|  |
| --- |
| M226b – Objektorientiert implementieren |
| Projekt - PacManGrande |
| Bildergebnis für Pac-Man"  Abbildung : Titelbild - Quelle: https://www.mentalfloss.com/article/90920/10-fast-facts-about-pac-man |
| **Dokumentattribute**   |  |  |  |  | | --- | --- | --- | --- | | Autor | Marrazza Mario | Adresse | Teufenerstrasse 41 | | Klasse | IFZ-826-003a | Dozent | Sven Schirmer | | Dateiname | M226b\_PacManGrande.docx | Modulnummer | M226b | | Ausgabe vom | 17.12.2019 | Abgabedatum | 17.12.2019 | |

Projektbeschreibung

PacManGrande ein von Mario Marrazza entwickeltes PacMan Spiel welches in Java programmiert wird und auf die PApplet Librarys von Processing zugreifft.

Änderungskontrolle

| Version | Datum | Ausführende Stelle | Bemerkung / Art der Änderung |
| --- | --- | --- | --- |
| 01 | 06.11.2019 | Mario Marrazza | Erstellung der Dokumentation |
| 02 | 06.11.2019 | Mario Marrazza | Tag 1 – Start des Projekts |
| 03 | 13.11.2019 | Mario Marrazza | Tag 2 – Dokumentationen & UML Klassendiagramm |
| 04 | 18.11.2019 | Mario Marrazza | Tag 3 - Verbesserungen an das UML Klassendiagramm |
| 05 | 25.11.2019 | Mario Marrazza | Tag 4 – Erzeugung der PacMan Entity und Bewegungsfunktion |
| 06 | 28.11.2019 | Mario Marrazza | Tag 5 – Erzeugung der Map |
| 07 | 01.12.2019 | Mario Marrazza | Tag 6 – Erzeugung der Map V2 |
| 08 | 10.12.2019 | Mario Marrazza | Tag 7 – Erzeugung der Map V3 |
| 09 | 12.12.2019 | Mario Marrazza | Tag 8 – Projekt neustart |
| 10 | 16.12.2019 | Mario Marrazza | Tag 9 – Schreiben des Map.txt Files |
| 11 | 17.12.2019 | Mario Marrazza | Tag 10 – Erzeugung der Entitys |
| 12 | 17.12.2019 | Mario Marrazza | Schlussreflektion / vervollständigung der Dokumentation |

Tabelle : Änderungskontrolle

Inhaltsverzeichnis

[1 Tag 1 – Start des Projekts 4](#_Toc27504116)

[1.1 Inhalt/ ausgeführte Arbeiten 4](#_Toc27504117)

[2 Tag 2 – Dokumentationen & UML Klassendiagramm 5](#_Toc27504118)

[2.1 Inhalt/ ausgeführte Arbeiten 5](#_Toc27504119)

[3 Tag 3 – Verbesserungen an das UML Klassendiagramm 7](#_Toc27504120)

[3.1 Inhalt/ ausgeführte Arbeiten 7](#_Toc27504121)

[4 Tag 4 – Erzeugung der PacMan Entity und Bewegungsfunktion 8](#_Toc27504122)

[4.1 Inhalt/ ausgeführte Arbeiten 8](#_Toc27504123)

[5 Tag 5 – Erzeugung der Map 9](#_Toc27504124)

[5.1 Inhalt/ ausgeführte Arbeiten 9](#_Toc27504125)

[6 Tag 6 – Erzeugung der Map V2 9](#_Toc27504126)

[6.1 Inhalt/ ausgeführte Arbeiten 9](#_Toc27504127)

[7 Tag 7 – Erzeugung der Map V3 10](#_Toc27504128)

[7.1 Inhalt/ ausgeführte Arbeiten 10](#_Toc27504129)

[8 Tag 8 – Projekt neustart 11](#_Toc27504130)

[8.1 Inhalt/ ausgeführte Arbeiten 11](#_Toc27504131)

[9 Tag 9 – Schreiben des Map.txt Files 12](#_Toc27504132)

[9.1 Inhalt/ ausgeführte Arbeiten 12](#_Toc27504133)

[10 Tag 10 – Erzeugung der Entitys 13](#_Toc27504134)

[10.1 Inhalt/ ausgeführte Arbeiten 13](#_Toc27504135)

[11 Abbildungverzeichnis 17](#_Toc27504136)

[12 Tabellenverzeichniss 17](#_Toc27504137)

[13 Literaturverzeichnis 17](#_Toc27504138)

[14 Selbstständigkeitserklärung 18](#_Toc27504139)

[15 Anhang 1](#_Toc27504140)

# Tag 1 – Start des Projekts

## Inhalt/ ausgeführte Arbeiten

* Einlesung im Buch
* Projekt Auswahl
* Einführung GitHub

Elesung im Buch

Ins Buch *“game-architecture-and-design-a-new-edition”* eingelesen und versucht etwas über den Vorgang zur Entwicklung eines Pac-Man Spiels zu erfahren. Erstes Bild über die Entwicklung sowie eine genaue Spielbeschreibung / auf was geachtet werden muss bei der Entwicklung des Spiels.

Projekt Auswahl

UML = Unified Modeling Language

Die UML Klassendiagramme sind vergleichbar mit den ERM (Entity Relation Model).

In den UML Klassendiagrammen, ist die Sichtbarkeit der Klassen ersichtlich und es werden den Klassen Attribute vergeben.

Einführung GitHub

Ein Repository mit dem Namen: «PacManGrande» wurde erstellt.

Das Repository wurde im Eclipse Workspace eingebunden. Es wurden 1. Klassen und Commits/Pushes getätigt.

# Tag 2 – Dokumentationen & UML Klassendiagramm

## Inhalt/ ausgeführte Arbeiten

* Erstellung der Dokumentationen
* Fertigstellung des Projektpflichtenhefts
* UML – Klassendiagramm

Erstellung der Dokumentationen

Basierend auf bisherigen Vorlagen wurden Dokumentationen erstellt für:

1. Projekt Pflichtenheft
2. Tages Journal

Darin wurden bereits besteimmte Ziele festgelegt und eine grobe Struktur aufgebaut sodass ich in Zukunft nur noch den Inhalt entsprechend befüllen muss.

Fertigstellung des Projektpflichtenhefts

Das Projektpflichtenheft soll meine Gedanken sowie Ziele festhalten und als Übersicht dienen was am Ende des Projektes erreicht werden soll und die potenziellen Risiken auflisten.

UML – Klassendiagram

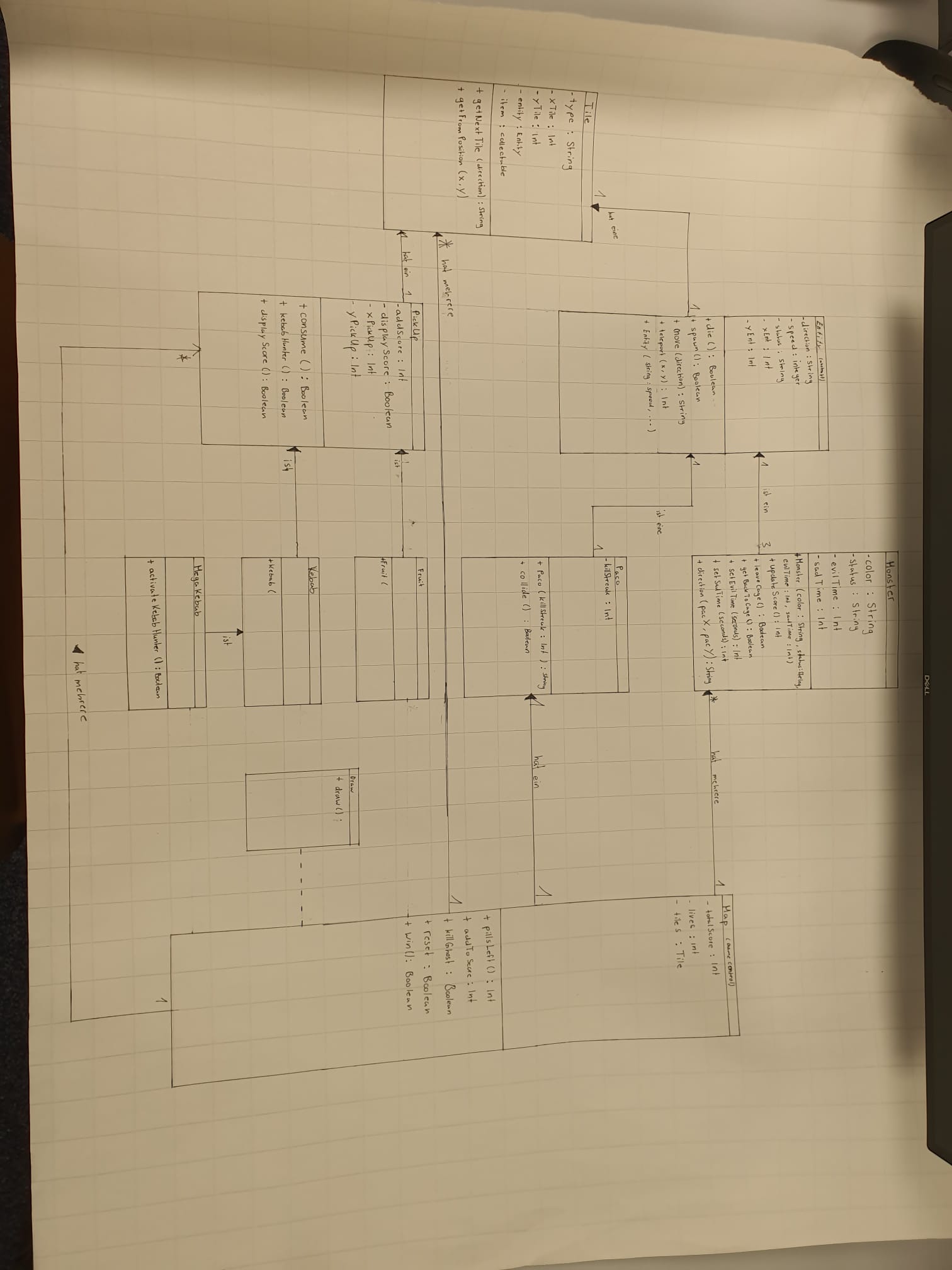


Abbildung : UML - Klassendiagramm

# Tag 3 – Verbesserungen an das UML Klassendiagramm

## Inhalt/ ausgeführte Arbeiten

* Fehleranalyse des Klassendiagramms
* Korrektur / Verbesserung des Klassendiagramms

Fehleranalyse des Klassendiagramms

Zu Beginn habe ich an einem erneuten Tag mit Herrn Schirmer mein UML Klassendiagramm angeschaut und allfällige Denkfehler sowie «Komplikationen» im Detail verfeinert.

Korrektur / Verbesserung des Klassendiagramms

Fehlerhafte Veerbungen sowie überflüssige Verbindungen der Tabellen wurden ausradiert und durch bessere Lösungen ersetzt.

Kardinalitäten sowie Sichtbarkeiten

# Tag 4 – Erzeugung der PacMan Entity und Bewegungsfunktion

## Inhalt/ ausgeführte Arbeiten

* Entity
  + PacMan
  + Monster
* PacMan – Bewegungsfunktion

Entity

Entity ist die Basisklasse für alle Charakter diese im PacMan Spiel benötigt warden.

Von der Entity Klasse werden die Attribute wie, Farbe, geschwindigkeit, grösse etc. vererbt.

PacMan

PacMan Objekt wurde in der Controller Klasse in der Draw Funktion erzeugt.

Monster

PacMan Objekt wurde in der Controller Klasse in der Draw Funktion erzeugt.

PacMan – Bewegungsfunktion

Der Geschwindigkeit wurde die tileSize übergeben sodass der PacMan sowie die Monster sich jeweils um 1 tile fortbewegen.

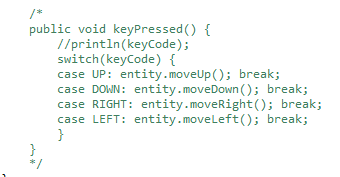


Abbildung : Codesnippet Stand - 17.12.2019

# Tag 5 – Erzeugung der Map

## Inhalt/ ausgeführte Arbeiten

* Erste Überlegungen
  + Was benötigt meine Map?
  + Auf was muss geachtet werden

Erste Überlegungen

Was benötigt meine Map?

Es Sollen Wände gezeichnet werden, PickUps die von PacMan aufegegessen werden können und ein Feld in dem die Monster «spawnen» können.

Auf was muss geachtet werden

Die Map soll in Raster mit Indexes (berechnet aus x und y) aufgeteilt werden sodass man die Positionen basierend auf den Index ausfindig machen kann. Die Wände sollen nicht begehbar sein. Die Pickups sollen verschwinden sobald der PacMan damit kollidiert.

# Tag 6 – Erzeugung der Map V2

## Inhalt/ ausgeführte Arbeiten

* Hartcodiertes Raster

Hartcodiertes Raster

Ich habe ein Raster mit 2 For Schleifen hartcodiert und anhand der x und y position angegeben wieviele tiles es auf der Map insgesamt gibt und berechnet wo sich welches Teil wann befindet.

Diese Tiles wurden jeweils mit grauer Farbe getrennt. Jedoch war es mir bisher nicht möglich eine Funktion zu entwickeln diese deklariert ob ein Tile eine wall ist oder nicht.

# Tag 7 – Erzeugung der Map V3

## Inhalt/ ausgeführte Arbeiten

* Effektive Berechnung

Effektive Berechnung

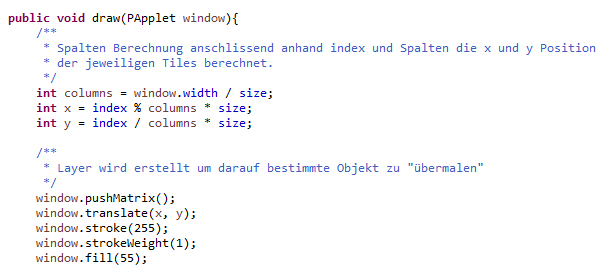


Abbildung : Codesnippet aus PacManGrande Stand: 17.12.2019

# Tag 8 – Projekt neustart

## Inhalt/ ausgeführte Arbeiten

* Entfernung unschönes Codes
* Neuorientierung

Entfernung unschönes Codes

Da mein bisheriger Anlaufsversuch gescheitert war und ich zu viel verwirrenden code hatte, welcher sich über die Zeit und diversen Versuchen aufgestappelt hat. Hatte mir Herr Schirmer geholfen den gesamten Code zu entfernen.

Neuorientierung

Herr Schirmer gab mir den Tipp die Map via Buffered Reader und einem .txt File in dem die Indexes mit Linebreaks eingetragen werde zu generieren.

Alle anderen tiles diese keine Wand sind, werden automatisch als PickUp gezeichnet.

# Tag 9 – Schreiben des Map.txt Files

## Inhalt/ ausgeführte Arbeiten

* Vorgang / schreiben des .txt Files

Vorgang / Schreiben des .txt Files

Da die Map mit Raster Grafiken und anhand der X und Y Position indexes berechnet wurden. Kann man so anhand einer «Mouseclick» funktion auswählen von welchem Tile man das Index haben möchte und im .txt File die Indexes abfüllen, welche man als «Wand» deklarieren möchte.

Es Wurde eine Rescource Folder erstellt und eine .txt Datei mit dem Namen: «paco\_map.txt» in dieser Text Datei wurden alle Indexes eingetragen welche eine Wand sein sollten. Diese Text Datei wurde im Funktionsaufruf der Buffered Reader Funktion übergeben.

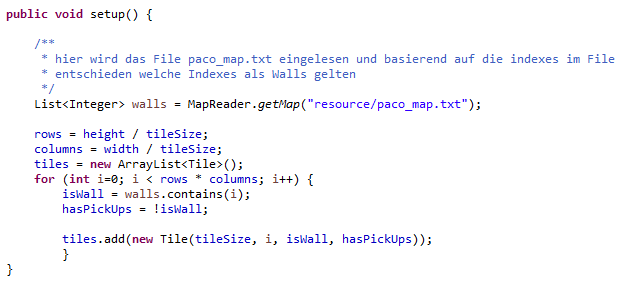


Abbildung : Funktionsaufruf - Buffered Reader



Abbildung : Buffered Reader Funktion

# Tag 10 – Erzeugung der Entitys

## Inhalt/ ausgeführte Arbeiten

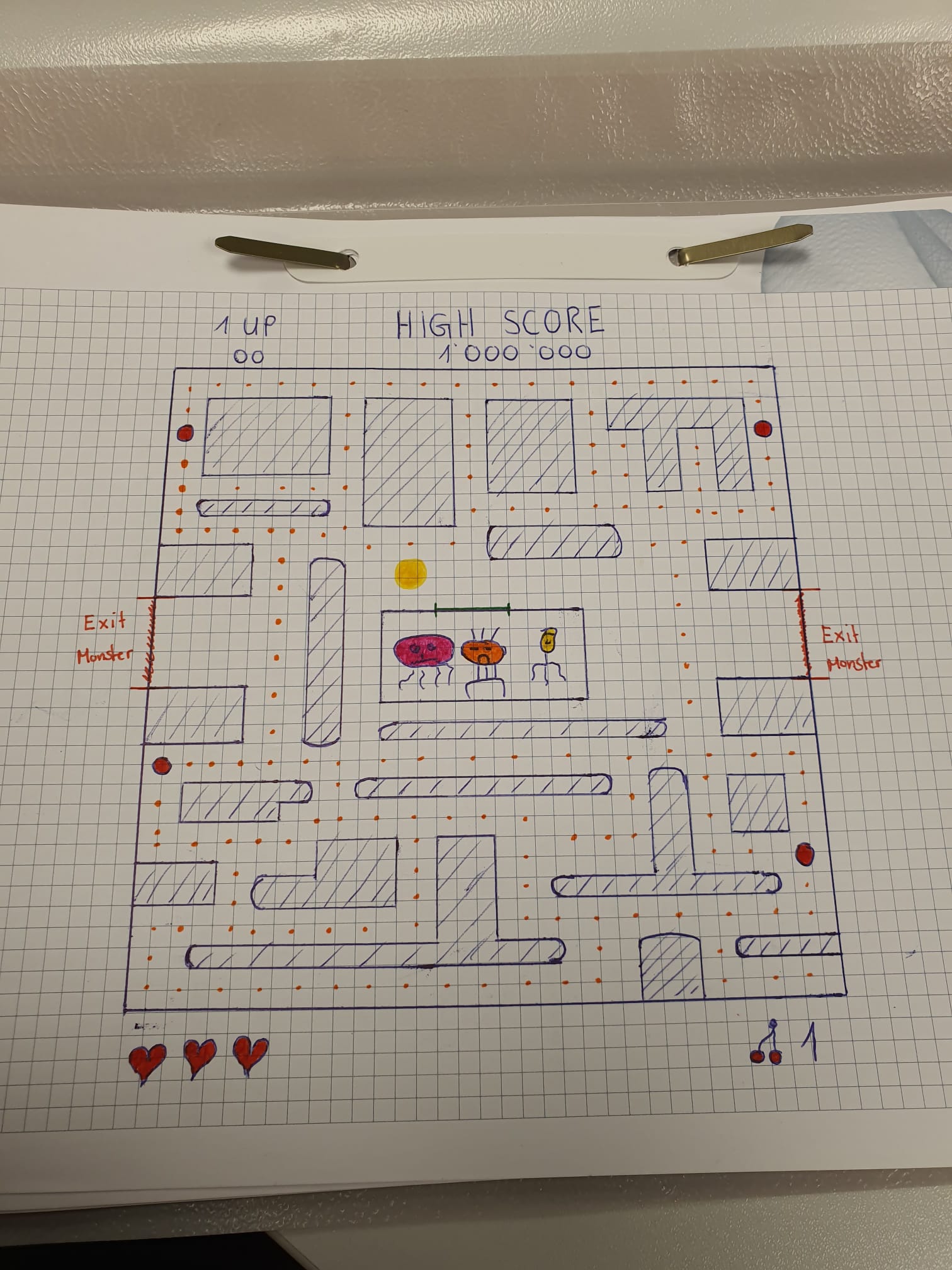
* Dokumentation Fertigstellung

Dokumentation Fertigstellung

Da ich leider noch einige Verbesserungen an meiner Map vornehmen musste und ich bei der Implementierung der Entitys haufenweise Fehler angetreten bin, habe ich mich entschieden am letzten Tag noch die Dokumentation sauber und ausführich zu vervollständigen.

Mockup Test

Zu Beginn des Projektes wurde ein MockUp gezeichnet um 1. Ideen grafisch darzustellen sowie die Machbarkeit und bestimmte Funktionalitäten zu testen.



Interface

**package** model;

**import** processing.core.PApplet;

**public** **interface** Draw {

**public** **void** draw(PApplet window);

}

Vererbung

public class Paco extends Entity{}

GitHub

Zur Codeversionierung wurde ein Repository in GitHub erstellt und stets aktualisiert / gepflegt: <https://github.com/MarioMarrazza/PacManGrande>

Schlussreflektion

Erfahrungswerte

Sehr vieles durfte ich bei der Entwicklung dieses Projektes mitnehmen und mir viele Erfahrungswerte aneignen welche mir in Zukunft bestimmt eine Lehre sein werden.

Es soll keine Ausrede sein, doch möchte ich den Fakt erwähnen das ich leider sehr viele Fehlstunden hatte in den Unterrichten des Moduls 226b.. Diese fehlende Kentnisse und Aneignung von Know-How hat mir viele Mühen bereitet bei der Arbeit an dem Projekt. Alles in einem kann ich vorallem das Sprichwort «Übung macht den Meister» mitnehmen und erkennen das es bei der Programmierung um nichts anderes als übung übung übung geht und man sich nur weiterentwickeln kann wenn man sich selbstständig hinsetzt und es versucht oder solange übt bis es klappt. (Nicht immer aber meistens)

Stolpersteine

Zu Beginn sah es sehr einfach aus und ich habe mir keine grossen Gedanken über die Machbarkeit des Projektes gemacht. Jedoch viel nach der Zeichnung des Mockups und Erstellung des Klassendiagramms sehr schnell auf das ich das Projekt doch sehr stark unterschätzt habe und es doch nicht so einfach ist wie man anfangs denkt.

Einer der grössten Stolpersteine für mich waren es den Anfoderungen entsprechend zu entwickeln. Bei dieser Arbeit bin ich mir oft Bewusst geworden stark an C# gewöhnt zu sein und bin mit der Verwendung von PApplet oder der Vererbungn in Java nicht ganz klar gekommen. Ich werde auf jeden Fall noch weitere daran arbeiten und versuchn das Spiel zu vervollständigen da ich diese Übung bestimmt brauchen werde.

Danksagung an Herrn Schirmer

Für Ihre Unterstützung und speziell für Ihren Aufwand den Sie in Ihrer Freizeit auf sich genommen haben um mir / uns zu helfen möchte ich mich recht Herzlich bei Ihnen bedanken Herr Schirmer! Ohne Ihre Hilfe wäre ich aufgeschmiessen gewesen und mir bei vielen Punkten die Haare aus dem Kopf gerissen.

# Abbildungverzeichnis

[Abbildung 1: Titelbild - Quelle: https://www.mentalfloss.com/article/90920/10-fast-facts-about-pac-man 1](#_Toc27504989)

[Abbildung 2: UML - Klassendiagramm 6](#_Toc27504990)

[Abbildung 3: Codesnippet Stand - 17.12.2019 8](#_Toc27504991)

[Abbildung 4: Codesnippet aus PacManGrande Stand: 17.12.2019 10](#_Toc27504992)

[Abbildung 5: Funktionsaufruf - Buffered Reader 12](#_Toc27504993)

[Abbildung 6: Buffered Reader Funktion 13](#_Toc27504994)

# Tabellenverzeichniss

[Tabelle 1: Änderungskontrolle 2](#_Toc27504987)

# Literaturverzeichnis

*ICT-Berufsbildung Schweiz*. (ohne Datum). Abgerufen am 27. 08 2018 von https://www.ict-berufsbildung.ch

Lambert, J., & Frye, C. (2016). *Microsoft Office 2016.* Heidelberg: dpunkt.verlag GmbH.

# Selbstständigkeitserklärung

Ich habe diese Arbeit selbständig verfasst und keine anderen als die angegebenen Quellen verwendet. Wörtlich zitierte Textstellen sind als solche gekennzeichnet und gemäss den Vorgaben im Abschnitt «Zitat- und Quellenangaben» auf 17 ausgewiesen und formatiert.

Ich bestätige, dass ich den in den Richtlinien zum Moduljournal Abschnitt «Zitat- und Quellenangaben» gelesen und verstanden habe. Mir ist klar, dass mein Moduljournal elektronisch auf Plagiate überprüft wird. Die Sanktionen beim Erstellen von Plagiaten sind mir bekannt.

St. Gallen, 17.12.2019



# Anhang

Bewertungsraster

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Marrazza, Mario** |  | | | |
| **Klasse** | **IFZ-826-003a** | | | |
| **Fachliche Dokumentation** | **Hanok** | **max. Pkt.** | **erreichte Punkte** | **Begründung bei Abzug** |
| Klassendiagramm enthält mind. 2 Vererbungen | 2 | 3 |  |  |
| Klassendiagramm enthält abstrakte Basisklasse | 2 | 3 |  |  |
| Klassendiagramm enthält mindestens ein Interface | 2 | 3 |  |  |
| Vererbung anhand Code-Snippet aufgezeigt | 3 | 3 |  |  |
| Abstrakte Klasse anhand Code-Snippet aufgezeigt | 3 | 3 |  |  |
| Interface anhand Code-Snippet aufgezeigt | 3 | 3 |  |  |
| Vererbung und Interfaces plausibel und sinnvoll eingesetzt | 1 | 6 |  |  |
| mind. sinnvolle 3 Unit-Tests für unterschiedliche Klassen vorhanden und dokumentiert | 4 | 6 |  |  |
| Mockup-Test vorhanden und beschrieben | 4 | 3 |  |  |
| Hilfestellungen aufgeführt |  | 3 |  |  |
| Coding Style: Javadoc |  | 2 |  |  |
| Coding Style: Namenskonventionen |  | 2 |  |  |
| Coding Style: Einrückungen |  | 2 |  |  |
| Assoziationen im Klassendiagramm korrekt |  | 5 |  |  |
| **Dokumentation formal** |  |  |  |  |
| Vollständig, inkl. Inhaltsverzeichnis |  | 1 |  |  |
| Allg. Beschreibung des Projekts |  | 2 |  |  |
| Fachlich richtig, inkl. Fachsprache |  | 3 |  |  |
| Sauberkeit, Layout |  | 3 |  |  |
| Profimässiger Tools-Einsatz (z.B. sinnvolle Formatierung Code-Snippets (kein DarkMode!), Abbildungen mit Bildunterschriften und Referenzen) |  | 5 |  |  |
| Nachvollziehbarkeit |  | 3 |  |  |
| Sprache (Stil/Orthographie) |  | 3 |  |  |
| **Codeversionierung** |  |  |  |  |
| Codeversionierung genutzt |  | 3 |  |  |
| sinnvolle Commits (Anzahl, Commit-Messages) |  | 5 |  |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Arbeitsjournal** |  |  |  |  |
| Vollständig (mind. Alle Lehrwerkstattblöcke vorhanden) |  | 5 |  |  |
| Ablauf nachvollziehbar (Ziele und Probleme ersichtlich) |  | 5 |  |  |
|  |  |  |  |  |
| **Fazit** |  |  |  |  |
| Reflexion: was habe ich gelernt, wo gab es Schwierigkeiten (analog reguläres WISS Moduljournal) |  | 3 |  |  |
| **Excellenz (max 5 Punkte)** |  | 5 |  | Serialisierung eingebunden und genutzt, Bewegungsablauf Snake; Zustandsautomat Spiel sauber beschrieben, DB-Anbindung erstellt |
| **Gesamt** |  | **93** | **0** |  |
| **Note** |  |  | **1.00** |  |

PS: Meine Handnotizen sin dim Anhang direkt eingefügt worden sodass ich diese nicht eingescannt drucken muss und sinnlos Papieren verschwenden..