

Dokumentacja Projektowa

system wspierający organizację KOMNATOWYCH WYDARZEŃ LARP

Uczestnicy LARP fizycznie przedstawiają POSTACIE w fikcyjnej scenerii. GRACZE przyjmują rolę POSTACI GRACZY (PC), które mogą sami stworzyć lub otrzymać od MISTRZÓW GRY (GM). Uczestnicy nazywani załogą, pomagają w przygotowaniu ŚRODOWISKA pełniąc rolę POMOCNIKÓW scenicznych, lub grając POSTACIE NIEZALEŻNE (NPC).

Autorzy: Piotr Heinzelman alb.146703
Status: przygotowanie
Wersja: 1.0
Data modyfikacji: 7.I.2024
Abstrakt: Dokument zawiera dokumentację metodyki, wymagań oraz projekt oprogramowania dla systemu wspierającego organizację komnatowych wydarzeń LARP

Historia zmian:

Wersja	Data	Kto	Opis
0.1	7.01.2024	Piotr Heinzelman	Utworzenie dokumentu.
	11.01.2024	Piotr Heinzelman	dodanie rejestru produktowego oraz scenariuszy przypadków użycia

1. Plan zadań w projekcie

1.1 Rejestr produktowy

Sprint 1	
2.1	Zarządzanie wydarzeniami LARP
2.5	Tworzenie profili odgrywanych postaci, akcji, zasobów i przedmiotów oraz udostępnianie informacji o nich ich graczom.
2.10	Zarządzanie przebiegiem gry
Sprint 2	
2.6	Zarządzanie terenem gry (pomieszczeniami), w tym – definiowanie znaczenia czujników i kodów QR
2.9	Zlecanie i rozstrzyganie efektów akcji w świecie gry
2.2	Zarządzanie zapisami na wydarzenie
Sprint 3	
2.7	Monitorowanie zasobów i przedmiotów posiadanych przez graczy.
2.8	Interakcja z otoczeniem za pomocą kodów QR lub odpowiednich czujników.
Sprint 4	
2.9	Zlecanie i rozstrzyganie efektów akcji w świecie gry
Sprint 5	
2.10	Zarządzanie przebiegiem gry
Sprint 6	
2.3	Obsługa płatności
2.4	Obsługa płatności BRAMKA

1.2 Rejestr sprintu 1

Tabela z zadaniami polegającymi na realizacji przypadków użycia wybranych do sprintu
Krótkie opisy poszczególnych zadań

Sprint 1 podstawowy szkielet aplikacji	
2.1	Zarządzanie wydarzeniami LARP Tworzenie nowych, edycja, zapis wydarzeń. Ogólny zarys interfejsu.
2.5	Tworzenie profili odgrywanych postaci, akcji, zasobów i przedmiotów oraz udostępnianie informacji o nich ich graczom. Przygotowanie ogólnego interfejsu tworzenia/edycji zapisu. Ogólny koncept interfejsu graficznego. Przygotowanie formatek listy i edycji postaci/akcji/zasobów i przedmiotów
2.10	Zarządzanie przebiegiem gry Wstępna wersja interfejsu edycji elementów systemu, wydarzenia, lokalizacji, postaci, zasobów i przedmiotów
Sprint 2 podstawowy szkielet aplikacji cz. 2	
2.6	Zarządzanie terenem gry (pomieszczeniami), w tym – definiowanie znaczenia czujników i kodów QR
2.9	Zlecanie i rozstrzyganie efektów akcji w świecie gry
2.2	Zarządzanie zapisami na wydarzenie

2. Opis wymagań funkcjonalnych systemu

- Zarządzanie wydarzeniami LARP – tworzenie wydarzeń ich opisów.
- Zarządzanie zapisami na wydarzenia LARP oraz selekcją uczestników.
- Obsługa płatności we współpracy z zewnętrznym systemem obsługi płatności.
- Tworzenie profili odgrywanych postaci, akcji, zasobów i przedmiotów oraz udostępnianie informacji o nich ich graczom.
- Monitorowanie zasobów i przedmiotów posiadanych przez graczy.
- Interakcja z otoczeniem za pomocą kodów QR lub odpowiednich czujników.
- Monitorowanie obecności uczestników na poszczególnych obszarach gry.
- Zarządzanie terenem gry (pomieszczeniami), w tym – definiowanie znaczenia czujników i kodów QR.
- Zlecanie i rozstrzyganie efektów akcji w świecie gry.
- Zarządzanie przebiegiem gry, w tym – obsługa historii wydarzeń.

GRACZ - osoba ze świata realnego.

<-- GUEST osoba bez przyznanej roli (niezalogowana)

<-- rola UŻYTKOWNIK - gracz o zwykłych prawach

<-- rola PISARZ - rola uprawniająca do zarządzania HISTORIĄ - czyli poszczególnymi scenariuszami rozgrywki (POSTACIE, AKCJE, ZASOBY, PRZEDMIOTY)

< -- rola ZARZĄDCA - rola uprawniająca to zarządzania:

• uprawnieniami w systemie,

• LOKALIZACJAMI (geograficzne, np: zamek Saski, Grudziądz)

• WYDARZENIAMI (czyli HISTORIA w LOKALIZACJI)

• OPŁATAMI

POSTAĆ - byt ogólny ze świata gry.

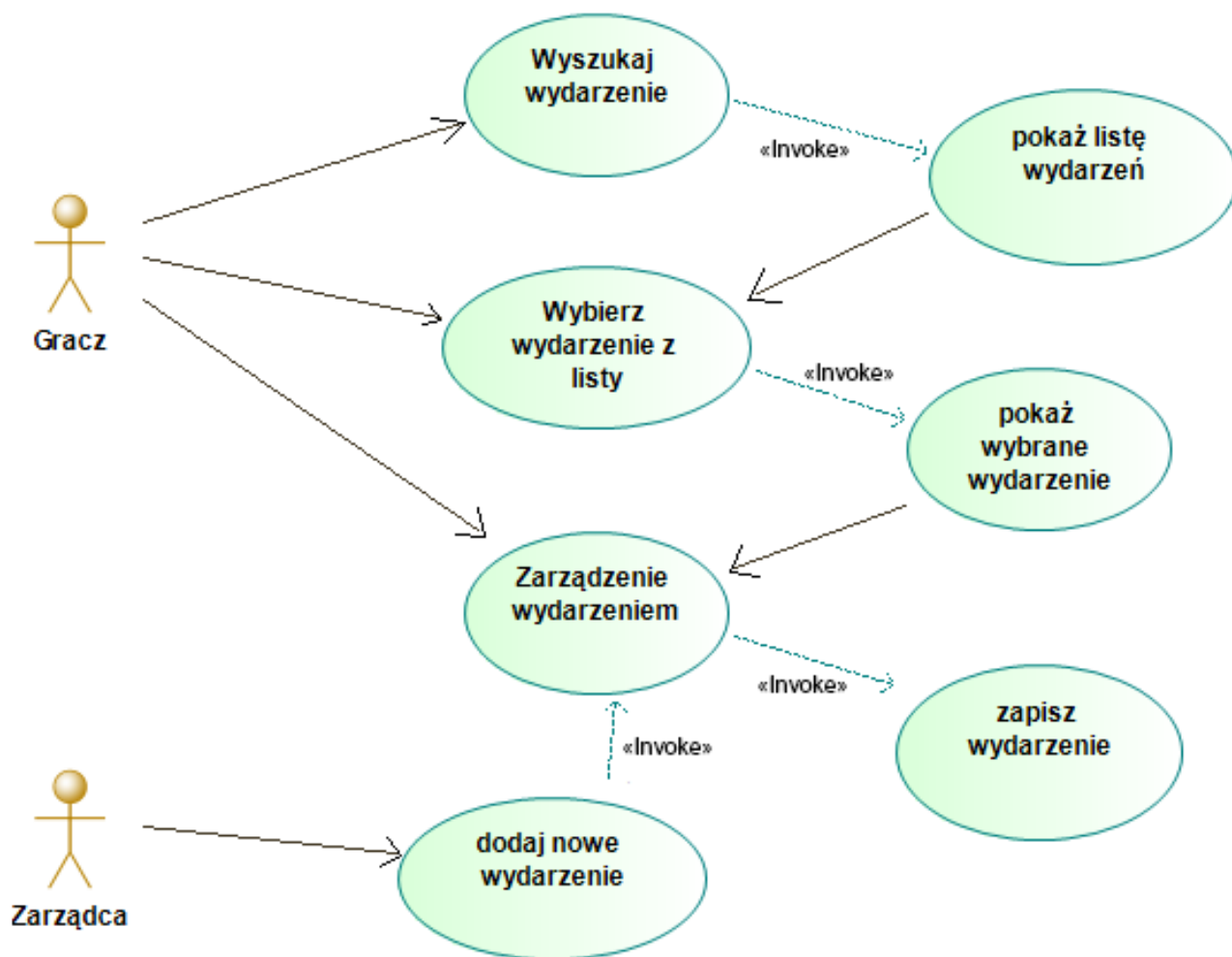
<-- PC - zwykły byt świata gry w który wciela się GRACZ, posiada statystyki (akcje, zasoby i położenie w teatrze działań).

<-- NPC - byt pomocniczy np. sprzedawca - sam nie angażuje się w rozgrywkę, pomaga PC w grze, (posiada położenie, zasoby i dozwolone akcje)

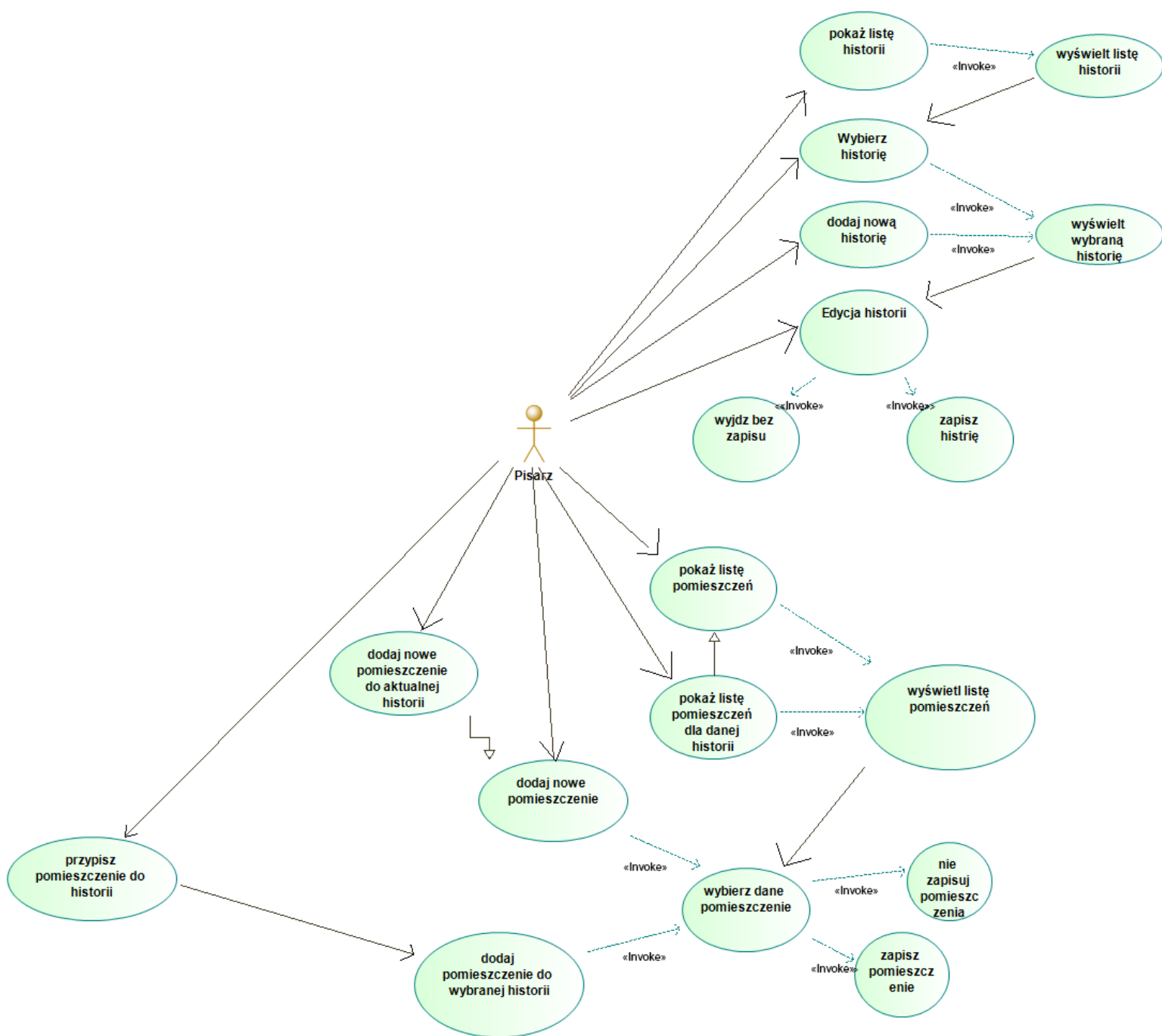
<-- GM - Mistrz gry - byt nadzorujący i rozstrzygający w grze. Posiada położenie w teatrze działań, dozwolone wszystkie akcje, w tym akcje specjalne.

2.1 Diagramy przypadków użycia

PU1: Zarządzanie wydarzeniami LARP



PU6: Zarządzanie terenem gry



2.2 Scenariusze przypadków użycia

PU1: Zarządzanie wydarzeniami LARP

{ GRACZ }

- zarządzanie WYDARZENIEM (sortowanie : wg. daty, lokalizacji, opłat, ilości wolnych miejsc)
- - - > invoke (pokaż listę wydarzeń)
 - JEŚLI { ZARZĄDCA }
 - - - > invoke (dodaj nowe wydarzenie)
- (wybierz wydarzenie)
- - - > invoke (pokaż informacje o wydarzeniu)
 - JEŚLI { ZARZĄDCA lub PISARZ }
 - - - > invoke (edytuj informacje o wydarzeniu)
 - - - > invoke (przypisz wydarzenie do lokalizacji)
 - - - > invoke (ustal datę rozpoczęcia wydarzenia)

PU2: Zarządzanie zapisami na wydarzenie

{ GRACZ }

- zapis na WYDARZENIE (sortowanie : - z możliwością zapisu)
- - - > invoke (pokaż listę wydarzeń)
- (wybierz wydarzenie)
 - JEŚLI { niezalogowany }
 - - - > invoke (zaloguj do systemu)
- JEŚLI { zalogowany }
- - - > invoke (zgłoś na wydarzenie jako GRACZ)
- JEŚLI { opłata !=0 && rola== UŻYTKOWNIK }
- - - > invoke (opłać: WYDARZENIE, GRACZ, RODZAJ PŁATNOŚCI)
 - JEŚLI { ZARZĄDCA }
 - - - > invoke (pokaż listę zgłoszonych użytkowników to wydarzenia)
 - (wybierz wpis)
 - - - > invoke (edytuj wpis : USUŃ WPIS = wyproś gracza z wydarzenia,
USUŃ OPŁATĘ = gracz gra za darmo , PRZYZNAJ RABAT = gracz przystępuje do gry taniej)

PU3: Obsługa płatności

{ UŻYTKOWNIK }

- zarządzanie MOIMI PŁATNOŚCIAMI
- - - > invoke (pokaż listę moich wydarzeń)
- (wybierz wydarzenie)
- - - > invoke (pokaż informację o wydarzeniu)
 - JEŚLI { STATUS PŁATNOŚCI == nieopłacona }
 - - - > invoke (opłać: WYDARZENIE, GRACZ, RODZAJ PŁATNOŚCI)

PU4: Obsługa płatności BRAMKA

{ SYSTEM }

- (opłać : WYDARZENIE, GRACZ, RODZAJ PŁATNOŚCI)
- (wybór interfejsu płatności : RODZAJ PŁATNOŚCI)
- - - > invoke (RODZAJ PŁATNOŚCI : opłać (WYDARZENIE, GRACZ))
- JEŚLI { BŁĄD PŁATNOŚCI }
- (zwróć informację o błędzie)
- JEŚLI { SUCCESS }
- - - > invoke (zapisz (uaktualnij) w systemie STATUS PŁATNOŚCI)
- (zwróć informację o płatności)
- - - > invoke (opłać: WYDARZENIE, GRACZ)

PU5: Tworzenie profili odgrywanych postaci, akcji, zasobów i przedmiotów oraz udostępnianie informacji o nich ich graczom.

```
{ PISARZ , ZARZĄDCA }  
- ( zarządzanie POSTACIE, AKCJE, ZASOBY, PRZEDMIOTY )  
- ( pobierz listę POSTACIE, AKCJE, ZASOBY, PRZEDMIOTY )  
- - - > invoke ( wybierz z listy POSTACIE, AKCJE, ZASOBY, PRZEDMIOTY )  
- - - > invoke ( edytuj POSTACIE, AKCJE, ZASOBY, PRZEDMIOTY )  
- - - > invoke ( zapisz POSTACIE, AKCJE, ZASOBY, PRZEDMIOTY )
```

PU6: Zarządzanie terenem gry (pomieszczeniami), w tym – definiowanie znaczenia czujników i kodów QR

```
{ PISARZ , ZARZĄDCA }  
- - - > invoke ( pokaż listę HISTORII (+filtr))  
{ SYSTEM }  
- ( pokaż listę HISTORII (uwzględniając filtr ) )  
{ PISARZ , ZARZĄDCA }  
- - - > invoke ( wybierz HISTORIĘ z listy )  
{ SYSTEM }  
- ( pokaż informacje o HISTORII )  
{ PISARZ , ZARZĄDCA }  
- - - > invoke ( dodaj nowe POMIESZCZENIE )  
{ SYSTEM }  
- ( zapisz POMIESZCZENIE do HISTORIA )  
- ( pokaż formatkę do edycji POMIESZCZENIA )  
{ PISARZ , ZARZĄDCA }  
- - - > invoke ( edytuj opis do POMIESZCZENIE )  
{ SYSTEM }  
- ( zapisz opis dla POMIESZCZENIE )  
{ PISARZ , ZARZĄDCA }  
- - - > invoke ( dodaj AKCJA do POMIESZCZENIE )  
{ SYSTEM }  
- (zapisz AKCJA do POMIESZCZENIE )  
{ PISARZ , ZARZĄDCA }  
- - - > invoke ( wyszukaj z listy QR-code lub czujnik )  
{ SYSTEM }  
- ( pokaż listę QR / czujników )  
{ PISARZ , ZARZĄDCA }  
- - - > invoke ( wybierz czujnik, przypisz do AKCJA )  
- - - > invoke ( dodaj opis do AKCJA )  
- - - > invoke ( dodaj warunki i parametry do AKCJA )  
{ SYSTEM }  
- ( zapisz przypisanie czujnika do AKCJA )
```

PU 7: Monitorowanie zasobów i przedmiotów posiadanych przez graczy.

```
{ GRACZ }  
- ( sprawdź stan rozgrywki )  
- - - > invoke ( moja postać )  
{ SYSTEM }  
- ( pokaż informacje o postaci - POŁOŻENIE, STAN OGÓLNY )  
- - - > invoke ( wybierz listę MOJE AKCJE, MOJE ZASOBY, MOJE PRZEDMIOTY, STAN WYDARZENIA )  
{ SYSTEM }  
- ( pokaż listę AKCJE | ZASOBY | PRZEDMIOTY | STAN WYDARZENIA )  
{ GRACZ }  
- - - > invoke ( wybierz z listy element AKCJE | ZASOBY | PRZEDMIOTY | STAN WYDARZENIA )  
{ SYSTEM }  
- ( pokaż informacje o wybranym elemencie )  
{ GRACZ }  
- - - > invoke ( powrót do listy )  
lub  
- - - > invoke ( powrót do informacji o postaci )  
lub  
- - - > invoke ( powrót do menu głównego )
```

PU 8: Interakcja z otoczeniem za pomocą kodów QR lub odpowiednich czujników.

```
{ CZUJNIK } -- czujnik  
- ( czujnik rejestruje obecność GRACZA w LOKALIZACJI )  
- - - > invoke ( wywołaj akcję SYSTEMU )  
{ SYSTEM }  
- - - > invoke ( aktualizuj pozycję GRACZA )  
- - - > invoke ( wyszukaj AKCJA dla LOKALIZACJI )  
- - - > invoke ( wykonaj AKCJA dla GRACZA )  
  
lub  
{ GRACZ }  
- ( odczytuje kod QR - uruchamia kod przypisany do kodu QR )  
- - - > invoke ( uruchom AKCJA dla GRACZ )  
{ SYSTEM }  
- - - > invoke ( wykonaj AKCJA dla GRACZA )
```

PU 9: Zlecanie i rozstrzyganie efektów akcji w świecie gry

```
{ CZUJNIK : rejestruje obecność GRACZA w LOKALIZACJI lub GM }  
- ( wywołaj akcję systemu )  
{ SYSTEM }  
- ( aktualizuj stan gry )  
- ( aktualizuj stan gracza )  
JEŚLI  
potrzebna decyzja GM lub GRACZ  
- ( pokaż formatkę z prośbą o dokonanie wyboru na urządzeniu GRACZA / GM )  
{ GM, GRACZ }  
- - - > invoke ( wybrana opcja )
```

PU 10: Zarządzanie przebiegiem gry

{ GM, GRACZ }

- (sprawdź stan gry)

- - - > invoke (sprawdź aktualny stan gry)

{ SYSTEM }

- (wyświetl aktualny stan gry dla GRACZ)

jeśli {GM}

- (pokaż listę graczy)

{ GM }

- - - > invoke (wybierz GRACZA / GRUPY GRACZY)

{ SYSTEM }

- (pokaż informacje o stanie GRY oraz stanie GRACZA / GRUPY GRACZY)

{ GM }

- - - > invoke { powrót do poprzedniego widoku }

lub

- - - > invoke { wyjście do menu głównego }

3. Słownik dziedziny problemu

- Zarządzanie wydarzeniami LARP – tworzenie wydarzeń ich opisów.
- Zarządzanie zapisami na wydarzenia LARP oraz selekcją uczestników.
- Obsługa płatności we współpracy z zewnętrznym systemem obsługi płatności.
- Tworzenie profili odgrywanych postaci, akcji, zasobów i przedmiotów oraz udostępnianie informacji o nich ich graczom.
- Monitorowanie zasobów i przedmiotów posiadanych przez graczy.
- Interakcja z otoczeniem za pomocą kodów QR lub odpowiednich czujników.
- Monitorowanie obecności uczestników na poszczególnych obszarach gry.
- Zarządzanie terenem gry (pomieszczeniami), w tym – definiowanie znaczenia czujników i kodów QR.
- Zlecanie i rozstrzyganie efektów akcji w świecie gry.
- Zarządzanie przebiegiem gry, w tym – obsługa historii wydarzeń.

GRACZ - osoba ze świata realnego.

<-- GUEST osoba bez przyznanej roli (niezalogowana)

<-- rola UŻYTKOWNIK - gracz o zwykłych prawach

<-- rola PISARZ - rola uprawniająca do zarządzania HISTORIĄ - czyli poszczególnymi scenariuszami rozgrywki (POSTACIE, AKCJE, ZASOBY, PRZEDMIOTY)

< -- rola ZARZĄDCA - rola uprawniająca to zarządzania:

- uprawnieniami w systemie,
- LOKALIZACJAMI (geograficzne, np: zamek Saski, Grudziądz)
- WYDARZENIAMI (czyli HISTORIA w LOKALIZACJI)
- OPŁATAMI

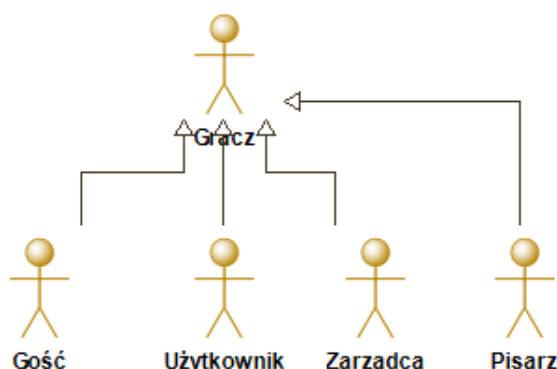
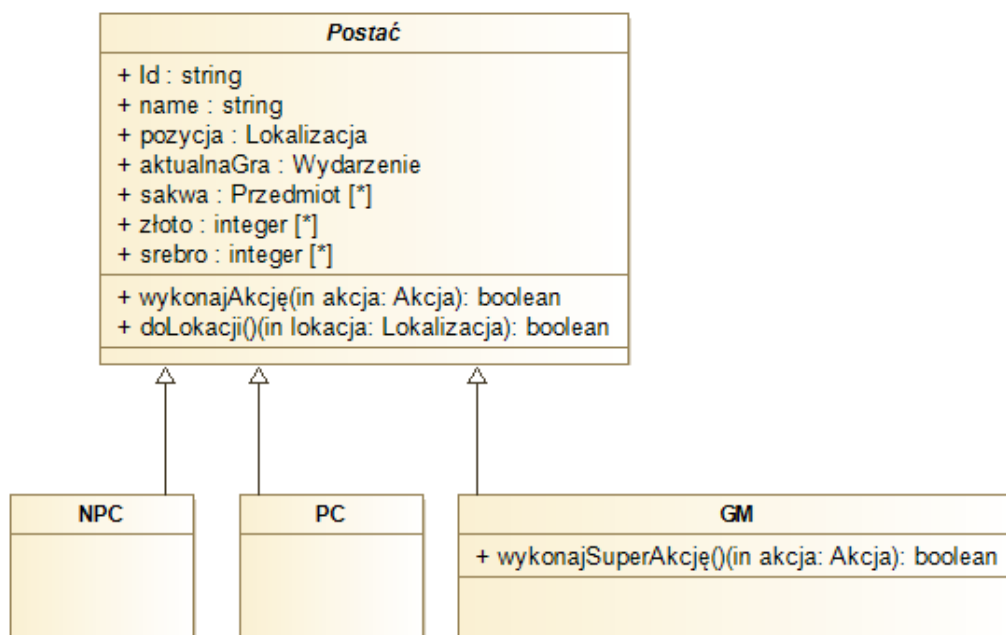
POSTAĆ - byt ogólny ze świata gry.

<-- PC - zwykły byt świata gry w który wciela się GRACZ, posiada statystyki (akcje, zasoby i położenie w teatrze działań).

<-- NPC - byt pomocniczy np. sprzedawca - sam nie angażuje się w rozgrywkę, pomaga PC w grze, (posiada położenie, zasoby i dozwolone akcje)

<-- GM - Mistrz gry - byt nadzorujący i rozstrzygający w grze. Posiada położenie w teatrze działań, dozwolone wszystkie akcje, w tym akcje specjalne.

Rysunek 3 : Diagram klas słownika dla obszaru dziedziny 1



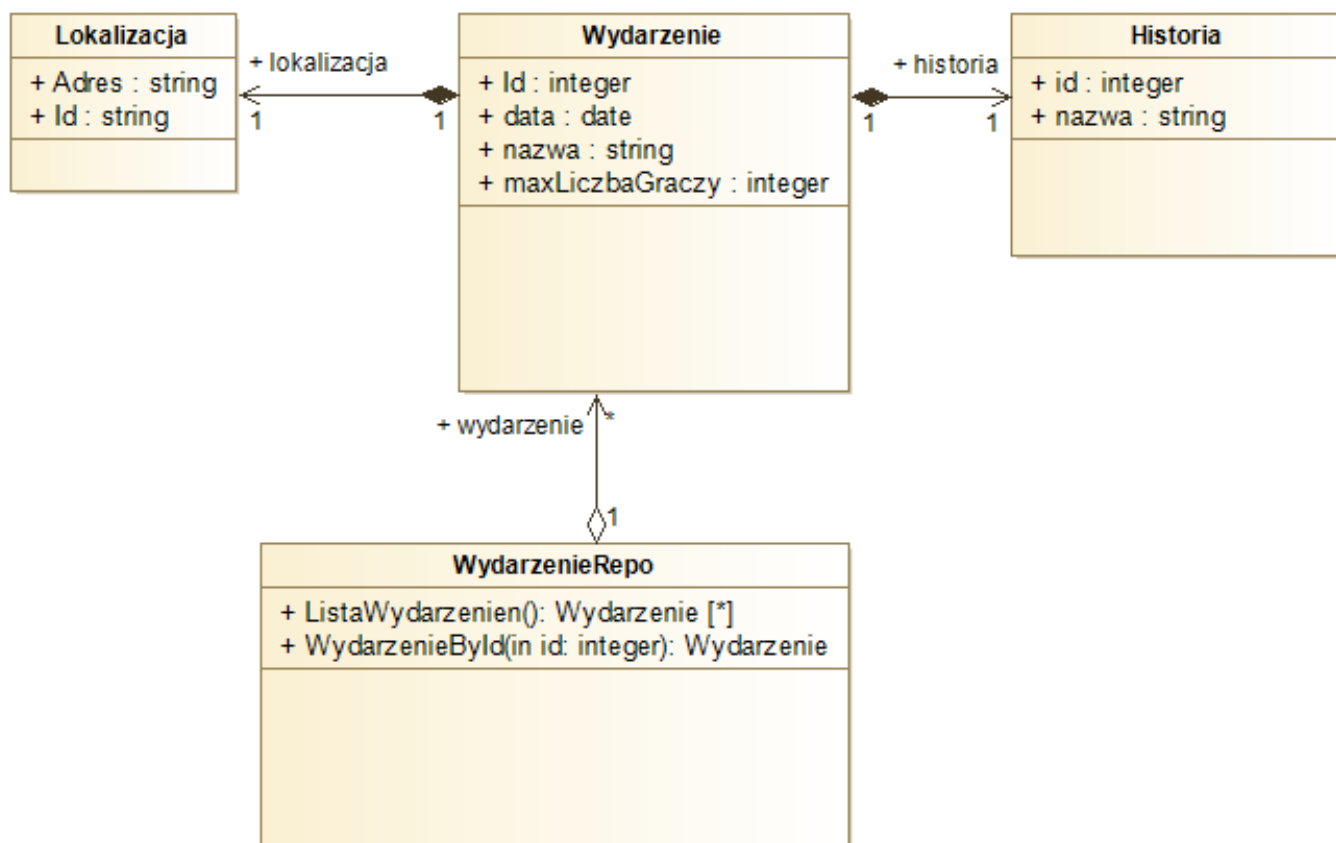
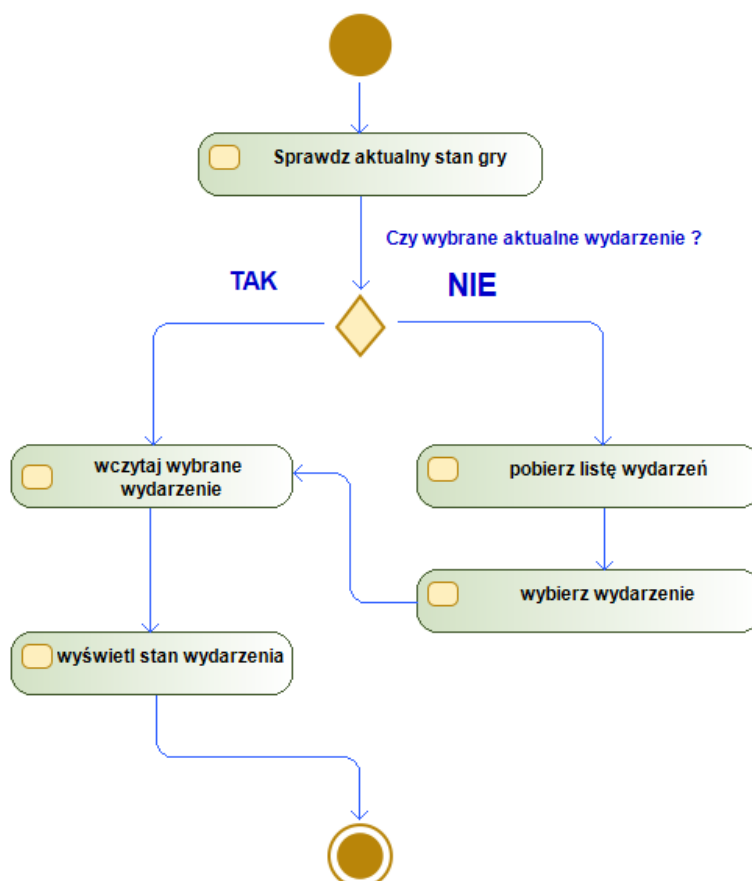


Diagram klas Wydarzenie, Historia, Lokalizacja i WydarzenieRepo

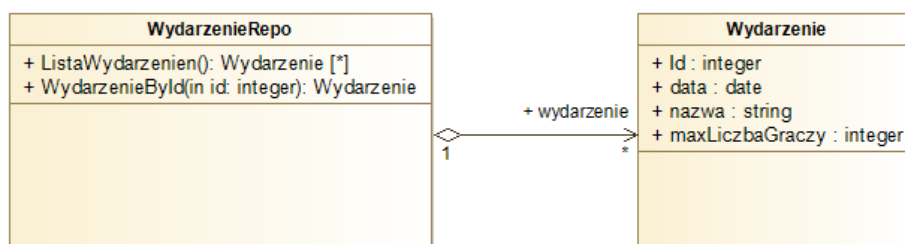
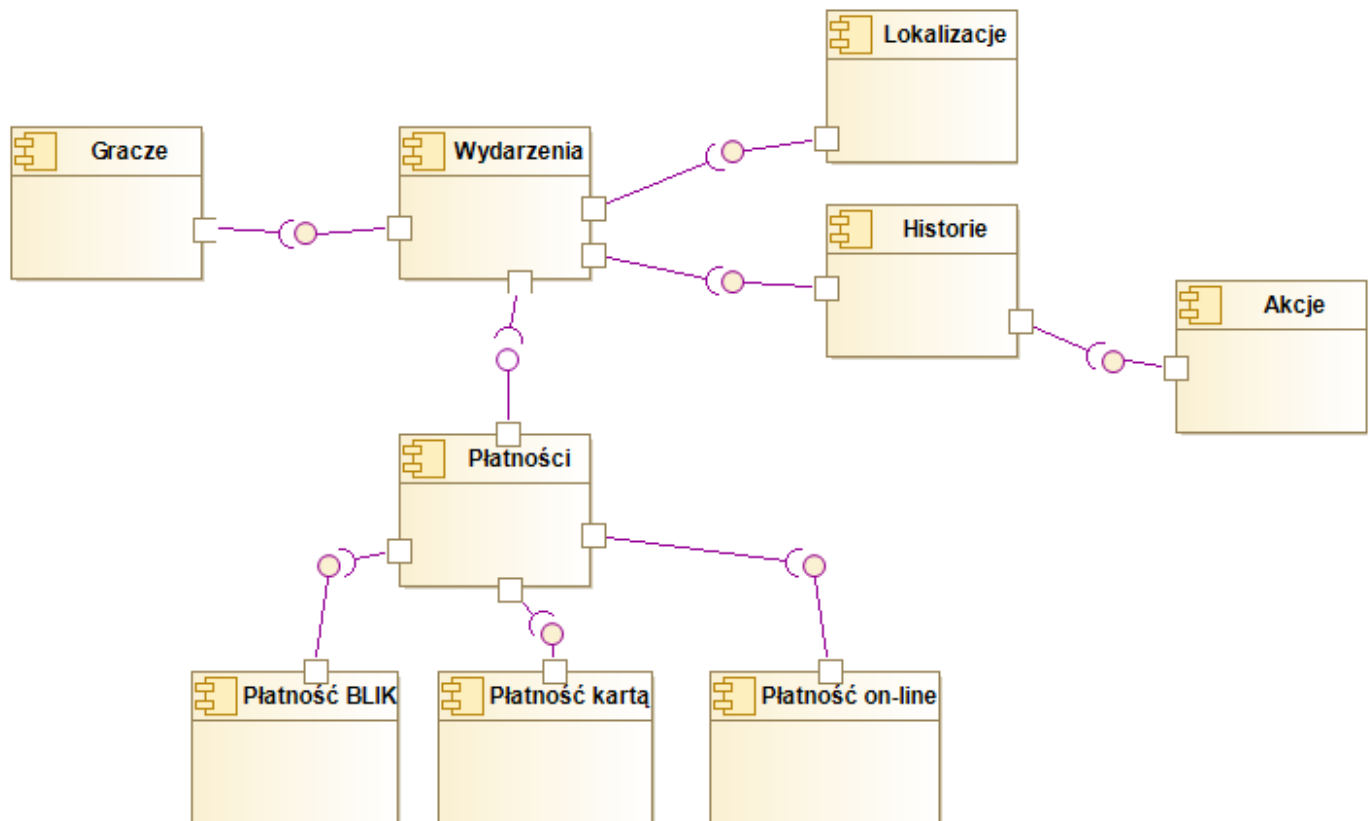


Model czynności PU 10: Sprawdzenie stanu gry

4. Projekt struktury systemu

Rysunek 4 : Diagram system wspierający organizację komnatowych wydarzeń LARP

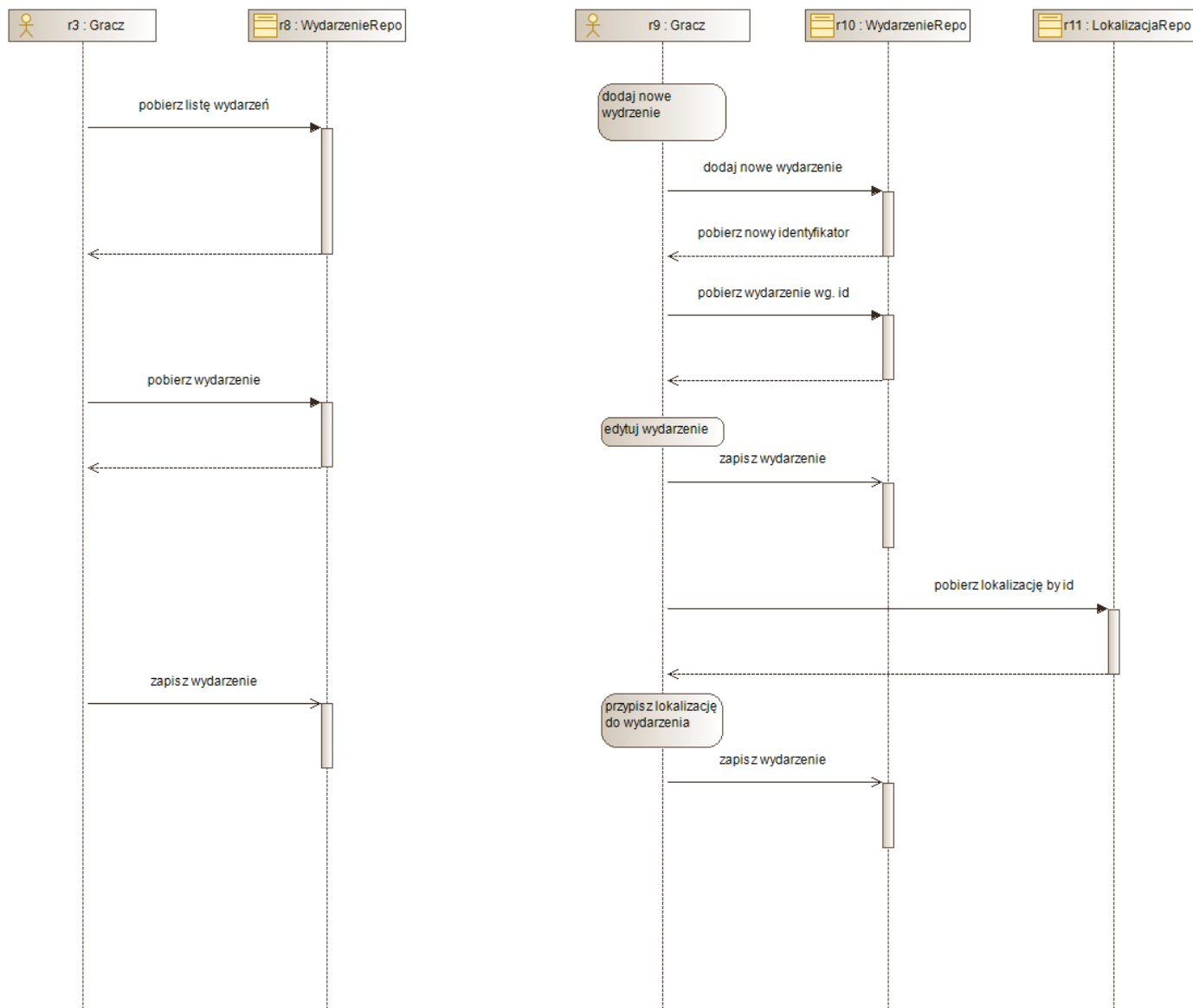
Centrum systemu stanowi moduł “Gracze” i “Historie” oraz “Akcje” tam odbywa się komunikacja główna z użytkownikiem i prowadzona jest cała akcja gry. Moduły “Lokalizacje”, “Wydarzenia” i “Płatności” używane są na raz przed rozpoczęciem gry, Lokalizacje niekiedy przy przeglądaniu gier archiwalnych. Płatności są zewnętrznymi systemami, Moduł Płatności jest właściwie *interfejsem*, lub *adapterem* dla modułów Płatność Blik, Płatność Kartą i Płatność on-line



5. Projekt dynamiki systemu

PU1: Zarządzanie wydarzeniami LARP

Diagramy ukazujące czasowe zależności przy tworzeniu i edycji Wydarzenia.



6. Struktura kodu systemu

6.1 Realizacja interfejsów

Tu wstawić diagram.

Rysunek 6 : Diagram klas przedstawiający realizację interfejsu I_XYZ

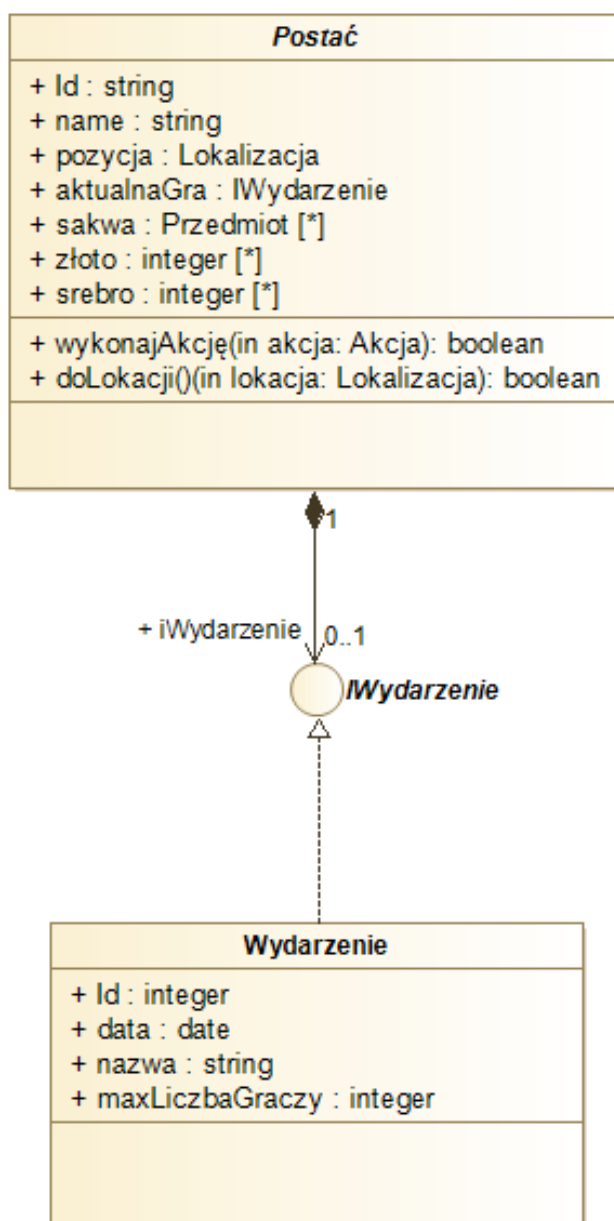
Kod interfejsu I_XYZ oraz klasy go implementującej, w języku programowania ABC, wygenerowany narzędziem AAA_UML.

Kod klasy czcionką Courier (Consolas, Lucida Console, Ubuntu Monospaced)

Kod klasy czcionką Courier

Kod klasy czcionką Courier

Itd...



6.2 Generacja kodu

Opis sposobu generacji kodu w narzędziu AAA_UML. Krótka instrukcja postępowania oraz przykład.

Aby wygenerować kod należy wybrać moduł lub klasę i wybrać -> Generate.

Można edytować wygenerowany kod, wygenerować JavaDOC

The screenshot displays the LARP - Modelio 4.1 application window. The top menu bar includes File, Edit, Configuration, Views, and Help. The left sidebar shows a project tree with folders for LARP, LocalModule, and LARP, and a sub-folder VP. The main workspace is divided into two panes. The left pane shows a UML class diagram with a class named 'Postać' and its associated attributes and methods. The right pane displays the generated Java code for the 'Postać' class, which is a JavaDoc-style class definition with various annotations and methods. A context menu is open over the 'Postać' class in the diagram, showing options such as 'Generate', 'Update model', 'Generate the JavaDoc', 'Visualize the JavaDoc', 'Create a compilation artifact', 'Create a java component', 'Create/Update auto-diagrams', 'Reverse', and 'Configuration'. The 'Generate' option is highlighted. The bottom status bar shows the text 'Type here to search'.

```
import com.modeliosoft.modelio.javadesigner.annotations.objid;

@objid("d9fbee-2a92-41aa-abba-db90f6e633")
public abstract class Postać {
    @objid("13531917-b35a-4ec2-a99d-59f6942586e4")
    public String Id;

    @objid("d9fbee39-0e8e-42d7-98c9-19c1325ee0af")
    public String name;

    @objid("0a1fd74a-385c-4849-bb3c-4872c058fadb")
    public Lokalizacja pozycja;

    @objid("f1d9578-f957-4093-b2c6-87e9695b5e32")
    public Wydarzenie aktualnaGra;

    @objid("1853695b-a625-4687-8cd6-80f61e80fb80")
    public Przedmiot[] sakwa;

    @objid("7525617e-98d2-4a96-988d-73d3cb15bd9d")
    public int[] zloto;

    @objid("0a9fdddf-c10b-4c7f-a651-b71b3300-9c1")
    public int[] srebro;

    @objid("25e5f2cd-cc3e-478e-aa20-39a5b1e6427e")
    public boolean wykonajAkcje(Akcja akcja) {
    }

    @objid("8af90086-1f03-4d00-a2b3-82dfe63ec88")
    public boolean doLokacji(Lokalizacja lokacja) {
    }
}
```

