**GRA „LABIRYNT”**

Stworzony w projekcie tym system jest **grą tekstową,** której celem jest eksploracja labiryntu oraz zbieranie skarbów i szukanie ich po ewentualnych kradzieżach złodziei, by na końcu móc dotrzeć do głównej komnaty – Świątyni Skarbów – w której znajduje się główny skarb (Korona Życzeń). By przejście do tej komnaty otworzyło się, gracz musi zdobyć wszystkie pozostałe skarby. W owej grze można zdobyć maksymalnie 1000 punktów, które przyznawane są za posiadane w inwentarzu gracza skarby. Podczas eksploracji labiryntu gracz musi również poszukiwać kluczy, bez których otworzenie drzwi łączących poszczególne komnaty jest niemożliwe.

**Spis klas i opis architektury systemu:**

System oparty jest na paradygmacie obiektowym i można go logicznie podzielić na następujące warstwy funkcjonalne:

- **model domenowy (świat gry)** – opisuje wszystkie obiekty istniejące w świecie gry, przechowuje dane oraz podstawowe reguły gry; składa się z następujących klas:

- Przedmiot,

- Klucz,

- Skarb,

- Przejscie,

- Drzwi,

- PrzejscieNaPunkty,

- Komnata,

- **postacie w grze** – reprezentują uczestników gry – zarówno gracza, jak i jego przeciwników (złodziei); składa się z następujących klas:

- Gracz,

- Stworzenie,

- **zarządzanie światem** – tworzy cały świat gry, rozmieszcza elementy oraz ustawia stany początkowe; składa się z następującej klasy:

- Mapa,

- **sterowanie grą** – obsługuje pętle gry, przetwarza polecenia użytkownika i zarządza przepływem informacji; składa się z następującej klasy:

- Rozgrywka.

**Przedmioty w grze:** Stworzona została klasa Przedmiot, której podklasami są: Skarb oraz Klucz.

**Komnaty i przejścia:** Stworzona została klasa Przejscie, której podklasami są: Drzwi oraz PrzejscieNaPunkty. Każde Przejscie pamięta komnatę, do której prowadzi w atrybucie komnata\_docelowa (asocjacja z Komnata). Ponadto, Drzwi pamiętają też drugą komnatę w atrybucie komnata\_docelowa2, by zachowywać informacje o obu komnatach, które dane Drzwi łączą (asocjacja z Komnata). Klasa ta pamięta również klucz, który pozwala na otworzenie danych drzwi w atrybucie pasujacy\_klucz (asocjacja z Klucz). Poza tym, klasa Drzwi wykazuje zależności z klasami Komnata i Gracz, gdyż wykorzystuje je jako argumenty swych operacji. Podobnie, występuje zależność klasy PrzejscieNaPunkty z klasą Gracz, gdyż przejście to korzysta z informacji o punktach danego gracza w swym działaniu.

Z kolei obiekty klasy Komnata tworzą graf przestrzenny, połączony poprzez obiekty typu Przejscie. Każda Komnata składa się z następujących części składowych, które przechowuje w odpowiednich atrybutach: przedmioty (atrybut: przedmioty, agregacja z Przedmiot), przejścia (atrybut: przejscia, agregacja z Przejscie) oraz stworzenia (atrybut: stworzenia, agregacja z Stworzenie). Ponadto, klasa Komnata wykazuje zależność z klasą Gracz, gdyż wykorzystuje ją jako argument operacji służącej do opisu komnaty, by stwierdzić, czy dane przejścia dla danego gracza są otwarte.

**Postacie w grze:** Klasa Gracz reprezentuje użytkownika gry. Pamięta on swoją aktualną pozycję w atrybucie aktualna\_komnata (asocjacja z Komnata). Ponadto, częścią składową każdego gracza jest jego inwentarz, w którym przechowuje wszystkie zebrane podczas rozgrywki przedmioty (atrybut: inwentarz, agregacja z Przedmiot). Klasa Gracz posiada również wiele metod, które pozwalają graczowi przede wszystkim na przechodzenie między komnatami, otwieranie drzwi oraz zbieranie/odkładanie przedmiotów.

W grze występuje również klasa Stworzenie, która reprezentuje złodziei. Podobnie do poprzedniej klasy, Stworzenie pamięta swoją aktualną pozycję w atrybucie komnata\_aktualna (asocjacja z Komnata). Dane Stworzenie porusza się losowo między komnatami oraz może wejść w interakcje z graczem: może ukraść skarb z inwentarza gracza oraz przenieść go do innej, również losowej komnaty. Wynika z tego, iż występują zależności między klasą Stworzenie a klasami Komnata oraz Gracz.

**Mapa i rozgrywka:** Klasa Mapa tworzy, dodaje i łączy ze sobą wszystkie elementy gry. Elementy te przechowuje w atrybutach: komnaty (kompozycja z Komnata), stworzenia (kompozycja z Stworzenie) oraz gracz (kompozycja z Gracz).

Z kolei klasa Rozgrywka jest głównym kontrolerem gry: uruchamia grę, przetwarza polecenia użytkownika, zarządza ruchem stworzeń i aktualizacją stanu oraz prezentuje mapę gry. Klasa Rozgrywka tworzy oraz pamięta mapę w atrybucie mapa (kompozycja z Mapa). Poza tym, dla klasy Rozgrywka występują także zależności z klasami Gracz (poprzez wywoływane metody) oraz Stworzenie (poprzez metody ruchu i interakcji).

**Opis każdej klasy:**

**1) Przedmiot** :

- znaczenie: klasa bazowa dla wszystkich przedmiotów w grze,

- atrybuty:

- nazwa (str) – nazwa danego przedmiotu,

- opis (str) – opis danego przedmiotu,

- metody:

- \_\_init\_\_(self, nazwa, opis) – inicjalizuje obiekt przedmiotu; przechowuje jego nazwę i opis,

- opisz(self) – zwraca tekstowy opis przedmiotu w formacie „nazwa: opis”,

**2) Klucz:**

- znaczenie: podklasa Przedmiot; reprezentuje klucz otwierający konkretne drzwi,

- atrybuty:

- nazwa (str) – dziedziczone z Przedmiot; nazwa danego klucza,

- opis (str) – dziedziczone z Przedmiot; opis danego klucza,

- metody:

- \_\_init\_\_(self, nazwa, opis) – tworzy obiekt klucza na bazie klasy Przedmiot, przekazując nazwę i opis,

- metody z klasy nadrzędnej Przedmiot,

**3) Skarb:**

- znaczenie: podklasa Przedmiot; specjalny rodzaj przedmiotu, który przyznaje graczowi daną ilość punktów,

- atrybuty:

- nazwa (str) – dziedziczone z Przedmiot; nazwa danego skarbu,

- opis (str) – dziedziczone z Przedmiot; opis danego skarbu,

- wartosc (int) - liczba punktów przyznawana graczowi za zdobycie danego skarbu,

- metody:

- \_\_init\_\_(self, nazwa, opis, wartosc) – tworzy obiekt skarbu na bazie klasy Przedmiot, dodatkowo przypisując wartość punktową, jaką przyznaje graczowi,

- opisz(self) – zwraca rozszerzony opis danego skarbu w formacie „nazwa: opis (Wartość: N pkt)”,

**4) Przejscie:**

- znaczenie: reprezentuje standardowe przejście między dwiema komnatami tylko w jednym kierunku,

- atrybuty:

- kierunek (str) – „północ”, „południe”, „wschód” lub „zachód”; oznacza, w jakim kierunku prowadzi dane przejście,

- komnata\_docelowa (Komnata) – obiekt komnaty, do której prowadzi dane przejście,

- metody:

- \_\_init\_\_(self, kierunek, komnata\_docelowa) – inicjalizuje obiekt przejścia; zapisuje kierunek oraz komnatę docelową,

- przejdz(self) – zwraca komnatę docelową – służy do zmiany lokalizacji gracza lub stworzenia,

**5) Drzwi:**

- znaczenie: podklasa Przejscie; reprezentuje dwukierunkowe domyślnie zamknięte przejście między komnatami, które wymaga klucza,

- atrybuty:

- kierunek (str) – dziedziczone z Przejscie; „północ”, „południe”, „wschód” lub „zachód”; reprezentuje pierwszy kierunek, w który prowadzą dane drzwi,

- komnata\_docelowa (Komnata) – dziedziczone z Przejscie; obiekt pierwszej komnaty, którą łączą dane drzwi (przejście przez drzwi do tej komnaty wykonuje się w kierunku *kierunek*),

- kierunek2 (str) - „północ”, „południe”, „wschód” lub „zachód”; reprezentuje drugi kierunek, w który prowadzą dane drzwi,

- komnata\_docelowa2 (Komnata) - obiekt drugiej komnaty, którą łączą dane drzwi (przejście przez drzwi do tej komnaty wykonuje się w kierunku *kierunek2*),

- pasujacy\_klucz (Klucz) – jedyny klucz, który może otworzyć dane drzwi,

- otwarte (bool) – przyjmuje wartość True, jeśli drzwi są otwarte oraz False, jeśli są zamknięte (domyślnie: False),

- metody:

- \_\_init\_\_(self, kierunek, komnata\_docelowa, pasujacy\_klucz, kierunek2, komnata\_docelowa2) – inicjalizuje obiekt drzwi jako przejście dwukierunkowe; wskazuje dwa kierunki, dwie komnaty i pasujący klucz,

- otworz(self, gracz) – jeśli gracz ma odpowiedni klucz w swoim inwentarzu, otwiera drzwi, ustawiając otwarte = True,

- zamknij (self) – zamyka drzwi, ustawiając otwarte = False,

- czy\_otwarte(self) – zwraca True lub False, w zależności od stanu drzwi (w atrybucie *otwarte*),

- przejdz(self, obecna\_komnata) – sprawdza, z której komnaty gracz przechodzi i zwraca drugą stronę (drugą komnatę), jeśli drzwi są otwarte,

**6) PrzejscieNaPunkty:**

- znaczenie: podklasa Przejscie; magiczne jednokierunkowe przejście, przez które gracz może przejść jedynie wtedy, gdy osiągnie określoną liczbę punktów; przejście to jest niedostępne dla obiektów klasy Stworzenie,

- atrybuty:

- kierunek (str) – dziedziczone z Przejscie; „północ”, „południe”, „wschód” lub „zachód”; oznacza, w jakim kierunku prowadzi dane przejście,

- komnata\_docelowa (Komnata) – dziedziczone z Przejscie; obiekt komnaty, do której prowadzi dane przejście,

- prog\_punktow (int) – minimalna liczba punktów potrzebna graczowi, by przejść przez dane przejście,

- metody:

- \_\_init\_\_(self, kierunek, komnata\_docelowa, prog\_punktow) – inicjalizuje obiekt specjalnego przejścia, które wymaga osiągnięcia określonej liczby punktów przez gracza,

- przejdz(self, gracz) – zwraca komnatę docelową, jeśli gracz spełnia warunek punktowy – służy do zmiany lokalizacji gracza,

- czy\_przejscie(self, gracz) – zwraca True, jeśli gracz osiągnął przynajmniej liczbę punktów z *prog\_punktow* oraz False w przeciwnym przypadku,

**7) Komnata:**

- znaczenie: pomieszczenie, w którym mogą znajdować się przedmioty, przejścia i stworzenia; komnaty - połączone poprzez obiekty typu Przejscie - tworzą graf przestrzenny, po którym gracz oraz stworzenia mogą się przemieszczać,

- atrybuty:

- nazwa (str) – nazwa danej komnaty,

- opis (str) – opis tekstowy danej komnaty,

- przedmioty (list) – lista przedmiotów znajdujących się aktualnie w danej komnacie,

- przejscia (list) – lista przejść znajdujących się w danej komnacie,

- stworzenia (list) – lista stworzeń znajdujących się aktualnie w danej komnacie,

- metody:

- \_\_init\_\_(self, nazwa, opis) – inicjalizuje obiekt komnaty z jej nazwą, opisem i pustymi listami przechowującymi przedmioty, przejścia i stworzenia,

- dodaj\_przedmiot(self, przedmiot) – dodaje przedmiot do danej komnaty,

- usun\_przedmiot(self, nazwa) – usuwa i zwraca z danej komnaty przedmiot o danej nazwie,

- opisz(self, gracz) – generuje tekstowy opis komnaty oraz znajdujących się w niej przedmiotów, stworzeń i przejść – z uwzględnieniem stanu drzwi i punktów gracza,

**8) Stworzenie:**

- znaczenie: przeciwnik gracza – złodziej, który porusza się losowo między komnatami i który z ustalonym prawdopodobieństwem może ukraść graczowi losowy skarb z jego inwentarza,

- atrybuty:

- komnata\_aktualna (Komnata) – komnata, w której aktualnie znajduje się dane stworzenie,

- metody:

- \_\_init\_\_(self, komnata\_poczatkowa) – inicjalizuje obiekt stworzenia; ustawia stworzenie (złodzieja) w komnacie początkowej,

- ruch\_losowy(self) – odzwierciedla losowy ruch danego stworzenia między komnatami, przy czym: stworzenie ma 30% szans na pozostanie w miejscu; stworzenie wybiera losowo dostępne z jego aktualnej komnaty przejście (za wyjątkiem drzwi, które są zamknięte oraz obiektów klasy PrzejscieNaPunkty, przez które mogą przechodzić wyłącznie gracze),

- spotkaj\_gracza(gracz, wszystkie\_komnaty) – metoda statyczna; odzwierciedla spotkanie danego stworzenia z graczem, przy czym: istnieje 20% szans, że stworzenie ukradnie graczowi skarb; jeśli stworzenie ukradnie graczowi skarb, skarb ten zostaje automatycznie przeniesiony do losowej innej komnaty (za wyjątkiem aktualnej komnaty gracza oraz Świątyni Skarbów, gdzie znajduje się główny skarb),

**9) Gracz:**

- znaczenie: reprezentuje gracza – jego lokalizację, ekwipunek, ruchy i interakcje,

- atrybuty:

- aktualna\_komnata (Komnata) – komnata, w której aktualnie znajduje się gracz,

- inwentarz (list) – lista posiadanych aktualnie przez gracza przedmiotów,

- metody:

- \_\_init\_\_(self, komnata\_poczatkowa) – tworzy obiekt gracza, ustawiając jego startową lokalizację i pusty inwentarz,

- przejdz(self, kierunek) – przechodzi w danym kierunku, jeśli jest to dla danego gracza możliwe (w przypadku obiektu Drzwi – jeśli są otwarte, w przypadku obiektu PrzejscieNaPunkty – jeśli gracz ma wystarczającą liczbę punktów),

- otworz\_drzwi(self, kierunek) – wywołuje metodę *otworz* na obiekcie Drzwi w danym kierunku, jeśli takie drzwi znajdują się w danej komnacie; otwiera dane drzwi, jeśli gracz posiada w inwentarzu odpowiedni klucz,

- rozejrzyj\_sie(self) – wywołuje metodę *opisz* na obiekcie Komnata pamiętanym w atrybucie *aktualna\_komnata*; wyświetla tekstowy opis aktualnej komnaty gracza wraz z występującymi w niej przedmiotami, przejściami i stworzeniami,

- sprawdz(self, nazwa) – wywołuje metodę *opisz* na przedmiocie o podanej nazwie, jeśli taki przedmiot znajduje się w aktualnej komnacie gracza lub w jego inwentarzu; pokazuje tekstowy opis przedmiotu z aktualnej komnaty gracza lub z jego inwentarza,

- wez(self, nazwa) – usuwa przedmiot o podanej nazwie z aktualnej komnaty gracza oraz umieszcza go w jego inwentarzu, jeśli taki przedmiot się tam znajduje,

- poloz(self, nazwa) – usuwa przedmiot o podanej nazwie z inwentarza gracza oraz umieszcza go w aktualnej komnacie gracza, jeśli taki przedmiot się tam znajduje,

- liczba\_punktow(self) - zwraca sumę punktów gracza za posiadane przez niego skarby,

- pokaz\_inwentarz(self) – wypisuje listę wszystkich przedmiotów gracza, które znajdują się w jego inwentarzu,

**10) Mapa:**

- znaczenie: buduje całą strukturę świata gry – komnaty, przejścia, klucze, stworzenia i gracza,

- atrybuty:

- komnaty (dict) – słownik, którego wartościami są utworzone komnaty w grze,

- stworzenia (list) – lista utworzonych stworzeń w świecie gry,

- gracz (Gracz) – obiekt utworzonego gracza,

- metody:

- \_\_init\_\_(self) – tworzy pustą strukturę świata gry – bez komnat, gracza ani stworzeń,

- zbuduj(self) – tworzy i dodaje wszystkie komnaty, klucze, skarby i przejścia oraz tworzy i ustawia gracza i stworzenia,

**11) Rozgrywka:**

- znaczenie: zarządza całą logiką gry – przyjmuje i wykonuje polecenia, przetwarza ruchy, wyświetla mapę oraz rozpoczyna i kończy grę,

- atrybuty:

- mapa (Mapa) – instancja świata gry; zawiera cały świat gry, w tym: gracza, komnaty i stworzenia,

- metody:

- \_\_init\_\_(self) – tworzy obiekt rozgrywki, inicjalizując i budując świat gry na podstawie klasy Mapa,

- wykonaj\_polecenie(self, wejscie) – interpretuje i wykonuje komendę gracza; dzieli tekst komendy na dwie części: polecenie i argument; na początku wywoływana jest metoda *ruch\_stworzen* – stworzenia wykonują swój ruch przed każdą turą gracza; następnie, program sprawdza komendę, przy czym możliwe są następujące polecenia: „rozejrzyj” – pokazuje opis aktualnej komnaty gracza, „sprawdź [przedmiot]” – pokazuje opis przedmiotu w aktualnej komnacie gracza lub w jego inwentarzu, „weź [przedmiot]” – przenosi przedmiot z danej komnaty do inwentarza gracza, „połóż [przedmiot]” – odkłada przedmiot z inwentarza gracza do danej komnaty, „przejdź [kierunek]” – próbuje przejść w podanym kierunku – jeśli się uda, wyświetla nową mapę, „otwórz [kierunek]” – próbuje otworzyć drzwi w podanym kierunku, „inwentarz” – wyświetla zawartość inwentarza gracza, „punkty” – wyświetla liczbę punktów zdobytych przez gracza, „koniec” – kończy grę; następnie, jeśli gra jeszcze się nie skończyła, na każdym stworzeniu znajdującym się w tej samej komnacie, co gracz, wywoływana jest metoda *spotkaj\_gracza* – stworzenie ma możliwość kradzieży skarbu z inwentarza gracza,

- ruch\_stworzen(self) – wywołuje metodę *ruch\_losowy* na każdym stworzeniu na mapie – stworzenia z ustalonym prawdopodobieństwem przemieszczają się między dostępnymi im komnatami,

- uruchom(self) – wyświetla ekran powitalny i dostępne dla gracza komendy oraz wyświetla mapę, a następnie w nieskończonej pętli (która przerywa się, gdy gracz wpisze komendę „koniec”) pobiera i przetwarza komendę od gracza; po zakończeniu wypisuje podsumowanie gry,

- wyswietl\_mape(self) – rysuje statyczną mapę gracza z zaznaczoną aktualną pozycją gracza – jest to format graficzny z odstępami i łączeniami między komnatami.