

WSPÓŁCZESNE JĘZYKI PROGRAMOWANIA

Prezentacja projektu nr 3 – Filip Brzeziński

CELE

- ◎ **Temat projektu:**

Interaktywna gra komputerowa „Segregacja odpadów”

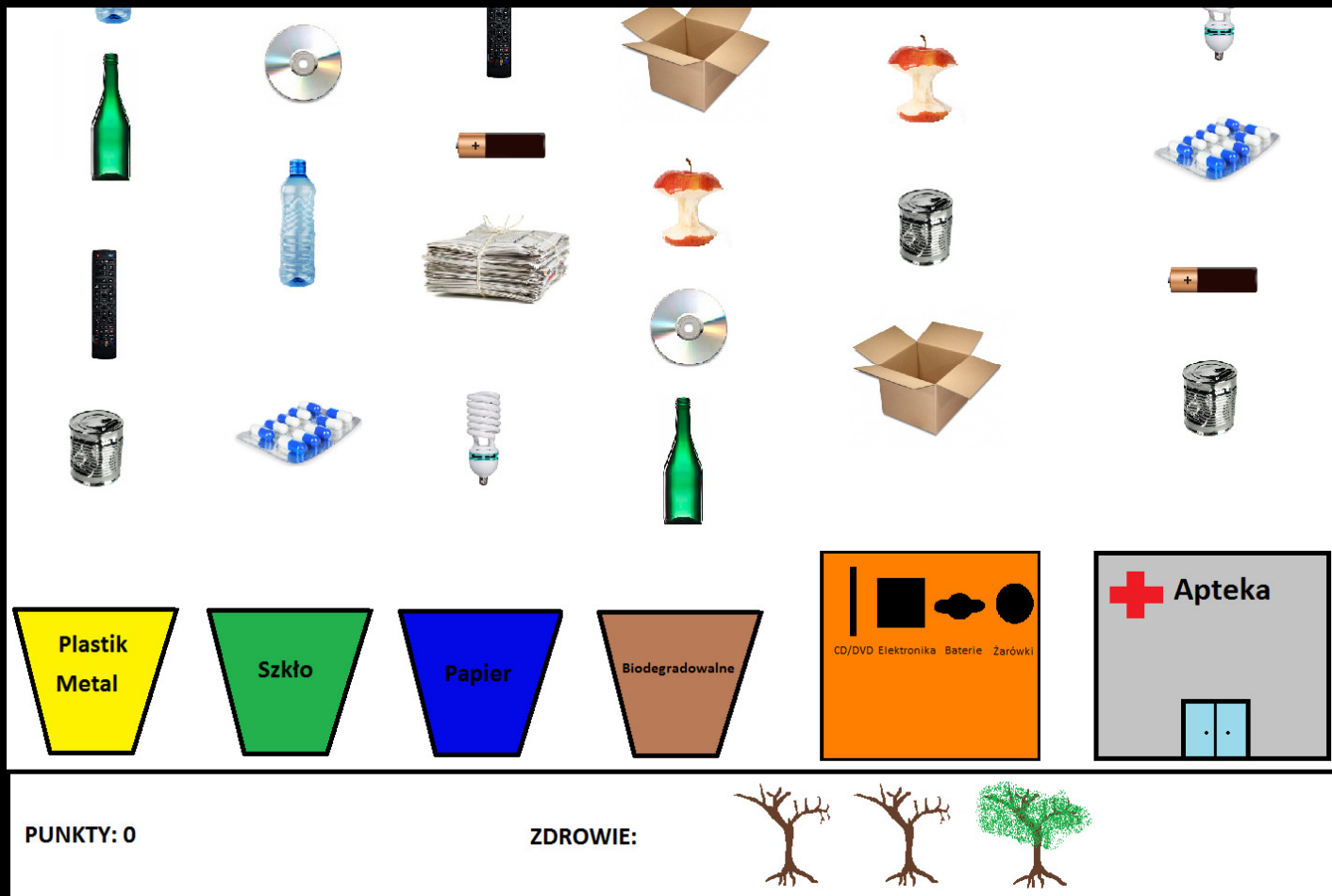
- ◎ **Cel:**

Celem projektu jest utworzenie gry, która ma edukować społeczeństwo jak odpowiedzialnie gospodarować odpadami. Warto wyćwiczyć u ludzi odruch wyrzucania konkretnych rodzajów odpadów do odpowiednich pojemników, oraz zwrócić uwagę, że nie wszystko da się tak łatwo posegregować. Istnieją takie odpady jak np. elektronika czy leki, których wyrzucenie do zwykłych pojemników może mieć fatalny wpływ na środowisko, a więc bardzo ważne jest uświadamianie o tym społeczeństwa.

- ◎ **Docelowi użytkownicy:**

Przede wszystkim dzieci i młodzież, gdyż jest to grupa najbardziej podatna na naukę poprzez granie w gry, ale równie dobrze może z niej korzystać przedstawiciel każdej innej grupy wiekowej.

Zakładany wygląd interfejsu graficznego



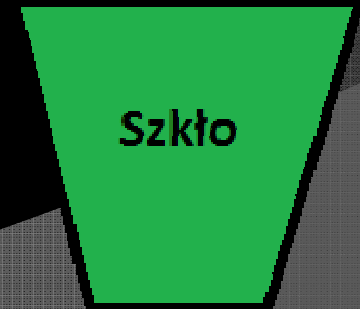
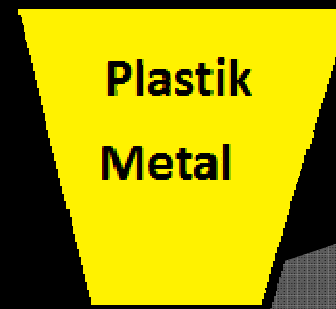
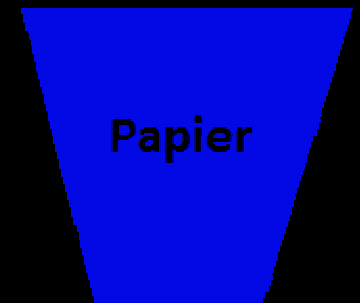
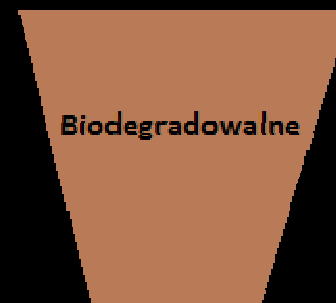
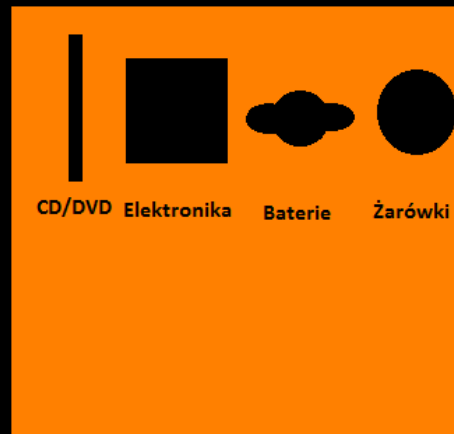
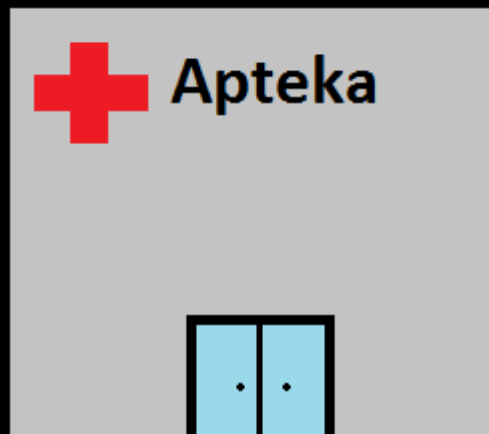
Aktualny wygląd interfejsu graficznego w trakcie działania gry



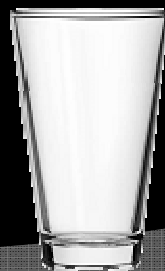
Zasoby gry:

Graficzne:

- Rysunki pojemników na odpady



- Zdjęcia odpadów



- Rysunki drzew symbolizujące ilość żyć (uschnięte oznacza utracone życie)



Zasoby gry:

Dźwiękowe:

Dwa pliki dźwiękowe odtwarzane odpowiednio gdy gracz dobrze lub źle zakwalifikuje odpad.

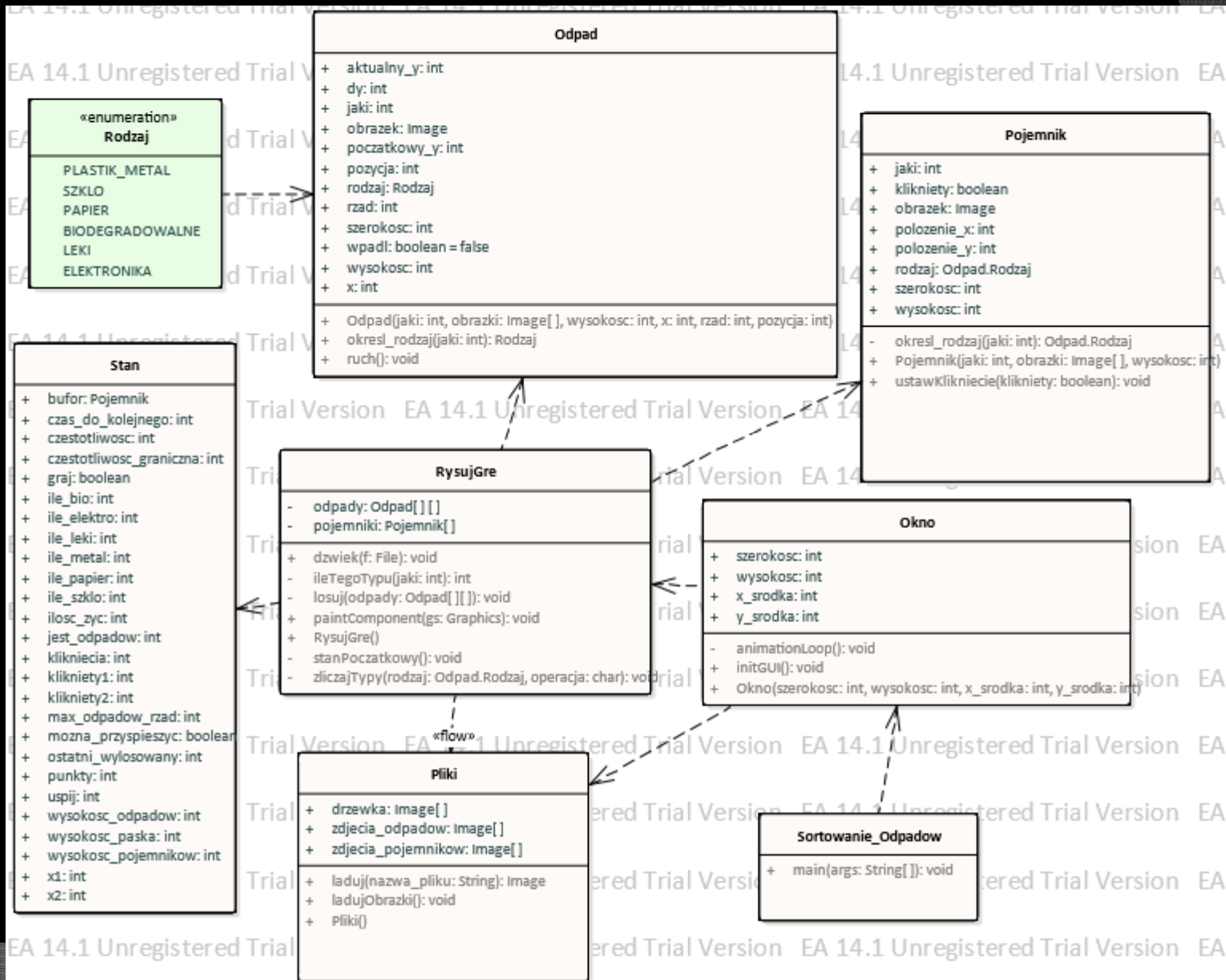
Składowanie danych

Planowane jest dodanie funkcji zapisującej do osobnego pliku wyniki gracza i odczytującej dane z tego pliku, aby wyświetlać ranking 10 najlepszych wyników gracza

Stan realizacji

1. Stworzono pliki zasobów
2. Zaprojektowano podział kodu na klasy (za pomocą programu Enterprise Architect)
3. Zrealizowano większość metod i klas, gra działa w pełni poprawnie i wymaga jedynie dodania menu, wcześniej wspomnianego zapisywania i wyświetlania wyników, oraz wyświetlania informacji o grze.

Stan realizacji – diagram klas



Stan realizacji – przykładowa funkcja losowania odpadów

```
private void losuj(Odpad [][] odpady) {  
    int ile_przejsc = 0;  
    int zapełnione_typy = 0;  
    int rzad;  
    int jaki;  
    boolean moze_powstac;  
    boolean utworzono_odpad = false;  
    Random losowanie = new Random();  
  
    jaki = losowanie.nextInt(Pliki.zdjecia_odpadow.length); //losujemy odpad  
    while(utworzono_odpad == false){  
  
        rzad = losowanie.nextInt(Pliki.zdjecia_pojemnikow.length); //losujemy rzad  
        while (Stan.ostatni_wylosowany == rzad){  
            rzad = losowanie.nextInt(Pliki.zdjecia_pojemnikow.length);  
        }  
        for (int i =0; i<odpady[rzad].length; i++){  
            moze_powstac = true;  
            if(odpady[rzad][i] == null){ //jezeli w tej komorce nic nie ma  
                for (int j =0; j<odpady[rzad].length; j++){  
                    if(odpady[rzad][j] != null){  
                        if(odpady[rzad][j].aktualny_y <= 0 ) //sprawdzamy jego pozycje zeby rysunki na siebie nie nachodzily  
                            moze_powstac = false;  
                    }  
                }  
            }  
        }  
    }  
}
```

Stan realizacji – przykładowa funkcja losowania odpadów

c.d.

```
if (moze_powstac != false){
    for (int k=0; k<Pliki.zdjecia_pojemnikow.length; k++){
        if(ileTegoTypu(k) > 1)
            zapelnione_typy++; // maksymalnie 2 takie same obiekty jednocześnie
        // w celu wiekszej roznorodnosci
        if(zapelnione_typy <Pliki.zdjecia_pojemnikow.length){
            while (ileTegoTypu(jaki) > 1){
                jaki = losowanie.nextInt(Pliki.zdjecia_odpadow.length);
            }
        }
        odpady[rzad][i] = new Odpad (jaki,Pliki.zdjecia_odpadow,Stan.wysokosc_odpadow, (rzad * (Okno.szerokosc/Pliki.zdjecia_pojemnikow.length)),rzad,i);
        utworzono_odpad = true;
        Stan.jest_odpadow ++;
        zliczajTypy(odpady[rzad][i].rodzaj, '+');
        Stan.ostatni_wylosowany = rzad;
    }
}
ile_przejsc ++;
if ((Pliki.zdjecia_pojemnikow.length*Stan.max_odpadow_rzad)< ile_przejsc) //oznacza to ze w zadnym rzędzie nie da sie już stworzyć nowego odpadu
    utworzono_odpad = true;// w takim przypadku należy opuścić petle i poczekać aż jakiś odpad zostanie usunięty

}
```

Stan realizacji – przykładowa funkcja zliczania odpadów danego typu

```
private void zliczajTypy(Odpad.Rodzaj rodzaj, char operacja){
    switch(rodzaj){
        case PLASTIK_METAL: {
            if(operacja == '+')
                Stan.ile_metal++;
            else if(operacja == '-')
                Stan.ile_metal--;
        }
        break;

        case SZKLO: {
            if(operacja == '+')
                Stan.ile_szklo++;
            else if(operacja == '-')
                Stan.ile_szklo--;
        }
        break;

        case PAPIER: {
            if(operacja == '+')
                Stan.ile_papier++;
            else if(operacja == '-')
                Stan.ile_papier--;
        }
        break;
    }
}
```

Stan realizacji – przykładowa funkcja zliczania odpadów danego typu c.d.

```
case BIODEGRADOWALNE: {
    if(operacja == '+')
        Stan.ile_bio++;
    else if(operacja == '-')
        Stan.ile_bio--;
}
break;

case LEKI: {
    if(operacja == '+')
        Stan.ile_leki++;
    else if(operacja == '-')
        Stan.ile_leki--;
}
break;

case ELEKTRONIKA: {
    if(operacja == '+')
        Stan.ile_elektro++;
    else if(operacja == '-')
        Stan.ile_elektro--;
}
break;
}
```

Stan realizacji – fragment kodu rysujący odpady oraz sprawdzający czy wpadły do pojemnika i jeśli tak , to czy do dobrego. Jednocześnie odtwarzany jest odpowiedni dźwięk oraz usuwany jest odpad, który wpadł do pojemnika.

```
for (int i =0; i<Pliki.zdjecia_pojemnikow.length; i++){
    for (int j=0; j<Stan.max_odpadow_rzad; j++){
        if(odpady[i][j] != null){
            g.drawImage(odpady[i][j].obrazek, odpady[i][j].x, odpady[i][j].aktualny_y, odpady[i][j].szerokosc, odpady[i][j].wysokosc, null);
            odpady[i][j].ruch();
        }
        if((odpady[i][j].aktualny_y+odpady[i][j].wysokosc) >= (Okno.wysokosc - Stan.wysokosc_pojemnikow + odpady[i][j].wysokosc*0.3)){
            if(odpady[i][j].rodzaj == pojemniki[odpady[i][j].rzad].rodzaj){
                Stan.punkty++;
                dzwiek(new File("Sounds/correct.wav"));
            }
            else {
                Stan.ilosc_zyc--;
                dzwiek(new File("Sounds/wrong.wav"));
            }
            zliczajTypy(odpady[i][j].rodzaj, '-');
            odpady[i][j] = null;
            Stan.jest_odpadow --;
        }
    }
}
```