



Projekt: Tetris

Gruppenprojekt Software Architektur

Inhaltsverzeichnis

1.	Einleitung	3
2.	Vorbereitung	3
3.	Spielbedingungen	4
4.	Anforderungen	5
5.	Konzept	6
6.	Realisation	6

1. Einleitung

Ziel dieses Projekts soll ein kleines, einfaches Tetris Spiel sein. Die Zusammenarbeit am Code wird über gitlab ermöglicht. Das Klassendiagramm soll als Basis für die Entwicklung dienen.

2. Vorbereitung

Lernen Sie die im Kurs eingerichtete git Umgebung anzuwenden. Schauen Sie dazu folgendes Video auf YouTube:

<https://youtu.be/HVsySz-h9r4>

Sobald Sie Sich fühlen im Umgang mit git und der Konsole (nicht grafisch!), können Sie beginnen, das Repository für Ihr Gruppenprojekt einzurichten:

Name: login.gruppe(Nr).tetris (passen Sie die Zahl gemäss Gruppe an)

Beispiel: login.gruppe1.tetris

Verbinden Sie Visual Studio mit Ihrem neu errichteten git-repository. Erstellen Sie eine neue Solution mit einem Windows Forms Projekt. Löschen Sie das vorhandene Form1.cs, Sie werden später ein eigenes hinzufügen.

Committen und Pushen Sie Ihre Solution nun auf den Server. Denken Sie dabei daran, eine aussagekräftige Beschreibung im Commit zu erstellen!

Nun sollen alle Projektmitglieder die neu erstellte Solution vom Server aus dem Repository pullen. Jedes Teammitglied sollte nun auf dem gleichen Stand sein.

Es wird von nun an erwartet, dass Sie sowohl den Code als auch die Dokumentation mindestens täglich vor Arbeitsende (ausstempeln) committen und pushen. Denken Sie daran, dass Sie am Ende des Projekts einen Nachweis liefern müssen, wer was gemacht hat.

3. Spielbedingungen

Diese Version von Tetris wird keine Formen verwenden, sondern sich an farbigen Panels orientieren:

- Befinden sich drei gleichfarbige Panels unter- oder nebeneinander, verschwinden sie
- Panels die über verschwindenden Panels sind, fallen auf das nächste Panel oder auf den «Boden» herunter
- Wenn Panels verschwinden, wird die Fallgeschwindigkeit der Panels erhöht
- Jedes Panel, das verschwindet ergibt eine Punktzahl mit Fallgeschwindigkeitsmultiplikator (Multiplikator und Basispunktzahl kann frei gewählt werden)
- Das Spiel ist dann zu Ende, wenn nur noch ein Panel in der obersten Zeile platziert werden kann

4. Anforderungen

Nr	Anforderung
FA1	Der Benutzer kann das Spiel starten
FA2	Der Benutzer kann die Schwierigkeit einstellen, indem er mehr Farben zulässt. Standard sollen vier Farben sein
FA3	Das Spiel verfügt über eine Pausenfunktion
FA4	Der Spieler kann sich nach Spielende im Highscore eintragen, muss aber nicht
NFA1	Das Spiel ist ansprechend gestaltet
NFA2	Die Icons sind passend gewählt
NFA3	Die Applikation braucht nicht mehr als 5 Sekunden zum Starten
NFA4	Der Highscore ist verständlich und ästhetisch gestaltet
NFA5	Das Spielfeld ist 8 Spalten breit und 13 Zeilen hoch
NFA6	Die Panele werden zufällig mit Farben gefüllt
NFA7	Die Ablaufsteuerung soll getrennt der Anzeige implementiert werden

5. Konzept

Erstellen Sie eine geeignete Software Architektur und beschreiben Sie in Ihrer Dokumentation, wie Sie die Anforderungen und Spielbedingungen umsetzen wollen. Denken Sie dabei auch an ein Layout für das UI.

Planen Sie die entsprechenden Arbeitspakete, wer was macht, indem Sie Issues in gitlab erstellen und diese dann direkt zuweisen. Bedenken Sie, dass jedes Teammitglied gleich viel zur Dokumentation sowie zum Code beitragen muss.

Sobald Sie mit dem Konzept fertig sind, besprechen Sie dieses mit der Kursleitung. Denken Sie daran, dass Sie nicht ohne Freigabe der Kursleitung mit der Implementation anfangen dürfen! Planen Sie Ihre Besprechung deswegen sorgfältig ein und vereinbaren Sie falls nötig, einen Termin.

6. Realisation

Realisieren Sie Ihr Projekt und dokumentieren Sie allfällige Stolpersteine oder Überlegungen. Comitten und pushen Sie den Code regelmässig und verwenden Sie gute Beschreibungen!