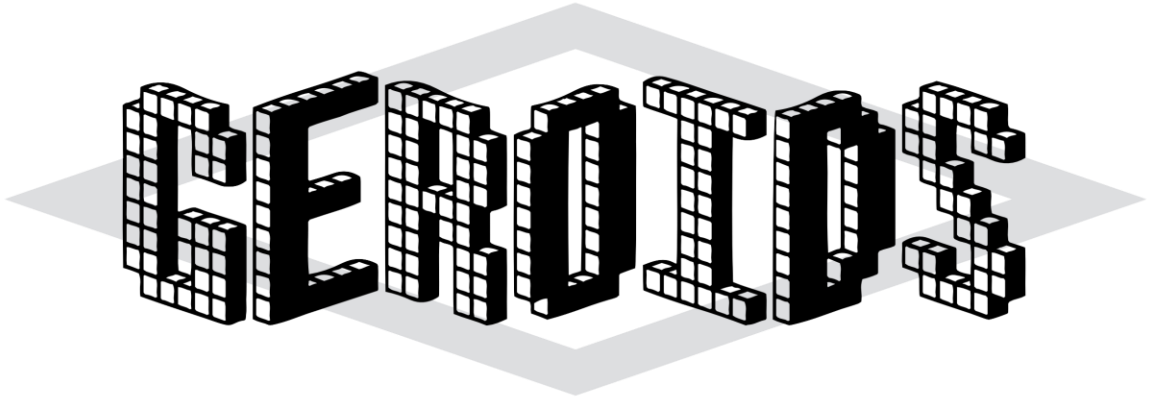

EIN 2D-ARCADEGAME FÜR KLEIN UND GROSS

Inhalt

- Aktueller Stand
- Erkenntnisse
- Darstellung des Spiels
- Fortschritt
- Demo
- Fragen

Aktueller Stand Iterationsplan

Iteration	Start	Ende	Meilenstein	Aufwand [h]	Ziele / Umsetzung
Inception Phase					
1	20.09.2016	04.10.2016	M1	50	Vision, Projektskizze, Präsentation
Elaboration Phase					
2	04.10.2016	18.10.2016		60	Entwicklungsumgebung, GUI-Entwurf
3	18.10.2016	01.11.2016	M2	80	Domänenmodell, UML-Diagramm, DB-Entwurf, SW-Architektur
Construction Phase					
4	01.11.2016	15.11.2016		100	UC3 Implementierung
5	15.11.2016	29.11.2016	M3	90	UC 1 & UC2 implementieren
6	29.11.2016	06.12.2016		60	UC4 & Datenbank Implementierung
Transition Phase					
7	06.12.2016	20.12.2016	M4	60	UC1-4 GUI-Testing, Prototyp-Release
Total				500	

Aktueller Stand aktuelle Iteration

Task #	Arbeitspaket	Aufwand [h]	Ist [h]	Verantwortlich
1	UC3 Tests	20		Alle
2	UC3 Refactoring Figure, Gamefield, Micro,	2	2	Matthias K.
3	UC3 Refactoring Account, Position	2	3	Valentin B.
4	UC3 Refactoring Game, Geroid, Projectile, Collisionhandler	2	10	Linda B.
5	UC3 Abbildung Objekte mit Bilder	15		Valentin B.
6	UC3 Score	5	3	Linda B.
7	UC3 Game Over Seite	5		Valentin B.
8	UC1 implementieren + Tests			Arben S. (PL)
9	UC2 implementieren + Tests	10		Linda B.
10	Wissenstransfer	4		Alle
11	Statusmeeting 2x	8		Alle
12	Javadoc aktualisieren	2	2	Alle
Total		90		

Erkenntnisse

- Kommunikation falls zusätzliche Arbeit geleistet wird
- Möglichst wenig Threads für weniger Komplexität und weniger Rechenaufwand
- Lokal keine Performanzprobleme
- Komplexität bei Erstellung einer ausgefeilten Login/Registrierung

Darstellung des Spiels

- Wie funktioniert Darstellung mittels Canvas
- Zusammenspiel Canvas und Java
- Flüssige Darstellung durch genügend viele FPS (60+)

:Figure

```java
/**
 * Translates the Object inclusive the position attribute into a JSONObject
 * @return JSONObject of figure
 */
@SuppressWarnings("unchecked")
public JSONObject toJSONObject(){
    JSONObject obj = new JSONObject();

    obj.put("position", this.position.toJSONObject());

    return obj;
}

public Position getPosition() {

    return this.position;
}
```

:Game

```
/*
 * Sends the new Values of Figure, all Geroids and all Projectiles via
 * webSocketHandler to the Client.
 */
@SuppressWarnings("unchecked")
private void sendNewValues() {
    JSONObject obj = new JSONObject();
    obj.put("Figure", figure.toJSONObject());
    obj.put("Geroids", this.geroidsToJSONArray());
    obj.put("Projectiles", this.projectilesToJSONArray());
    obj.put("Gameover", !isRunning);
    obj.put("Name", this.account.getNickname());
    webSocketHandler.sendMessage(obj.toJSONString());

}
```

Index.js

```
ws.onmessage = function(evt) {
    gamefield = JSON.parse(evt.data);
    console.log(gamefield);
};
```

```
function drawFigure(figureJSON) {
    context.drawImage(spaceShip, figureJSON.position.xStart
            * canvasXFactor, figureJSON.position.yStart
            * canvasYFactor, figureJSON.position.xLength
            * canvasXFactor, figureJSON.position.yLength
            * canvasYFactor);
}
```

Fortschritt

- Kollisionsdetektion vorhanden
- Darstellung von Spiel funktioniert

Demo

Fragen