|  |  |
| --- | --- |
| 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  rrrr  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 3](#_Toc148087948)

[2.1 Què és Python? 3](#_Toc148087949)

[2.2 Perquè Python ? 4](#_Toc148087950)

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

[2.4 Python vs altres llenguatges 5](#_Toc148087952)

[2.5 Com escriure codi de Python 6](#_Toc148087953)

[2.5.1 Entorn de desenvolupament (IDE) 6](#_Toc148087954)

[2.5.2 Control de Versions 6](#_Toc148087955)

[2.6 Què puc fer amb Python? 7](#_Toc148087956)

[2.6.1 Mòduls 7](#_Toc148087957)

[2.6.2 Usos 7](#_Toc148087958)

[3. Programació de Videojocs 9](#_Toc148087959)

[3.1 Introduccio 9](#_Toc148087960)

[3.2 Videojocs amb Python (Game Frameworks) 9](#_Toc148087961)

[3.2.1 Pygame 10](#_Toc148087962)

[3.3 Motors de desenvolupament (Game Engines) 11](#_Toc148087963)

[3.4 Comparativa 11](#_Toc148087964)

[4. El Joc 11](#_Toc148087965)

[4.1 Proces Creatiu 11](#_Toc148087966)

[4.2 Desenvolupament 11](#_Toc148087967)

[4.3 Documentacio 11](#_Toc148087968)

[5. Conclusions 12](#_Toc148087969)

[6. Bibliografia i Webgrafia 12](#_Toc148087970)

# Introducció

## Justificació

<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 2.1: Esquema que representa el procés pel qual passa un codi escrit en un llenguatge interpretat quan s’executa. (Font: “Learning Python – Mark Lutz, O’Reilly, 2013”)

## 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?

Actualment Python és un dels llenguatges de programació més utilitzats que podem trobar, ja sigui en projectes personals o professionals, i compta amb aproximadament un milió d’usuaris arreu del món. A més a més Python cada cop es troba integrat en més aplicacions, per exemple, fa poc que Microsoft ha anunciat que serà possible escriure codi en Python dins l’aplicació Excel sense cap instal·lació, això suposa una gran oportunitat per millorar el tractament de dades a Excel, fet que resultarà molt útil per als usuaris més recurrents de l’aplicació.

Python ja porta aproximadament 30 anys existint, això li ha permès consolidar-se com un llenguatge robust i actualitzat conegut mundialment. Encara que Python sigui un llenguatge que generalment s’utilitzi en projectes individuals o personals, també podem trobar que moltes companyies l’utilitzen en alguns dels seus projectes més ambiciosos, entre aquests projectes destaquen:

* Google, el qual fa ús de Python en el seu sistema de cerca de pàgines web
* Youtube, gran part del codi base d’aquesta plataforma es troba escrit en Python
* Dropbox, els servidors i clients d’aquest sistema d’emmagatzematge online es troben escrit principalment en Python
* BitTorrent, aquesta popular aplicació de distribució de fitxers es troba programada purament en Python.

## Python vs altres llenguatges

Existeixen una gran varietat de llenguatges de programació, cadascun d’ells té certs punts forts i característiques que els fan útils per a diferents tasques, pel que no es pot considerar que cap llenguatge sigui el més efectiu o útil, tampoc podem parlar estrictament de quin llenguatge és el més utilitzat ja que dependrà de quins projectes tinguin en compte a la hora d’obtenir les dades. És per aquesta raó que la manera més efectiva de determinar si un llenguatge està quedant més enrederit o està guanyant popularitat és observar la rellevància que té entre els programadors actuals. Un mètode per a fer això és utilitzar pàgines com “StackOverflow” on els usuaris comparteixen dubtes o problemes que presenta el seu codi, “StackOverflow” classifica quin a quin llenguatge pertany cada dubte que es troba a la web i utilitza això per determinar el llenguatge amb més “rellevància del moment.

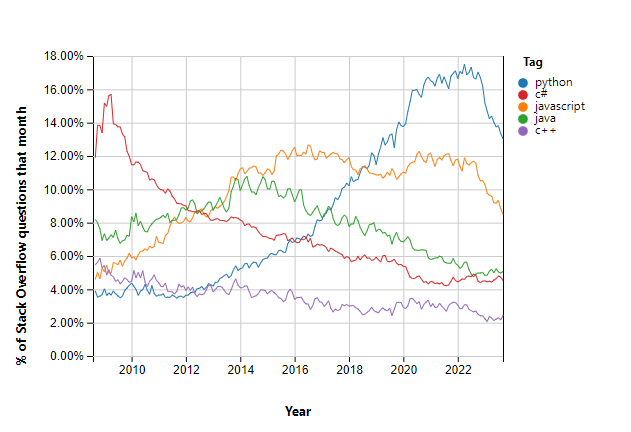


Figura 2.1, el gràfic mostra el percentatge de preguntes que s’han realitzat cada dos anys sobre diferents llenguatges a la pàgina “StackOverflow”, Font: (<https://insights.stackoverflow.com/trends?tags=python%2Cjavascript%2Cjava%2Cc%23%2Cc%2B%2B&utm_source=so-owned&utm_medium=blog&utm_campaign=gen-blog&utm_content=blog-link&utm_term=incredible-growth-python> )

El gràfic anterior (vegeu figura 2.1), proporcionat per la pròpia web de “StackOverflow”, ens mostra quin percentatge del total de preguntes realitzades cada mes pertany a cada llenguatge, l’extensió temporal del gràfic alberga des de l’any 2008 fins a l’actualitat. Aquest gràfic mostra el creixement que la comunitat de Python ha experimentat els últims anys, començant amb el percentatge de preguntes més baix l’any 2008 i arribant a l’actualitat amb el percentatge més alt de tots, aproximadament un 13% de preguntes.

Resulta obvi que Python ha aconseguit superar en popularitat a llenguatges que en un principi li feien ombra com serien C# o Java, els quals han anat perdut popularitat amb el pas dels anys, de fet actualment és possible utilitzar mètodes o variables escrites en C# dintre d’un programa escrit en Python si s’utilitza un mòdul concret. Un altre llenguatge que al igual que Python ha experimentat un creixement de popularitat, encara que no tan radical, ha estat JavaScript, un llenguatge interpretat que s’especialitza en la creació de pàgines webs i que s’utilitza ens molts projectes juntament amb Python.

## Com escriure codi de Python

### Entorn de desenvolupament (IDE)

Un IDE o “Integrated Development Environment” és un programa que s’especialitza en oferir un entorn on sigui possible escriure un codi i inclou vàries opcions i eines que permeten convertir-ho en una tasca molt més senzilla, com omplir automàticament algunes funcions, compilació automàtica, senyalitzar les diferents parts de la sintaxi amb colors i permetre al usuari executar el seu codi a la vegada que l’IDE marca els errors en cas que es produeixin. Un editor de codi corrent no te totes aquestes funcions, per això és molt recomanable utilitzar un IDE, ja que agilitzarà molt el procés i ofereix un entorn visual molt més senzill de comprendre.

L’IDE que s’ha utilitzat en el desenvolupament d’aquest codi ha estat Visual Studio Code, un IDE gratuït publicat per la empresa Microsoft. Existeixen varies raons que ens han fet creure que Visual Studio Code era la millor opció, lamés destacada és que permet afegir extensions, aquestes extensions tenen una gran varietat de funcions i permeten canviar l’aparença del editor de codi, integrar diferents llenguatges de programació o fins hi tot utilitzar una Intel·ligència Artificial per a que revisi el teu codi. Altres raons han estat que presenta el millor editor de codi que es pot trobar actualment i que presenta compatibilitat amb el programari Git, fet que ens permet realitzar un control de versions.

### Control de Versions

El control de versions ens permet guardar el nostre codi en diferents estats coneguts com a “versions”, d’aquesta manera quan es realitzen modificacions en un codi i es produeix algun tipus d’error es pot retornar a una versió anterior funcional sense preocupar-se de perdre el progrés ja realitzat. Es tracta d’una funció extremadament útil, sobretot en aquells projectes de dimensions més grans, ja que permet aplicar canvis i experimentar amb el codi sense posar en perill la integritat del projecte. Tot i així, també es recomanable utilitzar-lo en projectes més petits com seria el nostre cas, ja que és més segur i dona un caràcter més professional al desenvolupament del codi. (*Python Crash Course, 3rd Edition: A Hands-On, Project-Based Introduction to Programming , Eric Matthes, No Starch Press (2023)*)

Per a realitzar aquest control de versions s’ha utilitzat un programari conegut com a “Git”, el qual s’utilitza com un repositori local on s’emmagatzemen i organitzen totes aquestes còpies conegudes com a “Versions”. A més a més, Visual Studio Code ofereix és compatible amb “Git”, permetent crear i gestionar versions des del propi entorn de desenvolupament. També és possible anar més enllà utilitzant un repositori remot ubicat a la xarxa, això ens permet accedir al nostre codi des d’altres dispositius, una funció molt útil si es volen realitzar projectes en equip o si es perd el dispositiu original on es trobava el codi. El repositori remot que hem utilitzat a estat “GitHub”, ja que és aquell més popular entre els programadors i emmagatzema milions de codis online, d’aquesta manera hem aconseguit que el codi sigui accessible i executable des de qualsevol altre dispositiu.

Diagrama

Descripción generada automáticamente

Figura 2.2, diagrama que mostra el recorregut que una còpia experimenta fins arribar al repositori remot i les instruccions utilitzades per a fer-ho. Font: (https://support.nesi.org.nz/hc/en-gb/articles/360001508515-Git-Reference-Sheet)

## Què puc fer amb Python?

### Mòduls i Llibreries

Un mòdul és un fitxer de Python que té una sèrie d’instruccions i variables definides, Python ens permet importar aquest fitxer conegut com a mòdul a un segon fitxer, un cop haguem importat el fitxer podrem utilitzar totes les instruccions i valors que es troben definits al mòdul. Els mòduls són extremadament útils, ja que a través de la seva importació és possible utilitzar funcions molt complexes sense necessitat de definir-les, és per aquesta raó que resulten tan populars, permeten realitzar tasques molt específiques amb relativa facilitat.

Els mòduls relacionats entre si es poden agrupar en un tipus de fitxer conegut com a llibreria, les llibreries contenen varis mòduls per oferir el màxim de possibilitats i funcions que sigui possible al usuari. El seu funcionament és similar al dels mòduls, només cal importar el fitxer i tindrem accés a tot allò definit en els mòduls que inclou. La comunitat de Python ha estat creant un gran catàleg de llibreries durant els anys i és precisament gràcies a això que Python s’ha convertit en un llenguatge versàtil amb aplicacions en camps diversos.

### Usos

Les llibreries de Python ofereixen un gran nombre d’aplicacions i possibilitats als usuaris, però aquelles que més destaquen són les següents:

**Programació de Sistemes:** Python és un llenguatge ideal per a programar eines relacionades amb sistemes operatius ja que permet explorar directoris i fitxers, executar altres programes i realitzar varis processos al mateix temps. Existeixen llibreries com “Stackless” que ens permeten interactuar i modificar el nostre sistema operatiu amb molta més facilitat, convertint Python en un llenguatge idoni per tasques relacionades amb sistemes.

**Creació d’interfícies gràfiques (GUIs):** Python és un llenguatge orientat a objectes pel que també és una bona opció per construir interfícies gràfiques, a més a més, la versió bàsica de Python inclou una llibreria anomenada “Tkinter”, la qual permet crear interfícies que interactuen amb l’usuari i que poden ser executades en altres dispositius. En el cas de les interfícies de les pàgines webs es sol utilitzar la llibreria “Jython”, la qual combina el llenguatge JavaScript, molt utilitzat en el desenvolupament de webs, amb Python.

**Bases de dades:** Python és un dels llenguatges més adaptats al tractament de bases de dades ja que disposa d’una gran varietat de llibreries que simplifiquen aquesta tasca com “MySQL” o “PostgreSQL”, les quals permeten crear bases de dades i relacionar-les amb objectes creats en Python. També trobem llibreries com “PyMongo”, les quals ens permeten emmagatzemar aquestes bases de dades en el núvol.

**Programació numèrica o científica:** En el camp de les matemàtiques Python també ha anat guanyat rellevància durant aquests últims anys, fins a convertir-se en un dels llenguatges més utilitzats en les simulacions numèriques. La llibreria que més destaca dintre d’aquest sector és “NumPy” , la qual permet realitzar operacions matemàtiques d’alta complexitat a una gran velocitat, aquesta velocitat ha permès que Python sobrepassi a llenguatges compilats com C++. A través de “NumPy” és possible arribar a crear projectes com animacions o un entorn 3D on realitzar simulacions. Algunes llibreries utilitzen a “NumPy” com la seva base, com és el cas de “SciPy”, una llibreria utilitzada en el processament de dades experimentals.

**Intel·ligències artificials:** Amb l’auge que el sector de les intel·ligències artificials ha experimentat aquets últims anys Python s’ha vist obligat a adaptar-se a la programació d’aquestes, és per això que trobem llibreries com “PyBrain”, les quals ens permeten crear xarxes neuronals que poden acabar convertint-se en intel·ligències artificials relativament simples.

(Learning Python – Mark Lutz, O’Reilly, 2013)

# Programació de Videojocs

## Introducció

En l’actualitat la creació de videojocs es troba a l’abast de qualsevol persona, fins hi tot si no té experiència en el camp de la programació, gràcies a la gran quantitat d’eines disponibles que faciliten aquesta tasca. Aquestes eines es divideixen en dos grans subgrups, els “Game Frameworks” són llibreries o mòduls que permeten crear videojocs utilitzant purament codi escrit en llenguatges com Python o Java, mentre que els “Game Engines” o motors gràfics són entorns especialitzats que permeten crear videojocs de manera senzilla i solen permetre evitar la programació en gran part.

El desenvolupament d’un videojoc es troba constituït per molts passos diferents, la planificació, la creació dels gràfics i els sons, el disseny de nivells, la creació del software que el controla i molts altres que en la seva totalitat suposen un gran esforç. És per aquesta raó que molts estudis professionals opten per utilitzar motors gràfics que permeten estalviar temps que es pot dedicar a aquestes tasques externes a la programació, mentre que els “Game Frameworks” solen quedar reduïts a la creació de videojocs “indies”, els quals són jocs creats de manera independent per una sola persona, aquests jocs solen tenir unes dimensions més reduïdes i busquen destacar gràcies a algun tret o mecànica innovadora, és per això que utilitzen “Game Frameworks” que permeten programar el videojoc amb més profunditat.

## Videojocs amb Python (Game Frameworks)

Existeixen una gran varietat de “Game Frameworks” que funcionen amb diferents llenguatges de programació, la majoria orientat a objectes, però en aquest apartat es destacaran aquells que funcionen amb Python ja que presenten una major correlació amb el projecte realitzat. Encara que Python no sigui el llenguatge més utilitzat en els “Game Frameworks” aquest continua presentant diferents llibreries i mòduls que poden ser utilitzats en la creació de vídeojocs, a continuació trobem aquells més populars

**Pygame:** Pygame és la llibreria més popular i utilitzada per al desenvolupament de videojocs, la qual permet crear entorns interactius en dues dimensions, ja siguin aplicacions o videojocs. Es considera la llibreria més actualitzada i adaptada per aquesta tasca, a més a més, existeixen varis projectes oberts i tutorials oficials que permeten familiaritzar-se amb el seu funcionament de manera gratuïta. El seu rang de funcions pot resultar una mica escàs per desenvolupar un videojoc molt complex, però existeixen mòduls compatibles amb la llibreria que permeten simplificar la creació d’animacions, col·lisions i el disseny de nivells.

**Pyglet:** Pyglet és una llibreria multi-plataforma que ofereix un entorn per desenvolupar programes orientats a objectes com podrien ser videojocs o altres aplicacions visuals. Pyglet destaca en la creació de finestres emergents, ja que permet crear-ne vàries, controlar-les i fins hi tot treballar amb varis monitors a la vegada, també permet utilitzar arxius en una gran varietat d’informació. Per altra part, utilitzar Pyglet resulta més complicat i consumeix més temps que programar amb altres llibreries com Pygame, és per això que existeixen llibreries escrites en Pyglet com “Arcade” o “cocos2d” que simplifiquen la programació i la tornen més accessible.

**PyOpenGL:** Aquesta llibreria combina un programari conegut com a “OpenGL”, que s’utilitza en el disseny i creació de modelats en dues i tres dimensions, amb Python. Destaca perquè ofereix una gran varietat de mòduls i extensions que milloren el disseny d’aquests models. La majoria de videojocs creats amb aquesta llibreria utilitzen Pygame simultàniament, ja que PyOpenGL es concentra més en la creació de gràfics i no en el funcionament del videojoc com a tal.

**Pymunk:** Aquesta llibreria es troba construïda a partir d’un altra llibreria anomenada “Chipmunk”, ofereix la possibilitat de crear entorns en dos dimensions on es simulen físiques interactives, una funció molt útil si es volen crear videojocs relacionats amb moviments realistes com són els jocs de plataformes o carreres.

Existeixen més llibreries o mòduls que aporten funcions específiques capaces de simplificar el procés de creació del videojoc, però la majoria d’ells són poc populars i no suposen un gran canvi a l’hora de programar.

### Pygame

Pygame va ser creat l’any 2000 per Pete Shinners, un programador que fins aquell moment s’especialitzava en el llenguatge C, el qual va decidir intentar combinar Python i SDL (“Simple DirectMedia Layer”), una llibreria escrita en C que permet controlar els elements multimèdia que es troben en pantalla i que s’ha utilitzat en múltiples jocs comercials. El resultat d’aquesta combinació va ser una llibreria que permetia crear videojocs de manera relativament simple utilitzant únicament Python, la qual es va anomenar Pygame.

Es tracta d’un llenguatge útil però té un nombre de funcions reduïdes relacionades amb el funcionament del videojoc, per sort Pygame permet crear codis que gestionen cada un dels aspectes del joc així que si es necessita una funció concreta, aquesta la pot escriure un mateix. Molts usuaris de Pygame comparteixen aquestes funcions creades per a que així altres usuaris puguin obtenir els resultats desitjats sense tenir que crear la funció, tot i que si s’utilitzen funciones extretes d’internet sense comprendre el seu funcionament el més possible és que ens provoqui molts problemes quan avancem en el desenvolupament.

(Pete Shinners, 2017, *PygameIntro,* <https://www.pygame.org/docs/tut/PygameIntro.html>)

Texto, Carta

Descripción generada automáticamente

Figura 3.1, mostra de codi simple escrita utilitzant la llibreria Pygame, si s’executa apareix en pantalla l’animació d’una pilota rebotant. Font: (<https://www.pygame.org/docs/tut/PygameIntro.html>)

En la imatge anterior (vegeu figura 3.1) es troba una mostra de codi escrit mitjançant Pygame, aquest codi figura en la pàgina oficial de Pygame com a exemple d’un programa simple, ja que mostra les funcions més bàsiques de la llibreria. El primer que destaca en aquest codi o qualsevol altre escrit en Pygame es que es troba dividit en dos parts, la part escrita a partir de la instrucció “while True:” es troba dintre d’un bucle que executa constantment les instruccions del seu interior, aquestes instruccions solen ser aquelles que comproven si l’usuari interactua amb alguna tecla o element del joc ja que s’han d’executar constantment. La part anterior al bucle només s’executa un cop al inici, en aquesta part es solen definir les variables, funcions i classes que més endavant haurem d’utilitzar en el bucle.

La imatge anterior també exemplifica força bé com funciona l’ús de les llibreries com Pygame en els programes. Es poden observar varies instruccions que comencen amb la paraula “pygame”, el que es fa mitjançant això es cridar la llibreria Pygame ja importada en la primera línia del codi. Seguidament s’especifica quin dels mòduls de la llibreria es vol utilitzar, en la imatge podem observar mòduls com “Image” que s’utilitza per carregar i guardar imatges o el mòdul “Display” que s’utilitza per crear i modificar la superfície on es col·loquen les imatges. En últim lloc s’escriu quina de les funcions que inclou el mòdul es vol utilitzar, la funció “Load” que apareix en la imatge és una de les més senzilles, ja que només carrega la imatge que es proporcioni entre els parèntesis.

Resumint, les instruccions que utilitzen llibreries segueixen la següent estructura:

Llibreria.Mòdul.Funció()

Exemple: pygame.image.load()

## 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)