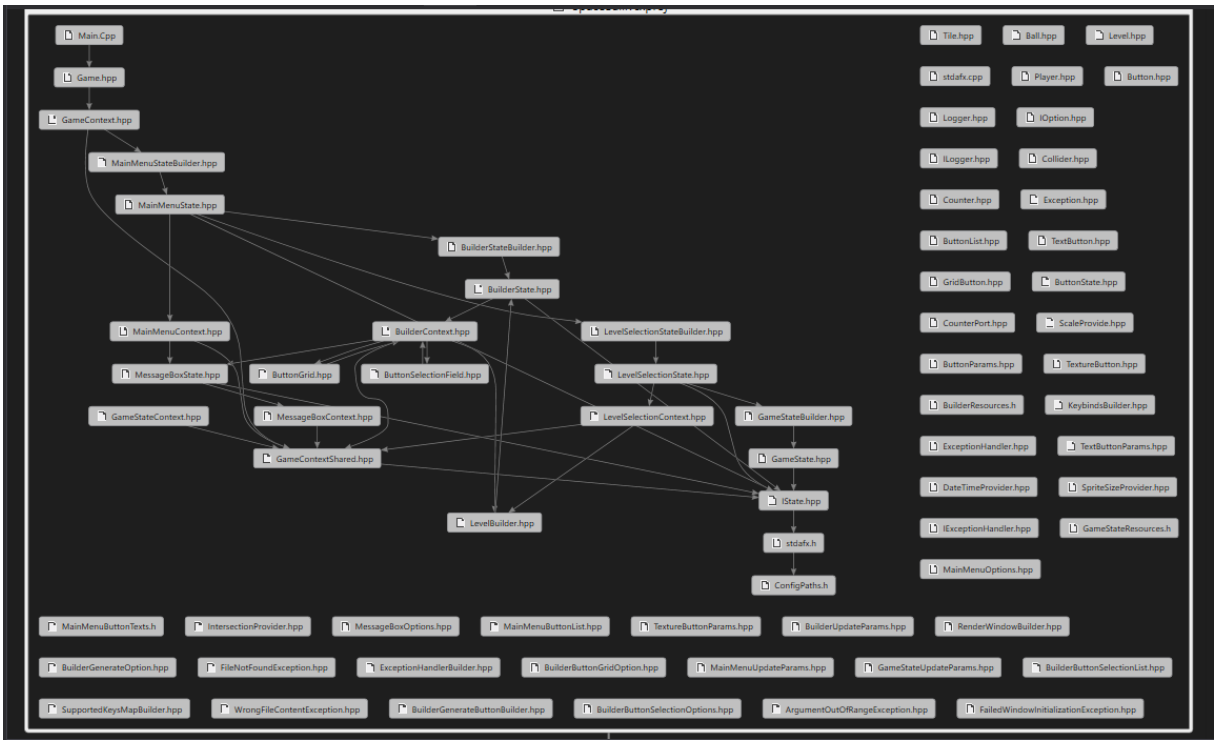
Projekt Space-Ball

Temat projektu	Gra typu arkanoid.
Autor	Aleksander Kijowski
Kierunek studiów	Informatyka
Rodzaj studiów	SSI
Semestr	IV
Prowadzący	Łukasz Dyga
GitHub	https://github.com/AleksanderKijowski/Space-Ball

Założenia projektu:

W ramach projektu powinna zostać zrealizowana w pełni funkcjonalna gra typu arkanoid napisana przy użyciu języka C++ i biblioteki graficznej SFML. Dodatkowo aplikacja powinna udostępniać narzędzie do własnoręcznego tworzenia poziomów .

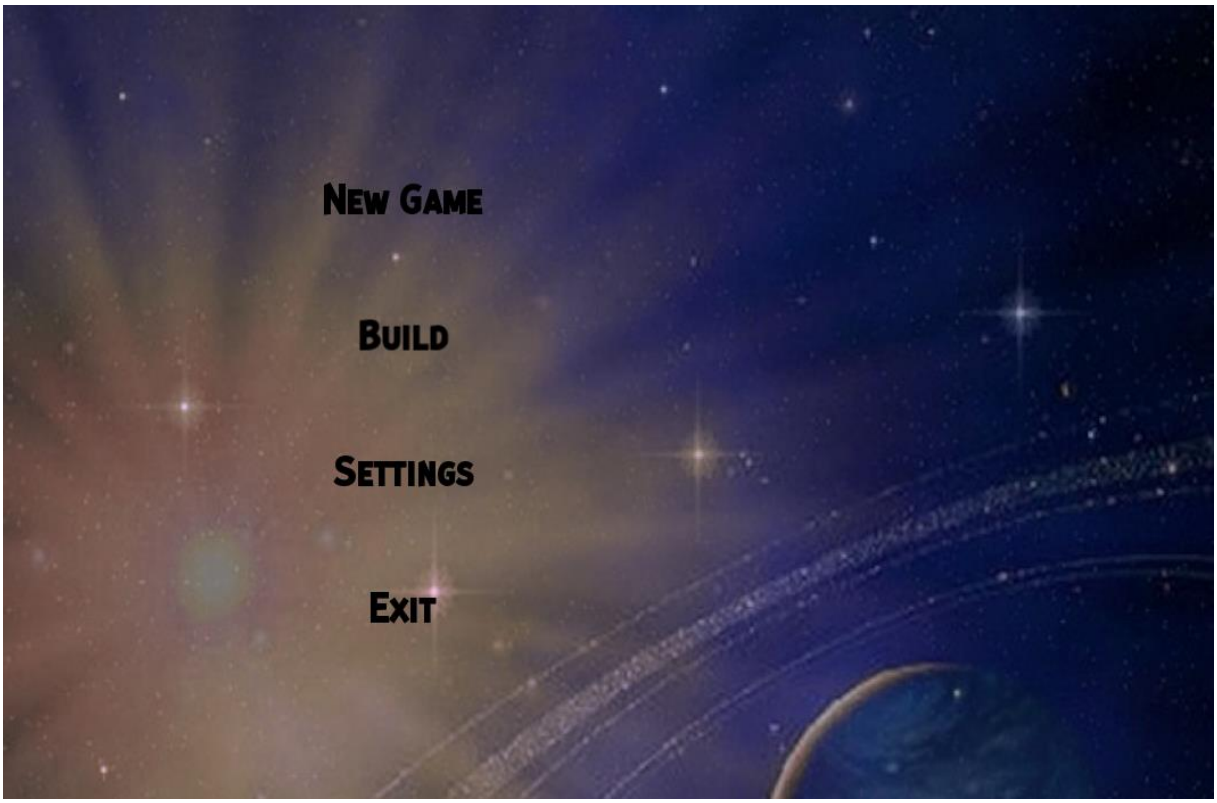
Diagram klas występujących w programie:



Rys. 1.1 Diagram klas.

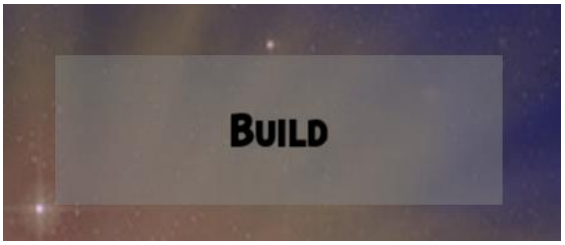
Korzystanie z aplikacji:

Aby skorzystać z aplikacji należy uruchomić plik .exe. Po jego uruchomieniu pokaże nam się widok menu głównego aplikacji. Wygląda ono w następujący sposób:



Rys. 1.2 Menu główne aplikacji.

Wybranie dowolnej opcji menu odbywa się poprzez naciśnięcie lewym klawiszem myszki po wcześniejszym najechaniu kursorem na odpowiedni przycisk (aplikacja sygnalizuję taki stan poprzez podświetlenie klawisza. Zostało to zaprezentowane poniżej:



Rys. 1.3 Prezentacja podświetlania klawiszy.

Aplikacja nie posiada predefiniowanych poziomów, więc aby pierwszy raz zagrać w grę najpierw należy zbudować poziom. Proces ten rozpoczynamy od wybrania pozycji „Build” z menu głównego.

Pokaże nam się wtedy następujący widok:

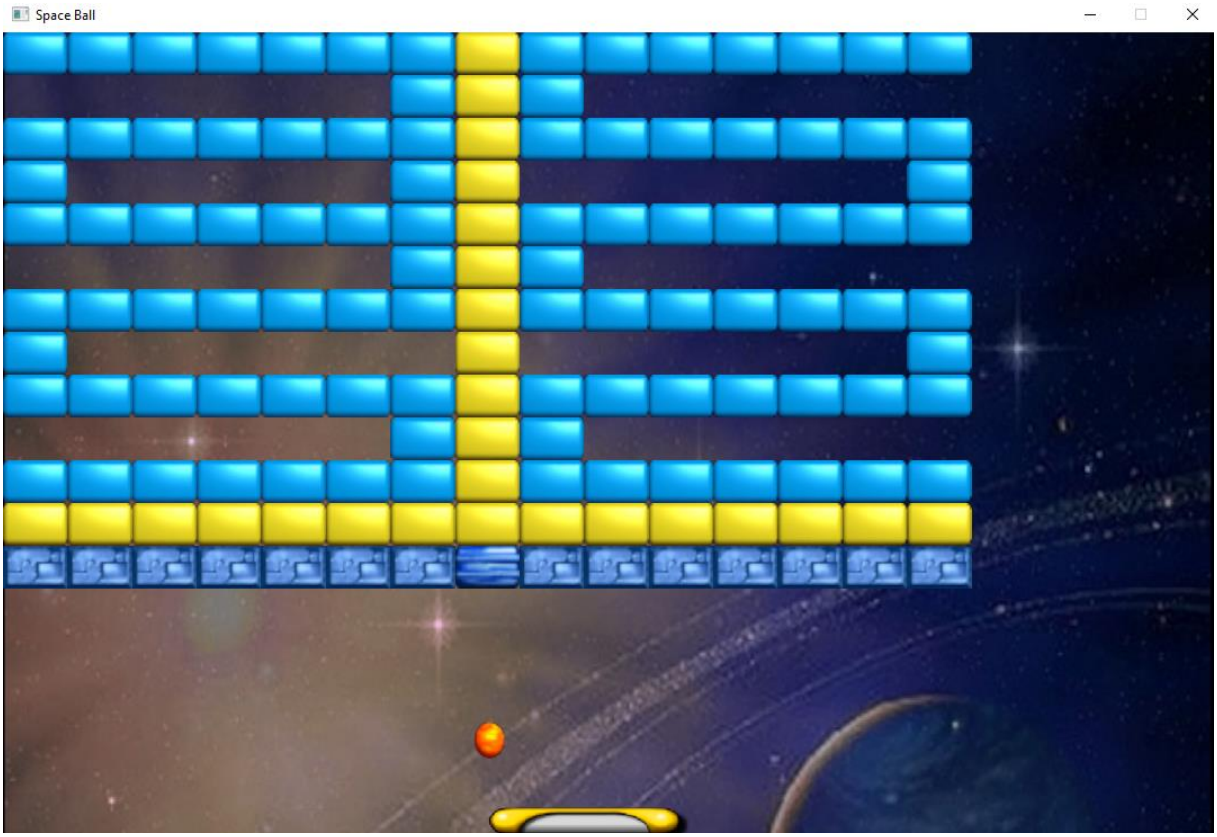
Należy poprzez kliknięcie wybrać któryś z klocków po lewej stronie.

Teraz możemy tworzyć poziom, wybrany przez nas klocek stawiamy na odpowiedniej pozycji poprzez naciśnięcie lewego przycisku myszy na odpowiedniej pozycji w kratce po prawej stronie. Możemy usuwać postawione klocki za pomocą prawego klawisza myszy. Dodatkowo w każdej chwili jest możliwa zmiana wybranego klocka poprzez wybranie innego klocka z listy. Po zakończeniu tworzenia poziomu naciskamy klawisz „Generate”. Pokażę, nam się odpowiedni widok informujący o utworzeniu nowego poziomu i informujący o nadanej mu automatycznie nazwie. Przedstawiono go poniżej:

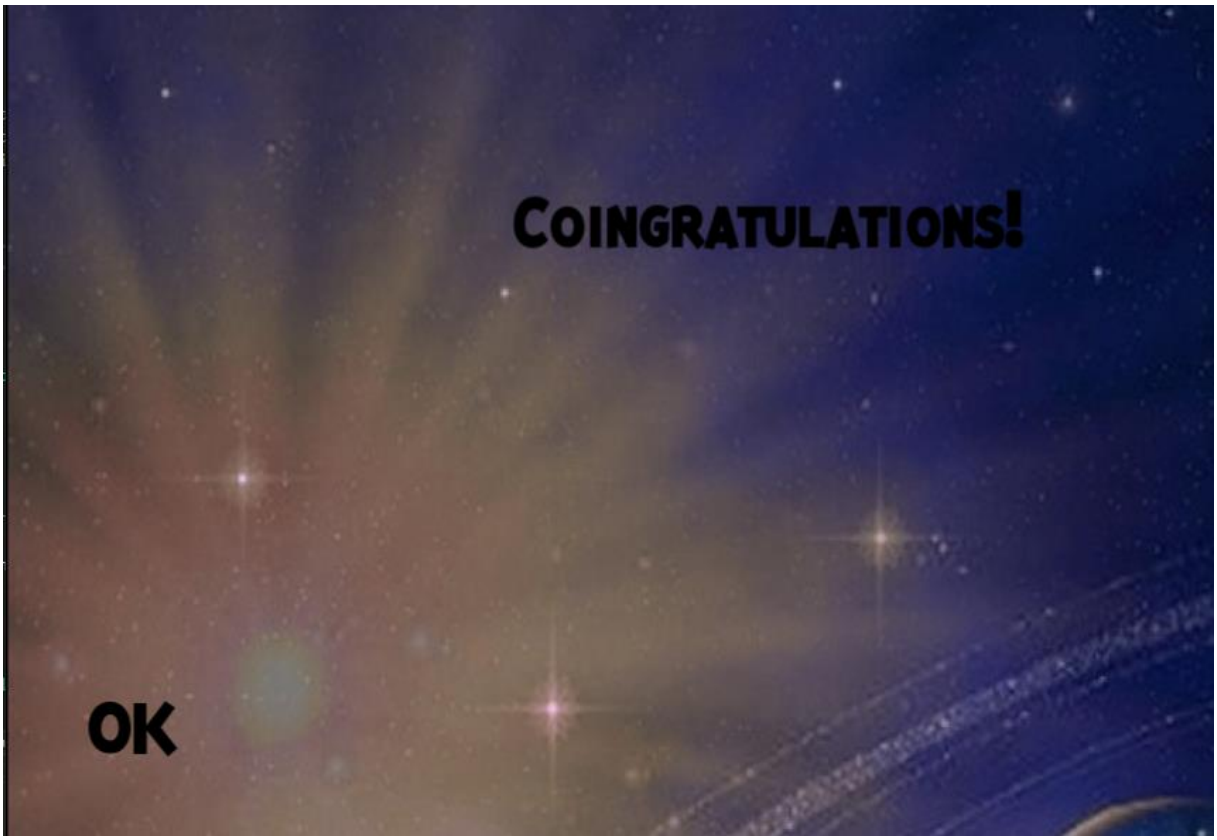
Potwierdzenia dokonujemy naciskając klawisz „OK”. Powracamy do menu głównego.

Teraz możemy rozpocząć nową grę. Naciskamy klawisz „New Game” i przechodzimy do widoku odpowiadającego za wybór poziomu.

Wybieramy poziom w który chcemy zagrać korzystając z przycisków „-” i „+”. Potwierdzamy naciskając klawisz „Ok”. Następuje przekierowanie do gry:



Grę rozpoczynamy poprzez naciśnięcie klawisza „spacja”. Sterowanie graczem odbywa się za pomocą strzałek. Po zbiciu wszystkich klocków lub trzykrotnym upadku piłeczki pokaże nam się okno informujące odpowiednio o sukcesie/porażce. Następuje przekierowanie do widoku menu głównego.

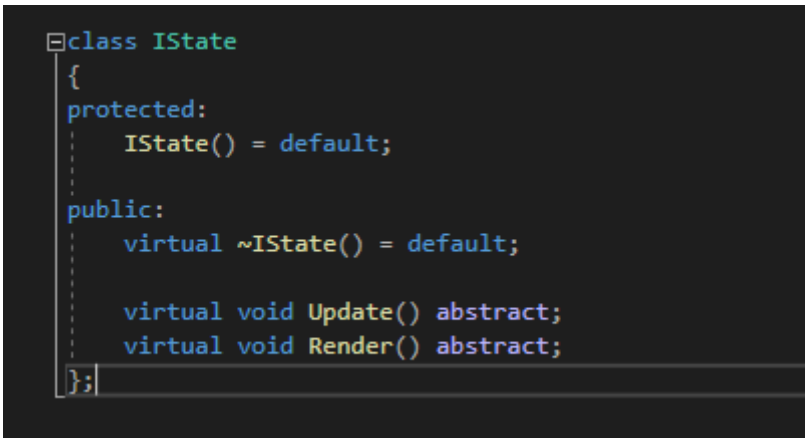


W menu zostały dwa nieomówione przyciski – „Settings” i „Exit”. Przycisk settings w tej wersji aplikacji nie został podłączony do żadnej funkcjonalności. Przycisk „Exit” wychodzi z aplikacji.

Rozwiązania i algorytmy w aplikacji:

Klasą zarządzającą aplikacji jest klasa game. Initializuję ona najistotniejsze struktury, mechanizmy i pętlę główną programu. Pierwszym takim mechanizmem jest mechanizm obsługi wyjątków ExceptionHandler oraz mechanizm logowania Logger. Przyjąłem strategię aby wszystkie przewidziane wyjątki w programie (np. brak pliku z danymi, brak załadowanej tekstury itp.) dziedziczyły po klasie czysto wirtualnej exception. Mechanizm ten gwarantuję, że wystąpienie któregoś z tych wyjątków nie spowoduję wyłączenia programu. Mechanizm logowania polega na zapisywaniu wiadomości przechowywanych przez wyjątki do pliku tekstowego, który nazwą odpowiada dacie wystąpienia wyjątku.

Kolejną istotną strukturą jest stos stanów (State’ów). Każdy widok w aplikacji nazywany jest stanem. Klasy te implementują interfejs IState.



Jak widać użyto tutaj polimorfizmu, jest to wygodne ponieważ klasa game nie potrzebuję znać aktualnego stanu a jedynie przekazać do niego kontrolę. Wybrano strukturę stosu ponieważ warto pamiętać kolejność wkładania stanów na stos w przypadku w którym chcemy umożliwić płynne przechodzenie wstecz pomiędzy widokami.

Stany aplikacji:

- MainMenuState
- GameState
- BuilderState
- LevelSelectionState
- MessageBoxState

Poniżej w kilku słowach opiszę za co każdy widok odpowiada.

MainMenuState:

Wyświetla przyciski