|  |  |
| --- | --- |
| Get Started with Pygame: A Beginner's Guide | Mediumpygame 2.5.0 - 🦋 delicious but a weird 🍄 — 24 Jun, 2023  desenvolupament d’ UN VIDEOJOC amb PYTHON I PYGAME  [Document subtitle] | Abstract  [Draw your reader in with an engaging abstract. It is typically a short summary of the document. When you’re ready to add your content, just click here and start typing.]  Gonzalez Moret Lluc  [Course title] |

Index

[1. Introducció 2](#_Toc148087943)

[1.1 Justificacio 2](#_Toc148087944)

[1.2 Objectius 2](#_Toc148087945)

[1.3 Procés 2](#_Toc148087946)

[1.4 Resultat 2](#_Toc148087947)

[2. Introducció a Python 2](#_Toc148087948)

[2.1 Què és Python? 2](#_Toc148087949)

[2.2 Perquè Python ? 3](#_Toc148087950)

[2.3 Qui fa servir Python? 4](#_Toc148087951)

[2.4 Python vs altres llenguatges 4](#_Toc148087952)

[2.5 Com escriure codi de Python 4](#_Toc148087953)

[2.5.1 Entorn de desenvolupament (IDE) 4](#_Toc148087954)

[2.5.2 Control de Versions 5](#_Toc148087955)

[2.6 Què puc fer amb Python? 5](#_Toc148087956)

[2.6.1 Mòduls 5](#_Toc148087957)

[2.6.2 Usos 5](#_Toc148087958)

[3. Programació de Videojocs 5](#_Toc148087959)

[3.1 Introduccio 6](#_Toc148087960)

[3.2 Videojocs amb Python (Game Frameworks) 6](#_Toc148087961)

[3.2.1 Pygame 6](#_Toc148087962)

[3.3 Motors de desenvolupament (Game Engines) 6](#_Toc148087963)

[3.4 Comparativa 6](#_Toc148087964)

[4. El Joc 7](#_Toc148087965)

[4.1 Proces Creatiu 7](#_Toc148087966)

[4.2 Desenvolupament 7](#_Toc148087967)

[4.3 Documentacio 7](#_Toc148087968)

[5. Conclusions 7](#_Toc148087969)

[6. Bibliografia i Webgrafia 7](#_Toc148087970)

# Introducció

## Justificacio

<Perque has triat aquest tema>

## Objectius

<Objectiu del treball>

E.g: Aprendre Python de forma lúdica. Coneixer l’entorn de programació (eines) i les practiques “professionals” (e.g: control de versions)

## Proces

Idea inicial. Com s’ha desenvolupat. Tria del llenguatge/llibreria/eines (vídeos/llibres de consulta

Model de joc triat (motiu)

## Resultat

<Estructura del treball presentat. Quins temes formen el marc teòric i quins temes formen la descripció de la part pràctica>

E.g:

-

Introduccio al Python i al desenvolupament de videojocs: Temes 2/3

El Joc: tema 4

Conclusions: tema 5

# Introducció a Python

## Què és Python?

Python és un dels llenguatges de programació més populars i utilitzats en l’actualitat, el qual destaca per la seva gran varietat d’aplicacions en la informàtica i per utilitzar una sintaxi senzilla molt similar a l’anglès, la qual el converteix en una de les millors opcions per a programadors principiants que busquen un entorn al que adaptar-se ràpidament.

Python fou inventat a principis de la dècada de 1990 pel programador neerlandès Gudio van Rossum, el qual es va basar en un llenguatge de programació ja existent anomenat ABC que mai va a resultar ser tan exitós com el seu successor. El seu nom està inspirat en la comèdia “Monty Python’s Flying Circus” que es retransmetia a la BBC, ja que Gudio Van Rossum era un gran aficionat a la sèrie i admirador dels Monty Python. Tot i això, molta gent relaciona aquest nom amb el rèptil que comparteix el seu nom amb el programari, el pitó, per això molts llibres o icones relacionats amb Python solen representar el llenguatge amb la imatge d’una serp, fins hi tot el propis logotip de Python consisteix en dues serps de color blau i groc respectivament.

La primera versió de Python es va llançar oficialment l’any 1991, però no va ser fins l’any 1995 que es va publicar una versió complerta del programari, en aquest cas es tractava de la versió 1.0, que ja permetia treballar amb classes, funcions i oferia una sintaxi senzilla, pel que va començar a guanyar popularitat pocs dies després de publicar-se. L’any 2000 Gudio Van Rossum va crear un equip amb altres programadors anomenat “BeOpen Python Labs”” i va publicar la versió 2.0 del llenguatge, la qual destacava per permetre la creació de llistes, un dels elements més importants del programari. Finalment durant l’any 2008 es va publicar la última gran actualització, la versió 3.0, la qual va corregir la majoria d’errors i inconvenients que el llenguatge havia presentat fins aquell moment. A partir d’aquest punt només s’han llançat actualitzacions a petita escala per a resoldre bugs o millorar la seguretat, concretament, la versió utilitzada en la realització d’aquest treball ha estat la versió 3.7.9.

En comparació amb altres llenguatges de programació com Java o C++, Python presenta una velocitat força inferior alhora de dur a terme l’execució del codi, això es deu al fet que Python pertany a un grup conegut com a “llenguatges interpretats”, aquests tipus de llenguatges necessiten que les seves instruccions siguin interpretades abans de poder-se executar, convertint-les en el que es coneix com “byte code”, un codi que es pot convertir al sistema binari per a que així finalment el pugi interpretar el propi ordinador (vegeu figura 1.1). Això provoca que aquests llenguatges tinguin un rendiment inferior comparats amb els “llenguatges compilats”, que poden ser executats directament. La principal avantatge que presenten els llenguatges interpretats com Python, és que al utilitzar un interpretador, no depenen completament del sistema i per tan poden ser molt més flexibles en aspectes com la sintaxi.

Una captura de pantalla de un celular con texto e imagen

Descripción generada automáticamente con confianza media

Figura 1.1: Esquema que representa el procés pel qual passa un codi escrit en un llenguatge interpretat quan s’executa.

## Perquè Python ?

Python ha anat guanyant popularitat amb el pas del temps des del seu llançament i actualment compta amb milions d’usuaris arreu del món, els quals formen una gran comunitat. Existeixen un gran nombre de raons per les quals Python resulta atractiu per a molts programadors, però a continuació es destaquen aquelles que solen tenir més pes:

**Sintaxi senzilla i entenedora:** Com s’ha mencionat prèviament, Python utilitza instruccions que presenten una gran semblança amb la llengua anglesa i que poden resultar intuïtives fins hi tot per a persones sense experiència amb el llenguatge. Gràcies a això els codis escrits en Python solen resultar molt més senzills de llegir i entendre que aquells escrits en altres llenguatges.

**Productivitat més alta:** Programar en Python sol resultar molt més eficient que programar en llenguatges compilats com Java o C++, això es deu al fet que Python és un llenguatge interpretat, per tan utilitza instruccions més senzilles i curtes que s’interpreten posteriorment. Com a referència, si tenim un mateix programa escrit en Python i C++, el programa escrit en Python serà de tres a cinc vegades més curt que aquell escrit en C++. Un codi més curt costa menys de programar i mantenir i per tan resulta molt més productiu.

**Portabilitat dels programes:** Un codi escrit en Python funciona en la gran majoria de dispositius, fins i tot si el dispositiu utilitza un sistema operatiu diferent, un codi escrit en un ordinador que utilitza Windows es pot executar sense necessitat d’alterar res en un ordinador que utilitza Linux, només és necessari que el dispositiu tingui el programari instal·lat.

**Suport de llibreries:** Python porta un grup de llibreries bàsiques per defecte que permeten al usuari afegir funcions més concretes en el seu codi, a més a més, permet importar milers de llibreries creades per altres usuaris que permeten resoldre tasques que el software bàsic de Python no podria. Python és un llenguatge amb un dels catàlegs de llibreries més ric i extens que existeix gràcies a la seva comunitat, això permet que Python sigui un programari utilitzat en una gran varietat de camps.

**Programació orientada a objectes:** Python és un llenguatge amb programació orientada a objectes, això permet al usuari crear una sèrie de classes que comparteixen uns mateixos trets i funcions i finalment crear objectes que tinguin certes propietats. Aquesta característica és un tret indispensable si es vol crear un videojoc, ja que agilitza i estructura molt més el funcionament del codi.

## Qui fa servir Python?

Ref1

<exemples d’empreses que fan servir Python>

*Google makes extensive use of Python in its web search systems.*

*The popular YouTube video sharing service is largely written in Python.*

*The Dropbox storage service codes both its server and desktop client software pri- marily in Python.*

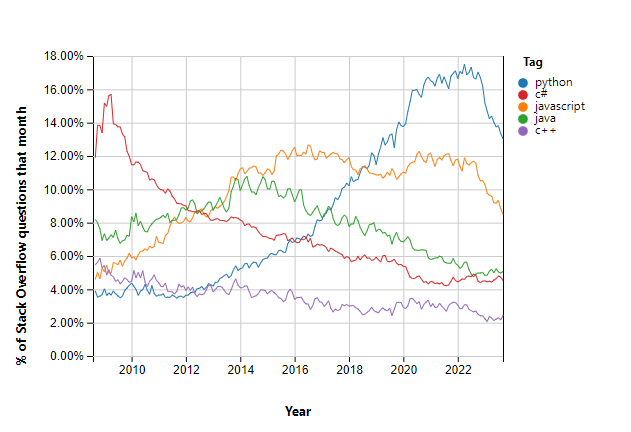
*The Raspberry Pi single-board computer promotes Python as its educational lan-guage.*

*The widespread BitTorrent peer-to-peer file sharing system began its life as a Python program.*

## Python vs altres llenguatges

Mostrar stackoverflow trends....

<https://insights.stackoverflow.com/trends?tags=python%2Cjavascript%2Cjava%2Cc%23%2Cc%2B%2B&utm_source=so-owned&utm_medium=blog&utm_campaign=gen-blog&utm_content=blog-link&utm_term=incredible-growth-python>



## Com escriure codi de Python

### Entorn de desenvolupament (IDE)

<explicar que es un IDE, tipus de IDEi explicar el IDE que has triat i perquè>

<https://www.simplilearn.com/tutorials/python-tutorial/python-ide>

<https://realpython.com/python-ides-code-editors-guide/>

### Control de Versions

<explicat qe es el control de versions, com tájuda en ell projecte. Com l´has integrat en el teu IDE>

*Version control software allows you to take snapshots of a project whenever it’s in a working state. When you make changes to project—for example, when you implement a new feature—you have the option of reverting back to a previous working state if the project’s current state isn’t functioning well.*

*Using version control software gives you the freedom to work on improvements and make mistakes without worrying about ruining your project. This is especially critical in large projects, but can also be helpful in smaller projects, even when you’re working on programs contained in a single file*. [Ref 5]

## Què puc fer amb Python?

### Mòduls

<Explicar que es un mòdul de Python (simple explicació)>

Ref <https://realpython.com/python-modules-packages/>

### Usos

Ref1

<es poden indicar noms de llibreries per cada cas si s’ escau>

Systems Programming/GUIs/Database Programming/Numeric and Scientific Programming/Gaming/Images/Data Mining/Robots

# Programació de Videojocs

## Introduccio

<http://ithare.com/programming-guide-for-video-gamers/>

<explicar les maneres que hi ha de fer-los: llenguatges de programació vs motors. Raw code (aka Game Frameworks) vs engines (aka Game Enginess)

Aquest video es molt interessant: explicant que es cada cosa

<https://www.youtube.com/watch?v=w2W6rE87Byw>

altres fonts

<https://www.linkedin.com/advice/0/how-do-you-learn-master-new-game-engine-framework-quickly>

## Videojocs amb Python (Game Frameworks)

<Llistat de les llibreries de Python per fer jocs>

<https://wiki.python.org/moin/PythonGameLibraries>

Grafica de percentatges d´´us

### Pygame

Explicacio mes detallada del pygame

Ref <https://www.pygame.org/docs/tut/PygameIntro.html>

## Motors de desenvolupament (Game Engines)

<Breu llista dels meus usats>

## Comparativa

*Es pot fer una taula amb el que surt al video*

# El Joc

## Proces Creatiu

Eines que vas triar (llenguatge/framework/IDE/control de versio).

Tema del joc.

## Desenvolupament

Controls

Limits

Personatges

Grafics

So

Objectiu

*Complementar cada descripcio amb video (??)*

## Documentacio

Manual tecnic del projecte. Com

<https://docs.python-guide.org/writing/documentation/>

Pot ser una bona idea executar Sphinx per documentar el programa (??)

<https://www.sphinx-doc.org/en/master/>

# Conclusions

Que es el que t’ha agradat mes o t’ha sorprès?

# Bibliografia i Webgrafia

Ref 1) Learning Python – Mark Lutz, O’Reilly, 2013

Ref 2) <https://en.wikipedia.org/wiki/Python_(programming_language)>

Ref 3) Python, PyGame, and Raspberry Pi Game Development, Sloan Kelly, Appress 2019

Ref 4) <https://code.visualstudio.com/docs/sourcecontrol/overview>

Ref 5) Python Crash Course, 3rd Edition: A Hands-On, Project-Based Introduction to Programming , Eric Matthes, No Starch Press (2023)