

A VPython használata 3D animációk készítéséhez

A VPythont szokatlanul könnyű használni navigálható, valós idejű 3D animációkat generáló programok készítéséhez. A Python programozási nyelven alapul, amelyet egyszerű kialakításának köszönhetően széles körben használnak a bevezető programozási kurzusokon, valamint a tudományban és az üzleti életben egyaránt. A glowsript.org weboldalon kattintsunk a "Példaprogramok" („Example programs”) elemre, hogy a gyakorlatban is lássuk a VPython használatát. A tinyurl.com/VPintro oldalon meg tudjuk hallgatni Bruce Sherwood 8 perces bevezetését a VPythonhoz. A GlowScript VPython használatához nincs szükség speciális szoftver telepítésére, mivel az a böngészőben fut. A VPython használható [a már telepített Python-nal](#) ("VPython 7") is.

Ez a dokumentáció a Python szintaxisa alapján készült, mivel szinte minden GlowScript felhasználó a Pythont használja programozáshoz, de GlowScript programokat JavaScriptben is lehet írni. [Ez a cikk](#) elmagyarázza, hogyan kell értelmezni a dokumentációt JavaScript programokban való felhasználásra.

A bal oldalon a VPython teljes dokumentációja megtalálható.

A gyors bevezetéshez nézzük meg a következő YouTube-videókat, de az angol nyelvű videók megtekintésekor ügyeljünk arra, hogy a VPython jelenlegi verziója esetén a modul neve "vpython", nem pedig "visual", a grafikus megjelenítő pedig a böngésző lapon, nem egy üres ablakban jelenik meg. (A GlowScript VPython esetében el is hagyhatjuk a vpython importálását.)

[3D objektumok](#) (Javier Carro, [Objetos 3D](#) - spanyol verzió)

[Változó hozzárendelés](#) (Javier Carro, [Asignación de Variables](#) - spanyol változat)

[Loopok](#) (Javier Carro, [Comenzando con Bucles](#) - spanyol változat)

[Loopok és animációk](#) (spanyol változat [Bucles y Animación](#), Javier Carro)

[Scaling tényezők](#) (spanyol változat [Factores de Escala](#) Javier Carro)

[Szintaxis hibák - debugging](#)

[Listák, 1. rész](#) (spanyol nyelvű változat [Listas 1](#) Javier Carro)

[Listák, 2. rész](#)

[GlowScript VPython oktatóprogramok](#) sorozatát készítette el Rhett Allain a mozgás számításos előrejelzésével összefüggésben egy bevezető fizika kurzus szintjén. Az oktatóanyagokban használt eszközök a [trinket.io webhelyen](http://trinket.io), a GlowScript VPython programok írását és futtatását lehetővé tevő notebookszerű környezetbe vannak beágyazva. A Trinkets segítségével könnyedén hozzáadhatjuk a GlowScript VPython szerkesztését és futtatását lehetővé tevő programot a saját weboldalunkon. Erre Allain a Wired magazin fizika témájú blogjában mutatott is példákat.

Első lépések

VPython program írásához jelentkezzünk be a glowsript.org weboldalon. Ajánlott Chrome vagy Edge böngésző használata, mivel ezek jelenítik meg a legjobban a hibaüzeneteket, bár a programokat minden böngészőben lehet írni és futtatni, beleértve az okostelefonokon és a táblagépeken futókat is. Bizonyos esetekben a programhibák csak akkor jelennek meg, ha megnyomjuk a Shift-Ctrl-J kombinációt a böngésző konzol megjelenítéséhez.

* A következő mondatot fogja látni: "Ön bejelentkezett, <felhasználónév> néven, a programjai itt találhatóak." Kattintsunk az itt szóra.

* Kattintson az Új program létrehozása (Create New Program) elemre, amely új programot hoz létre a Programjaim (MyPrograms) mappában. (A privát mappában lévő programok nem érhetők el mások számára.) Válasszunk nevet a program számára. A név nem tartalmazhat szóközt vagy alsóvonást.

* A GlowScript 3.0 VPython sor alatt írja be azt, hogy `box()`. A program futtatásához kattintson a Futtatás (Run) gombra. Fekete alapon fehér doboz fog megjelenni.

* Az egér jobb gombjával (vagy CTRL- bal gomb) forgassa el a kamerát a különböző szögekből történő megtekintéshez.

* A nagyításhoz és kicsinyítéshez használjuk a megfelelő két gombot, az ALT/OPTION - húzás, vagy az egérgörgő opciók egyikét. Érintőképernyő esetén: húzással vagy két ujjú forgatással; összecsapással/kiterjesztéssel navigáljunk a nagyításhoz, kicsinyítéshez.

* A balra/jobbra vagy fel/le navigáláshoz mozgassuk az egeret a SHIFT gomb lenyomva tartásával.

A szövegszerkesztő használata

Itt található a kereséshez, cseréhez stb. Tartozó [billentyűparancsok](#) listája. Szerkesztés közben nyomja meg a Ctrl-1 (Mac számítógépen a Cmd-1) gombot a program futtatásához ugyanabban az ablakban. Nyomja meg a Ctrl-2 (Mac számítógépen a Cmd-2), hogy egy külön ablakban futtassa a programot, amely lehetővé teszi a program és annak végrehajtása egymás melletti megtekintését. Ha megváltoztatja a programot, nyomja meg ismét a Ctrl-2 gombot a szerkesztő képernyőjén, hogy újraindítsa a programot az új változtatásokkal. A GlowScript az ACE szövegszerkesztőt használja. Mivel az ACE nem működik mobil eszközökön, ott egyszerűbb szerkesztőt használnak.

Különösen hasznos parancsikon a Ctrl- / (Cmd- / Mac esetén). Jelöljön ki egy vagy több sort a programjában, és ezzel a gombnyomással válthat, hogy ezeket a sorokat be vagy ki kommentálják-e. Ezenkívül válasszon ki egy vagy több sort, és nyomja meg a Tab billentyűt a behúzáshoz, vagy a Shift-Tab billentyűt a behúzáshoz.

Hagyja, hogy mások futtassák a programját

A program szövegének megtekintésénél kattintson a Program megosztása (Share this program) elemre, hogy megnézze, mások hogyan futtatják az Ön programját. Ahhoz, hogy az emberek futtatni tudják a programját, ahhoz a programnak nyilvános mappában kell lennie, vagy pedig exportálni kell azt a saját weboldalára. Valójában a megosztási oldalon elérhető kódot csak egyszerűen be kell illeszteni egy fájlba, és menteni ".html" kiterjesztéssel, majd a html fájlra való dupla kattintással futtathatjuk is a programot.

A bal oldalon elérhető opciók leírása

Bevezetés: a Python és a VPython alapjai.

VPython oktatóprogram: még több anyag a VPythonról, beleértve az animációkészítést is.

Bevezető videók : A fent felsorolt videók, amelyek könnyen elérhetők a Súgóból is

3D objektumokról: Hogyan néznek ki az ilyen objektumok?

Válasszon egy 3D objektumot: A henger, doboz stb. részletei. Áttekintéshez kezdje a [hengerrel](#) .

Munka a 3D objektumokkal: Az összes 3D objektumra alkalmazható módosítások: szín, anyag stb. A kulcsfontosságú elemek a következők:

Vektorműveletek: nagyság, pont- és kereszt, forgatás stb.

Grafikonok: Grafikonok készítése az adatokból.

Matematikai funkciók: faktoriális / kombinatív statisztikai függvényeket tartalmaz

Munkalap/események: ablakok létrehozása/módosítása; egér/billentyűzet események kezelése

Érdeemes felfedezni a számos GlowScript- [példaprogramot](#), amelyeket mások Python vagy JavaScript segítségével írtak. A VPython opció a [RapydScript-NG](#) Python-to-JavaScript fordítón alapszik.

Ahogy már korábban is szó volt róla, a [trinket.io](#) egy másik környezetet is kínál a GlowScript VPython programok írására. Kattintsunk az "Új Trinket" („New Trinket”) elemre, majd válasszuk a "GlowScript" lehetőséget, aztán a Python-t a normál szövegszerkesztéshez, vagy a Blockst a kért szerkesztéshez.

Általában a Python modulokat nem lehet importálni a GlowScript VPython programokba, mert azok JavaScript környezetben működnek. Ugyanakkor a Python "random" modult importálhatjuk VPython programba, amelyet a RapydScript-NG eszköz biztosít. Ez a Python-t JavaScriptté alakítja. Az „importálás randomból” nem használható, viszont a „random importálása” vagy a „random importálása rr-ként” (vagy más néven), illetve az „importálás randomból randint, random” (vagy más funkciók listája) működik. A Python random moduljának nem minden funkciója [érhető el](#).

[A Pythonban és/vagy VPythonban jártas felhasználók számára](#)

Ha Ön tapasztalt Python programozó és/vagy nagy tapasztalattal rendelkezik a régebbi „Klasszikus VPython” (VPython 6 vagy annál korábbi) alkalmazással kapcsolatban, akkor olvassa el [ezt a cikket](#) a különbségekről.

[A GlowScript használata internetkapcsolat nélkül; fejlesztők általi felhasználás](#)

Ez a [egy cikk](#), elmagyarázza, hogyan kell használni a GlowScriptet internetkapcsolat nélkül, beleértve azt is, hogy a fejlesztők hogyan használhatják ki ezt a képességet.

[A GlowScript architektúrája](#)

[Itt található](#) a GlowScript architektúra rövid áttekintése. A GlowScript architektúra sokkal szélesebb körű műszaki leírása pedig [itt érhető el](#), ideértve a témával foglalkozó workshop videókat is. Ez a [cikk](#) pedig leírja, hogyan valósult meg a pixelszintű átláthatóság és az egérválasztás.

[A VPython rövid története](#)

A VPython-t David Scherer hozta létre 2000-ben. 2011-ben David Scherer és Bruce Sherwood kezdeményezte a GlowScript, egy hasonló programozási környezet fejlesztését, amely böngészőben fut. Eredetileg a programokat JavaScript-ben kellett írni, de 2014-ben lehetővé vált a Python használata is. Jelenleg a GlowScript programokat Pythonban vagy JavaScriptben is meg lehet írni. Itt olvashatjuk el a [fontosabb változtatásokat](#), és itt található a projekt [nyílt forráskódja](#). Egy érdekes blogbejegyzés: "[A VPython fejlődésének idővonala](#)". A vezető fejlesztő Bruce Sherwood.

A VPython telepített Pythonnal is használható, köszönhetően a John Coady által 2014-ben kifejlesztett munkának, valamint John Coady, Ruth Chabay, Bruce Sherwood, Steve Spicklemire és Matt Craig folyamatos fejlesztésének. A GlowScript a VPython szintaxisát használja, de a szabványos Pythonnal kiegészítve, ezáltal hozzáférést biztosítva a szabványos Python-modulokhoz is. Ezt a VPython verziót "VPython 7"-nek hívják. Részletekért lásd a [vpython.org oldalt](#). Ez a dokumentáció mind a GlowScript VPython (böngésző alapú), mind a VPython 7 (telepített Python) esetén érvényes.

[Köszönetnyilvánítás](#)

Salvatore di Dio a <http://salvatore.pythonanywhere.com/RapydGlow> RapydGlow-val való kísérletezései során demonstrálta, hogyan lehet használni a RapydScript Python-to-JavaScript

fordítót GlowScript grafikával. Ez inspirálta a VPython (vpython.org) API megvalósítását a glowscript.org weboldalon. Ő szolgáltatta az operátorok túlterheléséhez a papercomp.js fájlt (PaperScript: <http://scratchdisk.com/posts/operator-overloading>) Jürg Lehni munkája alapján. Továbbá az operátorok túlterhelését és a szinkron kód írásának képességét is lehetővé tette a transform-all.js fájlal, melyek Bruno Jouhier (Streamline: <https://github.com/Sage/streamlinejs>) és Marijn Haverbeke (Acorn.js: <https://github.com/marijnh>) munkája alapján készültek. A VPython böngészőben való API támogatása Alexander Tsepkov (RapydScript: <https://github.com/atsepkov/RapydScript>) és Charles Law (böngésző alapú RapydScript: https://github.com/charleslaw/RapydScript_web) munkájának köszönhetően lehetséges.

2017 januárjában az eredeti RapydScript fordítót helyettesítette **RapydScript-NG**-vel Kovid Goyal, amely közelebb áll a valódi Python-szintaxis kezeléséhez. Goyal kivételesen támogató szellemiségben dolgozik, gyakran néhány órán belül kijavítja a beérkező hibákat. Továbbá ő a Caliber fejlesztője, amely egy széles körben használt eszköz az e-könyv formátumok közötti konvertáláshoz.

Amikor David Scherer és Bruce Sherwood 2011-ben elindította a GlowScript projektet, Scherer az operációs rendszer túlterhelését és a szinkron kódot az akkoriban létező könyvtárak felhasználásával valósította meg. 2015-ben szükségessé vált az újabb könyvtárakra való frissítés, mivel egyes böngészőkben nem sikerült a futtatás. Salvatore di Dio nagyon nagy segítséget nyújtott a frissítésben. 2017 januárjában a rendszert tovább frissítették.

A VPython dokumentációját Ruth Chabay, David Scherer és Bruce Sherwood készítette.

Fordította: kaszinoonline.net