

Spiel: LWB-Adventure

Zielbestimmung

Produktbeschreibung	<p>Es wird ein Spiel entwickelt, das an die Lehrerweiterbildung Informatik Berlin angelehnt ist. Der Spieler absolviert die vier Semester des Studiums, indem er kleine Mini-Games spielt, welche die Lehrinhalte der jeweiligen Semester thematisieren.</p> <p><u>MINI-SPIELE:</u></p> <ul style="list-style-type: none"> • <i>Muster-Spiel</i> (Funktionale Programmierung, 1. Sem.): Der Spieler muss bewerten, ob gegebene Muster und Argumente passen. • <i>Bauelemente-Spiel</i> (Technische Informatik, 1. Sem): Der Spieler muss durch Anklicken von Schaltern im Blockschaltbild mit NOT, AND, OR and XOR-Gattern alle Lampen anschalten. • <i>Super-ALP2-Escape</i> (Imperative und objektorientierte Programmierung, 2. Sem.): Die Spielfigur Vaderobi muss, mithilfe über die Tastatur in einen Texteditor eingegebene Befehle (<i>Laufen, (Ent-)Lasern, (Ent-)Mauern, Markieren, Links-/RechtsDrehen</i>), durch seine Welt bewegt werden. Ziel ist es, die linke obere Ecke des Spielfelds zu erreichen. Für die maximale Punktzahl müssen dabei bestimmte Bedingungen erfüllt werden (<i>z.B. möglichst wenige Befehle</i>). Es gibt verschiedene Level (Welten) und für jedes Level am Ende eine Punktzahl und eine Note. Am Ende des Mini-Games gibt es einen End-Bildschirm mit Punkte-Übersicht und Gesamtnote. • <i>Getränkeautomaten-Spiel</i> (Rechnernetze, 2. Sem.): Der Spieler muss mithilfe der Maus Münzen in einen Automaten einwerfen und dabei die richtige Anzahl und Reihenfolge der Münzen beachten, um das gewünschte Getränk für den Dozenten zu kaufen. Dafür muss er einen DEA oder NEA-Graphen lesen. • <i>SQL-Quest</i> (Datenbanksysteme, 3. Sem.): Ziel des Mini-Games ist es, der LWB-Adventure-Datenbank von Herk mithilfe von SQL-Anfragen einige Geheimnisse zu entlocken. Herk selbst ist Spielleiter und stellt verschiedene Aufgaben, die von Level zu Level schwieriger und komplexer werden. Die Anfragen werden über die Tastatur in einen Texteditor eingegeben (<i>richtige Syntax ist zu beachten, Leerzeichen spielen keine Rolle</i>). Bei richtiger Eingabe wird die entsprechende Ausgabe angezeigt, bei falscher Eingabe gibt es gestaffelte Hilfen und Punktabzug. Für jede Aufgabe gibt es vier Versuche, danach wird die richtige Lösung angezeigt und man kommt ohne Level-Punkte ins nächste Level. Für jedes Level gibt es eine Punktzahl und eine Note. Am Ende des Mini-Games
----------------------------	---

	<p>gibt es einen End-Bildschirm mit Punkte-Übersicht und Gesamtnote.</p> <ul style="list-style-type: none"> • <i>FachJargon</i> (Didaktik 3. Sem.): Der Spieler muss in zwei Leveln, unter Zeitdruck, zufällig erscheinende Bilder den entsprechenden Themen oder Stichworten zuordnen. • <i>theNETgame</i> (Rechnernetze, 4. Sem.): Der Spieler muss mithilfe der Maus ein Datenpaket durch ein Routernetzwerk navigieren und dabei Darth Schmidter und Routerdefekten ausweichen. • <i>BugAttack</i> (SWP 4. Sem.): Der Spieler muss mithilfe der Tastatur sich zufällig bewegende Bugs reparieren/abschießen.
Musskriterien	<ul style="list-style-type: none"> • Es gibt einen Startbildschirm mit einer Einführung. • Es gibt einen Hauptflur, von dem aus die Semester-Räume durch Klicken erreicht werden können. • Jedes Spiel startet und beendet innerhalb desselben, erstmals geöffneten Fensters (1200x700 Pixel) und hat eine grafische Ausgabe. • Alle Spiele resultieren in einer übergebenen Gesamtpunktzahl (uint32) und einer Spiel-Note (float32). • Zum Abschluss werden alle Resultate im Endbildschirm auf einem "Zeugnis" aufgelistet. • Es gibt einen Cheat (Supermensch als Nutzernamen eingeben) mit dem man automatisch die Note 4 für alle Spiele erhält und damit alle Räume betreten kann. • Es gibt Unterordner im Spiel für Bilder und Sounds. • Für jedes Minigame gibt es eine Testversion, um die Funktionalität des Spiels und der Klassen zu gewährleisten. • Mindestens ein Spiel pro Semester implementiert. <p><u>MINI-SPIELE:</u></p> <ul style="list-style-type: none"> • <i>Muster-Spiel</i>: mindestens 8 Muster-Abfragen, die über die Maus gesteuert sind -> als Multiple-Choice • <i>Bauelemente-Spiel</i>: mindestens 3 Level, mindesten 3 Gatter, Maussteuerung; • <i>Super-ALP2-Escape</i>: mindestens 3 Level, Anwendung der Grundkenntnisse der Befehle zur robi-Steuerung, Spiel-Steuerung durch Texteingabe; • <i>Getränkeautomaten-Spiel</i>: jeweils 1 Level pro Dozent, mindestens 2 Automatengraphen, Maussteuerung; • <i>SQL-Quest</i>: mindestens 5 Level mit ansteigendem Schwierigkeitsgrad, Anwendung der SQL-Kenntnisse, Spiel-Steuerung durch Texteingabe, Ausgaben-Anzeige für richtige Lösung, Punktabzug und Hilfe(n) bei Falscheingaben; • <i>theNETgame</i>: mindestens 3 Level mit unterschiedlichen Graphen, Maussteuerung; • <i>FachJargon</i>: mindestens 2 Level mit Zeitbegrenzung. Reaktion von WtheK auf richtige/falsche Antworten • <i>BugAttack</i>: mindestens 3 Level, Bugs bewegen sich zufällig, Punktabzug gibt es durch fortschreitende Zeit und wenn sich Bugs bewegen und dabei den Code essen. Bugs haben unterschiedliche Geschwindigkeit.

Wunschkriterien	<p><u>Hauptspiel:</u></p> <ul style="list-style-type: none">• ergänzt werden könnte noch die Implementierung einer Besten-Liste (nach Punkten) für das gesamte LWB-Adventure und ggf. auch für die einzelnen Mini-Games• Die Mini-Spiele werden grafisch ansprechend gestaltet.• Es gibt einen fünften "Raum" für die Zeugnisübergabe.• Es können in den Räumen verschiedene Gegenstände eingesammelt werden → mit diesen werden Features in den Minigames freigeschaltet oder neue Dialoge sichtbar.• Anpassung der Fenster-Größe an verschiedene Auflösungen.• Die Einbindung einer SQL-Datenbank für Abfragen zu einem der Spiele (SQL-Quest) wäre denkbar.• Schlüssel-Modus, sodass die Räume nur freigeschaltet werden, wenn Noten aus vorherigen Semestern ≤ 4.0.• Es soll die Möglichkeit geben, den Spieler-Namen und das Spieler-Bild auszuwählen.• Spielstände sollen gespeichert werden können.• Die verwendeten (individuellen, zum Teil gedoppelten) Klassen der Mini-Spiele können vereinheitlicht und angepasst werden.• Hintergrundmusik, die sich beim Verlassen der Spiele wieder ausschaltet• Vor dem Betreten des 4. Raumes muss die Hausordnung unterschrieben werden.• Weitere Minispiele für die anderen Vorlesungen (wurde in großem Umfang erfüllt), weitere Ideen: BSW-Abspeichergame; FP: Baumgame; RO - von-Neumann-Game; Alp2 - Sortiergame, SpaceInvader; Alp3 - 6-Sturmtruppler-/ Jedi-Game; AFL - Methoden-Game; NET - Multiplayer-Netzwerk-Game <p><u>MINI-SPIELE:</u></p> <ul style="list-style-type: none">• <i>Muster-Spiel</i> (FP): Muster und Argumente werden randomisiert angezeigt; bei Passung sollen Bindungen per Tastatur eingegeben werden können; als zweiter Abschnitt wird ein Muster-Memory (zusammengehörige Paare von Muster und Argument) implementiert in mehrstufigen Levels;• <i>Super-ALP2-Escape</i>: die Aufgabenstellung der verschiedenen Level noch mehr und anders variieren (mehr Progression, andere/mehr Befehle, ggf. auch kombiniert);• <i>SQL-Quest</i>: mehr Level mit noch mehr und anderen Abfragen (verschiedene Joins/...);• <i>BugAttack</i>: mehr Level; es werden neue Bugs generiert, wenn man nicht ganz in der Mitte trifft; es gibt Zusatzfähigkeiten wie Zielautomatik und mehrere Bugs auf einmal töten; es gibt nur eine begrenzte Zeit für jedes Level; <p>noch nicht implementiert: Verschiedene Arten von Bugs; Verbesserung der Fähigkeiten durch den Spieler; es gibt Upgrades die eingesammelt werden können (Zeitbonus, etc.);</p>
------------------------	---

	<p>der Code kann repariert werden; mehr Level; andere Musik pro Level</p> <ul style="list-style-type: none"> • <i>Getränkeautomaten-Spiel</i>: Es werden noch weitere und komplexere Graphen implementiert.
Abgrenzungskriterien	<p><u>Hauptspiel:</u></p> <ul style="list-style-type: none"> • Der Spieler selbst kann sich nicht innerhalb der Welt sichtbar bewegen (Spielfigur nicht dargestellt). • Die Spielfeldgröße ist 1200x700 Pixel festgelegt. • Ein Mehrspielermodus ist nicht vorgegeben. <p><u>MINI-SPIELE:</u></p> <ul style="list-style-type: none"> • <i>Muster-Spiel</i> (FP): • <i>Super-ALP2-Escape</i>: keine Steuerung mit Pfeilen/Shortcuts/Joystick o.a.; • <i>SQL-Quest</i>: kein Copy/Paste, keine Autovervollständigung bei der Texteingabe, keine alternative Steuerung außer der Texteingabe, keine Veränderung der Darstellung (des Spiel-Fensters); • <i>BugAttack</i>: keine Maussteuerung;

Produkteinsatz

Anwendungsbereiche	Das Spiel dient der praktischen Umsetzung aller gelernten Inhalte aus dem Weiterbildungsstudium Informatik. Es ist somit rein für den privaten Gebrauch bzw. die Verteilung unter Kommiliton:innen gedacht. Das ist vor allem vor dem Hintergrund des angefragten Wissens notwendig und sinnvoll.
Zielgruppen	Alle Teilnehmenden, Absolvent*innen und Dozent*innen der LWB Informatik und ggf. auch andere Informatik-Studierende/-Studierte oder Menschen mit informatischem und pädagogischem Interesse und den entsprechenden Fachkenntnissen. (Der Schwierigkeitsgrad der jeweiligen Spiele und Levels setzt natürliche Grenzen.)
Benutzeroberfläche	LWB-Adventure nutzt das grafische gfx-Paket (von Stefan Schmidt) und wird im dadurch erzeugten Fenster gespielt. Tastatur und Maus werden für die Eingaben genutzt.
Gesetze, Normen, Sicherheitsanforderungen, Plattformabhängigkeiten, ...	<p>Die Ausführung des Spiels erfolgt plattformunabhängig, sofern das gfx-Paket (nicht "gfxw") in einer Go-Entwicklungsumgebung im Source-Ordner liegt bzw. installiert ist. Hier wurde ein von Stefan Schmidt modifiziertes gfx-Paket verwendet, dieses kann Sounds stoppen.</p> <p>Alle verwendeten Bilder sind entweder selbst designt oder lizenzfrei von pixabay oder freepik.</p>

Technische Anforderungen

NotwendigeSoftware	Linux, Golang inkl. der Go-Pakete: gfx (unveröffentlicht, StoppeAlleSounds) von St. Schmidt, dateien, editor, editor1line, von St. Schmidt felder von O. Schäfer
NotwendigeHardware	x86-PC, Bildschirm, Tastatur, Maus
Produktschnittstellen	keine