

## Erlernen von Programmieren an Oberstufen der Gymnasien Österreichs durch Computergrafik-unterstützte Ausgabe

Mag. Christian Johannes Tomaschitz, BSc  
Software Engineering and Internet Computing

TU Wien Informatics  
Visual Computing and Human-Centered Technology (E193)  
Computer Graphics (E193-02)

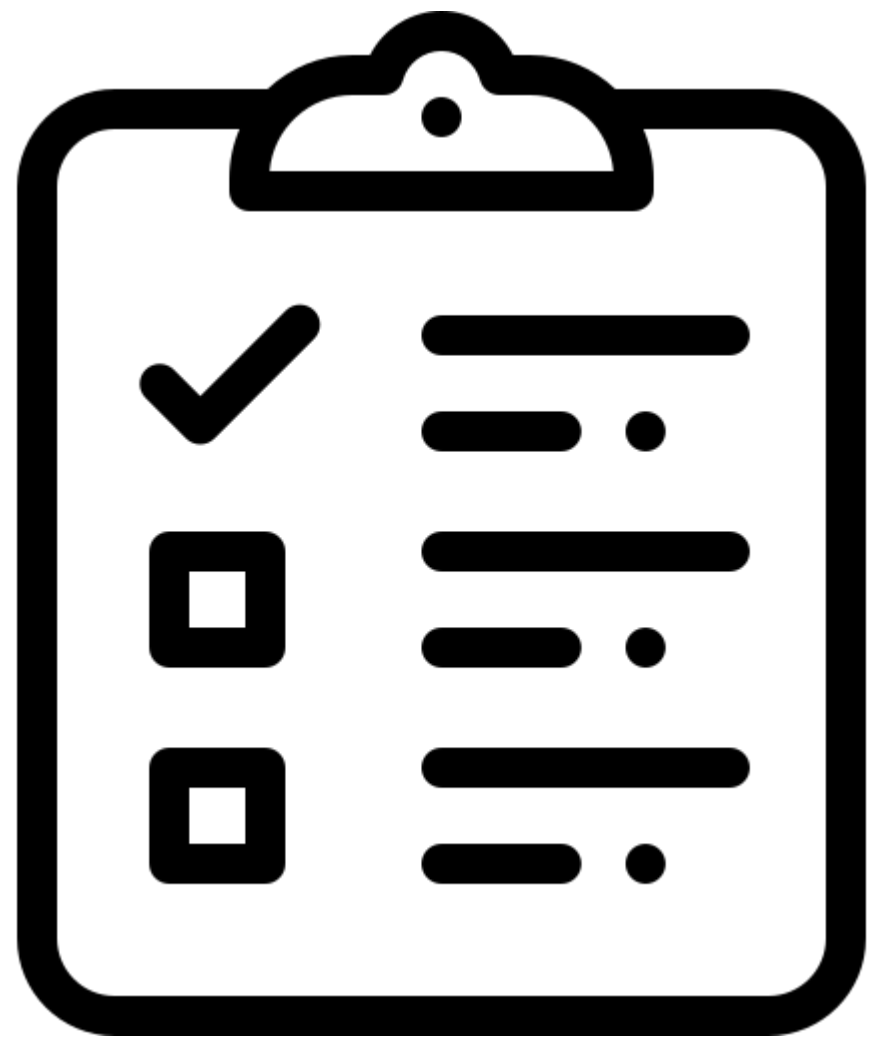
Supervisor: Associate Prof. Dipl.-Ing. Dipl.-Ing. Dr.techn. Wimmer Michael  
Contact: michael.e193.wimmer@tuwien.ac.at



In Oberstufen an Gymnasien an Schulen Österreichs ist die Zeit, welche pro Jahr für Programmieren eingesetzt werden kann, aufgrund des umfangreichen Lehrplans auf nur acht Wochen pro Jahr beschränkt. Aufgrund der mangelnden Zeit bekommen die SchülerInnen meist einen Theorieüberfluss zu hören und müssen im Selbststudium mit Übungen Programmieren lernen. In der Praxis geschieht dies meist so, dass ein oder zwei engagierte SchülerInnen die Übungen vorprogrammieren und der Rest der Klasse die Vorlage abtippt.

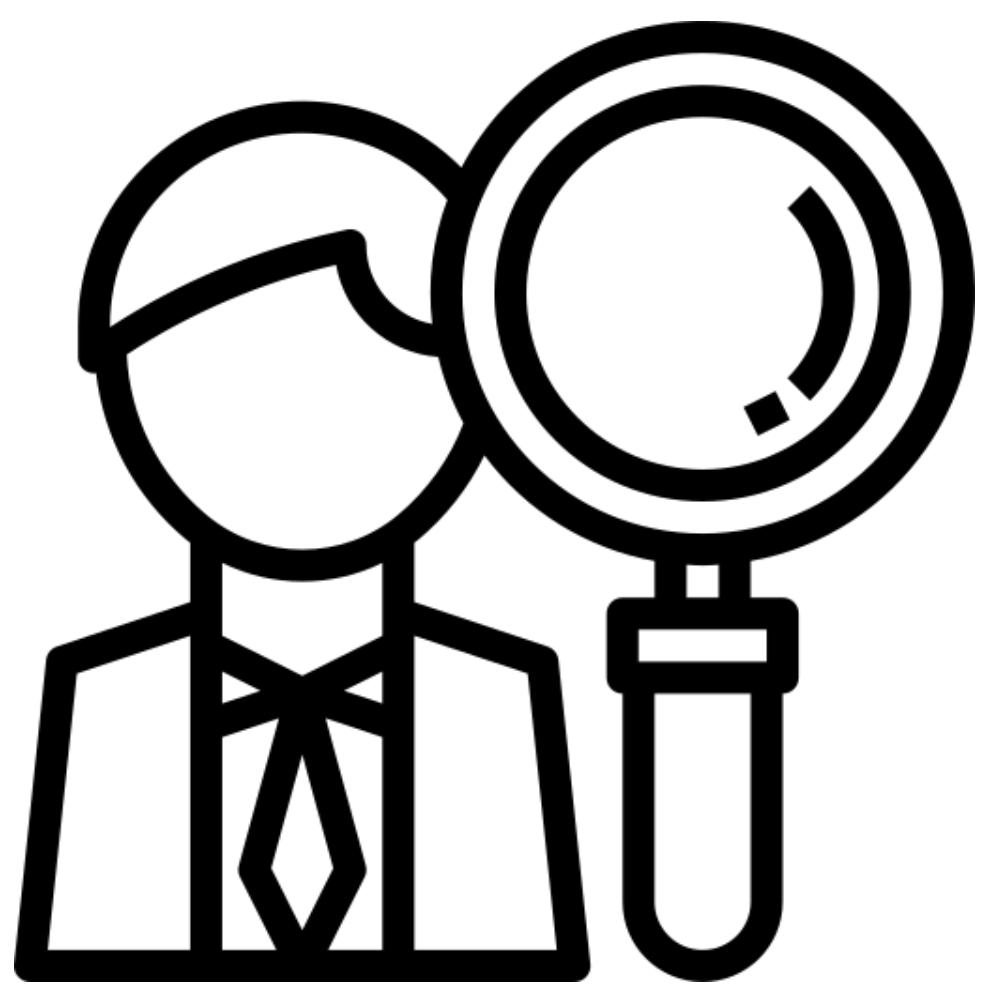
Es gilt also, den SchülerInnen im Unterricht schnell, effektiv und nachhaltig Wissen zu vermitteln und Kompetenz aufzubauen. Da aktives Lernen im Unterricht beim Programmieren mehr oder weniger durch Theorie, spielerisches Simulieren von Programmabläufen, teilweise kooperatives Arbeiten und manuelles Tippen von Code bereits vorkommt, wird hier der Fokus auf die meist vernachlässigte, visuelle Repräsentation von Programmabläufen und Code zur Steigerung der Effektivität gelegt. Es wird durch gezielte Verwendung von Computergrafik eine Methode entwickelt, welche das Verstehen, Reproduzieren und Anwenden beim Schüler, bei der Schülerin durch visuelle Darstellung beschleunigt.

### Ziel ist die Steigerung der Lerneffizienz beim Erlernen des Programmierens durch geschickte Abänderung bei der Ausgabe von ausgeführtem Code und grafischer Repräsentation des Ergebnisses



- 1 Die SchülerInnen steigen mit Grundkenntnissen im Umgang mit dem Computer in das Programmieren ein, und am Ende aller Einheiten sind die Anforderungen des Lehrplans zum Programmieren erfüllt.
- 2 Mit einer wöchentlichen Einheit von zweimal 50 Minuten auf acht Wochen verteilt wurde die Unterrichtssequenz angepasst auf den Lehrplan ausgelegt.
- 3 Dies wird durch zwei Testgruppen geprüft, welche zum einen textbasiertes Programmieren mit textueller Ausgabe durchführen und zum anderen visuell basiertes Programmieren mit grafischer Ausgabe erlernen.
- 4 Eine Überprüfung der gewonnenen theoretischen und praktischen Fertigkeiten am Ende der Unterrichtssequenz zeigt das Ergebnis.
- 5 Die Interpretation der Ergebnisse soll Erkenntnis über die Unterrichtsmethode mit grafisch basierter Ausgabe bringen.

### Eingesetzte wissenschaftliche Verfahren sind Literaturrecherche, Umfrage und Experiment



Das Erlernen des Programmierens mit visueller Unterstützung ist ein interessantes Gebiet, in dem Potential für nachhaltiges, schnelles Lernen steckt. In der Informatik gab es an Universitäten bereits Versuche, StudentInnen beim Einstieg ins Programmieren mit grafischen Hilfsmitteln zu unterstützen.

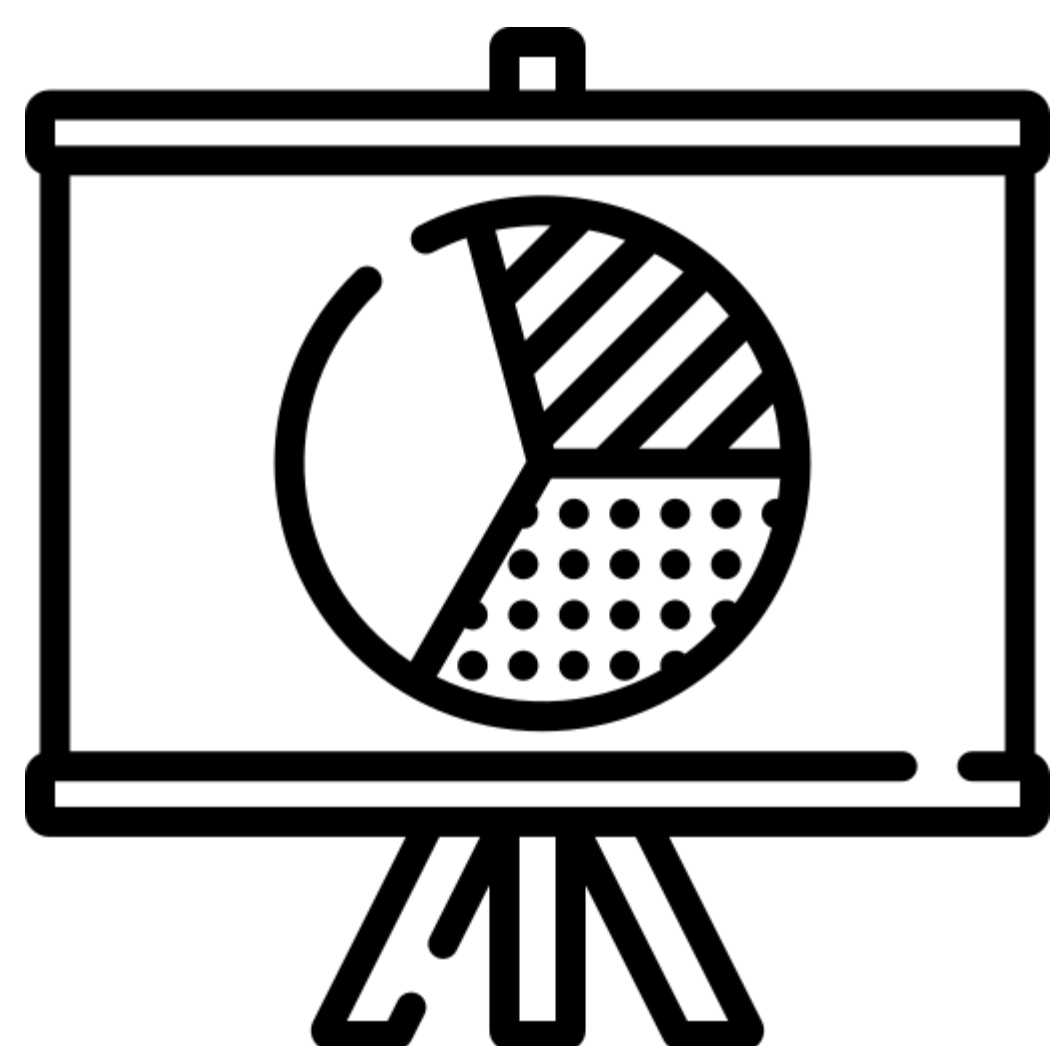
Auch an Unterstufen für SchülerInnen im Alter zwischen 10 und 14 Jahren wurde mit Scratch gearbeitet, um ihnen die Arbeitsweise des Programmierens nahezubringen. Sowohl die Experimente an den Universtäten als auch die Versuche in der Unterstufe der Schule haben deutlich gezeigt, dass Programmieren mit visueller Unterstützung effektiver ist als das Programmieren ohne visuelle Unterstützung.

Die Lerngeschwindigkeit beim Lernenden wird deutlich erhöht und das Verständnis wird besser.

Klassisch	Grafisch
<pre>var x = 100; if (x &gt; 100) {   console.log("BIG"); } if (x &lt; 100) {   console.log("SMALL"); } if (x == 100) {   console.log("EXACT"); } x = x*2; if (x &gt; 100) {   console.log("BIG"); } else {   console.log("SMALL OR EXACT"); }</pre>	<pre>var posX = 50; var posY = 50; var dirX = 1; var wWidth = 600; var wHeight = 600;  function setup() {   createCanvas(wWidth, wHeight); }  function draw() {   background(220);   circle(posX, posY, 90);   if (posX &gt; wWidth) {     dirX = -1;   }   if (posX &lt; 0) {     dirX = 1;   }   posX = posX + 1 * dirX; }</pre>

Einheit 2 von 5: Beispiele zu if-Anweisungen (Beispiel 1 von 3)

### Die erhofften Ergebnisse: die SchülerInnen mit visueller Ausgabe konnten im Gegensatz zu den SchülerInnen mit textueller Ausgabe eine deutliche Steigerung der Ergebnisse erzielen



Die Methode mit der visuellen Ausgabe ist im Unterricht effizient, da die SchülerInnen sich im Gegensatz zur textuellen Ausgabe an vieles erinnern und viel wiedergeben können. Es ist ein deutlicher Unterschied der Programmierkenntnisse in der „grafischen“ Gruppe gegenüber der „klassischen“ Gruppe zu erkennen.

Zu Beginn, bei der Vortestung, ist der Unterschied zwischen den beiden Testgruppen gering, 0,06 zwischen den beiden Mittelwerten. Bei der abschließenden Testung ist der Unterschied zwischen den beiden Testgruppen mit 0,78 zwischen den beiden Mittelwerten zu Gunsten der „grafischen“ Gruppe deutlich größer.

Die SchülerInnen wirken durch die grafischen Inhalte interessierter und motivierter und entwickeln ein besseres Vorstellungsvermögen beim Programmieren.

