

Zusätzliche technische Informationen zum CP 1.70

Inhaltsverzeichnis

1	Modding	2
1.1	Implementierung von .mod- und .nod-Dateien	2
1.1.1	Einleitung: Grundkenntnisse über die Data-Dateien	2
1.1.2	Erste Erweiterung: Dateiendung ".cpt"	2
1.1.3	Modding-Erweiterungen: Dateiendungen ".mod" und ".nod"	2
1.1.4	Zusätzliche Informationen: Was zu beachten ist, wenn man Dialoge modifiziert	3
1.2	Einige neue Parameter in den .info-Dateien	4
1.2.1	CondItemAmounts=0	4
1.2.2	CondHasSkill	4
1.2.3	CondPlayerKnowsNot	5
1.2.4	CondWearsItem	5
1.2.5	Erase	5
2	Neue Parameter in der ge3.ini	6
2.1	DisableDEP	6
2.2	Threads.Priority	6

1 Modding

Wir haben eine kleine Anzahl neuer Features für diejenigen von euch eingeführt, die Modifikationen für Gothic 3 erstellen möchten.

1.1 Implementierung von .mod- und .nod-Dateien

1.1.1 Einleitung: Grundkenntnisse über die Data-Dateien

Sicher kennt ihr die Datenorganisation von Gothic 3: Alle Spieldaten werden in Dateien mit der Endung **".pak"** aufbewahrt. Beispiel: "Music.pak". Wenn ihr den Inhalt einer solchen Datei ändern (editieren) möchtet, habt ihr dazu zwei Möglichkeiten:

- 1) Ersetzen der ganzen .pak-Datei (dies unpraktisch sein, wenn die Datei sehr groß ist und ihr nur wenig verändert habt), oder
- 2) Beibehalten der alten .pak-Datei und lediglich Hinzufügen einer neuen "Generation" der .pak-Datei.

Die nächste "Generation" der Datei Music.pak ist Music.p00, gefolgt von Music.p01, dann Music.p02 usw.

Ihr müsst dabei immer sicherstellen, dass es keine Lücke in der Aufeinanderfolge der Generationen gibt, weil die Engine mit dem Einlesen aufhört, wenn sie keine Datei mit der nächsthöheren Nummer finden kann.

Beispiel: Wenn es im Data-Ordner drei Musikdateien namens Music.pak, Music.p00 und Music.p02 gibt, wird das Spiel nur bis Music.p00 lesen und Music.p02 ignorieren, weil die ".p01"-Datei fehlt.

1.1.2 Erste Erweiterung: Dateiendung ".cpt"

Mit CP 1.70 haben wir das Feature "Alternatives Balancing" ("AB") eingeführt. Dafür mussten wir die geänderten Daten des AB von den originalen Daten des normalen Spieles abgrenzen. Dies wurde durch die Einführung der neuen Dateiendung **".cpt"** möglich.

Alle Dateien mit dieser Endung werden im Spiel nur dann geladen, wenn der Schalter "Alternatives Balancing" auf "An" gestellt hat. Wenn das AB deaktiviert ist, werden die .cpt-Dateien vom Spiel ignoriert.

Wird das AB angestellt, lädt die Engine zuerst die Inhalte der .pak-Dateien und deren Generationen und anschließend alle .cpt-Dateien. Die Inhalte der .cpt Dateien überschreiben also die entsprechenden Daten der .pak-Dateien.

Wenn es eines Tages weitere Generationen der .cpt-Dateien geben sollte, dann werden diese die Endungen ".c00", ".c01" etc. haben.

1.1.3 Modding-Erweiterungen: Dateiendungen ".mod" und ".nod"

Unsere Bitte an alle Gothic 3 Modder ist, dass ihr **alle .pak- und .cpt-Dateien so lasst, wie sie jetzt sind**. Bitte tauscht sie nicht aus und ergänzt keine weiteren Generationen! Um eine genaue Abgrenzung zwischen den Dateien des "Originalspiels" und den Dateien der Modifikationen zu ermöglichen, haben wir zwei andere Dateiendungen eingeführt. Diese sollen es den Spielern/Usern erleichtern, Überblick über die Inhalte ihres Data-

Ordners zu behalten und Modifikationen zu erkennen. Wir hoffen, dass dies auch für die Modder selber hilfreich ist. Wenn es in Zukunft doch noch mal neue "richtige Patches" geben sollte, dann können diese Patches die Reihe der .pak- und .cpt-Dateien fortführen, ohne das Risiko einzugehen, Dateien von Modifikationen zu löschen oder zu überschreiben.

Es gibt drei Regeln, die ihr verstehen und im Hinterkopf behalten müsst für die Entscheidung, wie ihr eure Data-Dateien benennt:

- **.mod**-Dateien führen die Reihe der .pak-Dateien fort und werden von der Engine *auf jeden Fall* geladen.
- **.nod**-Dateien sind die "Modding-Version" der .cpt-Dateien, was heißt, dass ihr Inhalt *nur bei eingeschaltetem Alternativem Balancing* vom Spiel geladen wird.
- Jede Datei mit einer "Modding"-Dateiendung wird über jede "normale" Dateiendung "gewinnen".

Hier ist eine komplette Übersicht der Ladereihenfolge von Gothic 3:

- (1) .pak-Dateien mit allen Generationen
- (2) .cpt Dateien mit allen Generationen (nur bei aktivem "Alternativen Balancing" geladen)
- (3) .mod Dateien mit allen Generationen
- (4) .nod Dateien mit allen Generationen (nur bei aktivem "Alternativen Balancing" geladen)

Natürlich tragen die Generationen der .mod-Dateien die Endungen ".m00", ".m01", etc. und die Generationen der .nod-Dateien die Endungen ".n00", ".n01" usw.

1.1.4 Zusätzliche Informationen: Was zu beachten ist, wenn man Dialoge modifiziert

Um Dialoge zu ändern oder hinzuzufügen, ist es nötig, zwei Gruppen von Dateien zu editieren/hinzuzufügen: Infos und Strings (bzw. die stringtable.ini, um genau zu sein).

In beiden Fällen werden eure modifizierten Dateien nur dann vom Spiel wahrgenommen, wenn ihr **binäre Versionen** dieser Dateien erstellt und sie auch in das Data-Verzeichnis kopiert. (Oder wenn ihr die Dateien Projects_compiled.p00 und Projects_compiled.cpt löscht oder umbenennt, aber das würde den Ladevorgang des Spieles deutlich verlangsamen!)

Und so wird's gemacht:

- 1) Erstelle einen neuen Unterordner "Infos" im "Data"-Ordner deiner Gothic-3-Installation.
- 2) Erstelle einen neuen Unterordner "Strings" im "Data"-Ordner deiner Gothic-3-Installation.
- 3) Stelle sicher, dass deine modifizierte Stringtable.ini und **alle** Info-Dateien (diejenigen, die du geändert/hinzugefügt hast und alle unveränderten Info-Dateien!) in den richtigen Ordnern unterhalb des "Data"-Ordners liegen.
- 4) Starte Gothic 3, dann starte ein neues Spiel
- 5) Beende Gothic 3
- 6) Öffne den Ordner Gothic III\Data\Infos
- 7) Dort gibt es nun die Datei "_compiledinfos_G3_World_01.bin" – benenne sie in "compiledinfos_G3_World_01.bin" um (das "_" ganz vorne löschen)
- 8) Öffne den Ordner Gothic III\Data\Strings
- 9) Dort gibt es nun die Datei "_stringtable.bin" - benenne sie in "stringtable.bin" um (das "_" ganz vorne löschen)

- 10) Erstelle einen neuen Ordner "Projects_compiled" irgendwo anders auf deiner Festplatte
- 11) Kopiere oder verschiebe die beiden .bin-Dateien (Punkte 7 und 9) in diesen neuen Ordner
- 12) Erstelle eine .pak-Datei von diesem Ordner mit dem Tool G3PakDir.exe. Das Ergebnis sollte eine Datei mit Namen " Projects_compiled.pak" sein
- 13) Benenne die Datei in "Projects_compiled.mod" oder "Projects_compiled.nod" um – abhängig von deinen Absichten mit deiner Modifikation
- 14) Kopiere diese Datei nun in deinen "Data"-Ordner und vergiss nicht, sie in deinem Modpaket zusammen mit deiner Infos.mod und Strings.mod (oder *.nod...) zu veröffentlichen
- 15) Lösche die Ordner "Gothic III\Data\Infos" und "Gothic III\Data\Strings" wieder!

1.2 Einige neue Parameter in den .info-Dateien

Einige Modder baten uns, die Möglichkeiten des Dialog-Designs zu erweitern. Wir konnten aus Zeitmangel nur wenig umsetzen, aber wir hoffen, dass die Neuerungen gelegentlich kommen.

1.2.1 CondItemAmounts=0

Die altbekannte Bedingung "CondItemAmounts" steht mit den Parametern "CondItem" und "CondItemContainer" in Beziehung. Beispiel: Die Einträge

```
CondItemContainer=Markus
CondItems=It_FiremageCup
CondItemAmounts=1
```

besagen, dass dieser Dialog verfügbar ist, wenn der NPC "Markus" mindestens einen Feuerkelch (It_FiremageCup) in seinem Inventar hat.

Allerdings war es bis jetzt nicht möglich, einen Dialog genau dann zu ermöglichen, wenn ein NPC **nicht** das bestimmte Item irgendwo in seinem Inventar hatte.

Mit dem CP 1.70 werdet ihr das abfragen können, indem ihr "CondItemAmounts=0" einsetzt.

Achtung: Seid vorsichtig bei Händlern! Sie mögen das interessante Item vielleicht nicht ihrem "Plünderinventar" haben, aber es könnte trotzdem irgendwo in ihrem Handelsinventar herumliegen. Dies könnte zu unerwarteten Dialogkonstellationen führen.

1.2.2 CondHasSkill

Die Bedeutung dieser neuen bedingung sollte eigentlich klar sein.

Ihr könnt mehrere Fertigkeiten – und übrigens auch Zauber - hintereinander hängen, getrennt durch Semikolons.

Syntax:

```
CondHasSkill=Perk_Barter;Spl_IceBomb
```

(alle Fähigkeiten und Zauber werden ohne das Präfix "It_" angegeben!)

In diesem Beispiel wird der Dialog erst möglich sein, wenn der Held die Fähigkeit "Feilschen" und den Zauber "Eisexplosion" gelernt hat.

Achtung: Diese neue Bedingung ist immer dem Helden zugeordnet, keinem NPC! (NPCs verfügen sowieso nicht über Fähigkeiten oder Zauber.)

Und: Die Bedingung ist nur erfüllt, wenn der Held die Fähigkeiten und Zauber wirklich von einem Lehrer gelernt hat. Boni von Rüstungen, Waffen etc. zählen nicht!

1.2.3 CondPlayerKnowsNot

Diese neue Anweisung ist das genaue Gegenteil der bekannten Anweisung "CondPlayerKnows". Das bedeutet, dass die Bedingung erfüllt ist, wenn ein bestimmtes GameEvent noch nicht zuvor gesetzt wurde oder es schon wieder gelöscht (cleared) wurde.

Syntax:

```
CondPlayerKnowsNot=TheHeroDoesntKnowThis
```

1.2.4 CondWearsItem

Dieser neue Parameter bezieht sich immer auf den Parameter "CondItemContainer". Und in jeder Info-Datei kann man maximal nur einen "ItemContainer" angeben.

Die Bedeutung dieser Bedingung sollte klar sein: Wenn ein bestimmter NPC einen bestimmten Gegenstand trägt, z.B. eine Waffe, einen Ring, ein Amulet usw., wird der Dialog verfügbar sein.

Beispiel:

```
CondItemContainer=PC_Hero  
CondWearsItem=Body_Druid
```

Ergebnis: Der Dialog dieser Info-Datei kann nur aktiviert werden, wenn der Held eine Druidenrobe trägt.

Bitte stellt sicher, keine Situationen einzubauen, in denen ihr mehr als einen CondItemContainer in einer .info-Datei benötigt! Und beachtet bitte, dass sich der CondItemContainer einerseits auf die Parameter "CondItems" und "CondItemAmounts" und andererseits auch auf "CondWearsItem" bezieht. Es gibt ein kleines Risiko, diese beide Dinge durcheinander zu bringen.

1.2.5 Erase

Im Gegensatz zu den gerade erwähnten vier .info-Datei-Neuerungen, ist dieses kein Bedingungsparameter, sondern ein neuer Befehl für die InfoScript-Zeilen. So wie "Say", "Give", "SetGameEvent" etc.

Mit "Erase" könnt ihr "heimlich" eine bestimmte Anzahl von Items aus dem Inventar eines bestimmten NPC löschen. (Dieser Befehl wurde ursprünglich für das Feature "Kühe melken" erstellt, um jeweils eine leere Flasche aus dem Inventar des Helden zu löschen, wenn er eine Kuh melkt, ohne dass dabei eine Nachricht angezeigt wird.)

Syntax:

```
InfoScript_Commands=Erase  
InfoScript_Entities1=player  
InfoScript_Entities2  
InfoScript_IDs1=It_Gold  
InfoScript_IDs2=100  
InfoScript_Texts
```

Wir können nichtüberblicken, ob dieser Befehl auch in anderen Zusammenhängen sinnvoll ist. Bitte missbraucht ihn nicht – seid fair zu den Spielern. :-)

2 Neue Parameter in der ge3.ini

Es hat eine bescheidene Anzahl von Änderungen in der ge3.ini gegeben. Zwei neue Einträge benötigen einige weitere Erklärungen.

2.1 DisableDEP

Einige Spieler, insbesondere Vista-Nutzer, konnten Gothic 3 nicht spielen, sondern erhielten stattdessen ein Nachrichtfenster mit der Meldung "Genome Gaming System funktioniert nicht". Der Hauptgrund hierfür ist ein Sicherheitsmechanismus von Vista (und einige andere Betriebssysteme) namens "Data execution prevention", oder kurz: DEP. Dieser Schutzmechanismus kann manuell in der Systemsteuerung des Betriebssystems ausgeschaltet werden. Für diejenigen unter euch, die es nicht generell abschalten möchten oder die nicht wissen, wie es geht, haben wir den Parameter "DisableDEP" in der ge3.ini eingefügt.

Wenn dieser Parameter auf "true" steht, wird (hoffentlich) die besagte Fehlermeldung nicht mehr angezeigt werden.

Wenn ihr sowieso kein Problem mit dieser Meldung hattet, dann belasst es bitte bei dem vorgegebenen Wert "false".

Ihr benutzt diesen Parameter auf eigenes Risiko!

Es könnten trotzdem noch Probleme während des Spielens auftreten.

2.2 Threads.Priority

Mit diesem Schalter könnt ihr Gothic 3 eine höhere Priorität im Vergleich zu anderen Applikationen verleihen, die ebenfalls gerade auf eurem Rechner laufen, z.B. Adobe Reader. Die Haupt-Absicht der Prioritätssteigerung von Gothic 3 ist, mehr CPU-Zeit für das Spiel bereitzustellen. Dies sollte eine bessere Performance zur Folge haben, aber garantieren können wir das nicht.

Die Werte sind:

0=Normal (default),

1=Above Normal,

2=Highest

Probiert es aus, wenn ihr mögt, aber wenn ihr irgendeinen unliebsamen Nebeneffekt bemerkt, dann macht eure Änderung wieder rückgängig. Wir wissen aus unseren Tests, dass ein Prioritätenwert von 1 ("above normal") bei schwachen/langsamen Rechnern Probleme mit der Spielsteuerung hervorrufen kann, was sich in immer stärkeren Verzögerungen aller Tastatur- oder Mouse-Aktionen zeigt.

Daher: Alle Änderungen der Thread.Priority geschehen auf euer eigenes Risiko.