

KOSZALIN, 22.01.2015 r.

Dokumentacja końcowa projektu

Temat: KwadrateX

Zespół projektowy: Paweł Biezuński

Łukasz Flak

Tomasz Barański

SPIS TREŚCI:

- 1 Wybór grupy projektowej
- 2 Wybór tematu
- 3 Protokół założycielski, kary, obowiązki
- 4 Zakres projektu
- 5 Opis funkcji
- 6 Mapa myśli
- 7 Mapa koncepcyjna
- 8 Identyfikacja aktorów
- 9 Diagram przypadków użycia
- 10 Specyfikacja funkcji
- 11 Diagram klas UML
- 12 Diagramy sekwencji
- 13 Diagramy przepływu danych
- 14 Szczegółowy opis przypadków użycia
- 15 GUI
- 16 Typy okien
- 17 Rozmiary/kolory/czcionki
- 18 Opis obszarów okien
- 19 Implementacja 5 funkcji
- 20 Niezawodność
- 21 Plan testów
- 22 Testy
- 23 Podsumowanie testów
- 24 Instrukcja użytkownika
- 25 Raport dla klienta
- 26 Narzędzia programistyczne

1.Wybór grupy projektowej:

Skład grupy: Łukasz Flak, Tomasz Barański, Paweł Biezuński

2.Wybór tematu

KwadratX to gra na podstawie znanej japońskiej zabawki Tamagotchi.

Cel projektu - napisanie aplikacji KwadratX, dzięki której można uzyskać zaliczenie z przedmiotu Inżynieria Oprogramowania.

3.PROTOKÓŁ ZAŁOŻYCIELSKI

W dniu 19.10.2013 roku w Koszalinie odbyło się zebranie osób mających na celu utworzenie grupy projektowej.

Obecni na zebraniu:

Paweł Biezuński

Łukasz Flak

Tomasz Barański

po zapoznaniu się z potrzebami oraz wymaganiami klienta, ustalili że:

§ 1 Grupa projektowa składa się z trzech członków:

Paweł Biezuński

Łukasz Flak

Tomasz Barański

§ 1.1 Poszczególnym członkom zostają przydzielone następujące role:

Łukasz Flak – kierownik projektu, programista

Tomasz Barański - programista

Paweł Biezuński - dokumentalista, tester

§ 1.2 Do obowiązków poszczególnych członków należy:

Kierownik projektu:

- kontrolowanie postępów pracy i ustalanie harmonogramu zadań
- przydzielanie zadań projektowych

Programista:

- pisanie kodu programu
- korekta błędów wykrytych przez testera

Dokumentalista:

- tworzenie dokumentacji projektu
- obsługa repozytorium

Tester:

- testowanie poszczególnych modułów aplikacji i wykrywanie błędów
- optymalizacja kodu programu

§ 2 Zespół zostanie rozwiązany po zakończeniu projektu w terminie przewidywanym na 06.01.2015 r.

§ 2.1 Projekt będzie realizowany w oparciu o standardy inżynierii oprogramowania.

§ 2.2 Za zakończenie realizacji projektu uznaje się ukończenie wszystkich etapów projektowania i wdrożenie aplikacji u klienta.

§ 2.3 W przypadku nie ukończenia projektu w przewidywanym terminie ustalony zostaje termin ostateczny na 20.01.2015 r.

§ 3 Celem projektu jest stworzenie aplikacji o nazwie **KwadrataX**, będącej odwzorowaniem popularnej w Japonii zabawki Tamagotchi.

§ 3.1 Aplikacja wstępnie jest realizowana na sprzedaż jako gra dla dzieci w wieku szkolnym, rozwijające zachowania opiekuńcze, regularność, odpowiedzialność itp.

§ 3.2 Aplikacja będzie realizowana w języku Java, przy użyciu środowiska Eclipse.

§ 3.3 Repozytorium aplikacji znajduje się na serwerze github.com pod nazwą **KwadrataX**.

§ 4 W celu utrzymania dyscypliny w zespole ustalony zostaje kar dyscyplinarnych:

§ 4.1 Za zachowania utrudniające realizację projektu, przede wszystkim nie wywiązywanie się w terminie z powierzonego przez kierownika projektu zadania bez usprawiedliwionej przyczyny, członek grupy otrzymuje **Ostrzeżenie**.

§ 4.2 Za zachowania nie mające wpływu na realizację projektu, jednak wprowadzające nieodpowiednią atmosferę w zespole, członek grupy otrzymuje **Upomnienie**.

§ 4.3 Otrzymanie **3 Upomnień** jest równoważne z otrzymaniem **Ostrzeżenia**.

§ 4.4 Członek grupy, który otrzymał już Ostrzeżenie (3 Upomnienia) w przypadku kolejnej niesubordynacji może zostać usunięty z grupy projektowej.

§ 4.5 Ostateczną decyzję o Upomnieniach, Ostrzeżeniach i ewentualnym wykluczeniu z zespołu podejmuje kierownik projektu po konsultacji z resztą grupy projektowej.

§ 5 Ważne decyzje będą podejmowane podczas spotkań za pomocą demokratycznego głosowania.

§ 5.1 Mniej istotne sprawy będą rozstrzygane drogą e-mailową lub na forum grupy.

§ 5.2 Każdy członek grupy ma prawo zgłaszać swoje uwagi odnośnie wykonywanych zadań (zarówno swoich jak i pozostałych członków grupy) do kierownika projektu, który z kolei zobowiązany jest poddać owe uwagi pod zespołową dyskusję.

4. OPIS FUNKCJI

- **KARMIE NIE** - do dyspozycji mamy kilka potraw, oraz przekąski i słodycze. To, czy i jak bardzo nasz ulubieniec jest głodny, sprawdzamy na wyświetlaczu, po czym decydujemy czym nakarmić naszego zwierzaka. Pamiętajmy, że nadmiar słodczy źle wpływa na wagę i figurę oraz może doprowadzić do bólu zębów, co bardzo psuje humor Tamagotchi. Jednak od czasu do czasu deser lodowy lub czekoladka wprowadza wirtualne zwierzątko w dobry nastrój.
- **ZAKUPY** - w wirtualnym sklepie znajdziemy wszystko czego może potrzebować Tamagotchi. Są tam wyszukane potrawy i przekąski, zabawki, gadżety i niespodzianki. Zapłacimy za nie punktami zdobytymi w grach.
- **SPRZĄTANIE** - Tamagotchi dba o higienę sam. Nie raz zaskoczy nas siedząc w wannie, czy myjąc zęby. Mimo to, o jego otoczenie należy dbać regularnie - jeśli tego nie zrobimy czeka nas niemiła niespodzianka.
- **GRY I ZABAWY** - aby uprzyjemnić czas Tamagotchi, można zagrać w jedną z kilku prostych gier, które mają też inną ważną funkcję. Za zebrane podczas zabawy Kwadra-punkty, możemy robić zakupy w sklepiku.
- **LECZENIE** - gdy nasz Tamagotchi źle się czuje, daje nam o tym znać. Pokazuje się wtedy koło niego ząb albo czaszka. Korzystamy wtedy z apteczki zawierającej leki na wszelkie dolegliwości.
- **KOMUNIKACJA** - to wirtualne zwierzątko, za pomocą wifii, potrafi nawiązać kontakty, przyjaźnie, a nawet zakochać się w innym KwadracieXami. Zwierzaki mogą odwiedzać się, przynosić sobie upominki i bawić się razem. Ta funkcja jest niezwykle istotna, gdyż zakochane KwadratyXy mogą

mieć potomstwo. Prowadzi to do powstania kolejnych generacji Kwadratów, które są zapisywane w specjalnym drzewie genealogicznym.

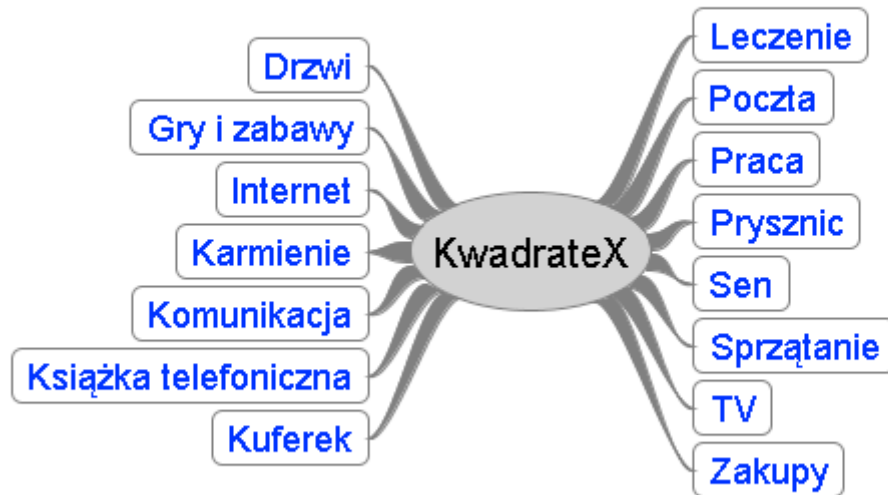
- **INTERNET** - wizytę w KwadraTown umożliwia nam specjalny kod. Po zalogowaniu się, możemy odwiedzić centrum handlowe i zrobić duże zakupy (kupione rzeczy pojawią się na wyświetlaczu naszego Kwadrata), pójść do kina i obejrzeć jeden z aktualnie wyświetlanych filmów, odwiedzić salon gier, by wygrać upominki, a nawet zwiedzić świat korzystając z ofert biura podróży. Możliwości zabawy są niemal nieograniczone.
- **SEN** - dzięki wbudowanemu zegarowi Kwadrat odróżnia dzień od nocy i wie, kiedy powinien pójść spać. Nie zapomnijmy zgasić mu światła na noc, bo obudzi się zmęczony i niewyspany.
- **PRACA** - Tamagotchi chodzi do przedszkola, szkoły podstawowej, a jako osoba dorosła - rozpoczyna pracę. Gdy dorosnie będziesz mógł grać w mini gry związane tematycznie z wykonywanym przez niego zawodem, a wyniki jakie osiągniesz podczas zabawy będą miały wpływ na wysokość zarobków oraz ilość punktów życiowych Twojego podopiecznego
- **POCZTA** - ciasteczka szczęścia, jakie otrzymuje Tamagotchi są pełne niespodzianek.
- **KSIĄŻKA TELEFONICZNA** - zawiera listę przyjaciół Kwadrata oraz historię jego rodziny
- **KUFEREK** - zawiera pamiątki i itemy
- **TV** - dostępne są trzy kanały - telezakupy, kanał randkowy i podróżniczy.
- **PRYSZNIC** - Tamagotchi może się umyć
- **DRZWI** - nowa funkcja, dzięki której Tamagotchi wychodzi do miasta lub do sklepu.

5.Opis funkcjonalny:

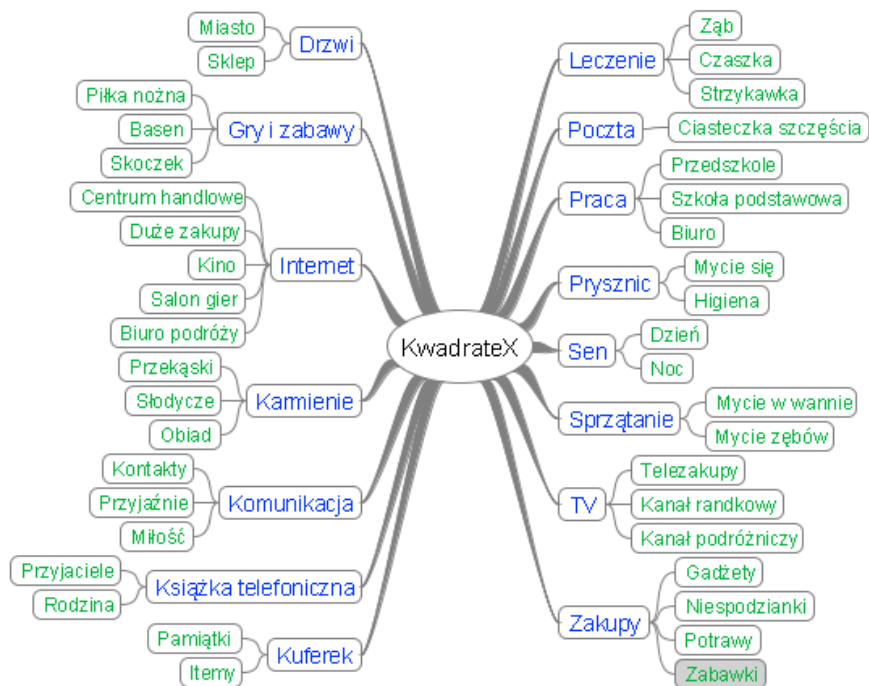
- **KARMIENIE**
- -wybór pokarmu
- -poprawienie kondycji paska życia
- **ZAKUPY**
- -zakup przedmiotów
- -wymiana z przyjaciółmi przedmiotów
- -wydawanie zgromadzonej gotówki
- **SPRZĄTANIE**
- -mycie stworka
- -poprawa kondycji paska higieny
- **GRY I ZABAWY**
- -zbieranie kwiatków
- -podskakiwanie ponad przeszkodami

- **-poprawa kondycji paska zadowolenie**
- **LECZENIE**
- -użycie strzykawki w celu wyleczenia stworka
- -poprawa kondycji paska zdrowia
- **KOMUNIKACJA -**
- -nawiązanie kontaktów z innymi użytkownikami
- -wymiana upominków między stworkami
- -możliwość spłodzenia potomka :D
- **SEN**
- użycie funkcji łóżka, aby by położyć stworka spać
- -poprawa kondycji paska sen
- **PRACA -**
- -zarabianie pieniędzy
- **POCZTA** - ciasteczka szczęścia, jakie otrzymuje Tamagotchi są pełne niespodzianek.
- **KSIĄŻKA TELEFONICZNA** - zawiera listę przyjaciół KwadrataXa oraz historię jego rodziny
- **KUFEREK** - zawiera pamiątki

6. Mapa myśli



7. Mapa konceptualna

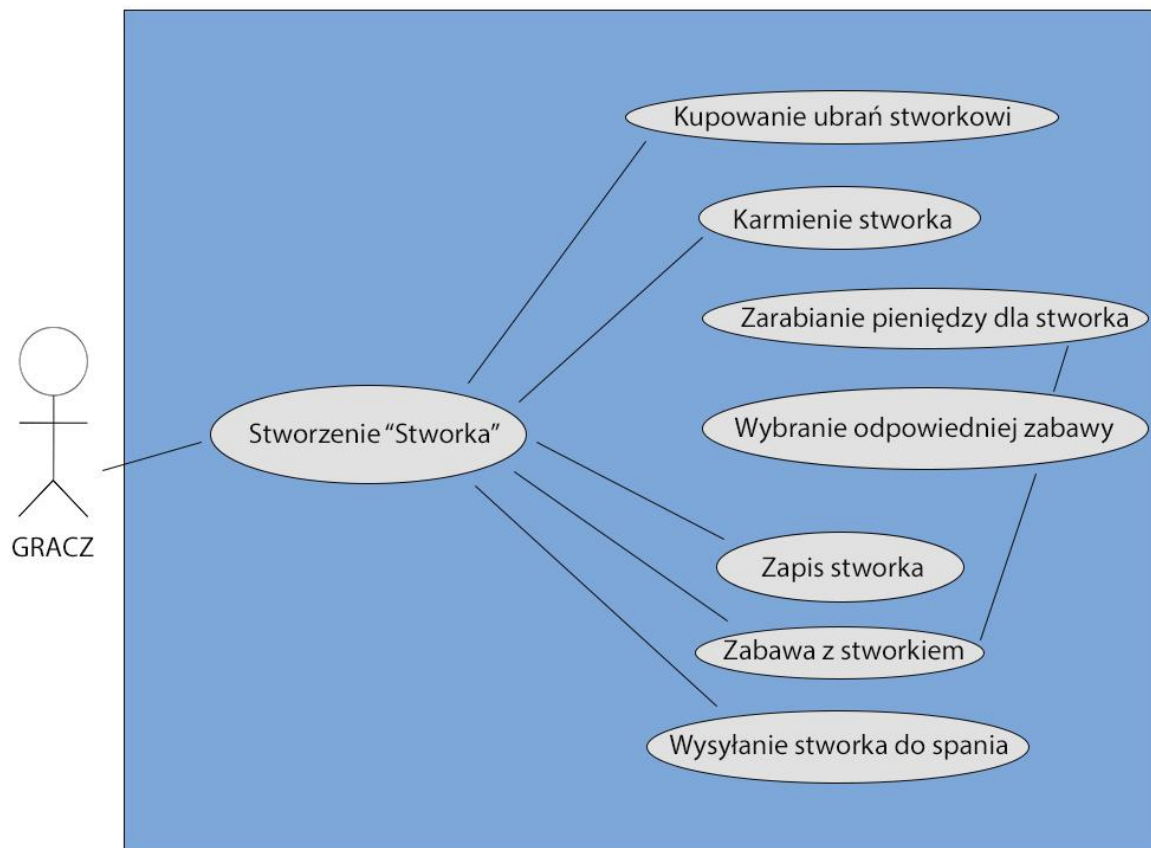


8.Identyfikacja aktorów

Nazwa : Gracz

Krótki opis: Osoba grająca w grę KwadrateX

9.Diagram przypadków użycia



10.Specyfikacje funkcji

public void obslugaBlednejPozycji()

Funkcja służy do sprawdzania czy dany Component jest w obszarze widzialnym dla użytkownika. Jeśli Component będzie poza obszarem widzialnym, wtedy na konsoli zobaczymy uwagę jaki Component jest poza obszarem i ile wystaje px poza widzialny obszar, aby programista mógł poprawić pozycje Componentu

public void ustawRozmiarKomponentu()

Funkcja, będzie ustawiała rozmiar odpowiedniego Componentu za pomocą pól składowych w klasie

public void pobierzObrazek()

Funkcja będzie pobierała obrazek z strumienia IO i zapisywała go pod odpowiednie pole składowe w Componencie, aby ten mógł później skorzystać z tego obrazka

public void wczytaj_font()

Funkcja jest odpowiedzialna za wczytanie odpowiedniej czcionki dla Componentu. Pobrana czcionka będzie zapisana w polu składowym klasy aby później łatwiej ją pobrać

public void uruchom_najechanie_na_przycisk_akcja()

Funkcja będzie wykonywała zmianę koloru tekstu umieszczonego w odpowiednim Componencie. Użytkownik który "najedzie" myszką na obszar Componentu, spowoduje zmianę koloru czcionki tekstu Componentu

public void ustawPozycjeKomponentu()

Funkcja ustawia Component na odpowiedniej pozycji w obiekcie JLabel. Będzie to wykonywała za pomocą parametrów podanych w konstruktorze klasy.

public void stworzEkranGlowny()

Za pomocą tej funkcji będziemy tworzyć ekran główny odpowiedniego okna(widoku dla użytkownika). Funkcja ta będzie umieszczała wszystkie Componenty na swoim miejscu i będzie wyświetlać Ekran(setVisible(true)) dla użytkownika.

public void dodajKomponentyEkranuGlownego()

Funkcja będzie wywoływać dwie metody które będą tworzyć przyciski i grafikę na Ekranie użytkownika

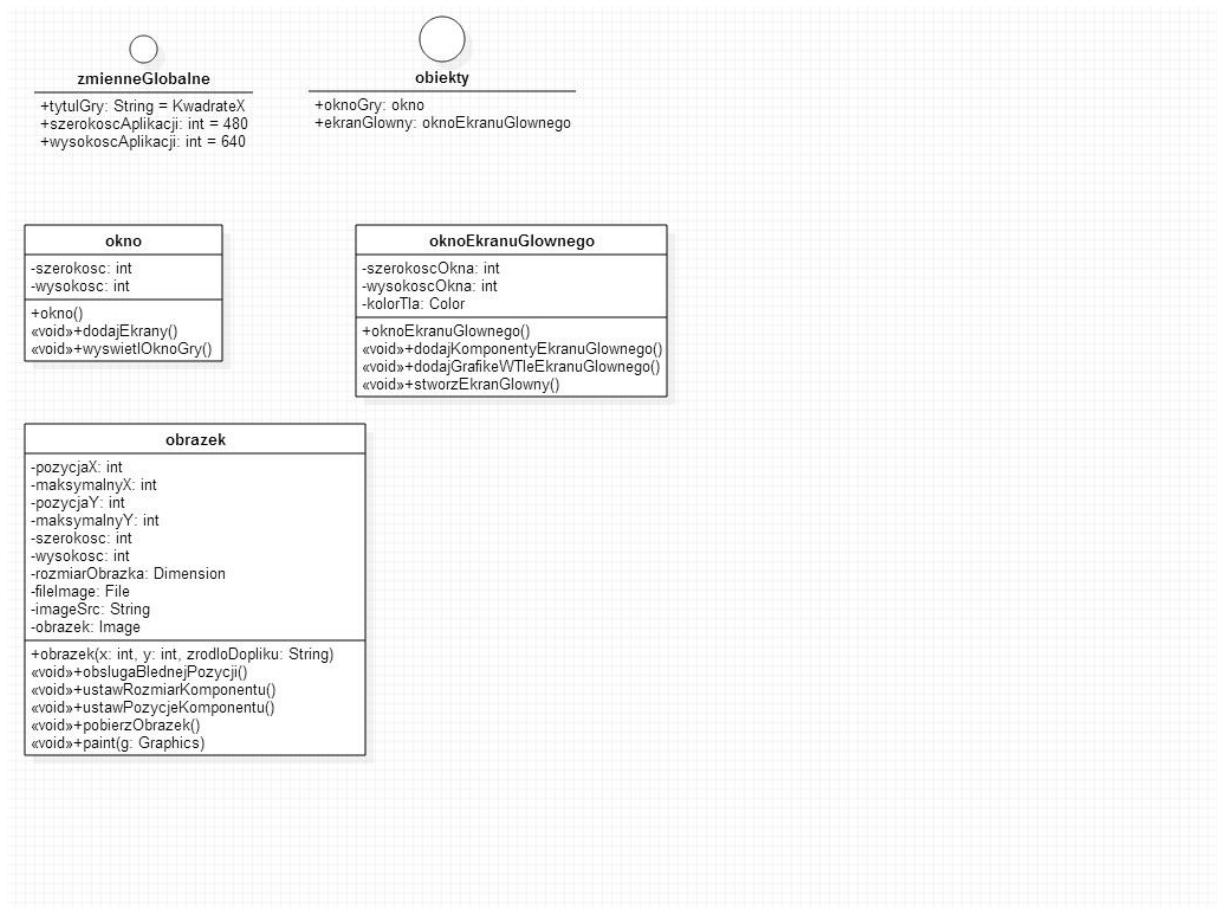
public void dodajPrzyciskiEkranuGlownego()

Za pomocą tej metody będziemy dodawać do odpowiedniego ekranu Componenty przycisków

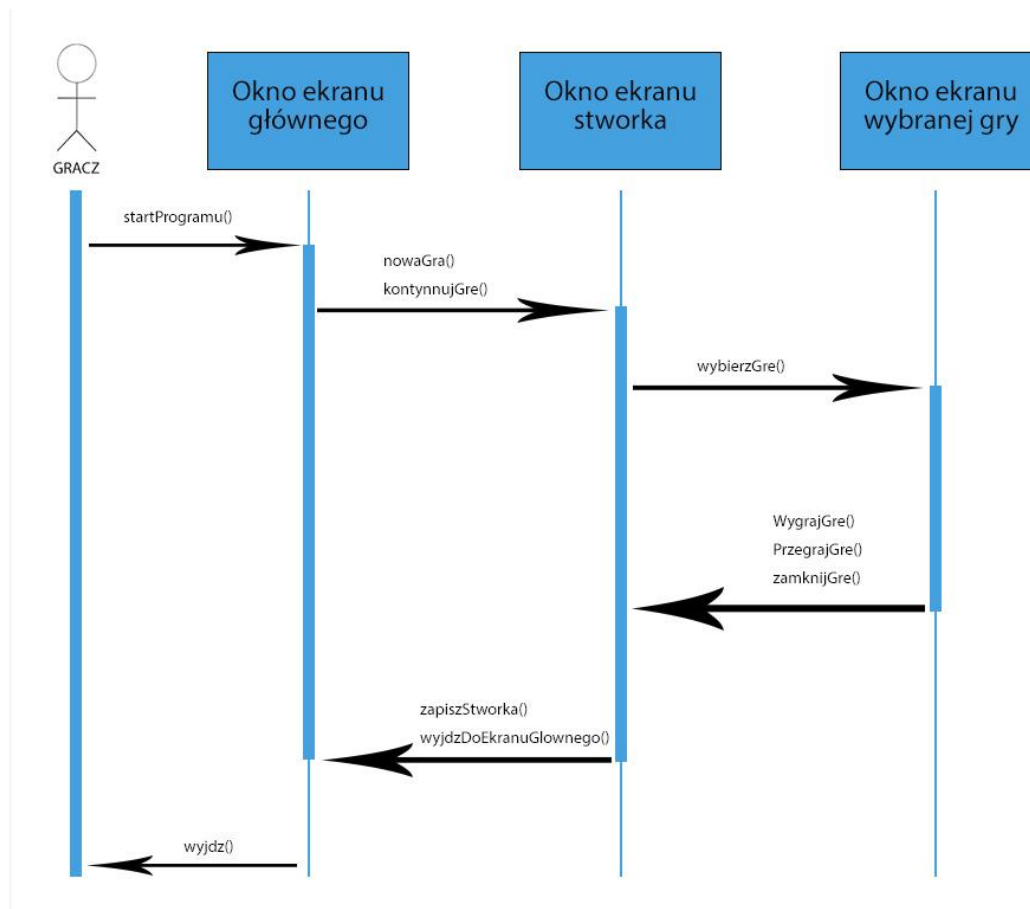
public void dodajGrafikeWTleEkranuGlownego()

Ta funkcja będzie dodawać na ekran Componenty grafiki widocznej dla użytkownika

11. Diagram klas UML



12. Diagram przepływu informacji

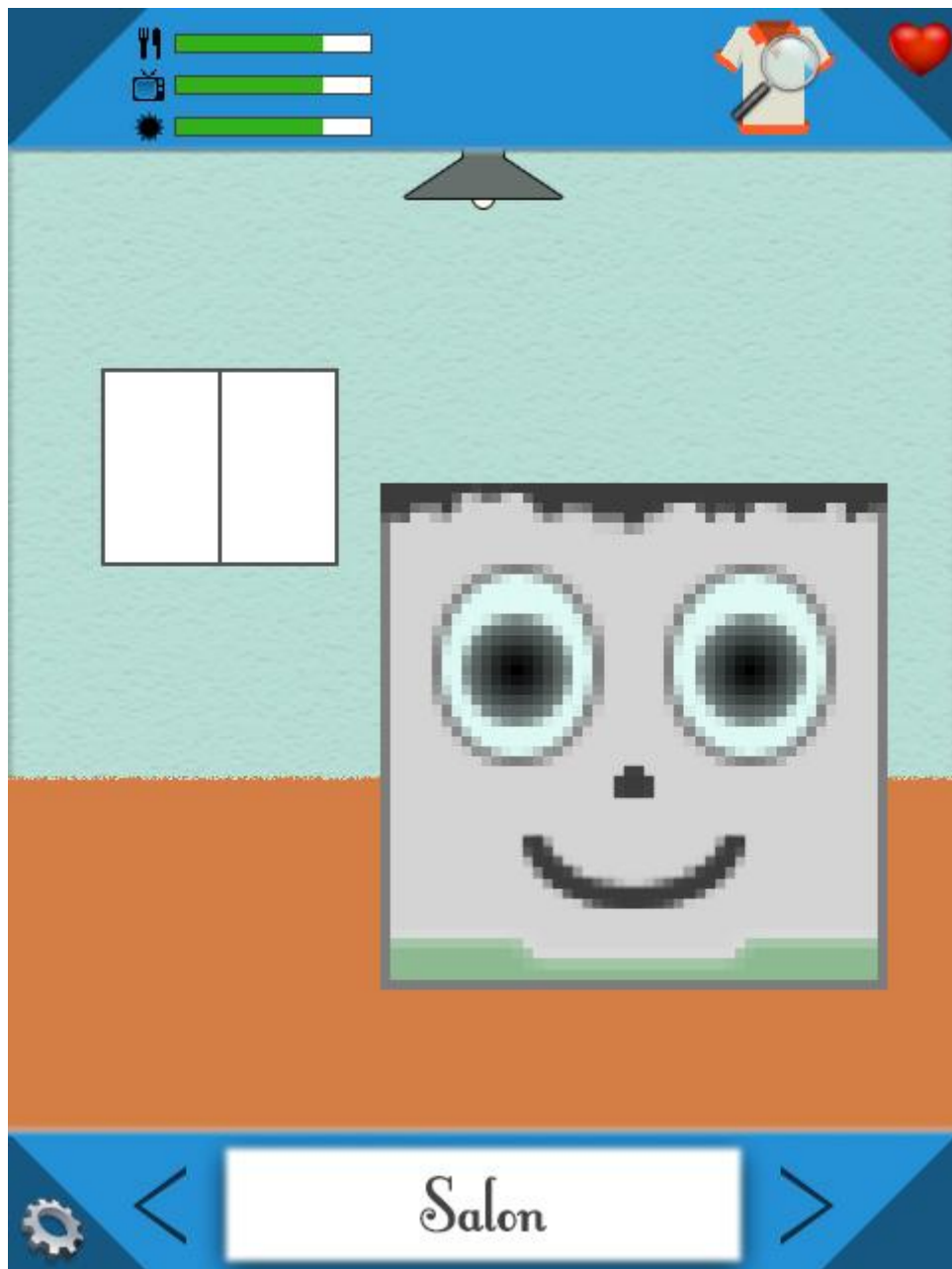


13. Szczegółowy opis przypadków użycia

Przypadek użycia	Opis
Stworzenie "stworka"	Klikając przycisk "Nowa gra" zostaje utworzony nasz stworek który w swoim obiekcie będzie miał odpowiednie dla niego dane(stopnie nastroju, wiek itd.)
Wysłanie "stworka" do spania	Po kliknięciu przycisku "Idź spać", ekran aktualny zostanie przygaszony, stworek zamknie oczy i jego nastrojnik "zmęczenie" zacznie powoli wzrastać
Zabawa z "stworkiem"	Po wybraniu odpowiedniej gry, podczas naszej zabawy z stworkiem jego nastrojnik "zabawa" będzie sukcesywnie rósł
Zapis "stworka"	Po kliknięciu "Zapisz stworka", wszystkie dane z pola składowego obiektu stworka zostaną zapisane do pliku tekstowego "Mój stworek.txt"
Wybranie odpowiedniej zabawy	Po wybraniu przycisku "Pobaw się z stworkiem" zostaniemy odesłani do menu gier, z którego możemy wybrać odpowiednią zabawę
Zarabianie pieniędzy dla "stworka"	Podczas grania w grę możemy zdobyć pieniądze dla "Stworka", np. poprzez zebranie spadającego z nieba przedmiotu.
Karmienie "stworka"	Po wybraniu odpowiedniego jedzenia z "Łodówki", nastrojnik stworka "głód" poprawi się
Kupowanie ubrań "stworkowi"	Dzięki zarobionym pieniądzu możemy kupić dla stworka w garderobie różnego rodzaju rzeczy

14.GUI





15.Czcionki, kolory, rozmiary

W grze użyliśmy darmowej czcionki: FontleroyBrown.ttf

Czcionka ta występować będzie w dwóch rozmiarach i różnych kolorach:

- dla wszystkich przycisków: rozmiar - 48px, kolor: #414141, kolor po najechniu myszką: #2490d5
- dla zwykłego tekstu: rozmiar - 12px, kolor: #414141, kolor po najechniu myszką: #2490d5

16. Niezawodność

1. Funkcja odpowiedzialna za wczytanie pliku .ttf z czcionką

```
public void wczytaj_font(){  
    try {  
        this.fontPrzycisku = fontPrzycisku.createFont(Font.TRUETYPE_FONT, new  
        FileInputStream("src/fonty/FontleroyBrown.ttf"));  
    } catch (FontFormatException e) {  
        e.printStackTrace();  
    } catch (IOException e) {  
        e.printStackTrace();  
    }  
    this.fontPrzycisku = this.fontPrzycisku.deriveFont(46f);  
}
```

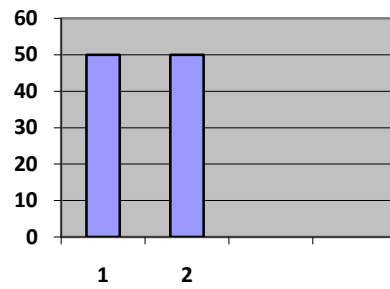
Jeśli font nie zostanie pomyslnie załadowany zostaniemy o tym poinformowani przez konsolę w środowisku programistycznym

17. TESTY

LP	Nazwa testu	Opis testu	Kryterium zaliczenia	Sposób oceny
1.	Sprawdzenie poprawności utworzenia nowej gry.	Sprawdzenie, czy po wciśnięciu przycisku nowa gra ,gra się uruchomi	Test uważa się za zaliczony, gdy po każdorazowym rozpoczęciu nowej gry, gra się rozpocznie	1 - test zaliczony 0 - test nie zaliczony
2.	Sprawdzenie poprawności działania przycisku kontynuuj	Sprawdzenie czy po wciśnięciu kontynuuj gra będzie kontynuowana.	Test uważa się za zaliczony, gdy wciśnięcie przycisku spowoduje kontynuację gry	1 - test zaliczony 0 - test nie zaliczony

Podsumowanie:

Wszystkie testy zostały przeprowadzone pośród 50. różnych użytkowników. Wyniki testów należy przedstawić na wykresie i podać analizie która ma zdecydować czy produkt jest skończony czy należy wprowadzić jakieś poprawki.



Moduł "Okno główne" przeszedł wszystkie testy.

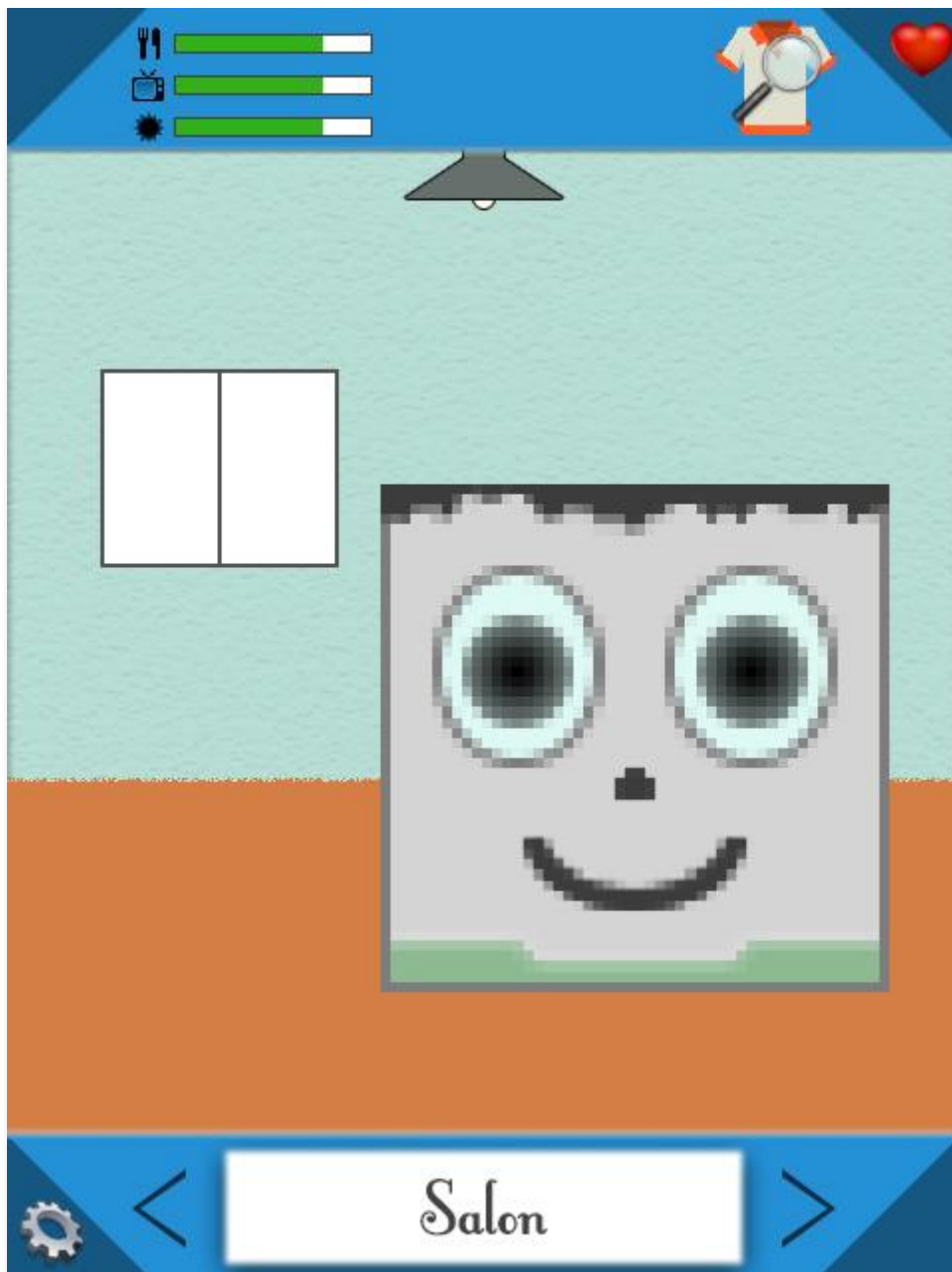
18. Dokumentacja użytkownika



okno główne aplikacji

Okno główne aplikacji pozwala na wybranie trybu gry. Nowa gra pozwala rozpocząć grę od nowa.

Kontynuuj pozwala na kontynuowanie wcześniej rozpoczętej gry. Wyjdź umożliwia zakończenie działania aplikacji.



Tryb salonu.

W tym trybie mamy dostępne wszystkie moduły gry. Ikonka trybiku pozwala na zmianę ustawień gry.

Strzałki obok białego pola z nazwą modułu pozwalają zmienić otoczenie.

Paski stanu ukazują aktualną kondycję stworka.

Koszulka z lupą pozwala na otworenie modułu szafa.

19.Narzędzia programistyczne

1. Adobe Photoshop CS6 - Stworzenie GUI i grafik dla naszego programu
2. Eclipse Luna Java EE- Kompilacja kodu źródłowego
3. Star UML - Tworzenie różnego rodzaju diagramów
4. Microsoft Word 2007 - Tworzenie dokumentacji