



Softwareprojekte im Informatikunterricht

Softwareprojekte: Chancen & Herausforderungen

- Großer Nutzen für Schüler – im Erfolgsfall
 - Design- und Testphasen (und deren Bedeutung) erleben
 - Implementierung: deutliche Vertiefung der Programmierfähigkeiten
 - Projekt- und Teamdynamik
- Hoher Zeitaufwand
 - für Schüler(innen),
 - Ggf. auch für Lehrer(innen)
- Gefahr des Scheiterns
 - Komplexität der Aufgabe
 - Technische (Detail-) Probleme häufig im Vordergrund

1

Erfahrungsaustausch

Erfahrungsaustausch (Mind-Map)



Projektideen: Welche Software-Projekte bzw. Langzeit-Hausaufgaben nutzen Sie im Unterricht?



Ratschläge guter Praxis: Wie organisieren Sie Projekte?
Was hat sich bewährt, was weniger?



Ratschläge und Erfolgsfaktoren

Bekanntes Fachwissen anwenden

- Softwareprojekte bieten schon für sich genommen Herausforderungen für Schüler(innen) dar, z. B.
 - Selbstorganisation
 - Gruppendynamik
 - Anwendung bekannten Wissens im neuen Kontext
 - Umgang mit langen Quelltexten
 - ...
- Projekte sollten daher mit vorhandenem Fachwissen lösbar sein.



Projekte und Unterricht müssen zusammenpassen – inhaltlich und methodisch.

Klare Aufgabenstellung

- Verschiedene Erwartungsniveaus definieren, z. B.
 - Mindestanforderung als klar nachprüfbare Anforderungen
 - Möglichkeiten für (kreative) Vertiefung / Erweiterung (inhaltlich und ggf. im Notenniveau).
- Bewertungskriterien im Vornhinein offen legen



Beispiel: Chatbot ELIZA

Phasierung

- Projekt in Phasen strukturieren, z. B.
 - Design – Implementierung – Test
 - Spielidee – Spiel-„Engine“ – Spielumsetzung
- Meilensteinergebnisse definieren und absichern
 - gemeinsame (!) Zwischenergebnisse bringen alle Schüler(innen) auf gleichen Stand
 - ggf. Schülerlösungen durch qualitätsgesicherte Musterlösung ergänzen



Herausforderung: Projektcharakter bewahren.

Beispiel „Classbook“



Classbook

For Good Friends Only!

- **Projektidee:**
Implementierung eines sozialen Netzwerkes mit PHP/SQL
- **DB-Design:** Erarbeitung eines Daten(bank)modells im Plenum
- **Verteilte Entwicklung:** Programmierung einzelner Tools durch (Datenerfassung, Datenauswertung, Suche)



Schauer, Hanno: Das Wissen von Facebook – Implementierung und Auswertung eines sozialen Netzwerkes in der Sekundarstufe II. LOG IN, Bd. 30, Nr. 166/67, 2010, S. 56-62.

Ergänzungsaufgaben

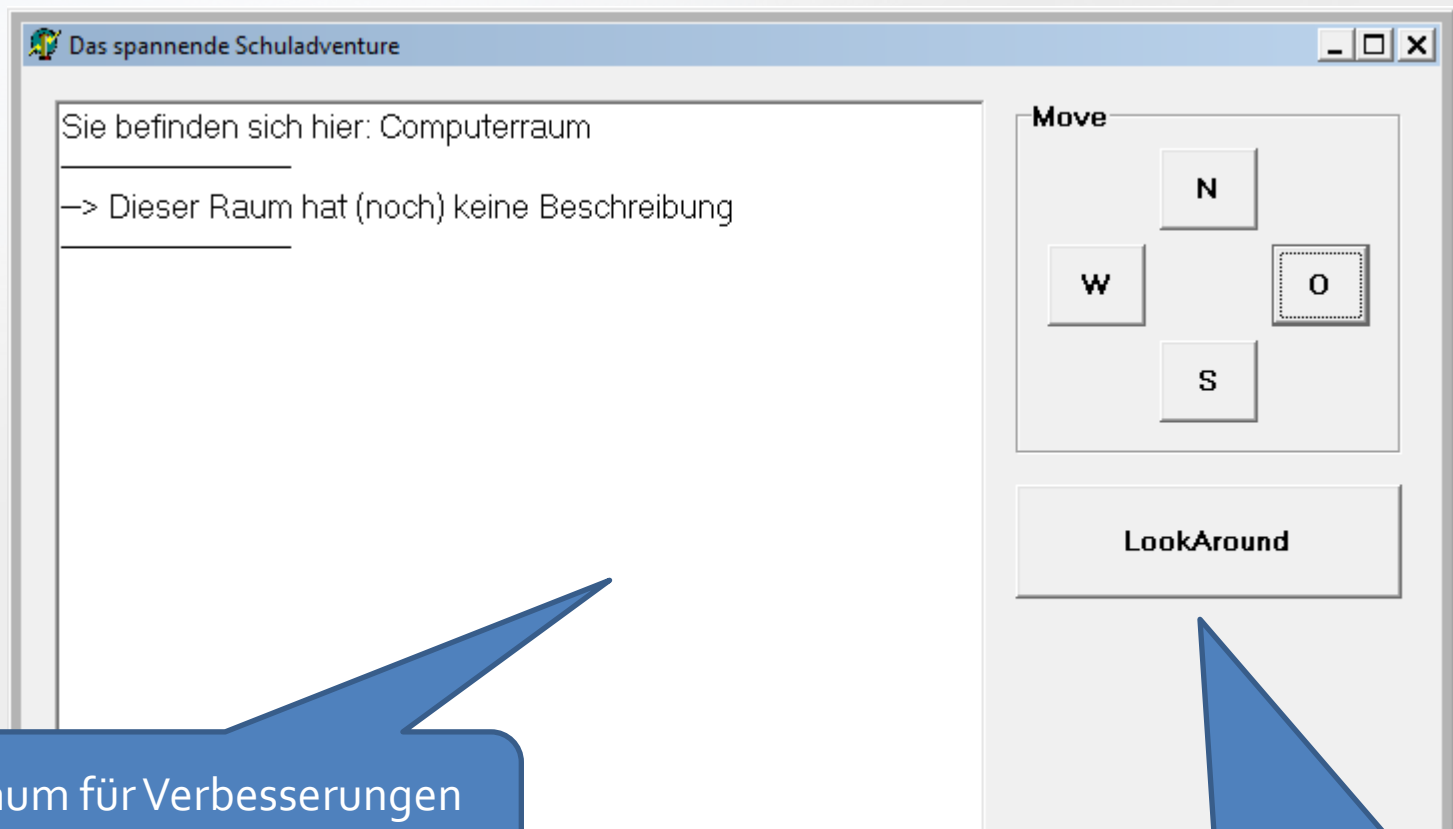
- Es hat sich bewährt, Schüler(inne)n ein Grundgerüst für die Implementierung geben – z. B.
 - Gemeinsame Datenbank (inkl. Konnektoren)
 - Funktionsfähige grafische Oberfläche
 - Bereits implementierte „Service“-Funktionen
- Ergänzungsaufgaben:
 - Grundgerüst als Starthilfe insb. für schwache Schüler
 - Erweiterungen auf unterschiedlichen Anforderungsniveaus



Beispiel „School Adventure“¹

¹In Anlehnung an eine Idee von Martin Jakobs, siehe <http://www.martinjakobs.de/pages/objektorientierte-programmierung.php>

Beispiel „School Adventure“



Viel Raum für Verbesserungen
und eigene Ideen!

Funktionsfähige grafische
Oberfläche erleichtert Einstieg

Projektergebnisse nutzen

- Nutzen Sie die Begeisterung der Schüler!
- Wenden Sie Projektergebnisse an ...
 - gelungene Spiele im Unterricht spielen lassen
 - Mit spannenden Chat-Bots unterhalten
 - Suchmaschin(en) in einer Fallstudie anwenden
- ... bzw. nutzen Sie diese im weiteren Verlauf einer Reihe (Beispiel: Classbook-Erweiterung)

Projekte von Anfang an

Projekte sollten mit den Fähigkeiten der Schüler wachsen.

Beispiel:

1. reine Programmierung (z.B. ELIZA)
2. Softwaremodelle implementieren (z.B. SchoolAdventure)
3. Software auf Basis eines eigenen Designs implementieren (z.B. Classbook)



Faustregel: Eine Langzeithausaufgabe pro Halbjahr



**Vielen
Dank!**