

Bachelorarbeit

**Erweiterung des CT Game Studios um einen „Open Stage“
Modus**

Daniel Rose
Matrikelnummer: 2270435

**UNIVERSITÄT
DUISBURG
ESSEN**

Abteilung Informatik und angewandte Kognitionswissenschaft
Fakultät für Ingenieurwissenschaften
Universität Duisburg-Essen

18. Mai 2018

Betreuer:
Prof. Dr. H. U. Hoppe
Sven Manske
Sören Werneburg

Inhaltsverzeichnis

1	Einleitung	1
1.1	Motivation	1
1.2	Aufgabenstellung	1
1.3	Aufbau der Arbeit	1
2	Grundlagen	3
2.1	Computational Thinking	3
2.2	CT Game Studio	3
2.3	Effektives Game-based-learning	3
2.3.1	Motivation und Engagement	3
2.3.2	Guided vs Discovery Learning	4
2.3.3	Social Environments und Gruppenstruktur	4
2.3.4	Scaffolding/Integration in Klassenraum	4
2.4	Verwandte Projekte	4
2.4.1	Blockly Games	4
2.4.2	RoboCode und RoboBuilder	5
2.4.3	Dragon Architect	5
3	Ansatz	7
3.1	Ein Lernszenario	7
3.2	Aufgabenanalyse	7
3.3	Vergleich zu verwandten Spielen	7
4	Implementierung	9
5	Evaluation	11
6	Zusammenfassung und Ausblick	13
	Literaturverzeichnis	15

Abbildungsverzeichnis

1 Einleitung

”Wie Facebook 4 Mio. Datensätze verloren hat” (Fakenews 2017). ”ÄI wird uns zerstören” (Fakemag, 2015). ”Big Technology invests 500 M to bring CS into schools” (Mustermag, 2017). Die Informatik umgibt nahezu alle Menschen, sie transformiert unseren Alltag und unseren Arbeitsplatz. Dies erfordert eine Gesellschaft, die in Kontrolle der Technologie ist, und weiß, wie damit umzugehen ist. Dabei fangen wir gerade erst an, die Informatik auch in dieser Reichweite und angemessenen Umfang in die Schulen zu bringen.

Mit ”Computational Thinking” hat Jeanette Wing einen einflussreichen Forschungsansatz gestellt zu der Frage ”Welche Fähigkeiten werden in der Informatik gebraucht?”. Unter diesem Begriff haben sich eine Reihe von Curricula und Werkzeugen gesammelt, die versuchen, diese Fähigkeiten Schülern und anderen Interessierten näher zu bringen.

Mit RoboPlanet wurde ein Spiel entwickelt, bei dem der Lernende mittels einem zugänglichen visuellen Programmiersprache einen Roboter programmiert, um verschiedene Spielziele zu erreichen. Das Spiel besitzt einen Storymodus, bei dem dem Spieler schrittweise neue Programmierkonzepte beigebracht werden. Zu Ende der Story kann der Roboter komplexe Probleme lösen.

Im Rahmen dieser Arbeit wird RoboPlanet um einen offenen Spielmodus erweitert, bei dem der Spieler seinen Roboter gegen andere Roboter antreten lässt. Im Training entwickelt der Schüler Strategien, um die Gegner, die wiederum verschiedene Kampfstrategien besitzen, zu besiegen. In einem klassenübergreifenden Wettbewerb werden die entwickelten Roboter gegeneinander antreten. Durch die Erweiterung soll motiviert werden, und fortgeschrittene, nahezu unbegrenzte Programmierung gefördert werden.

Im Folgenden wird zunächst dargestellt, wie Computational Thinking durch RoboPlanet gefördert wird. Dazu wird die Theorie und verwandte Arbeiten vorgestellt. Wir untersuchen die Bestandteile von Computational Thinking, und die Mechanismen mit denen Schüler diese Bestandteile lernen, bzw. Lernen dieser Bestandteile gefördert werden. Daraufhin wird exploriert, welchen Mehrwert RoboArena bringt, welche Konzeption daraus entsteht, und wie dieses technisch umgesetzt wird.

Test [Ikeda et al., 1997]

1.1 Motivation

1.2 Aufgabenstellung

1.3 Aufbau der Arbeit

2 Grundlagen

2.1 Computational Thinking

Unter Computational Thinking wird Kognitionsprozess oder Gedankenprozess verstanden, der durch die Fähigkeit, in Form von Dekomposition, abstrahierend, evaluierend, algorithmisch und generalisierend zu denken, reflektiert wird. Ziel des Prozesses ist es, ein Problem so dar zu stellen, dass es von einem Computer gelöst werden kann. Im Folgenden sollen diese Denkweisen im Einzelnen vorgestellt werden.

CT-Aktivitäten (z.B. Debugging) ...

2.2 CT Game Studio

CT Game Studio ist eine Lernumgebung zur Vermittlung von Computational Thinking Kompetenzen.

Game-based-learning

Blockbasierte Programmierung

...

2.3 Effektives Game-based-learning

Welche Eigenschaften weisen Spiele auf, die sich positiv auf Lernerfolg auswirken?

2.3.1 Motivation und Engagement

Games engage players on three main fronts: 1) The structure of the game provides motivation and the urge to solve problems for the problem's sake alone. 2) The backstory or narrative provides the believability or authenticity of engagement. 3) Characterization makes the player's role in the narrative believable so that the player can engage fully in the game

Flow besteht aus Herausforderung und Fähigkeiten und wirkt sich positiv auf das Engagement aus

Engagement wirkt sich positiv auf Lernerfolg aus

Wiederholtes Spielen wirkt sich positiv auf Lernerfolg aus -> Wiederspielbarkeit ist wichtig

2.3.2 Guided vs Discovery Learning

...

2.3.3 Social Environments und Gruppenstruktur

- Sharing von Ergebnissen - Kollaboration/Kompetitiv/vs single player (Goal structures?)

2.3.4 Scaffolding/Integration in Klassenraum

Use-create-modify

Debriefing?

2.4 Verwandte Projekte

Das CT Game Studio ist einer von vielen Ansätzen, Computational Thinking in Spielumgebungen zu lehren. Im Folgenden soll eine Auswahl von solchen Spielen vorgestellt werden.

2.4.1 Blockly Games

Blockly Games ist eine webbasierte Kollektion von sechs Minispielen, mit unterschiedlichen Mechaniken. Die Programmierung ist blockbasiert, wobei die verfügbaren Blöcke an die Mechanik des jeweiligen Spiels angepasst sind.

In *Puzzle* wird durch ein einfaches Assoziationspiel die Grundlagen der Blockbasierten Programmierung näher gebracht.

In den weiteren Spielen wird eine Spielfigur gesteuert. (Erklärung der UI/Ausführen-Reset-Zyklus/Instruktion). Dabei bestehen die folgenden vier Spiele aus jeweils zehn Levels, die den Spieler schrittweise an neue Konzepte heran führen und größere, komplexere Herausforderungen stellt.

In *Maze*...

- Maze hat Limit an Blocks um Abstraktion zu fordern, und ein Auswahlmenü zur Auswahl eines Avatars (Personalisierung)
- Levels können frei ausgewählt werden, das Spiel zeigt an welche Levels abgeschlossen wurden
- Der resultierende Javascript Code wird nach erfolgreichem Abschluss des Levels angezeigt
- Start/Reset

- Neue Konzepte und Interaktionsmöglichkeiten werden durch Hinweise eingeführt und erklärt. Teilweise tauchen auch Hilfestellungen auf, wenn das angenommen wird, dass der Spieler nicht weiter kommt. Bird, Turtle, Movie und Pond Tutor starten jedes Level mit einer Instruktion zur Lösung des Levels. Sie kann über einen dedizierten Hilfe-button erreicht werden.

2.4.2 RoboCode und RoboBuilder

...

2.4.3 Dragon Architect

...

Der Vollständigkeit ein Spiel vorstellen, dass nicht Blockbasiert ist? Z.B. Lightbot oder Program Your Robot

3 Ansatz

3.1 Ein Lernszenario

Informatikunterricht, Lehrer stellt Spiel vor, Schüler spielen Storymodus, dann Strategiemodus (Gameplay im Detail?), dann veranstaltet Lehrer Turnier, Spiele werden analysiert, Schüler verbessern ihre Strategien (Gewinnstrategie im Training verfügbar?), erneutes Turnier

3.2 Aufgabenanalyse

Folgende Fragen beantworten

- Wie sollten Turniere für den größten Lernerfolg aufgebaut werden?
- Wie sollten Kämpfe innerhalb des Turniers aufgebaut werden?
- Wie können Strategien entwickelt werden?
- Wie können Strategien für ein Turnier bereitgestellt werden?
- Wie wird das Verhältnis zwischen Schüler und Klasse hergestellt?
- Wie werden Turniere erstellt und daran teilgenommen?
- Welche Strategien sind im Trainingsmodus verfügbar?
- Wie können bestimmte CT-Aktivitäten unterstützt werden (z.B. Debugging)?

3.3 Vergleich zu verwandten Spielen

<https://docs.google.com/spreadsheets/d/1G4Rfo7FaiMg8xyQcF4hESy2Pk8S5qGSmIHvEniQA-fc>

4 Implementierung

5 Evaluation

6 Zusammenfassung und Ausblick

Literaturverzeichnis

- [Ikeda et al., 1997] Ikeda, M., Go, S., and Mizoguchi, R. (1997). Opportunistic group formation. In *Proceedings of the Conference on Artificial Intelligence in Education (AI-ED)*, pages 167–174, Amsterdam. IOS Press.