

---

# Projekt MasterMind

## Projekt Management Plan

---

Eine Projektarbeit des Studiengangs Informatik im Modul Programmieren 2

Projekt	Projekt MasterMind	
Dokument	Projekt Management Plan	
Auftraggeber	Hochschule Luzern, Technik & Architektur	
Modul	Programmieren 2	
Projektgruppe	<b>Simon Duss</b> Kantonsstrasse 37 6207 Nottwil  Tel. +41 79 455 33 79 simon.duss@stud.hslu.ch	<b>Christoph Enzmann</b> Schibiweg 11 6006 Luzern  Tel. +41 77 424 87 17 chris@amigo-online.ch
	<b>Fabio Figlioli</b> Moosmattstr. 54 6005 Luzern  Tel. +41 76 367 11 05 fabio.figlioli@stud.hslu.ch	<b>Philipp Flückiger</b> Haldenring 8 6020 Emmenbrücke  Tel. +41 76 453 03 20 philipp.flueckiger@stud.hslu.ch
	<b>Thomas Galliker</b> Panorama 6123 Geiss  Tel. +41 79 504 80 70 thomas.galliker@stud.hslu.ch	
	<b>Prof. Roger Diehl</b>	
Dozenten		
Letzte Änderung	21. Mai 2009, 23:15:00 Uhr	

# Inhalt

<b>1</b>	<b>Übersicht .....</b>	<b>4</b>
1.1	Ausgangslage .....	4
1.2	Projektauftrag .....	4
1.3	Vorgehen .....	4
1.3.1	Iterationsschritte .....	4
1.3.2	Vorgehensweise .....	4
1.3.3	Geforderte Abgaben .....	5
<b>2</b>	<b>Projekt Organisation.....</b>	<b>6</b>
2.1	Rollen und Zuständigkeiten.....	6
2.2	Projektplan .....	7
2.3	Aufwandschätzung.....	8
<b>3</b>	<b>Risikoanalyse.....</b>	<b>9</b>
3.1	Komplexität .....	9
3.2	Zeitaufwand.....	9
3.3	Datenverlust .....	9
<b>4</b>	<b>Qualitätssicherung .....</b>	<b>9</b>
4.1	Dokumentenversionisierung .....	9
4.2	Code Versionskontrolle .....	9
4.3	Testautomatisierung .....	9
4.4	Anhang 1: Spielregeln Mastermind .....	10
4.4.1	Allgemeines .....	10
4.4.2	Bewertungsregeln.....	10
4.4.3	Optionen .....	10

## Abbildungsverzeichnis

Error! No table of figures entries found.

## Tabellenverzeichnis

Error! No table of figures entries found.

## Dokumentenversionisierung

Version	Datum	Autor	Beschreibung
1.0	22.04.2009	Galliker	Dokument von Vorlage erstellt
1.1	22.04.2009	Galliker	Projektauftrag eingefügt
1.2	22.04.2009	Duss	Projektplan erstellt
1.3	23.04.2009	Galliker	Projektplan bearbeitet und eingefügt
1.4	24.04.2009	Galliker	Risikoanalyse und Qualitätssicherung

# 1 Übersicht

## 1.1 Ausgangslage

Im Zusammenhang mit dem Modul "Programmieren 2" (PRG2) führt die Hochschule Luzern Technik & Architektur ein praxisorientiertes Softwareprojekt durch. Dabei erhalten die Projektgruppen einen bestimmten Programmierauftrag, welchen sie zusammen umsetzen müssen. Der Schwerpunkt dieses Auftrags liegt sowohl bei der eigentlichen Programmierung sowie bei der projektorientierten Vorgehensweise

## 1.2 Projektauftrag

Entwickeln Sie ein Mastermind Programm, das sowohl die Rolle des Setzers / Bewerter als auch die Rolle des Raters übernehmen kann. Beide Rollen sollen wahlweise vom Computer oder vom Spieler übernommen werden können, d.h. für beide Funktionen braucht es eine Spiel-Engine und ein User-Interface. Jederzeit soll ein begonnenes Spiel abspeicherbar bzw. ein gespeicherter Spielstand wieder ladbar sein. Ein grafisches User Interface (GUI) ermöglicht ein benutzerfreundliches (und attraktives) Spielvergnügen.

## 1.3 Vorgehen

Da der gestellte Auftrag Aspekte enthält, die aufwendig werden können (z.B. das GUI) bietet sich ein iterativ-inkrementelles Vorgehen an. Zu Beginn konzentriert man sich auf den wesentlichen Kern der Funktionalität und fügt in jedem Iterationsschritt weitere Funktionalitäten hinzu.

### 1.3.1 Iterationsschritte

Folgende Iterationsschritte werden im Rahmen des Projekts vom Auftraggeber vorgegeben und können im Verlauf des Projekts ergänzt und reorganisiert werden.

Schritt	Bezeichnung	Resultate
1	Mastermind Game Engine	<ul style="list-style-type: none"><li>• Der Computer gibt eine Kombination von Farben oder Mustern vor, nimmt die Versuche des Users entgegen und bewertet diese. Das User Interface ist einfach und erfolgt über die Konsole. Die Optionen des Spieles sind fix vorgegeben.</li><li>• Es liegt ein Konzept für das graphische User Interface (GUI) vor.</li></ul>
2	Graphical User Interface	<ul style="list-style-type: none"><li>• Ein (einfaches) graphisches User Interface wird zur Verfügung gestellt.</li></ul>
3	Save Game Option	<ul style="list-style-type: none"><li>• Der aktuelle Spielstand kann abgespeichert werden, so dass das Spiel zu einem späteren Zeitpunkt fortgesetzt werden kann.</li></ul>
4	Advanced Game Options	<ul style="list-style-type: none"><li>• Der User kann zwischen verschiedenen Optionen für das Spiel wählen (Bsp: Anzahl Farben, Farben doppelt oder nicht, graphische Formen statt Farben, ...)</li></ul>
5 (Optional)	Artificial Intelligence	<ul style="list-style-type: none"><li>• Der User kann eine Farbkombination vorgeben, die vom Computer geraten wird. Hierbei bewertet der User die Versuche des Computers.</li></ul>
6 (Optional)	Multiplayer Game Mode	<ul style="list-style-type: none"><li>• Das Spiel kann übers Netzwerk gespielt werden.</li></ul>
7 (Optional)	Java Mobile MasterMind	<ul style="list-style-type: none"><li>• Das Spiel kann auf einem Java Mobile Phone (Java Micro Engine) gespielt werden.</li></ul>

### 1.3.2 Vorgehensweise

Die folgenden Tasks 1 bis 4 beschreiben Tätigkeiten, die in jeder Iteration für die Erfüllung des Auftrags notwendig sind. Da ein iterativ-inkrementelles Vorgehen gewählt wird, sind diese Tätigkeiten in den einzelnen Iterationsschritten typisch wiederholt/aufbauend durchzuführen.

In den Theorieeinheiten erhalten Sie genauere Informationen über den Detaillierungsgrad der zu erstellenden Dokumente. Es geht in diesem Projekt nicht darum, eine Vielzahl umfangreicher Dokumente zu erstellen, sondern nur das Wesentliche zu dokumentieren und die "Funktionsweise" und den Nutzen der einzelnen Dokumente zu verstehen.

## Task 1: Projekt Management Plan

Gehen Sie von den Vorgaben des gegebenen Projektplans aus. Erstellen Sie bis Semesterwoche 9 [10] einen Projekt Management Plan (max. 2 Seiten). Daraus müssen insbesondere die Rollen und Verantwortlichkeiten klar ersichtlich sein.

### Task 2: Kundenanforderungen/Systemspezifikation

Im Rahmen des ersten Iterationsschrittes erstellen Sie das Kundenanforderungs-Dokument. Erstellen Sie bzw. erweitern Sie das Systemspezifikations-Dokument für jeden Iterationsschritt.

### Task 3: Umsetzung

Implementieren Sie Ihr Projekt. Halten Sie sich bei der Implementation an den "Program style guide" von Appendix J unseres Lehrbuches. Dokumentieren Sie Ihre Klassen mit Javadoc.

### Task 4: Test

Erstellen Sie einen Test Plan.

Unit-Tests: Einzelne Klassen können Sie mit Hilfe von JUnit-Tests und/oder mit Hilfe ihrer main()-Methoden testen.

System-Tests: Es sind mindestens 5 Testfälle für Ihre Anwendung zu spezifizieren, durchzuführen und zu protokollieren.

### Task 5: Lessons Learned

Erstellen Sie begleitend zum Projekt ein kleines Dokument "Lessons Learned". Darin halten Sie kompakt fest:

- Was hat gut geklappt?
- Was hat nicht so gut geklappt?
- Was sollten Sie in einem zukünftigen Projekt genauso wieder machen?
- Was müssen Sie verbessern? Wie?

### 1.3.3 Geforderte Abgaben

Die Abgabe erfolgt auf ILIAS, und zwar in den Ordner "Briefkasten" Ihres Kurses.

- Projekt Plan mit Ihren Ergänzungen
- Kundenanforderungs-Dokument
- Systemspezifikations-Dokument
- Source Code mit JavaDoc
- Test Plan
- Lessons Learned

Bei jedem Artefakt müssen die Autoren / Autorinnen ersichtlich sein. Rechtschreibfehler sind die Ausnahme. Dokumentationshinweise und -vorlagen finden Sie auch unter: <http://edu.enterpriselab.ch/htagil>  
Die Abgabe und das Feedback zur Projektarbeit werden in den einzelnen Kursen etwas unterschiedlich erfolgen. Genaueres erfahren Sie von den Dozierenden.

Termin	Beschreibung
SW9	Kick-Off "Meeting" im Unterricht.
SW10	Projekt-Management Plan (PMP), Systemspezifikation (1. Entwurf).
SW11	Systemspezifikation, Software-Architektur, Console-basierte Game Engine.
SW12	GUI 1. Version.
SW13	GUI erweitert, Store & Recall von Spielständen, Testspezifikation, Quellprogramme.
SW14	Quellprogramme, Protokoll der Systemtests, "Lessons learned".

Der/die Dozierende der jeweiligen Kursdurchführung legt die Details für exakte Abgabetermine innerhalb der Woche, Art und Ort der Abgabe (einsehbar im ILIAS, auf Papier, gezipptes Projekt, u.s.w.) fest. Findet das Kick-Off nicht in der SW 9 statt, so passt der/die Dozierende den Plan entsprechend an.

## 2 Projekt Organisation

### 2.1 Rollen und Zuständigkeiten

<b>Kontakt</b>	<b>Telefon</b>	<b>E-Mail Adresse</b>
Simon Duss Kantonsstrasse 37 6207 Nottwil	+41 79 455 33 79	simon.duss@stud.hslu.ch
Christoph Enzmann Schibiweg 11 6006 Luzern	+41 77 424 87 17	chris@amigo-online.ch
Fabio Figlioli Moosmattstr. 54 6005 Luzern	+41 76 367 11 05	fabio.figlioli@stud.hslu.ch
Philipp Flückiger Haldenring 8 6020 Emmenbrücke	+41 76 453 03 20	philipp.flueckiger@stud.hslu.ch
Thomas Galliker Panorama 6123 Geiss	+41 79 504 80 70	thomas.galliker@stud.hslu.ch

## 2.2 Projektplan

	Duss	Enzmann	Figlioli	Flückiger	Galliker	SW 9	SW 10	SW 11	SW 12	SW 13	SW 14
<b>Iterationsschritt 1 - MasterMind Game Engine</b>											
Projekt-Management-Plan erstellen	C	C	C	C	R						
Systemspezifikationen	C	C	R	R	C						
Konzept für GUI	C	C	C	C	R						
Use-Case definieren	C	C	R	C	C						
Game Engine entwickeln	S	R	S	S	S						
Test	R	R	S	S	S						
Lessons Learned	R	S	S	S	S						
<b>Iterationsschritt 2 - Graphical User Interface</b>											
Software-Architektur	R	R	S	S	S						
MasterMind GUI entwickeln	R	S	S	S	S						
Design von GUI Objekten	R	S	S	S	S						
Settings GUI entwickeln	S	S	S	S	R						
Test	R	S	S	S	S						
Lessons Learned	S	R	S	S	S						
<b>Iterationsschritt 3 - Save Game Option</b>											
Save/Load des Spielstandes		R									
Save/Load von Einstellungsprofilen					R						
Test	S	S	S	S	R						
Lessons Learned	S	S	R	S	S						
<b>Iterationsschritt 4 - Advanced Game Options</b>											
Umsetzung	S	S	S	R	S						
Test	R	R	R	R	R						
Lessons Learned	S	S	S	R	S						
<b>Iterationsschritt 5: Artificial Intelligence</b>											
Umsetzung		R									
Test	S	R	S	S	S						
Lessons Learned	S	S	S	S	R						
<b>Iterationsschritt 6: Multiplayer Game Mode</b>											
Umsetzung		C			R						
Test	R	R	R	R	R						
Lessons Learned	R	S	S	S	R						
<b>Iterationsschritt 7: MasterMind Mobile Edition</b>											
Umsetzung		R									
Test	R	R	R	R	R						
Lessons Learned	S	R	S	R	R						

### Legende

- R Responsible
- A Accountable
- S Supportive
- C Consulted
- I Informed

## 2.3 Aufwandschätzung

Iterationsschritt	Teilaufgabe	Aufwand	Subtotal
1	• Projekt Management Plan	4h	36h
	• Anforderungsanalyse	4h	
	• Systemspezifikation	4h	
	• Game Engine (Console Version)	16h	
	• GUI Konzept	4h	
	• Testplan erstellen	4h	
2	• Projekt Management Tasks	1h	26h
	• Umsetzung: GUI	24h	
	• Testplan ergänzen	1h	
3	• Projekt Management Tasks	1h	14h
	• Umsetzung: Game Save Funktionen	8h	
	• Anpassungen: GUI	4h	
	• Testplan ergänzen	1h	
4	• Projekt Management Tasks	1h	42h
	• Umsetzung: Advanced Game Options	40h	
	• Testplan ergänzen	1h	
5	• Projekt Management Tasks	1h	53h
	• Umsetzung: Artificial Intelligence	50h	
	• Testplan ergänzen	2h	
6	• Projekt Management Tasks	1h	43h
	• Umsetzung: Multiplayer Game Mode	40h	
	• Testplan ergänzen	2h	
7	• Projekt Management Tasks	1h	43h
	• Umsetzung: Java Mobile MasterMind	40h	
	• Testplan ergänzen	2h	
		<b>Total</b>	<b>257h</b>

## 3 Risikoanalyse

### 3.1 Komplexität

Es ist möglich, dass einige Iterationsschritte wegen ihrer zu hohen Komplexität nicht oder nur teilweise umgesetzt werden können. Der Zeitplan des Projekts ist zu knapp, sodass fehlendes Know-How nicht innerhalb nützlicher Zeit angeeignet werden kann.

### 3.2 Zeitaufwand

Da das Projekt in einem sehr kurzen Zeitraum stattfinden, bleibt für die Umsetzung der einzelnen Iterationsschritte nur sehr wenig Zeit. Das Projektziel, bzw. einzelne Teilziele können unter dem Zeitdruck vernachlässigt werden.

### 3.3 Datenverlust

Programmcode und/oder Dokumentationen können durch logische oder physische Ursachen zerstört werden. Sämtliche Mitglieder des Projekts sind für die Sicherung der projektrelevanten Daten eigenständig verantwortlich.

## 4 Qualitätssicherung

### 4.1 Dokumentenversionisierung

In jedem projektrelevanten Dokument wird eine Dokumentenversionisierung geführt. In dieser Versionisierung ist die Versionsnummer, der Autor, eine Kurzbeschreibung der Änderung sowie das Datum ersichtlich.

### 4.2 Code Versionskontrolle

Um Änderungen am Quellcode zu dokumentieren, wird die Code-Versionierungssoftware "Subversion", kurz SVN verwendet. Mit Hilfe dieser Software können Änderungen verfolgt und überprüft werden.

### 4.3 Testautomatisierung

In der Entwicklungsumgebung werden Testfälle programmiert, welche die Game Engine automatisch auf allfällige Fehler überprüft. Diese Testprozeduren müssen nach jeder Änderung durchlaufen werden um die Fehlerfreiheit zu garantieren.

## 4.4 Anhang 1: Spielregeln Mastermind

### 4.4.1 Allgemeines

Mastermind ist ein Spiel für zwei Spieler. Ein Spieler gibt eine Kombination von Farben vor, und der andere versucht diese in einer vorgegebenen Anzahl von Versuchen zu erraten. Der Vorgebende bewertet dabei jeden Versuch des Ratenden nach seiner Korrektheit mit schwarzen und weissen Punkten.

### 4.4.2 Bewertungsregeln

Schwarzer Punkt: richtige Farbe an der richtigen Stelle erraten Weisser Punkt: richtige Farbe aber an die falsche Stelle gesetzt Dabei hat die Reihenfolge der gegebenen Punkte nichts mit der Reihenfolge in der geratenen oder der vorgegebenen Kombination zu tun. Jede Farbe in der Vorgabe und im Versuch darf nur einmal zur Bewertung herangezogen werden.

Kombination	r	b	y	r	r
Versuch	r	y	b	b	b
Punkte					

### 4.4.3 Optionen

- Die Spieloptionen können mit diesen Parametern gewählt werden: Farben: 4-8 Länge der Kombination: 4-6 maximale Versuchsanzahl: 6-12 mehrfache Farben erlaubt: Ja/Nein
- Computer gibt vor
- Der Computer gibt eine zufällige Kombination mit den gewünschten Spezifikationen vor. Danach kann der Spieler seine Versuche starten, und diese vom Computer bewerten lassen.
- Hilfen Ratehilfe: Der Computer gibt aus den noch möglichen Kombinationen zufällig eine vor. Hilfe: Der Computer verrät die Farbe an der gewünschten Position. Lösen: Der Computer gibt die Kombination aus. Aus den Optionen und den benötigten Versuchen wird dann eine Punktzahl errechnet.
- Spieler gibt vor
- Der Spieler gibt eine Kombination vor, oder lässt sich eine zufällig erzeugen. Der Computer rät eine Kombination, und der Spieler bewertet diese dann. Bei Falscher Bewertung kommt es zu einem Fehler.