



Politechnika Koszalińska
Wydział Elektroniki i Informatyki

Sprawozdanie do projektu z przedmiotu:

„Projekt zespołowy”

Semestr VI, Studia Dienne
Kierunek: Informatyka
Rok akademicki 2015/2016

Temat projektu: Aplikacja służąca
zajęciu wolnego czasu odwzorowanej
na grze „Monopoly”

Projekt wykonali:
Dawid Gawiński
Oskar Gruszczyński
Paweł Gawron
Patryk Afelt



Spis treści

1. Protokół założycielski
2. Raport o wybrany instrumentarium zespołu projektowego
3. Raport klienta
4. Harmonogram projektu
5. Raport dla Zleceniodawcy
6. Pełna wersja specyfikacji wymagań zgodna z zaleceniami IEEE
7. Styl firmowy
8. Notatka o narzędziach CASE
9. Diagramy
10. Wybór narzędzi RAD
11. Dokument detaliczny projektu
12. Podsumowanie projektu



1. Protokół założycielski

1. Dnia 24.02.2016 została utworzona grupa projektowa w celem wykonania zadania programistycznego „University Monopoly”.
2. Przewidywana data zakończenia projektu: 08.06.2016r. Jeśli projekt nie zostanie zakończony w pierwszym terminie to przewidywany jest drugi termin zakończenia projektu: 15.09.2016r. Zespół zostanie rozwiązany po zakończeniu projektu w odpowiednim terminie.

3. W skład grupy wchodzi:

Paweł Gawron
Oskar Gruszczyński
Dawid Gawiński
Patryk Afelt

4. Tytuł projektu:

Gra desktopowa „University Monopoly”.
Aplikacja przeznaczona jest dla rozrywki przez posiadające funkcje pozwalające miło spędzić czas wraz z rodziną bądź przyjaciółmi.

5. Kierownik projektu:

W pierwszej turze głosowania zostały zaproponowane dwie kandydatury:

- I. Dawid Gawiński
- II. Patryk Afelt

Za pierwszą kandydaturą oddane były 2 głosy, a za drugą 1 głos. Jedna osoba wstrzymała się od głosu.

W drugiej turze głosowania na kierownika projektu został wybrany Dawid Gawiński poprzez jawne głosowanie, w którym otrzymał 3 głosy za, zaś jedna osoba nadal wstrzymała się od głosu więc zasadą większości wygrał głosowanie.

6. Opis projektu:

Celem przedsięwzięcia jest stworzenie działającej aplikacji w oparciu o standardy inżynierii oprogramowania. Projekt będzie realizowany do momentu zakończenia wszystkich etapów projektowania.

7. Cel projektu

Celem projektu jest stworzenie aplikacji lokalnej wzorującej się na klasycznej grze „Monopoly” zgodnie z wymogami klienta.



8. Zasady współpracy w grupie:

- a. Wszystkie przydzielone nam zadania zobowiązujemy wykonywać terminowo.
- b. Wszyscy angażujemy się w pracę i jesteśmy odpowiedzialni za wyniki.
- c. Szanujemy oraz wspomagamy się nawzajem.
- d. Doceniamy pracę i zaangażowanie innych.
- e. Uwagi zgłaszamy do kierownika projektu.
- f. Podnosimy motywację i zaangażowanie w zespole. Nie używamy wulgarnych wyrażen przeciwko drugiej osobie.
- g. Ważne, aby każdy członek zespołu otrzymał określoną funkcję.

9. Wybór technologii:

a. Środowisko programistyczne

- i. Eclipse 4.5.1

b. Języki programowania

- i. Java SE

c. Inne

- i. GIMP 3.21
- ii. Microsoft Office 2007
- iii. Dia 0.97.2
- iv. Gantt Project 2.7.1

10. Opis ogólny:

- a. Ogólne możliwości projektowania aplikacji,
- b. Prosty i czytelny interfejs,
- c. Możliwość grania w gronie kilku osobowym,
- d. Łatwość w obsłudze,
- e. Rozmieszczenie przycisków w menu w przyjazny sposób dla użytkownika,
- f. Nisko poziom zużycia zasobów procesora.

11. Rola i odpowiedzialność:

- a. Dawid Gawiński – pełni rolę jako kierownik projektu. Odpowiada za nadzór nad wykonaniem projektu zgodnie z harmonogramem prac, sporządzanie notatek do każdego etapu.



Jego rolą jest także przygotowanie odpowiednich warunków oraz sprzętu potrzebnego do realizacji projektu takich jak środowisko programistyczne, komputery.

- b. Paweł Gawron – pełni rolę jako programista (grafik). Odpowiada za wykonanie prostego i przejrzystego interfejsu graficznego aplikacji przyjaznego dla klienta.
- c. Oskar Gruszczyński – pełni rolę programisty. Jego rolą w projekcie jest wykonanie funkcji odpowiadających za silnik gry, czyli np. możliwość wydawania pieniędzy.
- d.
- e. Patryk Afelt – pełni rolę dokumentalisty, testera oraz programisty pomocniczego. Odpowiada za tworzenie dokumentacji potrzebnej w celach projektu oraz wykonuje testy jednostkowe oraz całkowite aplikacji. Dodatkowym zadaniem jest pomoc w ewentualnych problemach programowania aplikacji.

12. Nagrody i kary:

a. Nagrody:

Za pracę zgodnie z harmonogramem uczestnicy projektu otrzymują wynagrodzenie w postaci oceny z przedmiotu, która zostaje określona przez kierownika projektu dla poszczególnych członków zespołu.

b. Kary

Za nie postępowanie zgodnie z regulaminem oraz ustalonym harmonogramem prac, członek zespołu otrzymuje naganę od kierownika projektu. Trzecia nagana skutkującą, po wcześniejszym uzgodnieniu z resztą zespołu, wykluczenie danej osoby z projektu.

13. Rozwiązanie grupy

Rozwiązanie grupy następuje po zakończeniu prac projektowych.

14. Słownik pojęć i synonimów:

- a. **Gracz** – Użytkownik którego celem jest zdobycie jak największej ilości wirtualnych pieniędzy otrzymanych od innych graczy za postawienie ich pionków do gry na polach, które są w posiadaniu gracza. Gracz podczas swojej kolejki losuje liczbę od 1 do 6 na wirtualnej kostce do gry, a następnie porusza swój pionek o wyznaczoną przez kostkę liczbę pól.



- b. **Bank** – W takich rodzajach gier planszowych w rzeczywistości funkcję banku pełni jeden z graczy, który nadzoruje czy odpowiednio zostają przekazane pieniądze za kupno pól bądź domków, hoteli. W wersjach komputerowych rolę bankiera przejmują komputer.
- c. **Karty „Szansa”** – obiekty, które użytkownik losuje z dwóch talii kart występujących na planszy. Różnią się one od siebie tym, że niektóre z nich nakazują graczowi zapłacić odpowiednią kwotę do banku, zaś czasami trafiają się takie karty, że gracz otrzymuje pieniądze od przeciwników bądź z banku.
- d. **Plansza** – jest to obszar na którym występują pionki reprezentujące graczy i karty „Szansy” . Wymiary planszy oraz rozkład pól na niej stały nie zależnie od ilości graczy.

15. Rozstrzygnięcie sporów:

- a. Ważne decyzje są rozstrzygane przez demokratyczne głosowanie.
- b. Mniej ważne spory są rozstrzygane podczas rozmów na osobności z kierownikiem projektu i osób, których dotyczy konflikt.

16. Postanowienie końcowe:

- a. Do projektu dopuszcza się wprowadzenie najwyżej kilku zmian, które nie będą miały wielkiego wpływu na funkcjonalność niniejszego projektu.
- b. Zostaną sporządzone cztery kopie umowy, po jednym egzemplarzu dla każdej ze stron.
- c. Umowa zaczyna obowiązywać w dniu, w którym zostanie podpisana i obowiązuje do 15.09.2016r.

*Paweł
Gawron*

*Oskar
Gruszczyński*

*Dawid
Gawiński*

*Patryk
Afelt*



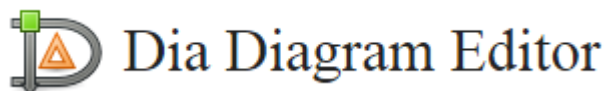
2. Raport o wybrany instrumentarium zespołu projektowego

1. Wybrane środowisko programistyczne – Eclipse (www.eclipse.org)



Eclipse - Platforma napisana w 2004 roku w Javie do tworzenia aplikacji typu rich client. Na bazie Eclipse powstało zintegrowane środowisko programistyczne do tworzenia programów w Javie, które jest razem z tą platformą rozpowszechniane. Projekt został stworzony przez firmę IBM, a następnie udostępniony na zasadach otwartego oprogramowania. W chwili obecnej jest on rozwijany przez Fundację Eclipse.

2. Narzędzie do projektowania diagramów UML – DIA (www.dia-installer.de)



DIA - program służący do tworzenia diagramów ogólnego przeznaczenia. Aktualnie posiada specjalne obiekty ułatwiające rysowanie diagramów zależności, diagramów UML, wykresów, diagramów sieciowych, schematów organizacyjnych oraz prostych obwodów.

3. Technologia wersjonowania dokumentacji projektowej – GIT



GIT – rozproszony system kontroli wersji. Stworzył go Linus Torvalds jako narzędzie wspomagające rozwój jądra Linux. Git stanowi wolne oprogramowanie i został opublikowany na licencji GNU GPL w wersji 2. Oferuje następujące funkcje:

- Praca off-line
- Wsparcie dla istniejących protokołów sieciowych
- Dobre wsparcie dla rozgałęzionego procesu tworzenia oprogramowania



4. Narzędzia do wspomagania pracy grupowej

GitHub (www.github.com)



GitHub – hostingowy serwis internetowy przeznaczony dla projektów programistycznych wykorzystujących system kontroli wersji Git. Stworzony został przy wykorzystaniu frameworka Ruby on Rails i języka Erlang. Serwis działa od kwietnia 2008 roku. W kwietniu 2011 ogłoszono, iż GitHub obsługuje 2 miliony repozytoriów. Github udostępnia darmowy hosting programów open source oraz płatne prywatne repozytoria.

GanttProject (www.ganttproject.biz)



GanttProject – jest to darmowy program, umożliwiający planowanie projektu za pomocą wykresu Gantta. Pozwala w łatwy sposób podzielić projekt na zadania, ustalić powiązania między nimi oraz zarządzać zasobami ludzkimi. GanttProject jest w całości napisany w języku Java, dzięki temu może być uruchomiony w każdym systemie operacyjnym obsługującym ten język, takim jak: Linux, Windows, Mac OSX i inne. Cały program opiera się na licencji GPL.



5. Narzędzia do tworzenia grafiki:

a. GIMP (www.gimp.org)



GIMP – darmowy program do tworzenia i obróbki grafiki rastrowej. Umożliwia malowanie, usuwanie tła obrazu (wycinanie), usuwanie obiektów ze zdjęć (klonowanie), zmianę rozmiaru i perspektywy oraz obrót obrazów, dokonywanie różnych operacji na kolorach, łączenie obrazów poprzez warstwy/maski oraz konwertowanie różnych formatów plików graficznych.

b. CorelDraw (www.coreldraw.com)



CorelDraw X5 – pakiet programów graficznych wchodzących w skład jednego produktu, służący początkowo do obróbki grafiki wektorowej, z czasem wzbogacany o kolejne programy dotyczące różnorodnych zastosowań grafiki komputerowej.

3. *Raport klienta*

1. Proces biznesowy – lokalny.

University Monopol – jest aplikacją lokalną, nie potrzebującą do pełnego funkcjonowania dostępu do Internetu.



Dlatego główny dochód z gry uzyskiwany będzie dzięki sprzedaży wersji pudełkowej w sklepach, takich jak:

- Media Markt
- Empik
- Media Expert
- inne (sklepy, które w swoim asortymencie posiadają gry komputerowe).

W obecnych czasach koszt pudełkowej wersji gry nie będzie wysoki.

Średnio 8zł za sztukę przy produkcji hurtowej.

Zakłada się, że gra będzie sprzedawana po 30 zł za pudełko. Co daje nam 22 zł zysku na pojedynczej sztuce.

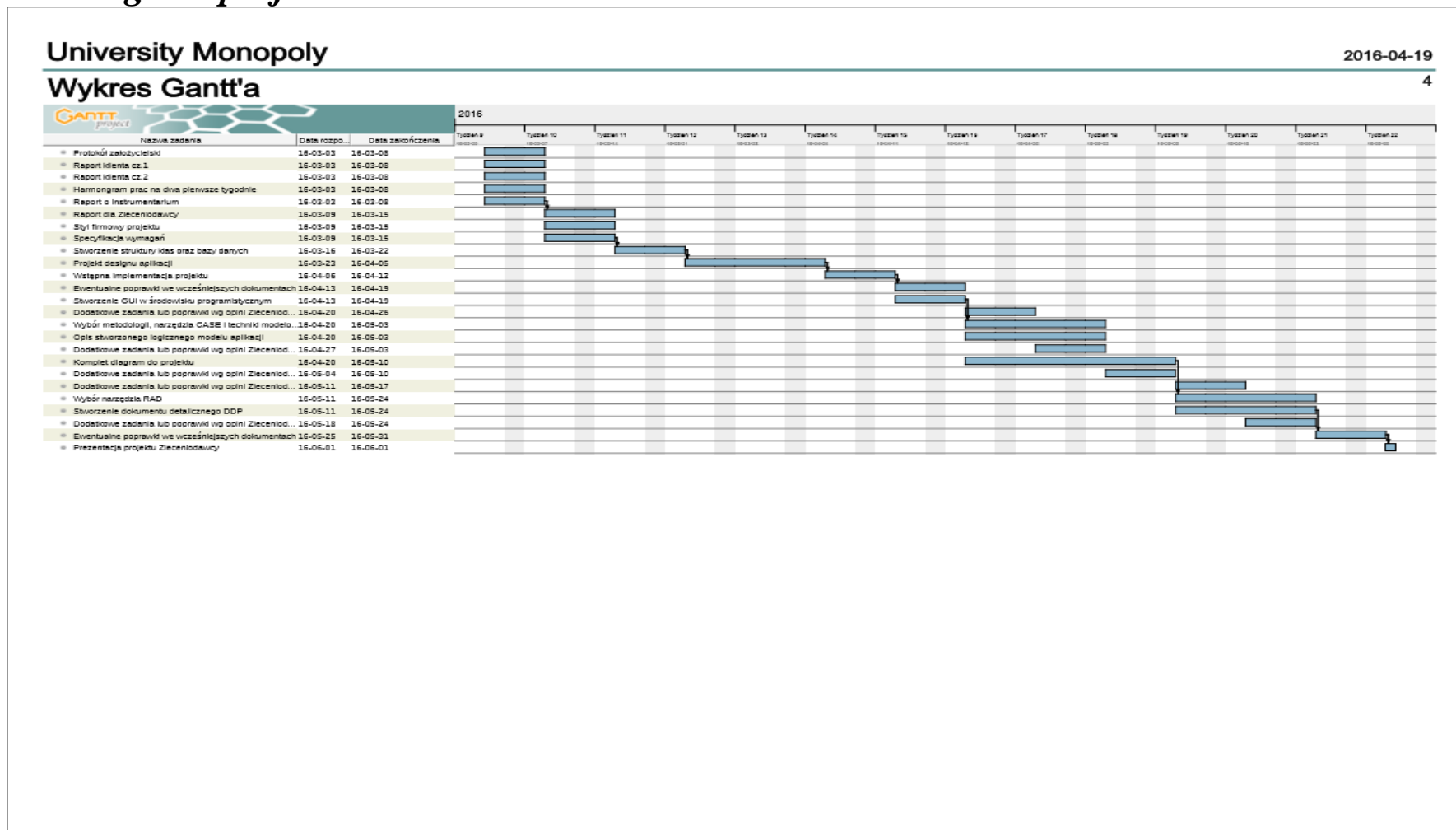
Przy 5000 wyprodukowanych egzemplarzy mamy średnio zysk w postaci 110000 zł.

2. Alternatywne źródło dochodu – STEAM.

Aby poszerzyć zakres sprzedaży możliwe będzie wejście w współpracę z internetową platformą gier Steam. Gra wydawana byłaby wtedy w wersji cyfrowej i zakup polegałby na kupnie elektronicznego klucza aktywacyjnego do gry. Platforma STEAM jest ogólnie znana na całym świecie, przez co dostęp do produktu miało by wiele więcej użytkowników.



4. Harmonogram projektu



Rysunek 1 Wykres stworzony na potrzeby projektu



5. Raport dla Zleceniodawcy

1. Cele przedsięwzięcia.

Celem naszego przedsięwzięcia jest stworzenie aplikacji lokalnej do urozmaicenia wolnego czasu spędzanego z rodziną bądź przyjaciółmi. Nasza gra będzie miała w ofercie dobór ilości graczy oraz będzie umożliwiał zakup dowolnych pól na planszy za wirtualne pieniądze zakładając, że nie są już zajęte przez innych graczy.

Aby zachęcić i przyciągać klientów aplikacja będzie miała intuicyjny i przyjazny dla oka interfejs graficzny ułatwiający rozgrywkę. Dzięki tym udogodnieniom skłonimy klientów do częstszego użytkowania naszej aplikacji.

2. Zakres przedsięwzięcia.

Zakres naszego przedsięwzięcia to określenie wymagań systemu od strony klienta oraz wymagań systemowych, wykonanie dokumentacji projektu, a co za tym idzie, fazy analizy oraz projektowania systemu. Oprócz tego będziemy testować aplikację, po czym zaimplementowana jako gotowy produkt, który spełni zarówno nasze oczekiwania jak i zleceniodawcy.

3. Systemy zewnętrzne.

Gra „University Monopoly” prócz stworzenia aplikacji jest rozszerzonym projektem co powoduje, iż powinien współpracować z technologiami obsługującymi cały projekt. Użyjemy technologii darmowych zezwalających na zastosowanie komercyjne. Nasz produkt będzie kompatybilny z technologią Java. Aplikacja będzie możliwa do uruchomienia na każdym systemie operacyjnym bez jakiegokolwiek zmiany graficznej oraz funkcjonalności, aby zapewnić dostęp dla każdego użytkownika, dzięki czemu maksymalnie poszerzymy grono klientów.

4. Szkicowy opis wymagań.

Aplikacja ma za zadanie wspomagać. Każdy użytkownik – klient powinien mieć możliwość zapoznania się z aktualną ofertą, a po pobraniu, wybrania określonego trybu gry i dopasowania ilości graczy..

5. Szkicowy opis systemu.

Nasza aplikacja będzie ogólnodostępna w formie aplikacji lokalnej. Na zewnętrznej platformie takiej jak Steam zostanie umieszczona aplikacja.



Dostęp do niej będzie możliwy także przez kupno gry w wersji pudełkowej w sklepach z grami komputerowymi np. „Empik”.

6. Szkicowy opis proponowanych rozwiązań.

Aplikacje do gry planszowej „University Monopoly” zaprogramujemy w środowisku programistycznym NetBeans IDE 8.0.2.

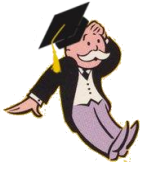
7. Wstępny kosztorys projektu.

W celu zobrazowania kosztów związanych z realizacją naszego projektu przygotowano szacunkowy kosztorys obejmujący koszty: sporządzenia dokumentacji, projektowania aplikacji, pracy ludzkiej, wdrożenia aplikacji i szkoleń.

Lp.	Usługa	Koszt [zł]
1.	Analiza wymagań i sporządzenie raportu	4500
2.	Opracowanie stylu firmowego	1500
3.	Zaprojektowanie architektury aplikacji	6000
4.	Zaprojektowanie i zaprogramowanie interfejsów graficznych	8000
6.	Zaprogramowanie aplikacji	16000
7.	Testowanie i optymalizacja aplikacji	6000
8.	Sporządzenie dokumentacji projektowej	7000
9.	Wdrożenie aplikacji	5000
10.	Szkolenie dla użytkowników	3500
11.	Koszty dojazdów	2000
Łączny koszt realizacji projektu:		59 500

*Koszty poszczególnych etapów tworzenia aplikacji zawierają pensję, które będą wypłacane członkom zespołu

W przypadku gdy zleceniodawca nie posiada zasobów sprzętowych i programowych niezbędnych do wdrożenia i utrzymania stworzonej aplikacji, do wyliczonej kwoty należy doliczyć koszt zakupu tych urządzeń i oprogramowania, a przedstawia się następująco:

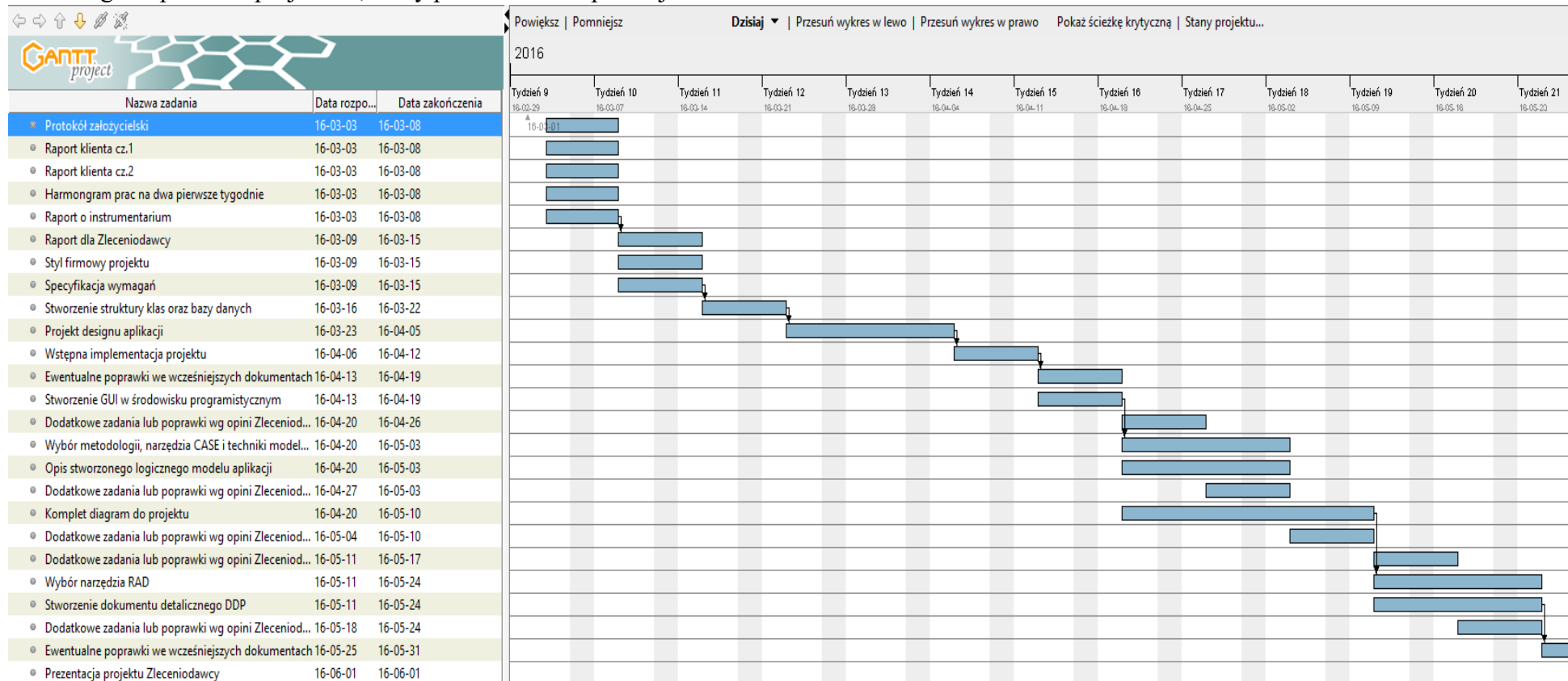


- Wypożyczenie 4 komputerów z systemem operacyjnym na okres 4-ech miesięcy:
19200 zł brutto
 - Wynajem lokalu na okres 4-ech miesięcy: 4000 zł
 - Wykupienie licencji na oprogramowanie: brak(oprogramowanie typu OpenSource)
- Razem: 23200 zł



8. Wstępny harmonogram prac.

Po zapoznaniu się ze wstępnymi wymaganiami dotyczącymi projektu i ich analizie w zespole projektowym sporządzono wstępny harmonogram prac nad projektem, który przedstawiono poniżej.



Rysunek 2 Wykres stworzony na potrzeby projektu



6. Pełna wersja specyfikacji wymagań zgodna z zaleceniami IEEE

1. Wprowadzenie

1.1 Cel

Celem tego dokumentu jest przedstawienie ogólnego oraz szczegółowego zarysu tworzonej aplikacji oraz funkcjonalności, które będzie posiadała.

1.2 Zakres

Projekt dotyczy stworzenia aplikacji desktopowej na komputery osobiste użytkowników, pozwalającej grać kilku graczom na jednym komputerze w grę University Monopoly.

1.3 Pojęcia

RAM - rodzaj pamięci występujący w urządzeniach przenośnych umożliwiający pracę na tym sprzęcie

System operacyjny - oprogramowanie zarządzające komputerem, tworzące środowisko do uruchamiania i kontroli zadań użytkownika.

PC – (ang. Personal Computer) Komputer osobisty.

1.4 Odniesienia

Standard IEEE 830

Strona internetowa Wikipedia.pl

1.5 Streszczenie

W dalszej części specyfikacji wymagań projektu zostanie w sposób szczegółowy opisana tworzona aplikacja oraz wszelkie funkcjonalności planowane do wykonania.

2. Opis ogólny

2.1 Perspektywa produktu

2.1.1 Interfejsy systemu

Interfejs systemu powinien być tak skonstruowany aby podczas rozgrywki główne komponenty były cały czas widoczne oraz możliwe było otworzenie nowych komponentów pozwalających podejrzeć interesujące użytkownika informacje nie zasłaniając głównego komponentu.

2.1.2 Interfejsy użytkownika

Interfejs użytkownika końcowego musi być przejrzysty i łatwy w obsłudze.



Każdy jego element powinien być umiejscowiony w taki sposób, aby ułatwić użytkownikowi jego obsługę. Elementy oraz ich funkcje powinny być należycie opisane oraz mieć odpowiedni wygląd, który usprawni korzystanie z aplikacji.

2.1.3 Interfejsy sprzętowe

Aplikacja wymaga komputera osobistego z systemem Windows z zainstalowanym środowiskiem wirtualnym Java.

2.1.4 Ograniczenia pamięci

Aplikacja będzie potrzebować około 100 MB pamięci RAM w celu bezproblemowego działania oraz około 50 MB wolnej przestrzeni dyskowej urządzenia do zainstalowania jej na urządzeniu.

2.1.5 Operacje

Użytkownik będzie potrzebował podstawowej wiedzy w ramach instalacji gry na urządzeniu, uruchamiania jej oraz obsługi. Wszystkie potrzebne informacje będzie można znaleźć w instrukcji obsługi dołączonej do gry.

2.2 Funkcje produktu

Aplikacja będzie pozwalała użytkownikowi rozegrać grę University Monopoly wraz z innymi graczami.

2.3 Charakterystyki użytkowników

Aplikacja skierowana jest do użytkowników chcących zagrać w grę typu Monopoly, głównie studentów Politechniki Koszalińskiej.

2.4 Ograniczenia

Aplikacja będzie posiadała ograniczenia odnośnie ilości graczy uczestniczących jednorazowo w rozgrywce.

2.5 Założenia i zależności

Założeniem projektu jest utworzenie gry pozwalającej studentom zagrać w grę, której akcja będzie rozgrywała się na Politechnice Koszalińskiej.

3.Opis wymagań

3.1 Funkcjonalność

3.1.1 Rozpoczęcie gry

- Wybór ilości graczy



- Wybór pionka
- Wylosowanie kolejności
- Ustawianie czasu gry

3.1.2 Rozgrywka

- Rzut kostką
- Możliwość zakupu pola
- Przegląd karty pola
- Możliwość licytacji pola
- Podgląd kart gracza
- Podgląd kart innych graczy
- Podgląd funduszy
- Budowanie domu
- Budowa hotelu
- Sprzedaż domu
- Sprzedaż hotelu
- Zastawa hipoteczna
- Regulowanie płatności
- Wykup hipoteki
- Handel między graczami
- Użycie kart specjalnych

3.1.3 Zakończenie gry

- Upływanie czasu
- Bankructwo
- Wygrana
- Warunki zwycięstwa w razie upływania czasu

3.2 Specyfikacje przypadków użycia

- **Wybór ilości graczy**

Gracz samodzielnie wybiera ilość osób które będą uczestniczyły w rozgrywce.

- **Wybór pionka**

Każdy gracz uczestniczący w rozgrywce będzie wybiera kolor pionka którego będzie używał w czasie gry.



- **Wylosowanie kolejności**

Gracze uczestniczący w rozgrywce samodzielnie losują kolejność w jakiej będą wykonywali swoje ruchy.

- **Ustawianie czasu gry**

Rozpoczynając rozgrywkę gracze ustawiają czas rozgrywki, która zakończy się w chwili upłynięcia określonego limitu czasowego.

- **Rzut kostką**

Każdy gracz samodzielnie wciska przycisk rzuć kostką, a system będzie generował losowe wartości z zakresu od 1 do 6.

- **Możliwość zakupu pola**

Gracze po najechaniu na odpowiednie pole będą dokonują zakupu pola, jeśli pole nie będzie posiadało właściciela.

- **Przegląd karty pola**

Każdy gracz może w dowolnej chwili obejrzeć dowolną kartę którą posiada w swojej talii.

- **Możliwość licytacji pola**

W razie bankructwa gracz ma możliwość sprzedaży jednej ze swoich kart graczowi który wygra licytację.

- **Podgląd kart gracza**

Każdy gracz w dowolnej chwili ma możliwość przejrzania ilości oraz wartości swoich kart, które będą wyświetlone obok siebie.

- **Podgląd kart innych graczy**

Każdy gracz w dowolnej chwili ma możliwość obejrzenia kart innych graczy.

- **Podgląd funduszy**

Gracze w każdej chwili ma możliwość sprawdzić ile posiadają pieniędzy.



- **Budowanie domu**

Gdy gracz trafi na pole oznaczające kartę którą posiada, może postawić na tym polu jeden dom, co zwiększy jego wpływy gdy inny gracz trafi na to pole.

- **Budowa hotelu**

W razie gdy gracz trafi na swoje pole na którym będzie już posiadał 4 domy, a na różnica domów pomiędzy polem na którym znajduje się gracz, a sąsiednimi, nie będzie większa niż jeden, gracz posiadający odpowiednią ilość pieniędzy, będzie mógł postawić hotel, który zwiększy jego wpływy, w razie gdy inni gracze trafią na to pole.

- **Sprzedaż domu**

W razie bankructwa gracz ma możliwość odsprzedażania jednego domu bankowi, aby otrzymać zwrot pieniędzy za dom.

- **Sprzedaż hotelu**

W razie bankructwa gracz ma możliwość odsprzedażania hotelu bankowi, aby otrzymać zwrot pieniędzy za hotel.

- **Zastawa hipoteczna**

Gracz który zbankrutuje ma możliwość zastawienia w hipotekę jednej ze swoich kart, za kwotę określoną na karcie. Karta ta nie będzie od tej pory należeć do gracza.

- **Regulowanie płatności**

Gracz, który najedzie na pole innego gracza jest zmuszony zapłacić drugiemu graczowi opłatę określoną na karcie oznaczającej dane pole.

- **Wykup hipoteki**

Gracz, który zastawił swoją kartę w hipotekę, ma możliwość wykupienia swojej karty, gdy ponownie najedzie na pole oznaczające daną kartę oraz będzie posiadał odpowiednią ilość pieniędzy.

- **Handel między graczami**

Gracze mają możliwość handlu swoimi kartami z innymi graczami. Warunki handlu będą ustalane przez graczy.



- **Użycie kart specjalnych**

W grze będą istniały karty specjalne, które będą posiadały specjalne właściwości, np.: gracz który trafił do więzienia, może wydostać się z więzienia natychmiast jeśli posiada kartę zwolnienia z więzienia i zdecyduje się jej użyć.

- **Upłynięcie czasu**

Gdy upłynie czas określony przez graczy na początku gry, gra kończy się.

- **Bankructwo**

Gracz kończy grę gdy zbankrutuje, nie będzie posiadał żadnych kart oraz pieniędzy.

- **Wygrana**

Gracz wygrywa, gdy posiada największy majątek po upłynięciu określonego czasu. Gra kończy się również, gdy reszta graczy zbankrutuje.

- **Warunki zwycięstwa w razie upłynięcia czasu**

Po upłynięciu określonego czasu zwycięzcą zostaje gracz posiadający największy majątek, w co wlicza się cena posiadłości oraz posiadane fundusze.

3.3 Wymagania dodatkowe

Klasyfikacja wymagań funkcjonalnych

Funkcjonalność	Typ
Wybór ilości graczy	Pożądane
Wybór pionka	Opcjonalne
Wylosowanie kolejności	pożądane
Ustawianie czasu gry	Wymagane
Rzut kostką	Wymagane
Możliwość zakupu pola	Wymagane
Przegląd karty pola	Wymagane
Możliwość licytacji pola	Pożądane



Podgląd kart gracza	Wymagane
Podgląd kart innych graczy	Wymagane
Podgląd funduszy	Wymagane
Budowanie domu	Wymagane
Budowa hotelu	Wymagane
Sprzedaż domu	Wymagane
Sprzedaż hotelu	Wymagane
Zastawa hipoteczna	Wymagane
Regulowanie płatności	Wymagane
Wykup hipoteki	Wymagane
Handel między graczami	Opcjonalne
Użycie kart specjalnych	Wymagane
Upływanie czasu	Wymagane
Bankructwo	Wymagane
Wygrana	Wymagane
Warunki zwycięstwa w razie upływu czasu	Wymagane

4. Słownik projektu

Gracz

Jest to osoba uczestnicząca w grze.

Karta pola

Karta posiadająca informacje o danym polu na planszy

Karta specjalna

Karta, która daje graczowi bonus, lub nakazuje zapłacić bankowi.



Zastaw hipoteczny

Karta przekazana bankowi za pożyczanie określonej na karcie sumy będąca gwarancją spłacenia długu.

Czas

Czas określony przez graczy na początku gry i pokazujący ile czasu pozostało do ukończenia rozgrywki.

Pionek

Znak na planszy o określonym przez gracza kolorze, oznaczający pole na którym aktualnie gracz się znajduje.

Kostka

System generujący liczby pseudolosowe z zakresu od 0 do 6.

Pieniądze

Liczby przedstawiająca aktualny stan budżetu graczy oraz banku.

Dom

Znak odpowiedniego koloru umieszczony na polu, oznaczający wybudowanie domu na danym polu.

Hotel

Znak odpowiedniego koloru umieszczony na polu, oznaczający wybudowanie hotelu na danym polu.

Bank

System reprezentujący działanie banku.

5. Identyfikacja aktorów

1) **Użytkownik** odpowiedzialny jest za:

- Wybór ilości graczy
- Wybór pionka
- Wylosowanie kolejności



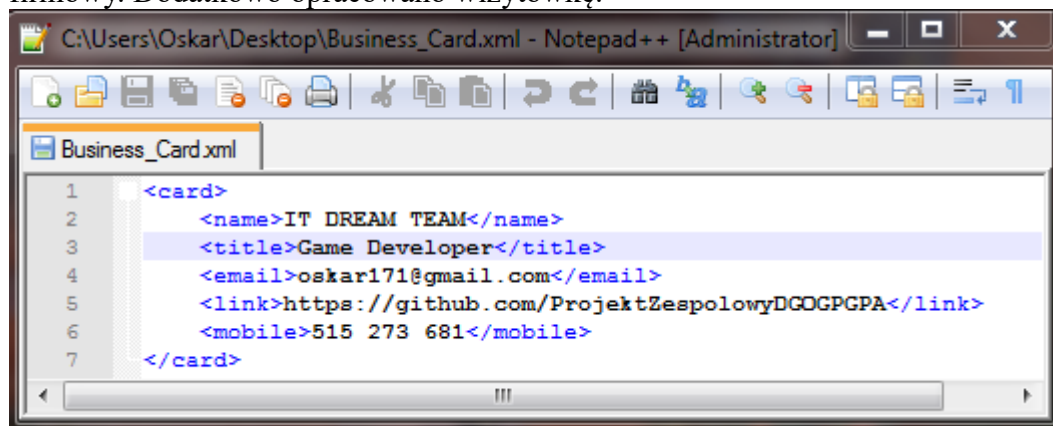
- Ustawianie czasu gry
- Rzut kostką
- zakup pola
- Przegląd karty pola
- licytacja pola
- Podgląd kart gracza
- Podgląd kart innych graczy
- Podgląd funduszy
- Budowanie domu
- Budowa hotelu
- Sprzedaż domu
- Sprzedaż hotelu
- Zastawa hipoteczna
- Regulowanie płatności
- Wykup hipoteki
- Handel między graczami
- Użycie kart specjalnych

2) **System** odpowiedzialny jest za:

- Określenie zwycięzcy w razie upłynięcia czasu
- Wylosowanie liczby oznaczającej rzut kostką
- Liczenie czasu
- Wydanie pieniędzy z banku
- Zatrzymanie gracza na określonej ilości tur gdy znajdzie się w więzieniu lub szpitalu

7. Styl firmowy

Struktura niniejszego dokumentu sporządzona została na podstawie stworzonego przez zespół projektowy stylu formalnego dokumentu, a graficzne wstawki prezentują papier firmowy. Dodatkowo opracowano wizytówkę.



Rysunek 3 Widok zaprojektowanej wizytówki



8. Notatka o narzędziach CASE

a. Metodologia

Jako metodologię prowadzenia projektu zdecydowano się na model iteracyjny. Jest to model pozwalający na skorzystanie z ciągłego kontaktu z klientem w celu wytworzenia oprogramowania bardziej pasującego do jego wymagań.

b. Narzędzia CASE

W ramach projektu wybrane zostały następujące narzędzia CASE:

- Repozytorium GIT github.com
- Środowisko programistyczne NetBeans 8.1

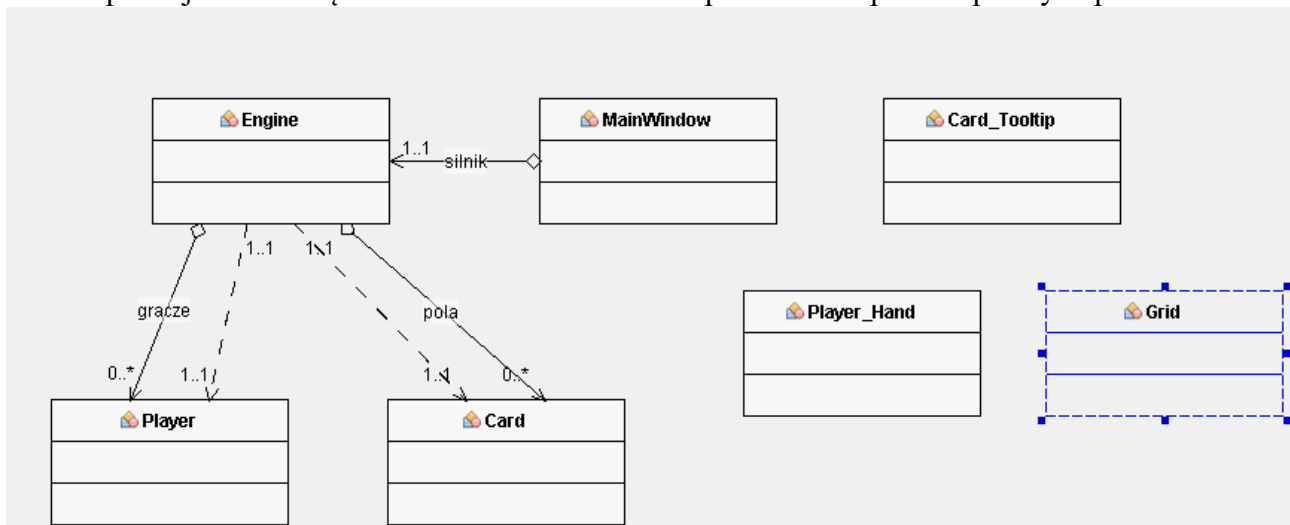
c. Metody modelowania

W projekcie zastosowana została metoda modelowania obiektowego. Pozwala to na stworzenie składowych systemów, które będą od siebie niezależne i przygotowane do wielokrotnego wykorzystania.

Pozwoli to na wykorzystanie utworzonego kodu w przyszłych projektach tego typu. Jako język modelowania obiektowego wybrano język UML, gdyż to jest język znany przez wszystkich członków grupy projektowej.

9. Diagramy

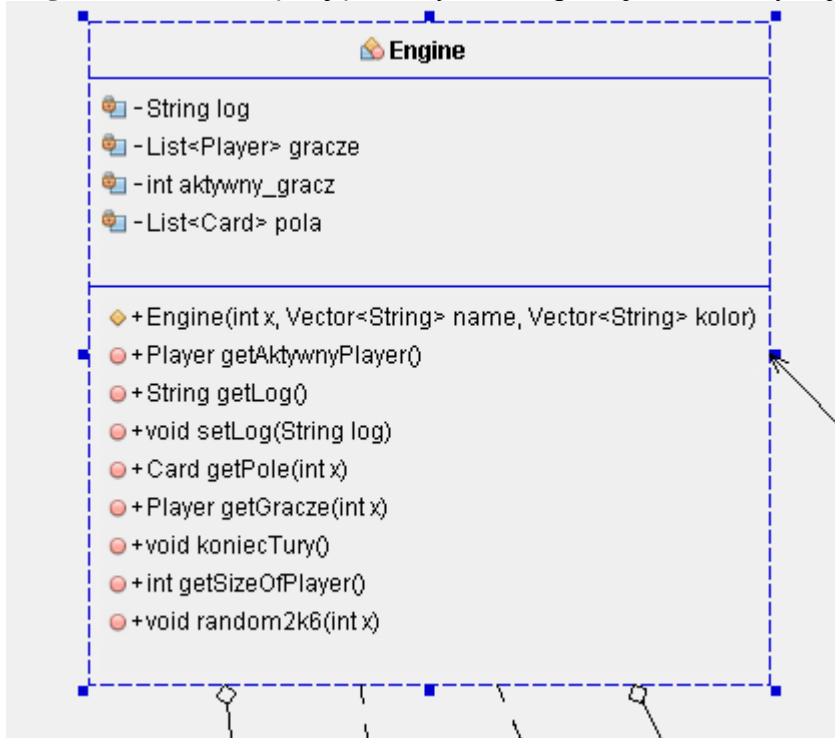
Nasza aplikacja składa się z czterech klas i trzech komponentów w pełni napisanych przez nas.



Rysunek 4 Diagram wszystkich klas i komponentów



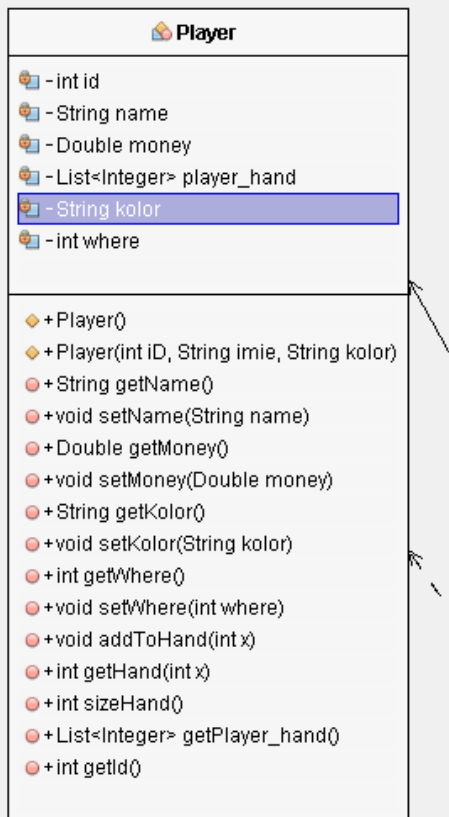
Engine – Klasa zarządzająca wszystkimi operacjami na danych jest to silnik naszej gry.



Rysunek 5 Szczegółowy widok klasy Engine



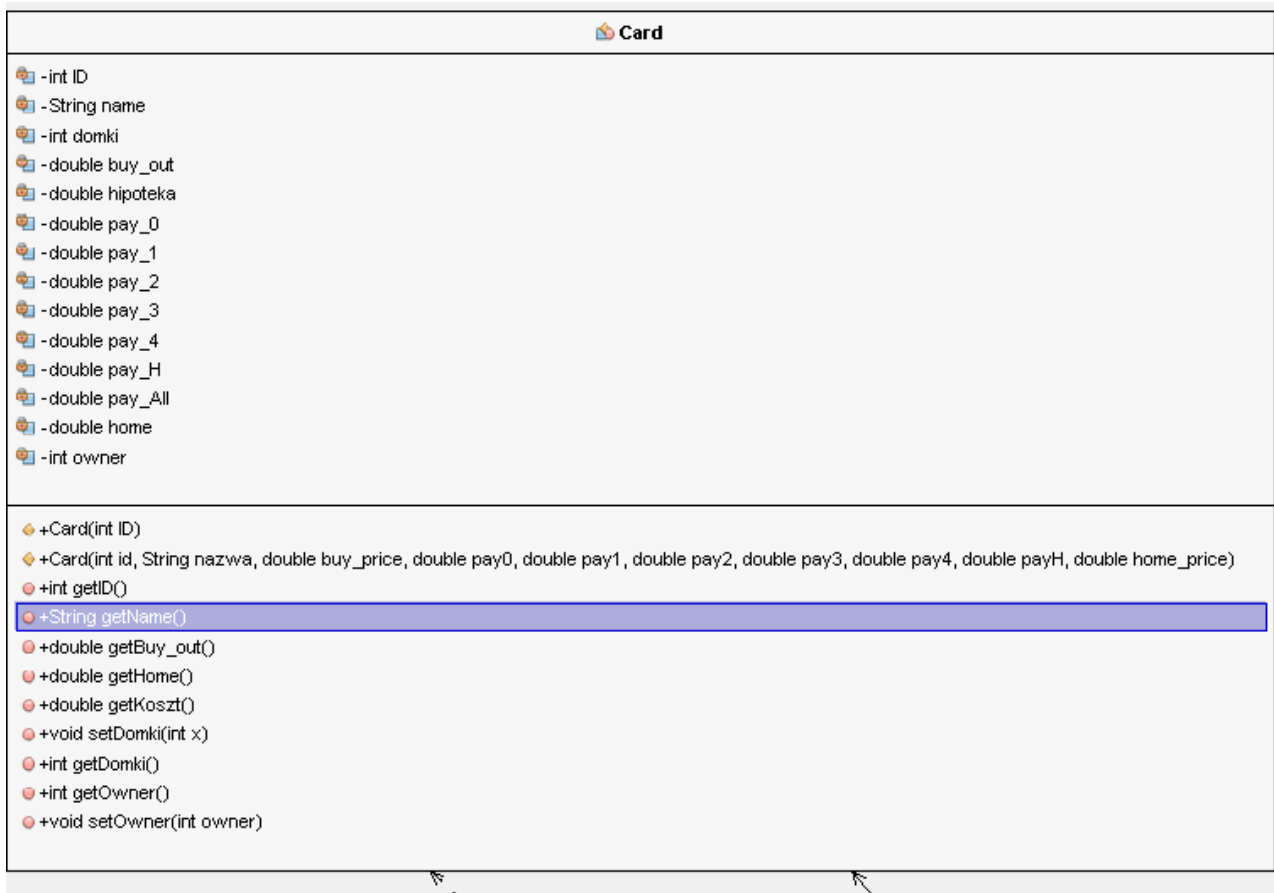
Player – Klasa zawierająca wszystkie dane na temat gracza.



Rysunek 6 Szczegółowy widok klasy **Player**



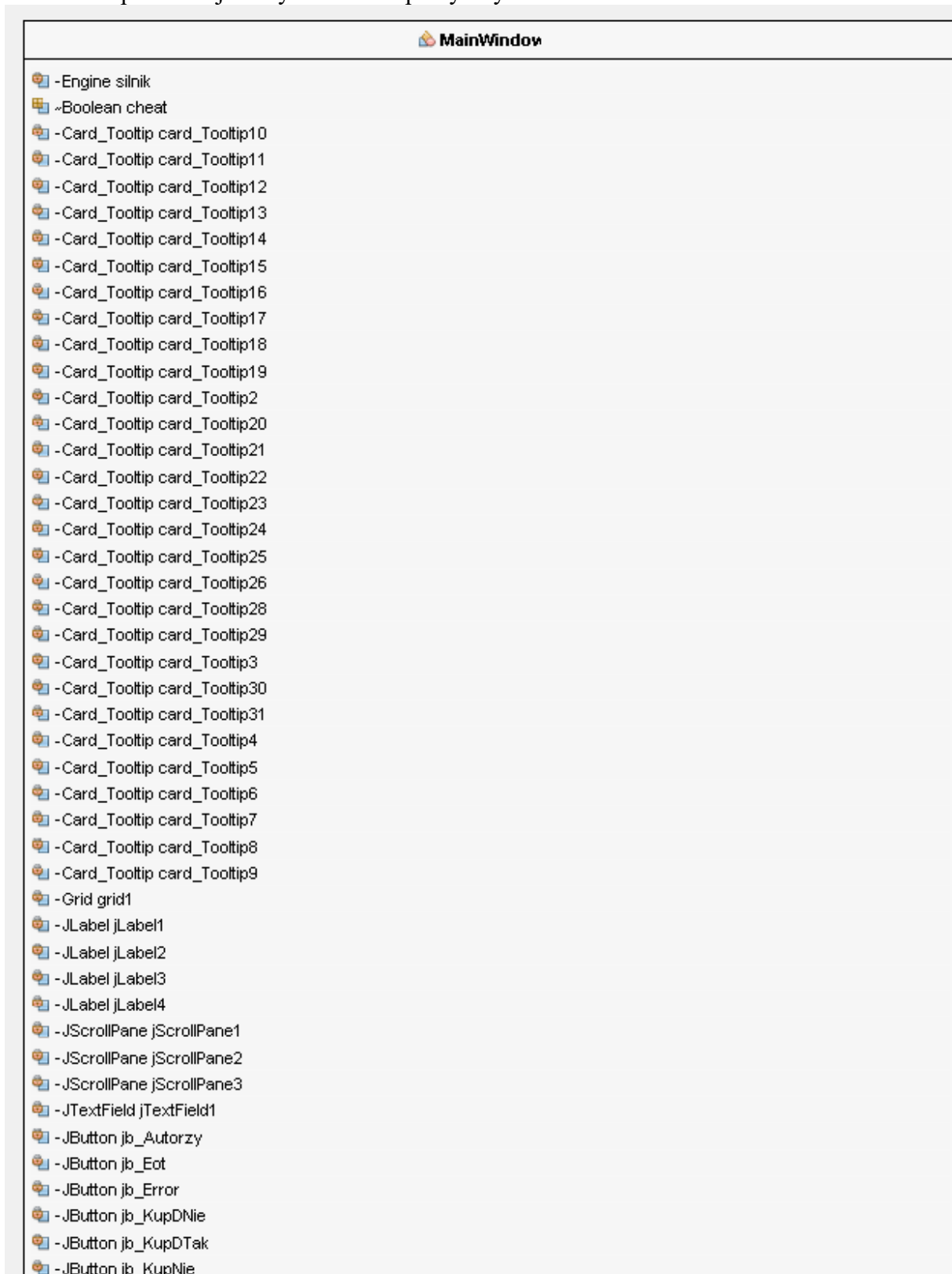
Card – Klasa opisująca pojedynczą kartę zawierającą metody ułatwiające dostęp do tych pól.



Rysunek 7 Szczegółowy widok klasy Card



MainWindow – Klasa która jest naszym głównym oknem są tu ustawione wszystkie przyciski i wizualna reprezentacja danych z trzech powyższych klas



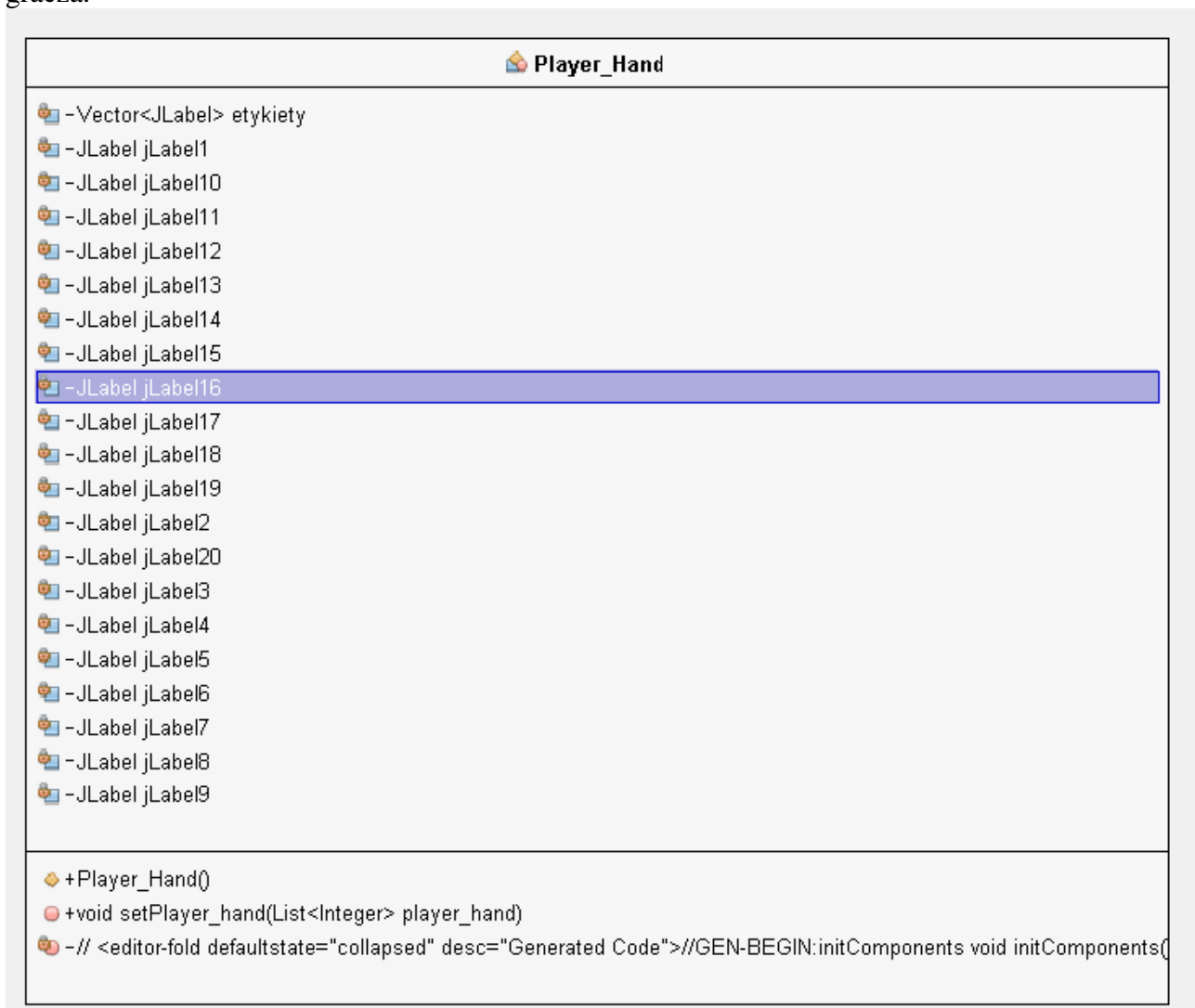
Rysunek 8 Szczegółowy widok klasy MainWindow cz.1



Rysunek 9 Szczegółowy widok klasy MainWindow cz.2



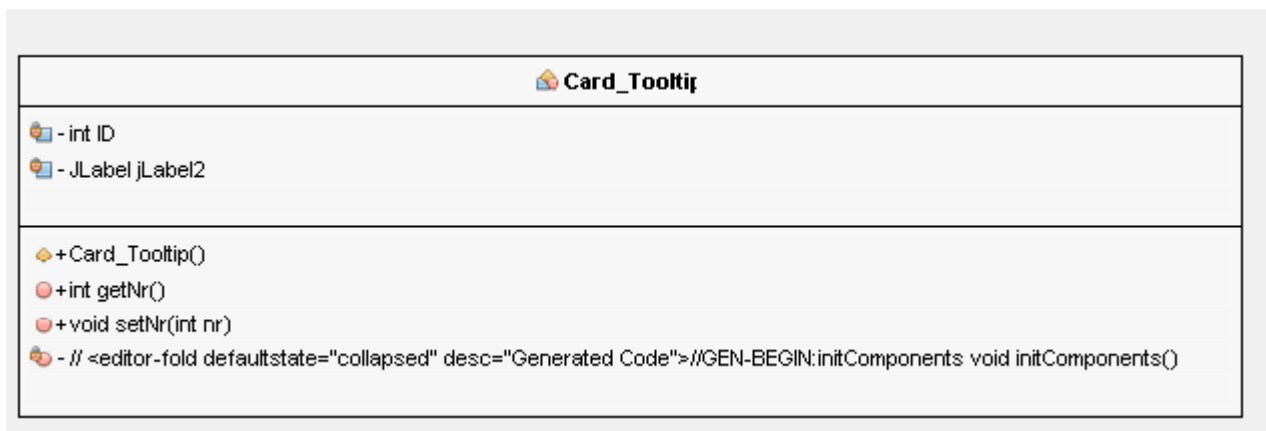
PlayerHand – Jest to komponent który służy do graficznego prezentowania aktualnego stanu kart gracza.



Rysunek 10 Szczegółowy widok komponentu PlayerHand

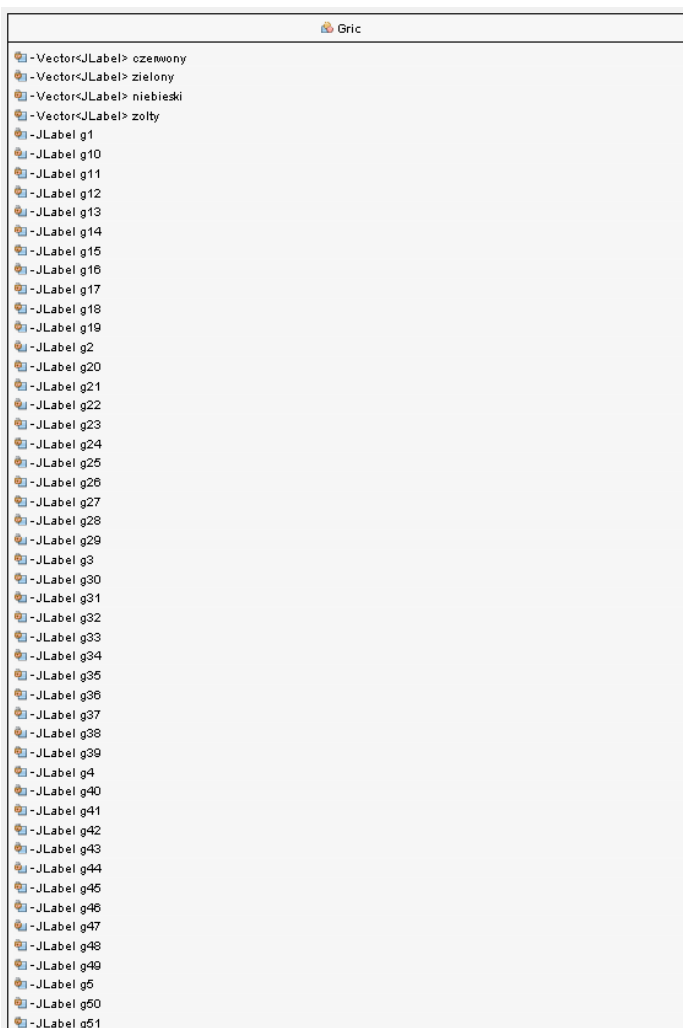


CardTooltip – Komponent stworzony do graficznego przedstawienia danych o polach na planszy

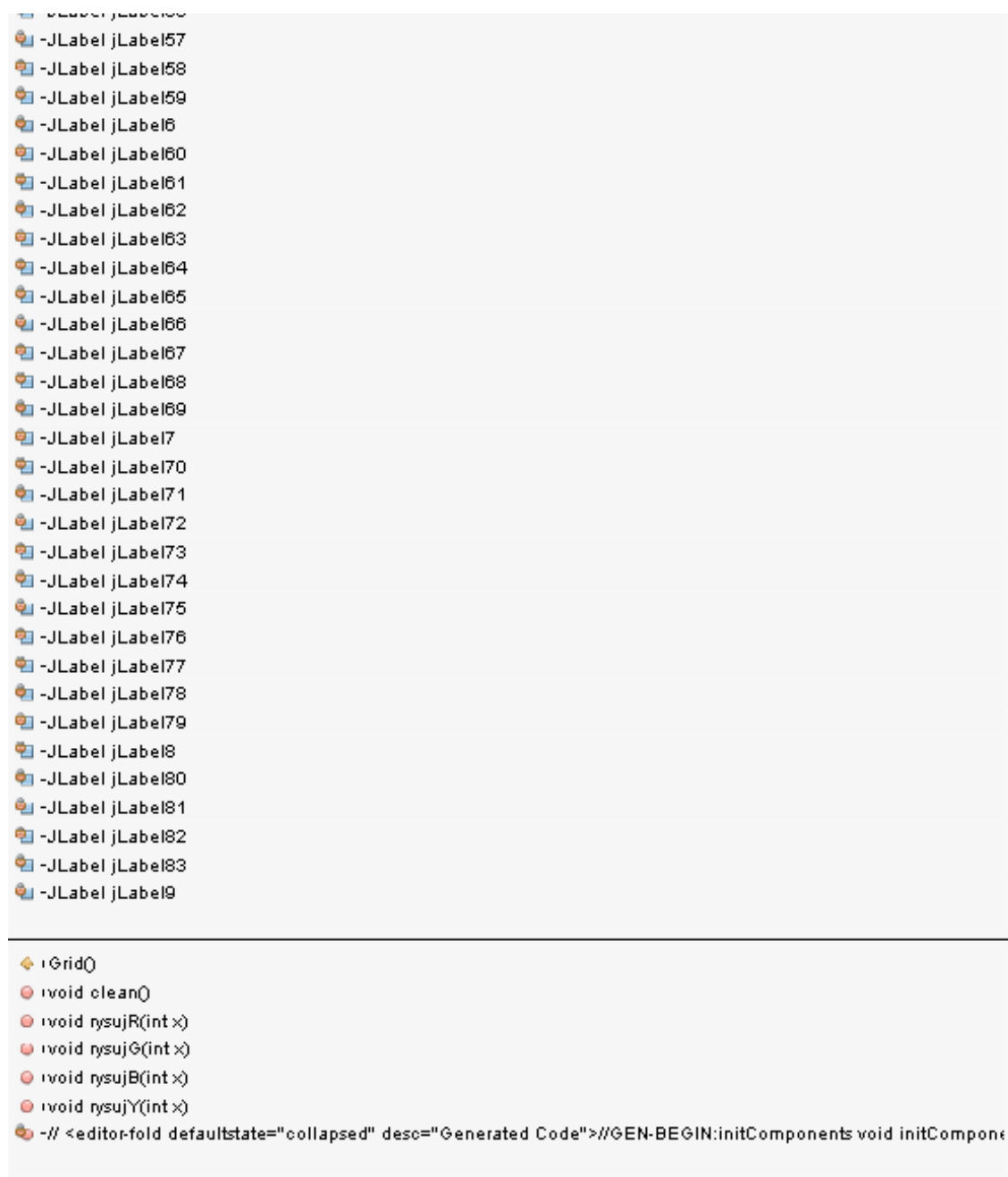


Rysunek 11 Szczegółowy widok komponentu CardTooltip

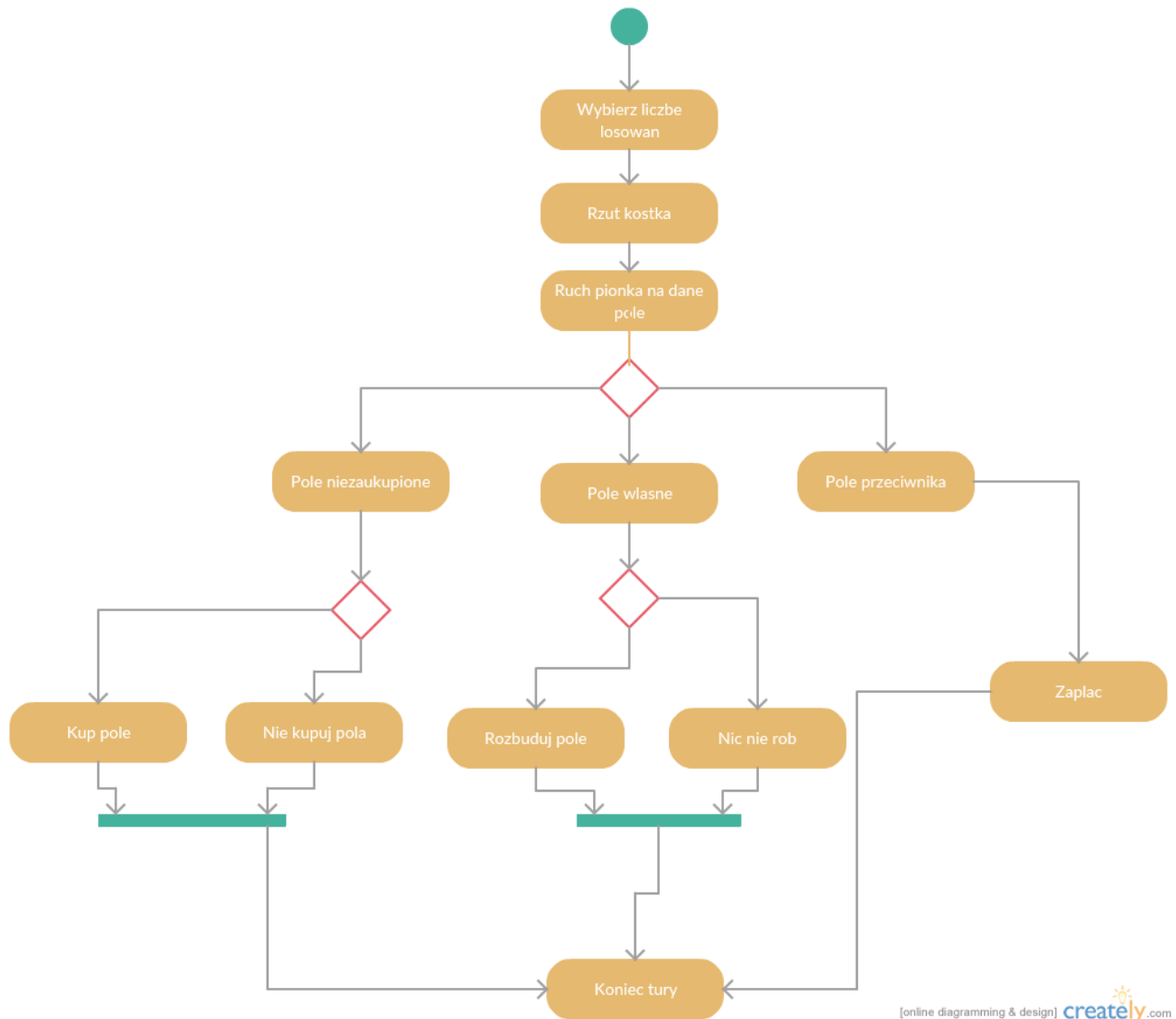
Grid – Komponent stworzony do graficznej reprezentacji naszej planszy dzięki niemu widzimy jaki gracz gdzie aktualnie się znajduje



Rysunek 12 Szczegółowy widok komponentu Grid cz.1

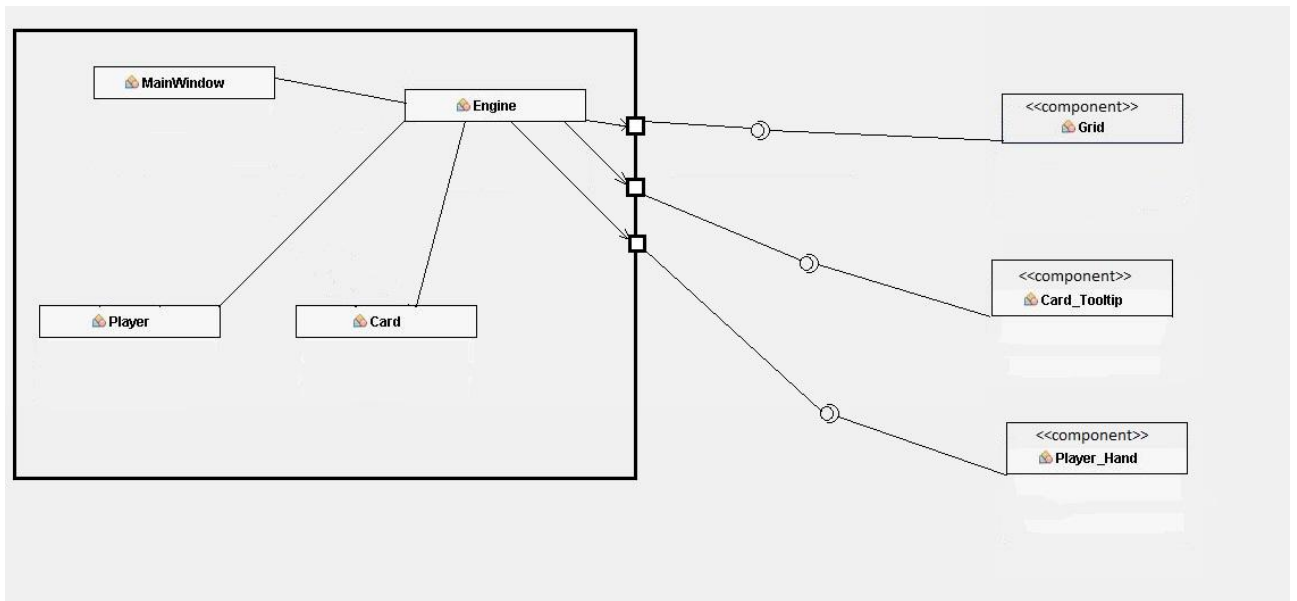


Rysunek 13 Szczegółowy widok komponentu Grid cz.2

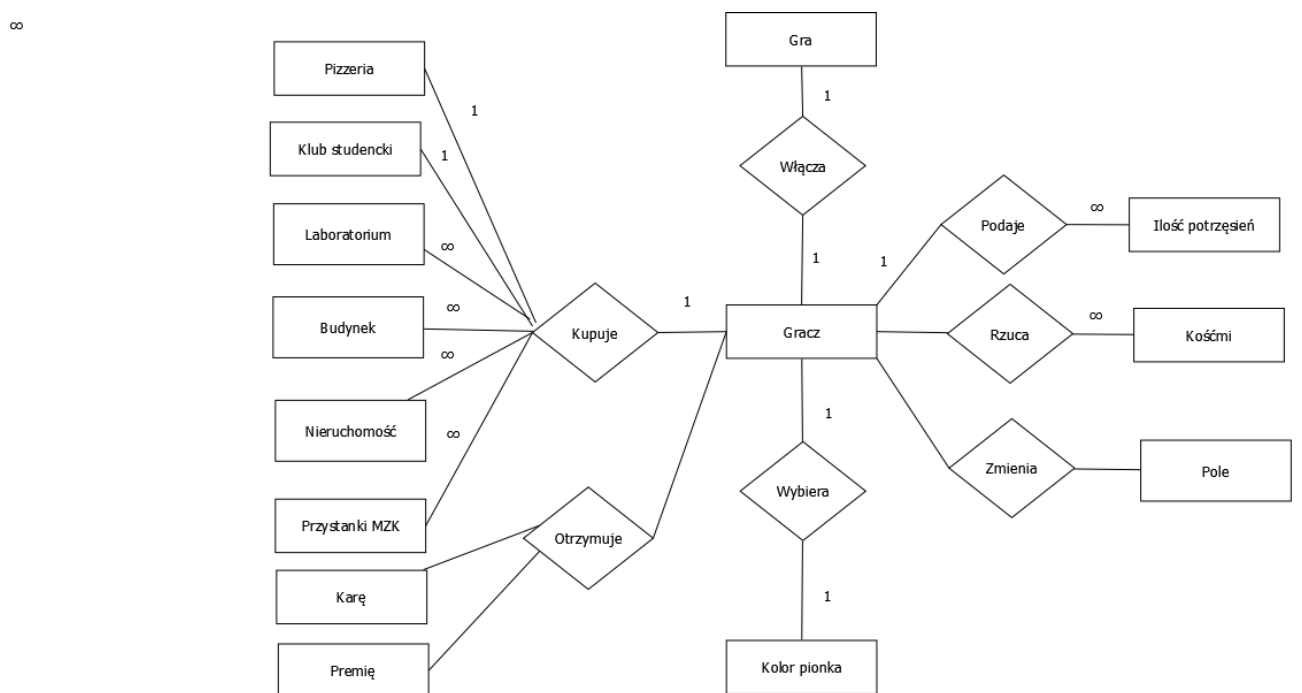


[online diagramming & design] creately.com

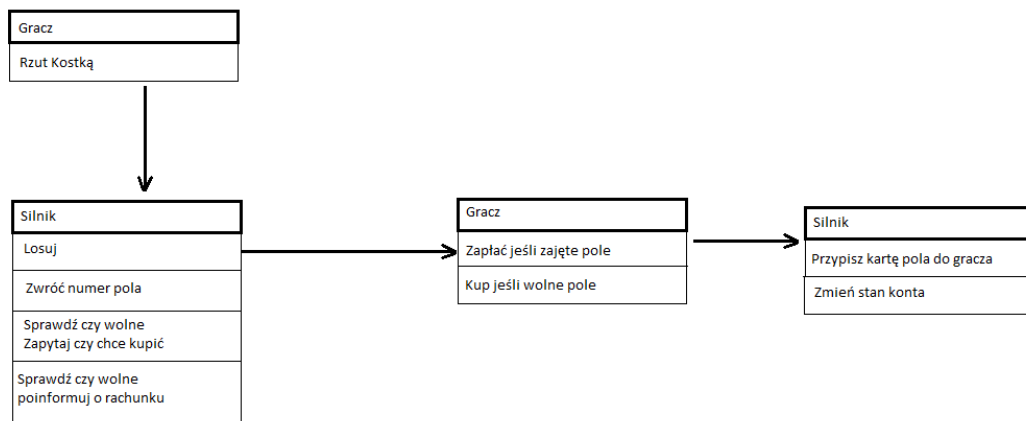
Rysunek 14 Diagram aktywności



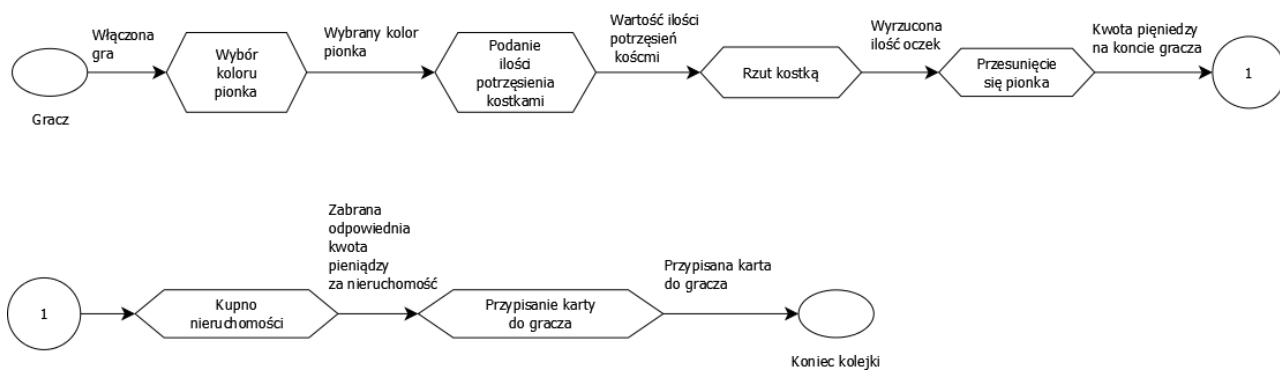
Rysunek 15 Diagram komponentów



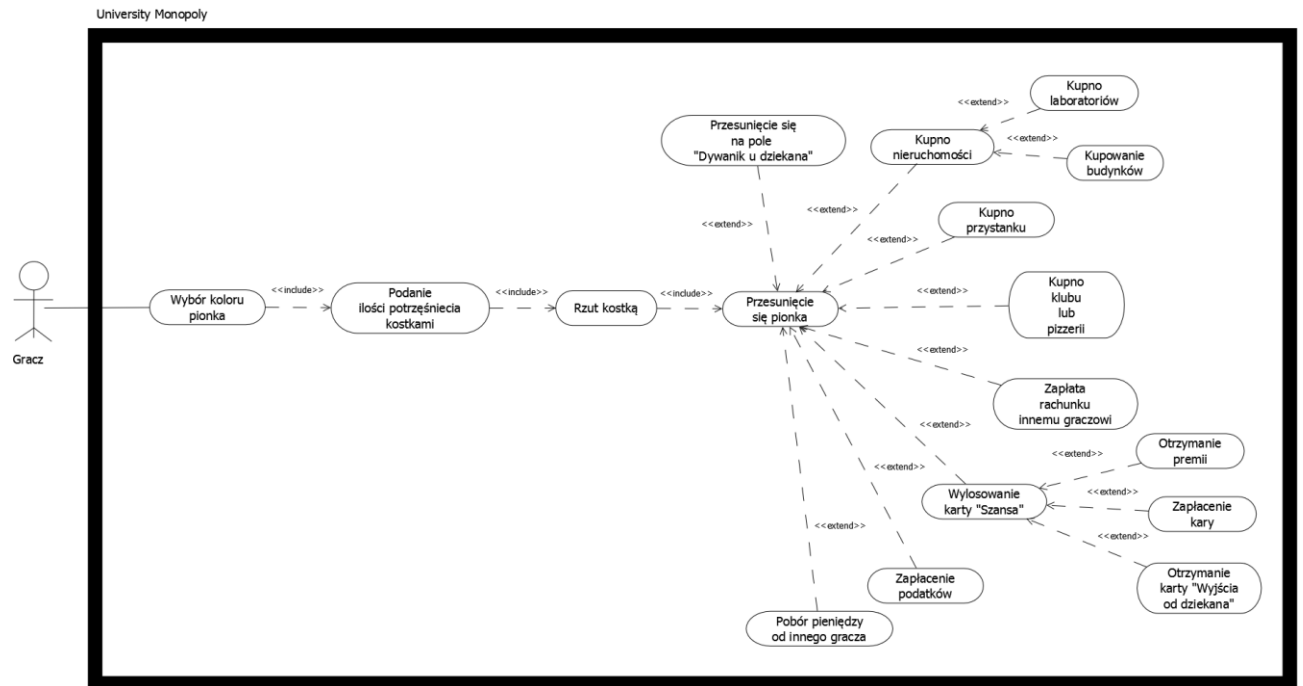
Rysunek 16 Diagram encja-związek



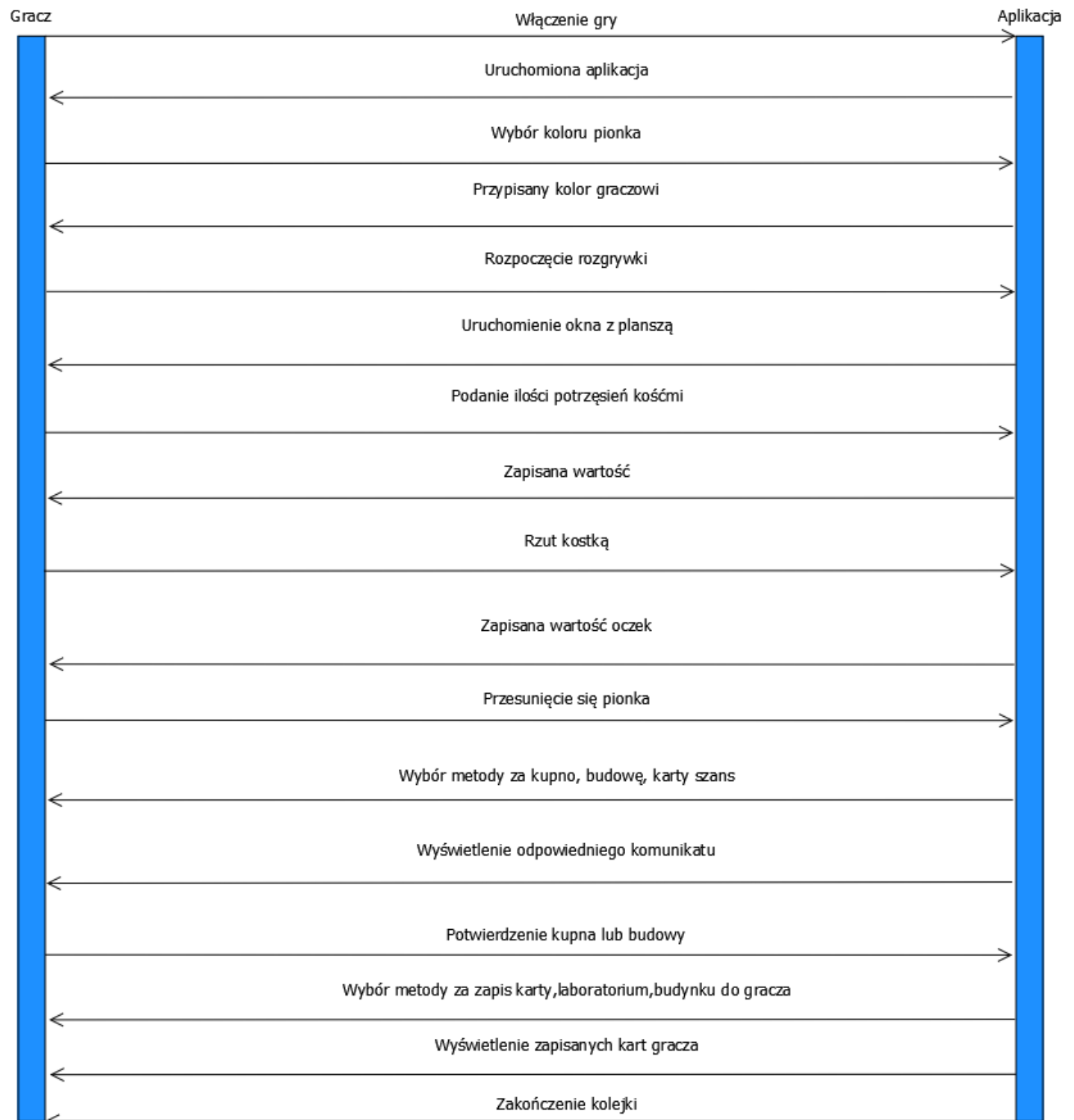
Rysunek 17 Diagram współpracy



Rysunek 18 Diagram przepływu danych dla kupna pola



Rysunek 19 Diagram przypadków użycia



Rysunek 20 Diagram sekwencji dla jednej kolejki z kupnem pola



10. *Wybór narzędzi RAD*

NetBeans IDE 8.1 - środowisko programistyczne oferujące wsparcie dla programowania komponentowego oraz języka Java. Środowisko to zostało wybrane z uwagi na możliwość tworzenia GUI z gotowych komponentów, których wygląd można później edytować oraz z uwagi na wsparcie dla języka Java, w którym został napisany projekt.



Rysunek 21 Przedstawienie wybranego narzędzia RAD

11. *Dokument detaliczny projektu*

STRESZCZENIE Niniejszy dokument detaliczny projektu opisuje detale pracy zespołu projektowego, który skupia się na stworzeniu gry komputerowej na wzór Eurobiznesu. Pierwsza część dokumentu zawiera opis ogólnych założeń projektowych, a także wykorzystywanych standardów i konwencji. Druga część opisuje specyfikacje wykorzystanych w projekcie komponentów. Wszystkie zmiany w dokumencie są odnotowane w historii zmian dokumentu zamieszczonej powyżej.

1.WPROWADZENIE

a) **CEL**

Dokument ten ma za zadanie sprecyzować sposób realizowanych prac. Określić założenia projektu, standardy, narzędzia i komponenty wchodzące w skład implementacji oraz opis realizacji tych komponentów.



b) ZAKRES

Założeniem projektu jest stworzenie gry komputerowej na podstawie gry Eurobiznes. Aplikacja ma być lokalna przeznaczona dla wielu graczy (od 2 do 4). Funkcje w grze które będą dostępne to możliwość wybrania ile razy losujemy wynik ruchu., przesunięcie pionka, nabycie pola, nabycie laboratorium, płacenie rachunków, skończenie rundy

c) OMÓWIENIE

Dokument ten powstał na bazie specyfikacji wymagań systemowych. Zawiera on definicje standardów, strategii i konwencji, które będą przestrzegane podczas realizacji projektu. Dalsza część dokumentu zawiera informacje o modułach i komponentach systemu i interfejsie graficznym aplikacji.

2.STANDARDY PROJEKTU, KONWENCJE, PROCEDURY

a) STANDARDY PROJEKTOWE

Podczas tworzenia projektu wykorzystaliśmy model przyrostowy tworzenia oprogramowania. Wybrany został ze względu na:

- Częsty kontakt z klientem
- Mniejsze ryzyko porażki całkowitej przedsięwzięcia
- Usługi o najwyższym priorytecie są dostarczane jako pierwsze
- Wczesne wykorzystanie przez klientów fragmentów systemu

b) STANDARDY DOKUMENTACYJNE

Wszystkie dokumenty projektu są tworzone na podstawie jednego firmowego szablonu.

c) KONWENCJE NAZEWNICZE

Nazewnictwo w projekcie jest ukierunkowane na prostotę i jednoznaczność. W implementacji została zachowana konwencja nazewnictwa Java. Pierwsze litery komponentu, twarda spacja i przeznaczenie np. `jta_NazwaGracza`

d) STANDARDY PROGRAMISTYCZNE

W projekcie będziemy wykorzystywać podejście obiektowe do programowania. Będziemy wykorzystywać wzorzec projektowy MVC(Model-View Controller). Zalety takiego podejścia:

- Brak zależności modelu od widoków aplikacji
- Łatwość dodawania oraz modyfikowania istniejących widoków bez wpływu na kluczową część systemu



e) NARZĘDZIA

Do realizacji projektu wykorzystamy język Java. Będziemy korzystać ze środowiska RAD(Rapid Application Development) w postaci NetBeans. Podczas projektowania będziemy korzystać z GitHub'a do środowiska Eclipse dzięki niemu będziemy mieć możliwość dostępu do systemu kontroli wersji Git z poziomu Desktopa. Podczas tworzenia dokumentacji, będziemy wykorzystywać:

- Visual Paradigm – program do tworzenia diagramów UML
- Microsoft Office w wersji 2007 – wykorzystany przy tworzeniu dokumentacji
- GanttProject – program do tworzenia harmonogramów
- EasyUml– plugin do NetBensa umożliwiając generowanie diagramu klas

f) SPECYFIKACJA KOMPONENTÓW

Engine – Klasa zarządzająca wszystkimi operacjami na danych jest to silnik naszej gry.

Card – Klasa opisująca pojedynczą kartę zawierająca metody ułatwiające dostęp do tych pól.

Player – Klasa zawierająca wszystkie dane na temat gracza.

MainWindow – Klasa która jest naszym głównym oknem są tu ustawione wszystkie przyciski i wizualna reprezentacja danych z trzech powyższych klas

PlayerHand – Jest to komponent który służy do graficznego prezentowania aktualnego stanu kart gracza.

CardTooltip – Komponent stworzony do graficznego przedstawienia danych o polach na planszy

Grid – Komponent stworzony do graficznej reprezentacji naszej planszy dzięki niemu widzimy jaki gracz gdzie aktualnie się znajduje

12. Podsumowanie projektu

Celem projektu było stworzenie złożonej aplikacji desktopowej. Pierwszą częścią było sporządzenie dokumentacji opierającej się na zamyśle skonstruowania aplikacji. Kolejnym etapem naszych zmagania w celu zoptymalizowania programu zostały przeprowadzone testy jednostkowe. Podczas testów nie wystąpiły problemy. Następnie poprawiono błędy w dokumentacji, lecz nie miały one większego znaczenia w kontekście całego projektu. Etap zwany testem użytkownika polegał na udostępnieniu wstępnej wersji aplikacji kilku użytkownikom w celu sprawdzenia czy nie wystąpią kolejne, niespodziewane przez zespół błędy programistyczne. Po odbyciu tego kroku aplikację przedstawiono klientowi wraz z pełną dokumentacją. Wszystkie założenia projektowe zostały w pełni zrealizowane i po rozmowie z przedstawicielem firmy naszego partnera biznesowego program został w pełni zaakceptowany.



W planach na przyszłość jest rozwinięcie programu, aby działał na wszystkich systemach operacyjnych proponowanych na komputery stacjonarne oraz laptopy.

13. *Źródło informacji*

Tworząc naszą aplikację oraz dokumentację zespół projektowy korzystał z informacji z źródeł takich jak:

- Serwis stackoverflow
- Witryna internetowa oryginalnej gry „Monopoly”
- Książka: „Java. Podstawy.” Wydanie VIII Autorzy: Cay S. Horstmann, Gary Cornell
- Książka: „Inżynieria oprogramowania” Autor: Andrzej Jaskiewicz
- Książka: „UML. Inżynieria oprogramowania.” Wydanie II Autor: Perdita Stevens
- Poradniki związane z tworzeniem diagramów
- Poradniki związane z tworzeniem harmonogramu projektu
- Strona internetowa z pomocą do obsługi GitHuba