

1. Record Nr.	UNINA9910455387003321
Titolo	Cognitive poetics [[electronic resource]] : goals, gains and gaps / / edited by Geert Brone, Jeroen Vandaele
Pubbl/distr/stampa	Berlin ; ; New York, : Mouton de Gruyter, c2009
ISBN	1-282-18799-6 9786612187995 3-11-021337-0
Descrizione fisica	1 online resource (568 p.)
Collana	Applications of cognitive linguistics, , 1861-4078 ; ; 10
Classificazione	ER 955
Altri autori (Persone)	BroneGeert <1978-> VandaeleJeroen
Disciplina	415
Soggetti	Cognitive grammar Poetics Electronic books.
Lingua di pubblicazione	Inglese
Formato	Materiale a stampa
Livello bibliografico	Monografia
Note generali	Description based upon print version of record.
Nota di bibliografia	Includes bibliographical references and index.
Nota di contenuto	Frontmatter -- Table of contents -- Cognitive poetics. A critical introduction -- Part I: Story -- Text worlds -- Cognitive approaches to narrative analysis -- Reflections on a cognitive stylistic approach to characterisation -- Part II: Figure -- Minding: feeling, form, and meaning in the creation of poetic iconicity -- Metaphor and figure-ground relationship: comparisons from poetry, music, and the visual arts -- Part III: Stance -- Deconstructing verbal humour with Construction Grammar -- Judging distances: mental spaces, distance, and viewpoint in literary discourse -- Does an "ironic situation" favor an ironic interpretation? -- Part IV: Critique -- How cognitive is cognitive poetics? The interaction between symbolic and embodied cognition -- Epilogue. How (not) to advance toward the narrative mind -- Backmatter
Sommario/riassunto	For more than two decades now, cognitive science has been making overtures to literature and literary studies. Only recently, however, cognitive linguistics and poetics seem to be moving towards a more serious and reciprocal type of interdisciplinarity. In coupling cognitive linguistics and poetics, cognitive poetics aim to offer cognitive

readings of literary texts and formulate specific hypotheses concerning the relationship between aesthetic meaning effects and patterns in the cognitive construal and processing of literary texts. One of the basic assumptions of the endeavour is that some of the key topics in poetics (such as the construction of text worlds, characterization, narrative perspective, distancing discourse, etc.) may be fruitfully approached by applying cognitive linguistic concepts and insights (such as embodied cognition, metaphor, mental spaces, iconicity, construction grammar, figure/ground alignment, etc.), in an attempt to support, enrich or adjust 'traditional' poetic analysis. Conversely, the tradition of poetics may support, frame or call into question insights from cognitive linguistics. In order to capture the goals, gains and gaps of this rapidly growing interdisciplinary field of research, this volume brings together some of the key players and critics of cognitive poetics. The eleven chapters are grouped into four major sections, each dealing with central concerns of the field: (i) the cognitive mechanisms, discursive means and mental products related to narrativity (Semino, Herman, Culpeper); (ii) the different incarnations of the concept of figure in cognitive poetics (Freeman, Steen, Tsur); (iii) the procedures that are meant to express or create discursive attitudes, like humour, irony or distance in general (Antonopoulou and Nikiforidou, Dancygier and Vandelanotte, Giora et al.); and (iv) a critical assessment of the current state of affairs in cognitive poetics, and more specifically the incorporation of insights from cognitive linguistics as only one of the contributing fields in the interdisciplinary conglomerate of cognitive science (Louwerse and Van Peer, Sternberg). The ensuing dialogue between cognitive and literary partners, as well as between advocates and opponents, is promoted through the use of short response articles included after ten chapters of the volume. Geert Brône, Katholieke Universiteit Leuven, Belgium; Jeroen Vandaele, University of Oslo, Norway.

2. Record Nr.	UNINA9910155379003321
Autore	Braun Daniel
Titolo	Lets play : minecraft : plugins entwickeln mit Python : programmieren lernen ohne vorkenntnisse // Daniel Braun
Pubbl/distr/stampa	Frechen, [Germany] : , : mitp, , 2017 ©2017
ISBN	3-95845-382-1 3-95845-381-3
Edizione	[1. Auflage.]
Descrizione fisica	1 online resource (296 pages) : illustrations
Disciplina	794.8
Soggetti	Minecraft (Game) Video games Adventure video games
Lingua di pubblicazione	Tedesco
Formato	Materiale a stampa
Livello bibliografico	Monografia
Note generali	Includes index.
Nota di contenuto	Cover -- Titel -- Impressum -- Inhalt -- Einleitung -- Kapitel 1: Minecraft-Server -- 1.1 Java installieren -- 1.2 Installation -- 1.2.1 CraftBukkit -- 1.2.2 Spigot -- 1.3 Konfiguration -- 1.4 Befehle -- 1.5 Verbinden -- 1.6 Updates -- Kapitel 2: Python -- 2.1 Programmiersprachen -- 2.2 Besonderheiten von Python -- 2.3 Einrichtung -- 2.3.1 Jython -- 2.3.2 PPLoader -- 2.4 Editor -- Kapitel 3: Das erste Plugin -- 3.1 Ordner anlegen -- 3.2 plugin.py -- 3.3 plugin.yml -- 3.4 Testen -- 3.5 Fehler finden -- 3.6 Entdecken -- Kapitel 4: Chat-Kommandos -- 4.1 Eigene Befehle definieren -- 4.2 Chat-Nachrichten versenden -- Kapitel 5: Variablen -- 5.1 Namen -- 5.2 Werte -- 5.2.1 Operatoren -- 5.2.2 Umwandlung -- 5.2.3 Runden -- 5.3 +1-Plugin -- 5.4 Listen und Arrays -- 5.5 Konstanten -- Kapitel 6: Schleifen -- 6.1 Kürbis-Plugin -- 6.1.1 Positionierung -- 6.1.2 Blöcke platzieren -- 6.2 Die verschiedenen Schleifen -- 6.2.1 for-Schleife -- 6.2.2 while-Schleife -- 6.2.3 Verschachtelte Schleifen -- Kapitel 7: Verzweigungen -- 7.1 if -- 7.2 else -- 7.3 elif -- Kapitel 8: Funktionen -- 8.1 Deklaration von Funktionen -- 8.2 Rückgabewerte -- 8.3 Parameter -- 8.4 Anwendungsbeispiel -- Kapitel 9: Bauen -- 9.1 Notunterkunft -- 9.1.1 Decke und Wände -- 9.1.2 Tür -- 9.1.3 Bett --

9.1.4 Fackel -- 9.2 Runde Objekte -- 9.2.1 Kreise -- 9.2.2 Kugeln --
 Kapitel 10: Schilder -- 10.1 Hängende Schilder -- 10.2 Stehende
 Schilder -- 10.3 Text festlegen -- 10.3.1 Farbe -- 10.3.2 Formatierung
 -- 10.4 Schilder-Plugin -- 10.4.1 Wiederholung: Listen -- 10.4.2 Das
 Plugin -- Kapitel 11: Listener -- 11.1 Grundgerüst -- 11.2 Spieler-
 Events -- 11.3 Kreaturen-Events -- 11.4 Block-Events -- 11.5
 Inventar-Events -- 11.6 Server-Events -- 11.7 Fahrzeug-Events --
 11.8 Wetter-Events -- 11.9 Welt-Events -- 11.10 Mehrere Listener in
 einem Plugin -- Kapitel 12: Klassen und Objekte.
 12.1 Die ganze Welt ist ein Objekt -- 12.2 Funktionen in Klassen --
 12.3 Zugriffskontrolle -- 12.4 Vererbung -- 12.5 Mehrfachvererbung
 und mehrstufige Vererbung -- 12.6 Bau-Plugin -- Kapitel 13: Crafting-
 Rezepte -- 13.1 Rezepte festlegen -- 13.2 Eigene Rezepte entwerfen
 -- 13.3 Feuerschwert -- 13.4 Enderbogen -- Kapitel 14: Informationen
 dauerhaft speichern -- 14.1 Konfigurationsdateien -- 14.1.1 Lesen --
 14.1.2 Schreiben -- 14.2 Objekte in Dateien speichern -- Kapitel 15:
 Eigene Spielmodi entwickeln -- 15.1 Schneeballschlacht -- 15.1.1
 Schneebälle verteilen -- 15.1.2 Schneebälle auffüllen -- 15.1.3 Punkte
 zählen -- 15.1.4 Punkte dauerhaft speichern -- 15.1.5 Highscore-Liste
 anzeigen -- 15.1.6 Vollständiger Quellcode -- 15.2 Sammelspiel --
 15.2.1 Aufbau des Plugins -- 15.2.2 Plugin starten -- 15.2.3 Spieler
 betritt den Server -- 15.2.4 Gegenstände zählen -- 15.2.5 Auftrag
 anzeigen -- 15.2.6 Vollständiger Quellcode -- Kapitel 16:
 Eigenständige Python-Programme -- 16.1 Python einrichten -- 16.2
 Grundgerüst -- 16.3 Ein- und Ausgabe -- 16.4 Quiz programmieren
 -- Anhang A: Befehlsreferenz -- Anhang B: Materialien -- Index.

Sommario/riassunto

Erschaffe deine eigene Welt: mächtige neue Waffen erstellen wie z.B. ein Feuerschwert, auf Knopfdruck Türme und ganze Häuser bauen, automatisch auf Veränderungen in der Welt reagieren Neue Crafting-Rezepte und eigene Spielmodi entwickeln wie z.B. eine Schneeballschlacht mit Highscore-Liste Mit umfangreicher Referenz zu allen Programmierbefehlen Du spielst schon lange Minecraft und denkst, du hast schon alles gesehen? Kennst du schon das Feuerschwert, den Enderbogen oder den Spielmodus » Schneeballschlacht«? Du willst auf Knopfdruck Türme, Mauern oder sogar ganze Häuser bauen? Vollautomatisch auf Geschehnisse in der Spielwelt reagieren? Mit eigenen Plugins kannst du all das und noch viel mehr entdecken und ganz nebenbei auch noch programmieren lernen. Python ist für Programmiereinsteiger besonders leicht zu lernen. Daniel Braun zeigt dir, wie du mit Python und Bukkit oder Spigot Erweiterungen für Minecraft programmierst, sogenannte Plugins, die du dann zusammen mit deinen Freunden auf deinem eigenen Minecraft-Server ausprobieren kannst. Dafür sind keine Vorkenntnisse erforderlich, du lernst alles von Anfang an. Nach dem Programmieren einfacher Chat-Befehle wirst du coole Plugins zum Bauen erstellen, so dass mit einem einzigen Befehl sofort z.B. ein fertiges Haus oder eine Kugel vor dir steht. Außerdem erfährst du, wie deine Plugins automatisch auf Geschehnisse in der Spielwelt reagieren können. Du kannst auch eigene Crafting-Rezepte entwerfen, um z.B. mächtige neue Waffen zu kreieren wie das Feuerschwert, das alles in Brand setzt, worauf es trifft. Am Ende lernst du sogar, wie du eigene Spielmodi entwickeln kannst, also ein Spiel im Spiel. Ob eine Schneeballschlacht mit Highscore-Liste oder ein Wettsammeln mit Belohnung für den Sieger, hier ist jede Menge Spaß garantiert. Für das alles brauchst du keine Vorkenntnisse, nur Spaß am Programmieren. Es beginnt mit ganz einfachen Beispielen, aber mit jedem Kapitel lernst du mehr Möglichkeiten kennen, um Minecraft nach deinen Wünschen anzupassen. Am Ende kannst du richtig in Python programmieren und

deiner Kreativität sind keine Grenzen mehr gesetzt, um deine eigene Minecraft-Welt zu erschaffen. Aus dem Inhalt: Bukkit und Spigot installieren und einrichten Eigene Befehle definieren und Chat-Nachrichten versenden Python-Grundlagen: Variablen, Schleifen, Verzweigungen, Funktionen, Klassen und Objekte Häuser, Kreise und Kugeln bauen Schilder errichten und beliebig beschriften Plugins programmieren, die au...
