Dokumentacja klas i przykładów użycia klas w grze Svarog

V\_0.15

Drzewo Projektu:

Svarog/

--collision/

----AABB.java

----Collision.java

--entity/

----Entity.java

----HP.java

----Player.java

----Transform.java

----XP.java

--game/

----Main.java

----SecondTestWorld.java

----StartWorld.java

----WorldLoader.java

--gui/

----font/

------Character.java

------Color.java

------Font.java

------FontMap.java

------TrueTypeFont.java

----GuiObject.java

----GuiPanels.java

----GuiRenderer.java

----TextureObject.java

--io/

----strings/

------IntAttribute.java

------StringAttribute.java

------StringUtilis.java

----Input.java

----TextLoader.java

----Timer.java

----Window.java

--language/

----InterfaceTranslation.java

----InterfaceTranslations.java

----LanguageLoader.java

--render/

----Animation.java

----Camera.java

----Model.java

----Shader.java

----Texture.java

--world/

----Door.java

----Tile.java

----World.java

1.Analiza Klas:

1. AABB.java //system wykrywania kolizji

public class AABB {

private Vector2f center; //środek obiektu

private Vector2f half\_extent; //połowa odległości

public AABB(Vector2f center, Vector2f half\_extent) { … } //konstruktor

public Collision getCollision(AABB box) { … } //zwraca obiekt Collision (dystans od obiektu i czy doszło do kolizji)

public Vector2f getCenter() { … } //getter pola center

public Vector2f getHalfExtent() { … } //getter pola halfExtent

public void correctPosition(AABB box, Collision data) { … } //koryguje kolizje

}

2.Collision.java //

public class Collision {

private Vector2f distance; //odległość wektorowa

private boolean isIntersecting; //logiczny bool czy doszło do zderzenia?

public Collision(Vector2f distance, boolean intersects) { … } //konstruktor

public void setDistance(Vector2f distance) { … } //setter pola Distance

public Vector2f getDistance() { … }//getter pola Distance

public void setIsIntersecting(boolean state) { … }//setter pola isIntersectiong

public boolean isIntersecting() { … } //getter pola isIntersection

}

3.Entity.java

public class Entity {

private static int auto\_increment = 0;

private static final float[] verticesArray = new float[] { … }; //Przechowuje tablice punktów

private static final float[] textureArray = new float[] { … }; //Przechowuje rogi płaszczyzny

private static final int[] indicesArray = new int[] {...}; //Przechowuje łączenia punktów w ściany (kwadrat to 2 trójkąty )

private Model model;

protected Animation animation;

protected Texture texture;

protected Transform transform;

protected Transform textureTransform;

private AABB bounding\_box;

private int id;

private boolean isStatic = true;

private boolean fullBoundingBox;

protected Direction currentDirection;

protected boolean[] isColliding = new boolean[2];

protected enum Direction {…}//enumerator 4 kierunków świata

public Entity(Animation animation, Transform transform, boolean fullBoundingBox) {…}//Konstruktor animacji

public Entity(Texture texture, Transform transform, boolean fullBoundingBox) { …} //konstruktor tekstury

private void setEntityProperties() {…} //Ustala skalę dla obiektu – aktora

public void move(Vector2f direction) {…} //Przemieszcza obiekt po mapie

public void setPosition(int x, int y) {…} //ustala pozycję w przestrzeni (x,y)

public void collideWithTiles(World world) {…} //Ustala kolizje (na podstawie odpowiedniego pliku maski w World) aby działały

public void collideWithEntities(World world) {…}//ustala kolizje z innymi obiektami typu entities

public void update(float delta, Window window, Camera camera, World world) {…} //blokuje gracza przed ewentualną możliwością wyjścia poza mapę

public void render(Shader shader, Camera camera, World world) {…} //renderowanie postaci

public Entity setIsStatic(boolean state) {…}//setter

public Entity setFullBoundingBox(boolean state) {…}//setter

public boolean isStatic() {…}//getter

public AABB getBoduningBox() {…}// getter

protected void setAnimation(Direction direction, Animation animation) {…}//setter

protected void setTexture(Texture texture) {…} //setter

4.HP.java //ogólne działanie:

//pole hp > 0 i hp <= maxHP

Pola:

private int hp; //posiada gettery I settery

private int maxHP; //posiada gettery I setter

public void AddHP(int health) {} //dodaje HP

public void DecreaseHP(int damage) {} //Odjemuje HP

5.Player.java

public class Player extends Entity {

private boolean setCamWithoutAnimation;

private int[] lastKeysPressed = new int[4];

private int lastPressedKey;

private String texturesPath;

private String fileName;

public Player(String texturePath, String filename, Transform transform, boolean fullBoundingBox) {…}konstruktor publiczny

@Override

public void update(float delta, Window window, Camera camera, World world) {…}//ststem sterowania graczem

private void setAnimation(Direction direction) {…}//pojawia się animacja uzależniona od kierunku poruszania się gracza

private void setTexture(Direction direction) {…}//zmiana tekstury gracza od pozycji w którą stronę świata jest skierowany

private int getNewPressedKey(int[] lastPressedKeys, int[] pressedKeys) {…}//metoda pomocnicza, przechowująca wciśnięte przyciski

private void setLastKeysPressed(int[] keysPressed) {…} //Metoda przechowująca kolejność wciskania przycisków

public int getPositionX() {..}//getter

public int getPositionY() {…}//getter

public void setSetCamWithoutAnimation(boolean setCamWithoutAnimation) {…}//setter dla wertości typu boolean

}

6. Transform.java

Pola:

private Vector3f position; //odpowiednie gettery I settery (float / Vector3f)

private Vector3f scale; //odpowiednie gettery I settery (float / Vector3f)

Ciekawszy getter:

public Matrix4f getProjection(Matrix4f target){…} //zwraca projekcję w postaci Matrix4f

7. XP.java

Pola:

private int xp = 0; //getter I setter(int)

public void AddXP(int points) {…}//metoda zwiększająca poziom XP

8.Character.java

public class Character {

//pola

char character;

int x;

int y;

int width;

int height;

Character(char character) {...} //konstruktor dla symbolu

int getX() {...}

void setX(int x) {...}

int getY() {...}

void setY(int y) {...}

int getWidth() {...}

void setWidth(int width) {...}

int getHeight() {...}

void setHeight(int height) {...}

char getCharacter() {...}

void setCharacter(char character) {...}

}

9.Color.java

public class Color {

private byte R;

private byte G;

private byte B;

public Color(byte red, byte green, byte blue) {...} //ustawia kolory

public void set(Color color) {...} //ustawia odpowiednie kolory

public byte getR() {...}

public void setR(byte r) {...}

public byte getG() {...}

public void setG(byte g) {...}

public byte getB() {...}

public void setB(byte b) {...}

}

10.Font.java

public class Font {

private String text;

private FontMap fontMap;

private BufferedImage fontImage;

private int stringWidth;

private int stringHeight;

public Font(String font) {...} //pobieta zdjęcie I plik fnt

public ByteBuffer getStringBuffer(String string, Color color) {...}//ustawia wszystkie parametry I układa pixele dla fontów do wyświetlania

public void setText(String text) {...}

public String getText() {...}

public int getStringHeight() {...}

public int getStringWidth() {...}

}

11.FontMap.java

public class FontMap {

List<Character> characters;

TextLoader text;

FontMap(String file) {...}//używając metody TextLoadera wczytuje fonty

int getWidth(char ch) {...}//zwraca szerokośćfontu,jeśli nie znajdzie zwraca -1

int getHeight(char ch) {...}//zwraca wysokość fonu, jeśli nie znajdzie zwraca -1

int getX(char ch) {...} //zwraca współżędną x zaczęcia się fontu na obrazku

int getY(char ch) {...}//zwraca współżędną y zaczęcia się fontu na obrazku

int getId(char ch) {...} //zwraca id fontu

private void loadCharacters() {...}//wczytuje wszystkie fonty

private int getAttribute(String line, String attributeName, char spacer) {...}

}

12. TrueTypeFont.java

public class TrueTypeFont {

private int fontSize;

private Font font;

private boolean antialiasing;

private Color color;

private int stringWidth;

private int stringHeight;

public TrueTypeFont(String fileName, int size, boolean antialising) {…} //setter

public BufferedImage getFontImage(char ch) {…}// zwraca zdjęcie fontu

public ByteBuffer getStringBuffer(String string) {…}//zwraca kolory pixeli

public int getFontSize() {…}//zwraca rozmiar fontów

public void setFontSize(int fontsize) {…} //setter rozmiaru fontów

public Font getFont() {…}//zwraca obiekt font

public void setFont(String fileName) {…}//tworzy font z pliku

public boolean isAntialiasing() {…}//zwraca booleana

public void setAntialiasing(boolean antialiasing) {…} //setter boolean

public int getStringWidth() {…} //zwraca długość

public int getStringHeight() {…}//zwraca wysokość

public Color getColor() {...}//zwraca kolor

public void setColor(Color color) {...} //ustawia kolor

}

13.GuiObject.java

public class GuiObject {

private static int auto\_increment = 0;

private int id;

private float width;

private float height;

private Transform transform;

private Vector2f relativeTransform;

private stickTo stickTo;

private State state;

private static final float scale = 16f;

protected GuiObject() {...}

protected GuiObject(int width, int height, stickTo stickTo) {...}

protected GuiObject(int width, int height, Vector2f position) {...}

public float getHeight() {...}

public void setHeight(int height) {...}

public float getWidth() {...}

public void setWidth(int width) {...}

public int getId() {...}

public Transform getTransform() {...}

public void setPosition(Vector2f position) {...}

public void setPosition(float X, float Y) {...}

public void move(Vector2f direction) {...}

public void move(float X, float Y) {...}

public static float getScale() {...}

private void configureTransform(Vector2f position) {...}

public stickTo getStickTo() {...}

public void setStickTo(stickTo stickTo) {...}

public Vector2f getMove() {...}

public State getState() {...}

Texture getTexture() {...}

public GuiObject setState(State state) {...}

}