

# Projektplan

Spiel: Snake

Muhammed Emin Boz | Modul 326

Projektstart: 26.03.

Projektende: 07.05.

| Aufgabe                                  | Von:       | Bis:       | Soll:   | Ist  |
|--|------------|------------|---|--|
| Use Case Diagramm erstellen und anfangen | 26.03.2020 | 02.04.2020 | Datei für das Use Case Diagramm erstellt und Use Case Fälle grob gezeichnet.  | Datei wurde erstellt und Fälle wurden grob gezeichnet.   |
| Sequenzdiagramm erstellen und anfangen   | 26.03.2020 | 26.03.2020 | Datei für das Sequenzdiagramm erstellt und einige Objekte und Funktionen grob gezeichnet haben.                             | Datei wurde erstellt und einige Objekte und Funktionen wurden grob gezeichnet                                  |
| Pflichtenheft erstellen und anfangen     | 26.03.2020 | 02.04.2020 | Datei für das Pflichtenheft erstellt und die angegebenen Punkte im Inhaltsverzeichnis alle beschrieben.                     | Datei wurde erstellt und alle Punkte, wenn auch manche grob, beschrieben.                                      |
| Klassendiagramm erstellen und anfangen   | 26.03.2020 | 02.04.2020 | Datei für das Klassendiagramm erstellt, Klassen und Attribute grob gezeichnet haben.  | Datei wurde erstellt, und einige Klassen und Attribute wurden grob gezeichnet.                                 |
| Pflichtenheft aktualisieren und ergänzen | 02.04.2020 | 07.04.2020 | Fehler im Pflichtenheft und fehlende Beschreibungen sollten ergänzt sein, und das Inhaltsverzeichnis soll angepasst werden. | Fehler und fehlende Beschreibungen wurden ergänzt, und das Inhaltsverzeichnis wurde dementsprechend angepasst. |
| Use Case Diagramm ergänzen               | 02.04.2020 | 07.04.2020 | Das Use Case Diagramm sollte alle richtigen Use Case Fälle, und dazu die richtigen Relationen haben.                        | Nach den bisherigen Informationen alle Fälle gezeichnet, jedoch Ergänzung in der Zukunft nötig.                |

|  |                     |                     |   |  |
|--|---------------------|---------------------|---|--|
| Kardinalitäten im Klassendiagramm zufügen, Relationen ergänzen, und unnötige Klassen im Klassendiagramm entfernen. | 15.04.2020<br>16:45 | 15.04.2020<br>17:48 | Das Klassendiagramm soll die nötigen Klassen, Kardinalitäten und richtigen Relationen haben. Zudem soll das Klassendiagramm ordentlich aussehen.                                  | Das Klassendiagramm hat die nötigen Klassen (unnötige wurden entfernt), Kardinalitäten wurden gesetzt und die Relationen zwischen den Klassen wurden korrigiert. Das Klassendiagramm wurde am Schluss noch visuell ordentlich designt. |
| Java Projekt und Klassen gemäss Klassendiagramm erstellen und das Projekt im richtigen Verzeichnis ablegen         | 15.04.2020<br>18:00 | 15.04.2020<br>18:10 | Das Java Projekt soll erstellt und in dem GitHub Ordner für Modul 326 abgelegt sein. Dann sollten die Klassen gemäss Klassendiagramm erzeugt werden.                              | Das Java Projekt wurde erstellt, im richtigen Ordner abgelegt und die Klassen gemäss Klassendiagramm wurden erzeugt.   |
| Attribute und Funktionen im Java Projekt und eine bewegende Spielfigur programmieren                               | 16.04.2020<br>16:00 | 16.04.2020<br>18:30 | Es sollten die richtigen Attribute gesetzt sein und es soll eine Spielfigur geben, die man mit den Pfeiltasten auf der Tastatur bewegen kann.                                     | Die richtigen Attribute wurden gesetzt und es wurde eine Spielfigur programmiert, die man mithilfe der Pfeiltasten bewegen kann.   |
| Vier Wände (Spielfeldrand) an denen man Kollidieren kann   | 16.04.2020<br>16:00 | 16.04.2020<br>18:30 | Vier Wände, bzw. die äussersten Spielfeldpositionen, an denen man bei dagegen gehen kollidiert.   | Vier Wände, bzw. die äussersten Spielfeldpositionen, an denen man bei dagegen gehen kollidiert.  |
| Zielobjekt (das Essen, welches von der Spielfigur aufgegessen wird) programmieren                                  | 17.04.2020          | 19.04.2020          | Ein Viereck, die eine andere Farbe als die des Spielers hat. Wenn die Spielfigur auf der gleichen Position ist, wie dieses Viereck, soll es an einer anderen Position erscheinen. | Ein Viereck mit einer anderen Farbe als die Spielfigur, die an einer anderen Position erscheint, wenn die Spielfigur auf der gleichen Position ist wie es selbst.  |

|  |                     |                     |  |  |
|--|---------------------|---------------------|--|--|
| Die Funktion für die Hinzufügung von Vierecken (Der Schwanz) hinter dem Kopf der Spielfigur (die man steuert). | 20.04.2020          | 21.04.2020          | Jedes Mal, wenn die Spielfigur Essen aufisst, also die Position des Spielers gleich wie die Position des Zielobjektes ist, soll hinter dem Kopf des Spielers ein Viereck hinzugefügt werden und hinterher bewegen. | Bei Aufessen des Zielobjektes wird ein neues Viereck hinter dem Spieler hinzugefügt und bewegt sich dementsprechend dem Spieler hinterher. |
| Selbst-Kollidieren programmieren   | 22.04.2020          | 22.04.2020          | Wenn die Position des Spielers die gleiche ist wie die des Schwanzes soll man von neu anfangen.  | Wenn die Position des Spielers die gleiche ist wie die des Schwanzes, fängt man von neu an.  |
| Die Spielfigur mit einem Bild zeichnen.  | 23.04.2020<br>14:20 | 23.04.2020<br>15:30 | Die Spielfigur soll ein 64x64 Pixel grosses Bild sein.   | Die Spielfigur ist eine 64x64 Pixel grosses Bild.  |
| Statistiken einfügen   | 25.04.2020<br>15:00 | 25.04.2020<br>16:30 | Punkte, Rekord und Gewonnene Runden anzeigen und addieren.   | Punkte, Rekord und Gewonnene Runden werden angezeigt.  |
| Menü programmieren   | 28.04.2020<br>12:00 | 28.04.2020<br>15:30 | Ein Menü mit drei JButtons wo man die Schwierigkeit auswählen kann.  | Ein Menü mit drei JButtons wo man die Schwierigkeit auswählen kann.  |
| Schwierigkeit Einfach und Mittel programmieren   | 30.04.2020<br>09:50 | 01.05.2020<br>18:00 | Der Gegner soll je nach Schwierigkeit verschieden gut spielen können.  | Der Gegner spielt je nach Schwierigkeit verschieden gut.   |
| Schwierigkeit Schwer programmieren   | 02.05.2020          | 04.05.2020          | Ein Gegner der schwerer ist zu besiegen als Mittel.  | Ein Gegner der schwerer ist zu besiegen als Mittel.  |
| Statistiken ergänzen / anpassen und Sieger darstellen  | 05.05.2020          | 07.05.2020          | Gewonnene Spiele und insgesamt gewonnene Spiele anzeigen und den Sieger darstellen (String).   | Gewonnene Spiele und insgesamt gewonnene Spiele und den Sieger werden angezeigt. (String).   |