Napisz program będący realizacją gry w stylu "Pacman". Przykład rozgrywki dostępny **tutaj**. W rozgrywce przeciwnicy raz na 5 sekund z prawdopodobieństwem 25% tworzą ulepszenia (np. +50% prędkości poruszania się itp.), które gracz może zebrać. Zaimplementuj min. 5 różnych niebanalnych ulepszeń. Program po uruchomieniu wyświetla menu główne składające się z opcji:

- *New Game*
- *High Scores*
- *Exit*

Po uruchomieniu nowej gry, gracz zostanie zapytany o rozmiar planszy na której chcemy grać. Należy zaimplementować możliwość utworzenia okna o rozmiarach od 10 do 100 wierszy/kolumn. Następnie w nowym oknie wyświetlana jest wygenerowana plansza gry. Musi być ona zrealizowana w oparciu o komponent *JTabel*, w którym w komórkach będziemy wyświetlać pola planszy. Obowiązkowe jest zaimplementowanie własnego modelu *JTable* opartego o *AbstractTableModel*. Implementacja planszy w inny sposób niż na JTable nie otrzyma żadnych punktów.

Należy zapewnić w pełni funkcjonalny interfejs graficzny. *CLI* może być tylko pomocą informacyjną dla programisty, ale nie może zachodzić tam żadna interakcja z użytkownikiem.

Podczas gry musi być również widoczny licznik punktów, czasu, żyć i inne potrzebne elementy interfejsu graficznego, które będą aktualizowane na żywo podczas rozgrywki.

W projekcie należy wykorzystać pliki graficzne oraz postarać się o spójny efekt wizualny całej aplikacji z uwzględnieniem wyglądu okien takich jak *Menu*, *High Scores* i *okno z rozgrywką*.

Obowiązkowe jest wykorzystanie plików graficznych w celu wizualizacji każdego elementu okna. Zrealizuj ruch postaci i wykonywane zadania poprzez proste animacje poklatkowe (np. animacja ruchu, animacja jedzenia itp). Przykład animacji: ***http://pj.edu.pl/pacmanAnim***. Zaimplementuj to samodzielnie w oparciu o wątki, a nie np. przez pliki GIF.

Wszystkie kwestie związane z czasem należy zrealizować z uwzględnieniem wątków (nie wolno wykorzystywać klasy *Timer*, *Executor* i innych). Należy zapewnić przemyślaną i poprawną synchronizację wątków. Nie można łączyć różnych funkcjonalności w jeden wątek.

Gra toczy się według wyżej wymienionych zasad. Należy zapewnić możliwość przerwania gry w dowolnym momencie poprzez ***złożony skrót klawiszowy*** (***Ctrl+Shift+Q***), który spowoduje powrót do menu głównego. Należy zapewnić, żeby skrót działał w **każdym** momencie rozgrywki.

Po zakończeniu gry, w nowym oknie gracz proszony jest o nazwę pod jaką ma być zapisany w rankingu. Zapewnij trwałość rankingu wykorzystując interfejs ***Serializable*** i zapis do pliku.

Po wybraniu opcji rankingu z menu głównego, zostaje on wyświetlony użytkownikowi. Ponieważ okno rankingu może być relatywnie duże, należy zadbać o paski przewijania.

Aplikację zaimplementuj z wykorzystaniem wzorca programistycznego MVC z pełną obsługą zdarzeń zaimplementowaną przez delegacyjny model obsługi zdarzeń

Wskazówki:

- Należy zadbać o wyjątki w programie i wyświetlać stosowne komunikaty użytkownikowi.
- Ranking należy zrealizować przy pomocy komponentu ***JList***.
- Należy zadbać o skalowalność okien aplikacji.
- Przy mniejszych i informacyjnych oknach można wykorzystać okna dialogowe.

Projekt opiera się o materiał z zakresu GUI.

Proszę pamiętać o wymogach formalnych opisanych w zasadach zaliczenia!

Uwaga:

- ***W przypadku otrzymania projektu niezgodnego w powyższymi wymaganiami lub ze znacznymi brakami w implementacji, z niezaimplementowaną rozgrywką lub niekompilującego się skutkować to będzie wyzerowaniem punktacji za projekt***
- ***Zabrania się wykorzystywania narzędzi WYSIWYG (tzw. Window/Scene Builder'ów).***
- ***Zabronione jest wykorzystanie narzędzi AI przy realizacji projektu***
- ***Proszę pamiętać, aby nie udostępniać swoich rozwiązań innym osobom.***
- ***Brak znajomości dowolnej linii kodu lub wysokie podobieństwo nadesłanego rozwiązania z innym rozwiązaniem skutkować będzie niezaliczeniem przedmiotu.***
- ***W ocenie projektu poza praktyczną i merytoryczną poprawnością będzie brana również pod uwagę optymalność, jakość i czytelność napisanego przez Państwa kodu.***
- ***Ważną częścią projektu jest wykorzystanie między innymi: dziedziczenia, kolekcji, interfejsów lub klas abstrakcyjnych, lambda-wyrażeń, typów generycznych oraz innych elementów omawianych w trakcie semestru***
- ***Poza praktyczną i merytoryczną poprawnością będzie brana również pod uwagę optymalność, jakość i czytelność napisanego przez Państwa kodu.***