**Gra planszowa Monopoly**

**Opis gry**:

Każdy gracz przed startem otrzymuje określoną sumę pieniędzy. Grę rozpoczyna się z pola START, następnie każdy z graczy wykonuje rzut dwiema kostkami. Suma rzutu oznacza ilość pól o które gracz poruszy się po planszy. Jeśli pole, na którym znalazł się gracz nie należy jeszcze do nikogo i jest nieruchomością możliwą do kupienia, gracz ma możliwość zakupu tego pola za określoną kwotę. Następnie gracze, którzy staną na tym polu będą musieli zapłacić mu czynsz. Jeśli gracz skompletuje wszystkie pola danego koloru czynsz na każdym z tych pól się podwoi. Dodatkowo gracz otrzyma wówczas możliwość kupa domów (maksymalnie 4 na pole) oraz hotelu. Koszt budowy domu/hotelu na danym polu jest podany obok ceny kupna pola. Każdy dom/hotel podwaja wysokość czynszu. Jeśli gracz stanie na polu, którego czynsz przewyższa ilość jego pieniędzy do końca tury ma szanse sprzedać część swoich inwestycji, by być w stanie zapłacić czynsz. Jeśli mu się to nie uda bankrutuje i traci możliwość dalszej gry. Wszystkie jego nieruchomości oraz pieniądze zostają przekazane graczowi, na którego polu zbankrutował. Jeśli gracz stanie na polu „Go to Jail” zostaje przeniesiony do więzienia, z którego wychodzi po 3 turach, lub po wyrzuceniu takiej samej wartości na obu kostkach. Za każde pełne okrążenie planszy gracz otrzymuje 200 jednostek waluty. Grę wygrywa osoba, która pozostanie jako ostatnia, czyli gdy wszyscy jej przeciwnicy zbankrutują.

**Realizacja projektu:**

Projekt został napisany w języku Python (wersja 3.6.5) przy pomocy interfejsu graficznego TKinter.

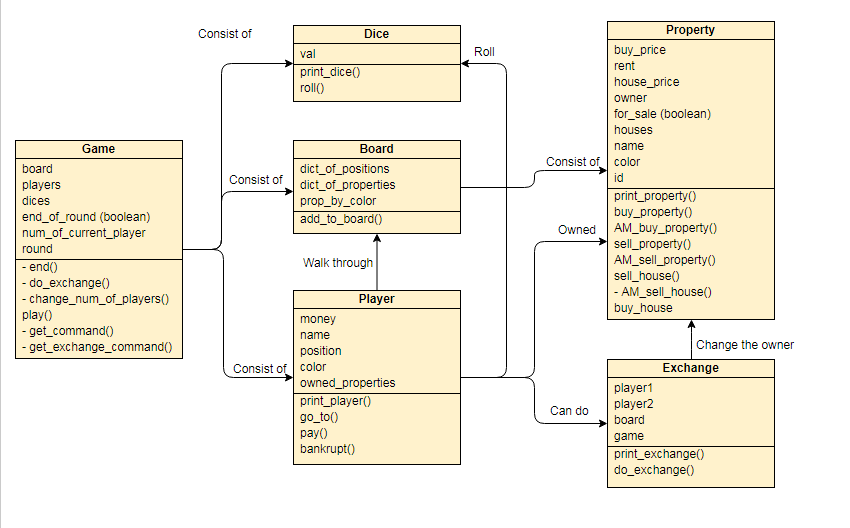
Pola reprezentowane są jako prostokąty. Na górze znajduje się ich nazwa oraz kolor tła oznaczający typ pola. Poniżej znajduje się nazwa właściciela danego pola z kolorem tła odpowiadającym kolorowi pionka tego gracza. Następnie znajduje się informacja o cenach odpowiednio: kupna pola, kupna domu/hotelu na tym polu, wysokości czynszu. W 2 ostatnich wierszach wyświetlane są domy/hotele obecne na tym polu oraz wyświetlane są pionki graczy stojących na tym polu.

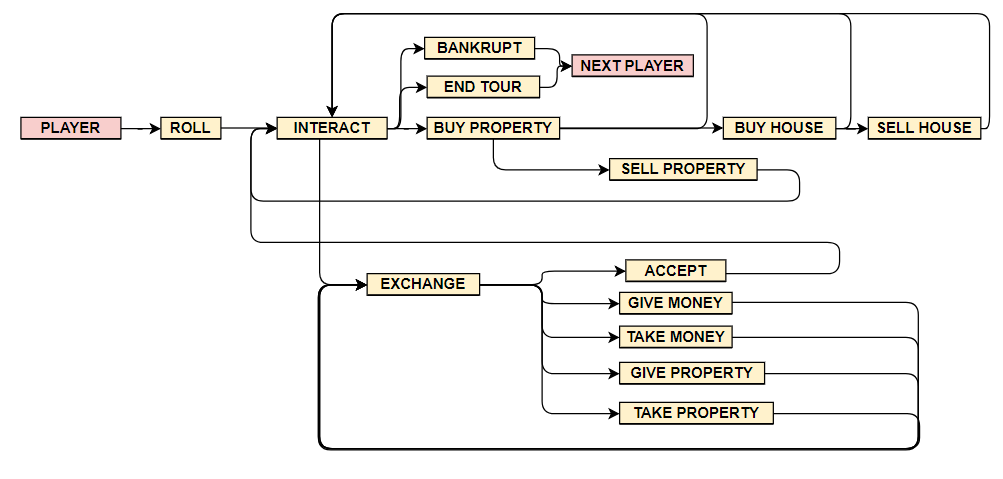
Gracz steruje za pomocą przycisków pojawiających się w odpowiednich momentach w środku planszy. Informacje o ilości pieniędzy poszczególnego gracza, o tym, który gracza ma aktualnie swoją turę oraz o wartości rzutów kostkami również wyświetlane są w środku planszy.

**Lista klas wraz z krótkim opisem:**

* **Property -** główna klasa reprezentująca pole. Jej podklasy odpowiadają polom danego typu:
  + **Specjal property –** pole nie przeznaczone na sprzedaż, np. START
    - **Go to Jail**
  + **Brown property**
  + **Light blue property**
  + **Pink property**
  + **Orange property**
  + **Red property**
  + **Yellow property**
  + **Green property**
  + **Dark blue property**
  + **Railway**
* **Player -** klasa reprezentująca gracza
* **Board –** klasa reprezentująca plansze, czyli zbiór pól w dwóch słownikach zapewniających szybki dostęp zarówno do pole o danym numerze, jak i numeru danego pola
* **Dice –** klasa kostki do rzucania oparta na pseudolosowości
* **Exchange –** klasa umożliwiająca wymianę oraz handel posiadłościami pomiędzy graczami
* **Game –** główna klasa odpowiadająca za całą rozgrywkę, zawierającą listę graczy, planszę oraz odpowiadająca za główną pętlę gry

**Diagram klas:**

****

**Funkcjonalność: **

**Wskazówki dla użytkowników:**

Nową grę tworzymy jako nową instancję klasy Game. Jedynym argumentem, jaki przyjmuje klasa Game jest lista graczy, których chcemy dodać do gry. Graczy, których planujemy dodać reprezentujemy jako dwuelementowe listy, gdzie pierwszy element to nazwa gracza (string), a drugi to kolor pionka gracza podany w języku angielskim (string). Maksymalna liczba użytkowników to 5 (ograniczenie jest spowodowane wyświetlaniem pionków na planszy). Grę rozpoczyna się wywołaniem metody play() na naszej instancji klasy Gra.

Użytkownik powinien postępować zgodne z diagramem Funkcjonalności. Wszelkie interakcje podejmuje się za pomocą przycisków.

**Wskazówki dla programistów:**

* **Rozwój:** Gra może zostać rozwinięta o następne typy pól poprzez dodanie nowych podklas dziedziczących po klasie Property lub Special property. Nowo utworzone pola można dodać do planszy w jej konstruktorze. Rozmiar planszy oraz rozstawienie pól można dowolnie edytować dostosowując do pożądanego poziomu trudności. Innym pomysłem na rozbudową projektu jest dodanie interfejsu` służącego do dodawania nowych graczy na początku nowej gry (wraz z możliwością wyboru koloru i losowaniem kolejności tur). Można również znieść limit graczy dodając blokadę wyświetlania większej ilości pionków na polu, niż 5.
* **Ponowne wykorzystanie:** Klasa Property wraz z podklasami oraz klasa Board może zostać wykorzystana do stworzenia gry o podobnych zasadach, jak Monopoly (np. Eurobiznes). Podobne zastosowania mogą mieć klasy Player oraz Exchange. Klasę Dice można wykorzystać w dowolnej grze planszowej.

Mateusz Markiewicz

298653