Schöner programmieren Teil 6 – Begrifflichkeiten und Pausen

Description

Im letzten Teil dieser Serie möchte ich auf ein paar übrig gebliebene Themen eingehen. Soll man bei der Namensgebung von Ressourcen und Variablen die Einzahl oder die Mehrzahl verwenden und wie sieht es mit längeren Pausen aus?

Einzahl oder Mehrzahl?

Die Namensfindung von Variablen und Ressourcen haben wir bereits im zweiten und im dritten Teil behandelt. Was manchmal ein Problem sein kann: Verwendet man besser die Einzahl oder die Mehrzahl? Das Problem stellt sich bei mir vor allem bei Scripten und in PHP bei Funktionsnamen. Ein Script, in dem alle globale Variablen stecken (kann ich nur empfehlen um den Chaos Herr zu werden) kann ich *scr_global_variabl*e oder *scr_global_variables* nennen. Oder ein Script, welches das Aussehen von bestimmten Wänden steuert (welches Sprite verwendet werden muss) kann ich *scr_red_wall* oder *scr_red_walls* nennen.

Auch über dieses Thema sollte man sich vor Projektstart genug Gedanken machen. Ich tue mich manchmal mit der Entscheidung ebenfalls schwer, habe mich aber darauf eingelassen, dass ich nur dann die Mehrzahl verwende, wenn es keinen Fall gibt, in dem keine Mehrzahl vorkommt. Um bei den oberen Beispielen zu bleiben: das Script für die Globale Variablen heißt bei mir *scr_global_variable***s**, weil ich immer mehrere habe. Das Script, welches das Aussehen der Wand steuert, heißt *scr_red_wall* , weil es zwar auch tausend Wände sein können, aber je nach Level kann es auch nur eine Wand sein. Außerdem bestimmt das Script das Aussehen eines einzelnen Steins und fragt nicht alle Steine in einer Schleife ab. Das kann übrigens ebenfalls ein Hinweis sein, ob man Plural oder Singular verwendet: hat das Script Schleifen? Wie bereist dargelegt, sollte es nicht vorkommen, dass man beide Namen verwendet, weil man damit in Rekordzeit sein eigenes Grab schaufelt.

Noch besser ist es jedoch, wenn man konsequent Namen verwendet, die genau aussagen, was die Ressource, die Variable oder das Script macht. Bei den globalen Variablen kann man sich darüber streiten, ob ein *scr_set_all_global_variables* wirklich besser ist, da man wahrscheinlich nur ein Script haben wird, in dem es ausschließlich um globale Variablen geht. Bei den roten Wänden wäre eine

Bezeichnung wie *scr_change_all_red_walls* mit Sicherheit besser, oder, um beim Singular zu bleiben: *scr_change_red_wall*.

Große Pausen

Das hat nun nichts mit Programmierstil zu tun, aber ist dennoch ein wichtiger Tipp an Anfänger. Wenn ihr eine Entwicklungsumgebung mit Programmiersprache wie den GM erlernt, dann macht bei euren Projekten keine große Pausen. Programmiersprachen sind wie Fremdsprachen: Wenn man sie nicht nutzt, vergisst man sie wieder und das umso mehr, je weniger man sie beherrscht hat. Wer sich also wirklich dazu entscheidet eine Sprache / Entwicklungsumgebung zu erlernen, sollte das eisern durch ziehen und nicht zwischendurch mehrere Wochen pausieren. Selbst wenn man nicht immer programmieren kann, sollte man zumindest immer wieder über die Programmiersprache Artikel lesen oder sich in Foren umsehen, damit man nicht zu vieles verlernt.

Pausen haben bei komplexen Projekten zusätzlich den Nachteil, dass man ziemlich lange braucht, um das Projekt als ganzes zu begreifen. Variablen, Ressourcen, Abhängigkeiten von Objekten und Scripten sind nicht immer leicht zu verstehen. Man steht oft vor einem riesigen Berg, mit einem noch größeren Fragezeichen darüber. Spätestens jetzt merkt man, ob man gut kommentiert, bezeichnet oder eingerückt hat. Doch selbst wenn man das alles getan hat, wenn man alle Tipps befolgt hat, stellt es eine große Herausforderung dar.

Es hat seine Grüne, warum viele, auch vielversprechende Hobby Projekte, niemals fertig werden. Das Leben, egal ob berufliche oder private Umstände, zwingen einen immer wieder zum pausieren und aus eine Projekt, welches in sechs bis zwölf Monaten fertig wäre, werden mehrere Jahre, bis es irgendwann ein schläft. Man holt es von irgend einer Backup-Disc wieder hervor, schaut es sich an, schüttelt den Kopf und schließt es wieder. Das sollte man aber nicht tun.

Alte Projekte haben den Vorteil, dass man die Gelegenheit bekommt, seine eigene Entwicklung zu begutachten. Das kann negativ ausfallen, wenn man bemerkt, was man alles verlernt hat. Es kann aber auch positiv sein, wenn man sieht, wie gut man mittlerweile wurde. Egal wie das persönliche Urteil aussieht, so hat man die Chance, das Projekt zu beleben und es besser zu machen. Sofern es die Zeit zulässt, sollte man sich auf das Abenteuer einlassen.

Mir passiert es eher selten, weil ich meistens meine Projekte abschließe, aber wenn mir doch ein altes Projekt in die Hände fällt, bei dem ich wieder Feuer und Flamme bin, dann gehe ich wie folgt vor. Ich öffne den ersten Raum, schaue, welche Objekte sich darin befinden und editiere sie dann. Nun gehe ich alle Events und Scripte nach und nach durch, bis ich wirklich verstanden habe, wie alles funktioniert. Dabei entdecke ich immer wieder interessante Dinge. Zum Beispiel, wie sich meine eigene Einrückung verändert hat oder Variablen, die ich so nicht mehr verwenden würde. Auch stoße ich immer wieder auf Problemlösungen, die ich heute anders machen würde oder auf ungenügende Kommentare. Diese Stellen werden nach und nach verbessert. So wird nicht nur das Spiel besser, wenn auch nur intern, ich verstehe dadurch auch das ganze Projekt und komme irgendwann an einen Punkt, an dem ich Neuerungen einbauen kann.

Eine Pause muss somit nicht immer ein Nachteil sein. Manchmal reichen die eigenen Fähigkeiten nicht aus, um ein Projekt so zu vollenden, wie es der eigene Anspruch verlangt. Nach einer längeren Pause kann man das Projekt mit neuem Wissen, mit neuen Ideen und frischer Begeisterung angehen. Und wenn man der Meinung ist, dass das alte Projekt zu chaotisch ist, sich aber immer noch für die Idee

begeistern kann, dann hat man vielleicht die Gelegenheit, das Projekt neu zu starten. Auch das kann eine sehr lehrreiche Erfahrung sein.

Date Created
8. November 2016
Author
sven