Dokumentacja projektowa

1. Opis projektu:
   Celem projektu było stworzenie prostej gry tekstowej, której motywem przewodnim jest wystrzeliwanie rakiet w kosmos, sterowanie nimi oraz kupowanie ulepszeń za zdobyte pieniądze.

2. Opis funkcjonalności:
   - Możliwość zmiany czcionki – wchodząc w ustawienia, mamy możliwość zmniejszyć lub zwiększyć czcionkę, aby dopasować wielkość okna gry do wielkości ekranu komputera
   - Sklep – przed startem rakiety mamy możliwość udania się do sklepu przy pomocy przycisku „Tab”, gdzie za zdobytą podczas gry walutę można dokonać ulepszeń rakiety
   - Sterowanie rakietą – aby rozpocząć grę, należy nacisnąć strzałkę w górę na klawiaturze, strzałki w lewo i prawo umożliwiają nawigowanie rakietą na boki, strzałka w górę – strzelanie z działka pokładowego
   - Prosta ekonomia – gracz otrzymuje walutę za każdy lot:
     - 0,5$ - za każdy pokonany wiersz mapy (tekstu) poniżej aktualnego rekordu
     - 1$ - za każdy wiersz mapy (tekstu) powyżej aktualnego rekordu
     - 20$ - jednorazowa nagroda za zestrzelenie samolotu, helikoptera lub sterowca
     - 100$ - jednorazowa nagroda za zestrzelenie spodka UFO

3. Szczególnie interesujące zagadnienia projektowe:
   - Wątek odpowiedzialny za sterowanie rakietą czasami nie zamykał się po zakończeniu/zapauzowaniu gry, przez co utrudniał on poruszanie się po menu (nasłuchiwał przyciskanych klawiszy), spowalniał również działanie aplikacji. Rozwiązaniem problemu jest flaga, którą wątek główny ustawia na wartość „true”, gdy rakieta zakończy lot.

     Pętla w wątku obsługującym sterowanie rakietą:

```
while (!Game.gameover && !Game.paused)
```

     Warunek gwarantujący wyjście z pętli, gdy zmienna flag = true:

```
if(flag){
    break;
}
```

Warunek gwarantujący zatrzymanie wątku obsługującego sterowanie rakietą, podczas pauzy:

```
else{
    mh.setFlag();
    this.pause();
    mh = new MoveHadler(screen, rocket, tg);
    mh.start();
```

- Funkcja sprawdzająca kolizję rakiety z przeszkodą (ptaki, samolot, ufo, etc.) napisana została w sposób, który umożliwia zmianę lub dodanie nowych ASCII Artów obiektów, zmianę ASCII Arta rakiety, bez edycji kodu.

```
public static boolean checkColisions(Rocket rocket, Obstacle obstacle){
    if(     !obstacle.isTransparent() &&
            ((rocket.getColumn()<(int)obstacle.getX()+obstacle.getArt().art[0].length()
            && rocket.getColumn()+Arts.ROCKET_FAST.art[0].length()> (int)obstacle.getX())
            ||
            (rocket.getColumn()>(int)obstacle.getX()+obstacle.getArt().art[0].length()
            && rocket.getColumn()+Arts.ROCKET_FAST.art[0].length()<(int)obstacle.getX())
            ||
            (rocket.getColumn() >= (int)obstacle.getX()
                    && rocket.getColumn()+Arts.ROCKET_FAST.art[0].length() <= (int)obstacle.getX()+obstacle.getArt().art[0].length()))
            &&
            obstacle.getY() > Menu.frameHeight-Arts.ROCKET_FAST.art.length-obstacle.getArt().art.length
            ){
        obstacle.collisionEffect(rocket);
        return true;
    }
    return false;
}
```

- Kolor czcionki pobrany przy pomocy polecenia TextColor.factory.fromString(„WHITE”) wbrew nazwie nie jest biały. Jest on jasnym odcieniem szarości. Co utrudniło rysowanie planszy.

Deklaracje kolorów:

```
private final TextColor gray = TextColor.Factory.fromString( value: "white");
private final TextColor white = new TextColor.RGB( r: 255, g: 255, b: 255);
private final TextColor black = new TextColor.RGB( r: 0, g: 0, b: 0);
```

- Przed każdym lotem generowana jest mapa z unikalnym ułożeniem gwiazd, chmur oraz przeszkód.

  Część funkcji odpowiedzialnej za generowanie nowej mapy.
  Na wygenerowanie chmury jest 0,009% (zgodnie z poniższym screenshotem) oraz po 33% na każdy jej warrant:

```
int randNum = rand.nextInt( bound: 10000);
switch (randNum) {
    case 500, 501, 502, 503, 504, 505, 506, 507, 508 -> {
        if (i >= Menu.frameHeight) {
            if (randNum == 500 || randNum == 501 || randNum == 506)
                Game.getObstacles().add(new Cloud(Arts.CLOUD1, j, i));
            else if (randNum == 502 || randNum == 503 || randNum == 507)
                Game.getObstacles().add(new Cloud(Arts.CLOUD2, j, i));
            else {
                Game.getObstacles().add(new Cloud(Arts.CLOUD3, j, i));
            }
        }
        temp.append(' ');
    }
```

  Najpierw losowana jest liczba z przedziału 0->9999, następnie w zależności jaki będzie wynik generowana jest jakaś struktura: przeciwnik, chmura lub gwiazda/puste pole. Struktury zapisywane są do listy, wraz z ich położeniem, a drukowane jest puste pole.

4. Instrukcja użytkowania:
   - Po przejściu do gry:
     - Przycisk Tab - przejście do sklepu:
       - Strzałka w górę/strzałka w dół – wybór ulepszenia
       - Enter – zakup ulepszenia, pod warunkiem, że pozwala na to stan konta, koszt każdego ulepszenia, to iloczyn 100 i poziomu ulepszenia (np. jeśli posiada się szybkostrzelność 3 poziomu, to zakup 4 poziomu będzie kosztować 300$)
       - Tab – powrót do rakiety
     - Strzałka w górę – start rakiety:
       - Strzałka w lewo/strzałka w prawo – manipulacja rakietą na boki
       - Strzałka w górę – strzał z działka pokładowego (liczba strzałów na minutę zależna od poziomu szybkostrzelności)
       - Esc – pauza, możliwość powrotu do gry lub do menu głównego
     - Esc - Powrót do menu głównego

- Menu:
    - Start/Kontynuuj – umożliwia przejście do gry, jeśli gra była wcześniej rozpoczęta, to wczytuje poprzedni stan
    - Nowa gra – umożliwia przejście o gry, rozpoczyna grę od początku
    - Opcje – przejście do opcji gry:
        - Wielkość czcionki – po naciśnięciu na klawiaturze przycisku Enter, umożliwia zmianę czcionki, strzałka w lewo – mniejsza czcionka, strzałka w prawo – większa czcionka
        - Powrót – powrót do menu głównego
    - Wyjście – wyjście z gry

- Ulepszenia rakiety:
    - Bak paliwa – wyższy poziom tej umiejętności umożliwia dłuższy lot
    - Siła ognia – wyższy poziom tej umiejętności sprawia, że pocisk zadaje przeciwnikom wyższe obrażenia
    - Pancerz – wyższy poziom tej umiejętności poprawia odporność na zderzenia z przeciwnikami, pociski przeciwników
    - Szybkostrzelność - wyższy poziom tej umiejętności zwiększa szybkość wystrzeliwania kolejnych pocisków

- Zdobywanie waluty – gracz otrzymuje walutę za każdy lot:
    - 0,5$ - za każdy pokonany wiersz mapy (tekstu) poniżej aktualnego rekordu
    - 1$ - za każdy wiersz mapy (tekstu) powyżej aktualnego rekordu
    - 20$ - jednorazowa nagroda za zestrzelenie samolotu, helikoptera lub sterowca
    - 100$ - jednorazowa nagroda za zestrzelenie spodka UFO

5. Wnioski:
    - Stworzenie ładnych wizualnie ASCII Artów jest zadaniem bardzo czasochłonnym.
    - Płynność animowanych ASCII Artów pozostawia wiele do życzenia.
    - Wygodny interfejs graficzny jest prostszy do stworzenia niż wygodny interfejs tekstowy
    - Gra konsolowa jest dużym wyzwaniem pod względem atrakcyjności wizualnej.

6. Samoocena:
   Stworzoną przez siebie aplikację oceniam na 5.