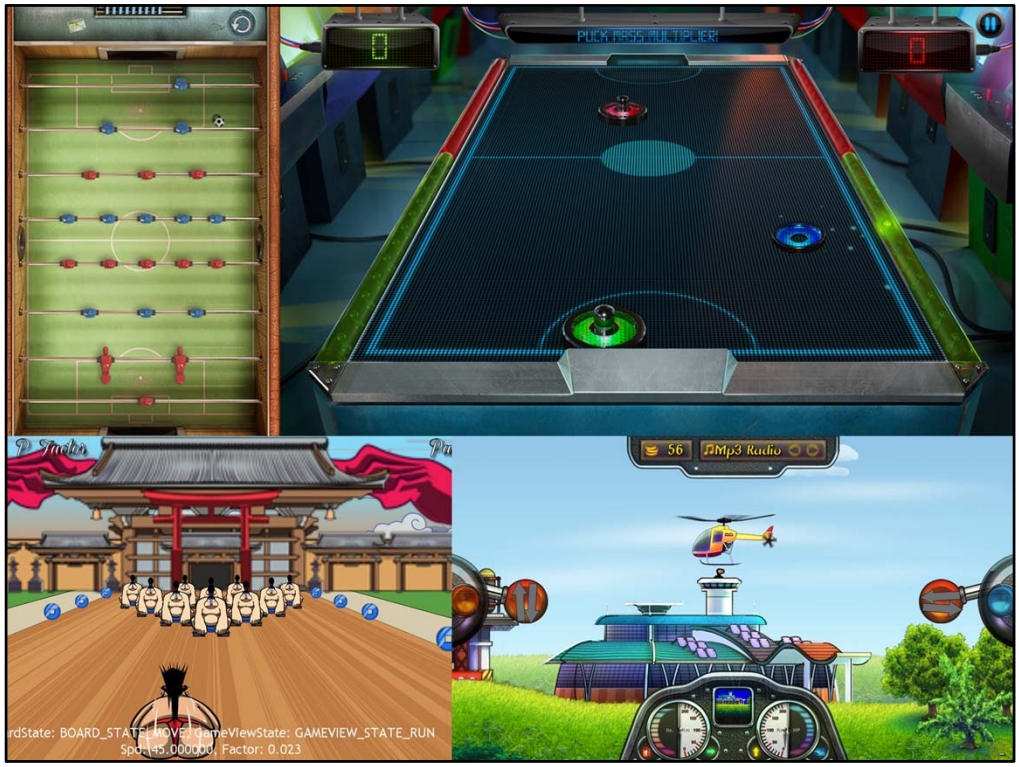
¡Bienvenidos al PyCon Venezuela! Esta charla es un compendio de cosas acerca de desarrollo de videojuegos para Python que he aprendido en mi labor como desarrollador de software y como docente. Python es parte de mi caja de herramientas al momento de enfrentarme al desarrollo de una solución tecnológica para alguna empresa o institución, y me complace compartir mi conocimiento con la audiencia de este evento.

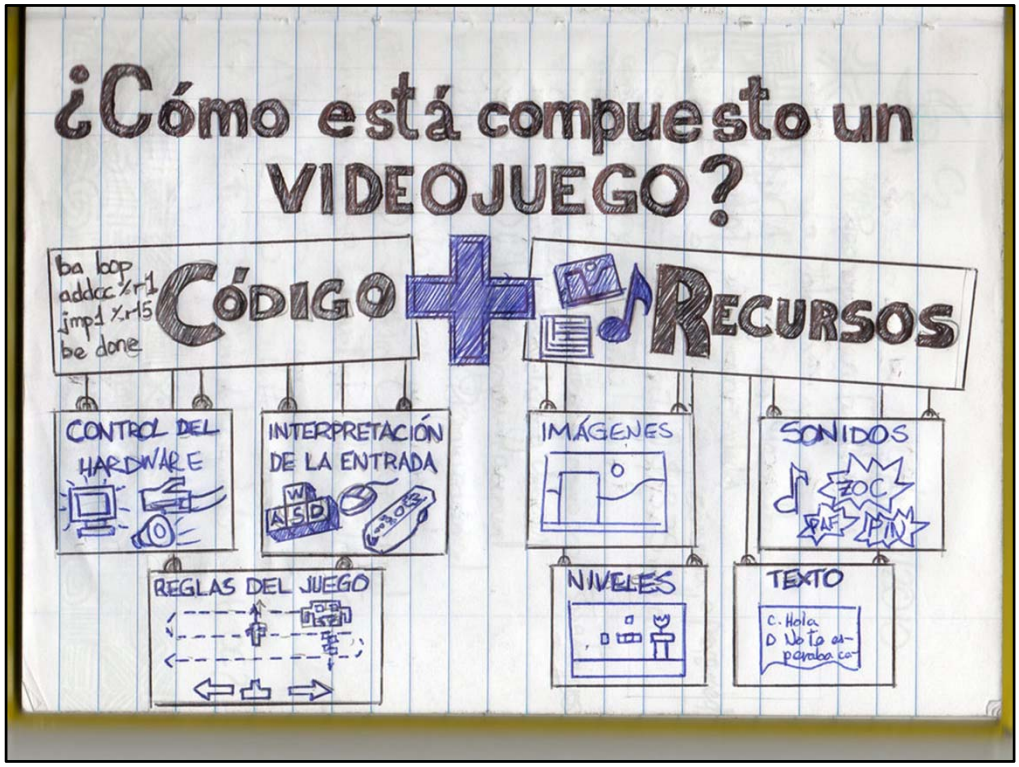
El tema de hoy es sobre Frameworks de desarrollo de videojuegos en Python, mi nombre es Ciro Durán, y mi especialidad son los videojuegos, específicamente su programación y diseño.

Llevo más de 10 años desarrollando videojuegos. Desde hace un año llevo adelante Mecaludens, el nombre bajo el cual estamos desarrollando videojuegos para dispositivos móviles. Desde 2006 escribo en El Chigüire Literario, un blog en español de programación de videojuegos. En el pasado fui co-fundador de Open English y miembro del equipo inicial de desarrollo de su plataforma tecnológica (2006-2011). He publicado varios juegos gratuitamente en mi página, http://www.ciroduran.com



Mis últimos proyectos han estado orientados al desarrollo de videojuegos para plataformas móviles, lo que me ha alejado un poco de Python temporalmente. La mayoría de mis juegos de 2012 los pueden buscar en el BlackBerry App World para dispositivos BlackBerry 10.

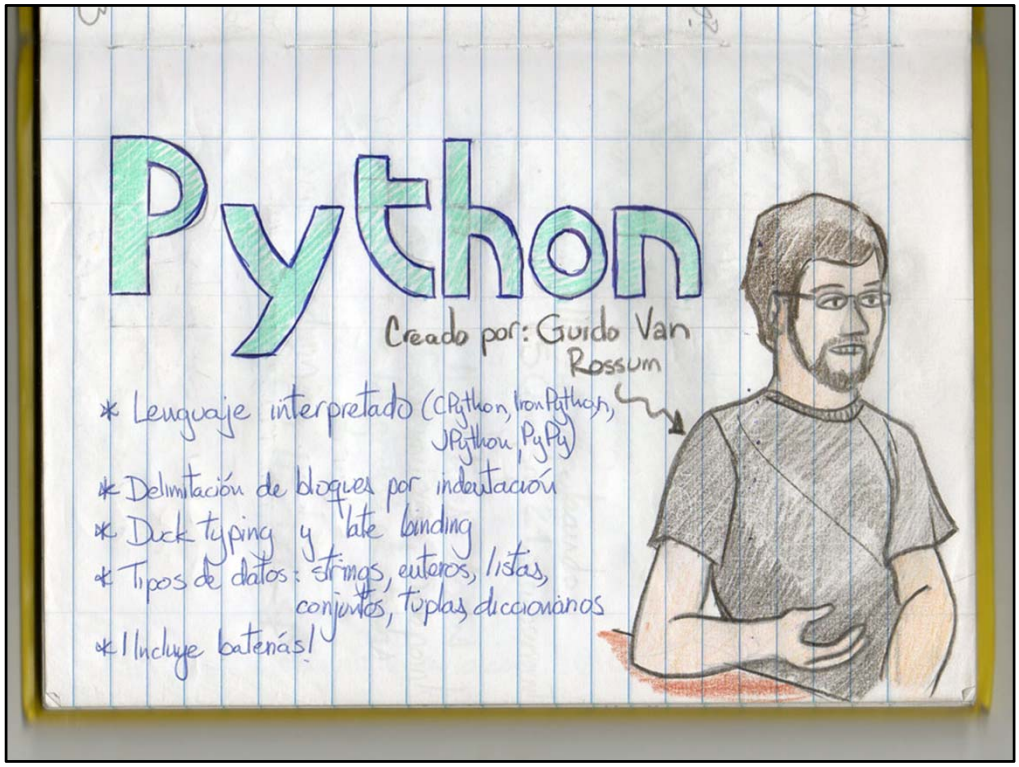
Desde 2009 organizo el Caracas Game Jam, un evento de 48 horas en el que multitud de estudiantes, profesionales y entusiastas se han anotado para hacer videojuegos. Todos los participantes son geniales. Revisen sus juegos en http://www.caracasgamejam.com



Para poder conocer dónde cabe Python en un proyecto de videojuegos, tenemos que saber cómo está compuesto un videojuego. Un videojuego tiene dos compontes, uno de código y otro de recursos.

El código es lo que llamamos “el software”, lo que controla el hardware para el juego (despliegue gráfico, comunicación con redes, sonido, etc.), lee los dispositivos de entrada (teclado, mouse, control de juegos, etc.), e implementa las reglas del juego (qué es el personaje controlado por el jugador, cómo se mueve, cuál es la condición para ganar, etc.).

Los recursos por otro lado son lo que emplea el código para presentar al jugador una narrativa, a través de imágenes, sonidos, texto, modelos 3D, y niveles, que es un nombre genérico para los archivos que integran todos estos elementos para presentárselos al jugador.



Python es un lenguaje de programación interpretado, creado por Guido Van Rossum. La característica que lo resalta de casi todos los lenguajes es la delimitación de bloques de código por la indentación, en vez de encerrar los bloques entre llaves (“{ }”) o pares de palabras (“begin … end”).

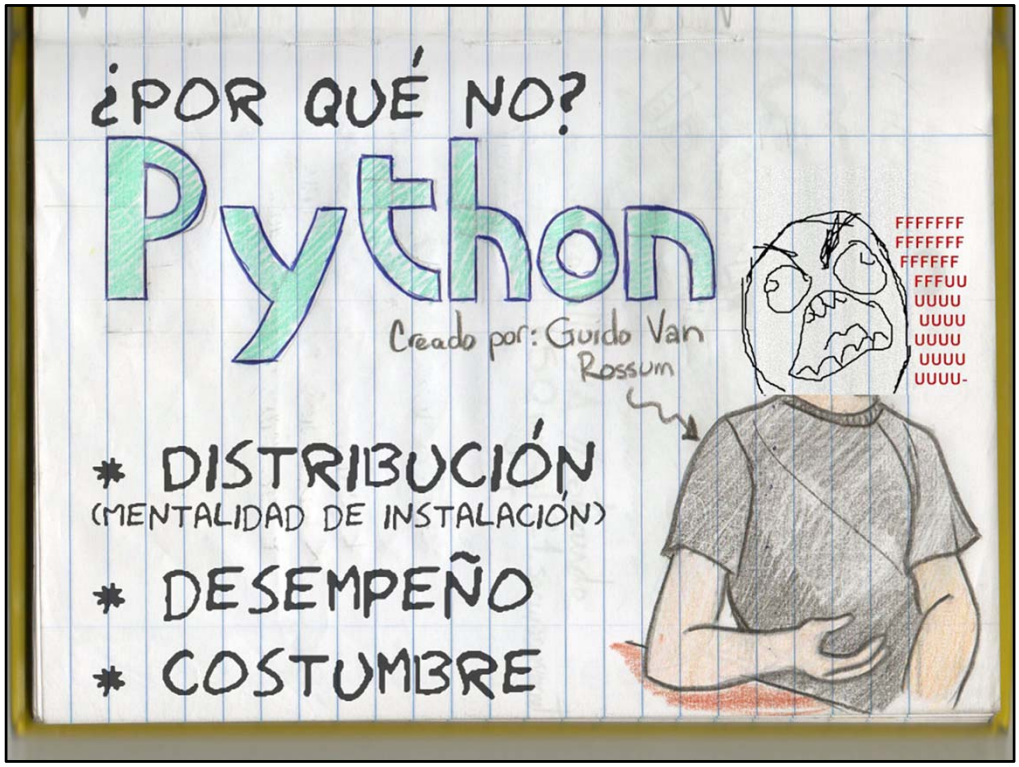
Al ser un lenguaje interpretado necesita un programa intérprete (valga la redundancia) que lea los scripts que escribimos y los convierta a instrucciones que la computadora entiende. CPython es la implementación más extendida actualmente, el intérprete está escrito en C. Hay otros intérpretes como JPython (Java), IronPython (.net) y PyPy (el intérprete está escrito en Python)

El lenguaje es multi-paradigma, orientado por igual al lado funcional, de objetos y procedimental. En un script puedes escribir una función y no tiene que estar dentro de un objeto (a diferencia de Java). Por otro lado, los objetos están tipificados dinámicamente al estilo duck typing, por lo que una instrucción valida sobre un objeto no depende de la herencia de otro objeto, sino de que los métodos o variables a los que se acceden dentro del objeto deben existir.

Los tipos de dato que maneja son strings, enteros, listas, conjuntos, tuplas, diccionarios, entre otros. Algunos de estos tipos de datos corresponden a un lenguaje de más alto nivel que, por ejemplo, C.

Por último, una frase popular de Python es que “¡incluye baterías!”, lo que significa que

Python incluye librerías de uso común que aceleran todo tipo de desarrollo.

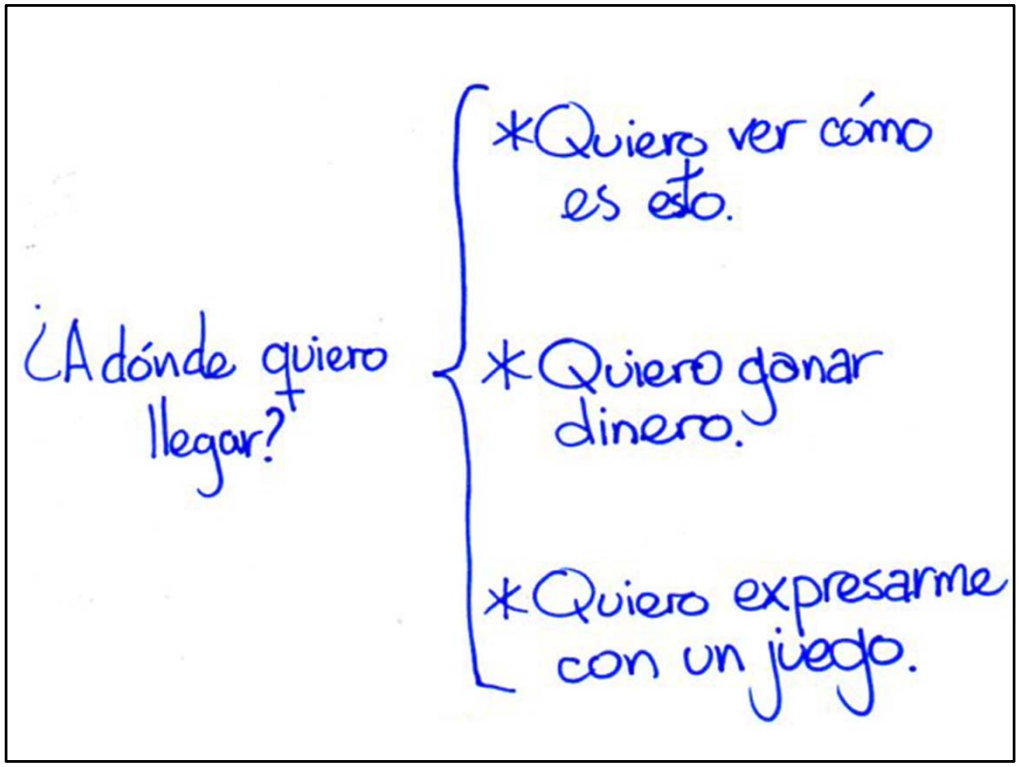


Ahora, ¿cuáles son los problemas que presenta Python en la actualidad?

Hay un problema en la distribución. Python, al heredar del pasado UNIX, posee una mentalidad de instalación, lo que significa que en muchos casos al recibir los scripts hace falta instalar todas las librerías que el código emplea. Esto puede causar conflictos de versiones, y dificultades al momento de instalar la aplicación. Cuando se distribuye en Linux se tiene la ventaja de que estas librerías se pueden especificar como dependencias del juego, por lo que los sistemas de paquetes pueden encargarse automáticamente de satisfacer las dependencias. En Windows y en Mac tenemos un poco de desventaja.

Otro problema puede ser el desempeño. Al ser un lenguaje interpretado, dependiendo del estado en el que esté el intérprete, puedes tener un software más rápido o más lento.

El último problema es el de costumbre. La mayoría de las personas que han aprendido un solo lenguaje tienden a emplear el mismo lenguaje para todas las oportunidades. Si este es tu caso, te recomiendo que aprendas otros lenguajes para ampliar tu repertorio de herramientas.



Entonces quieres desarrollar un videojuego. La primera pregunta que deberías hacerte antes de comenzar a poner la primera línea de código es “¿a dónde quiero llegar”? Con esto me refiero a cuáles son tus intenciones con hacer un videojuego.

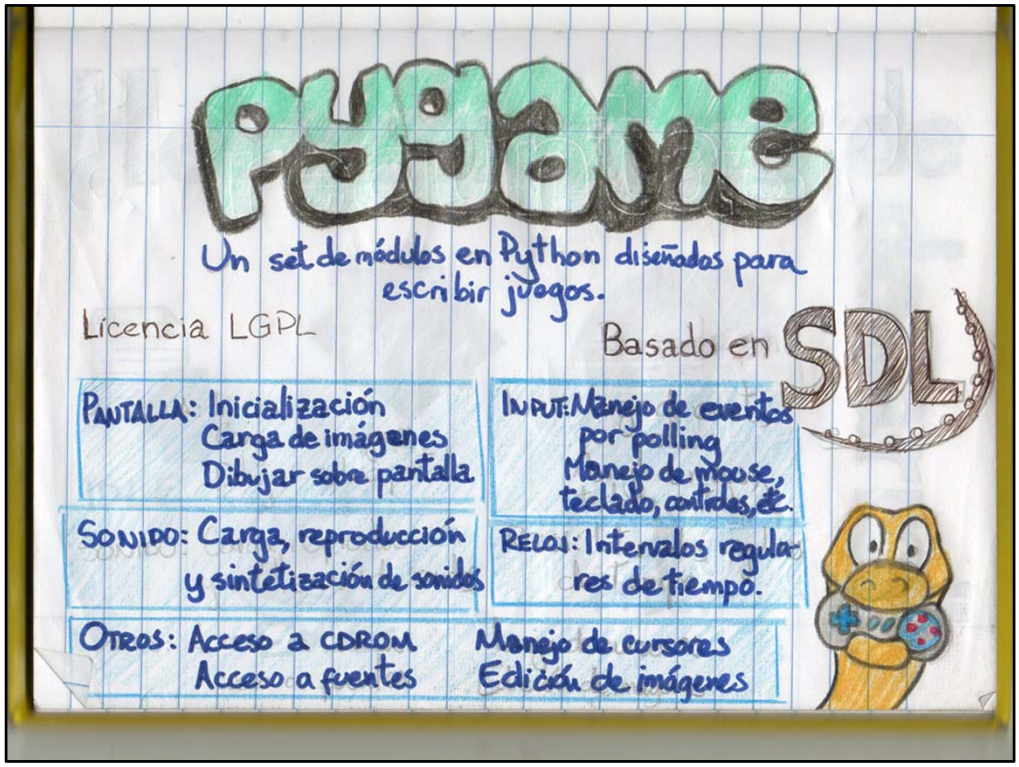
Puede ser que simplemente tengas curiosidad de como se hace un videojuego. Los videojuegos son piezas complejas a pesar de la emoción que producen, y la curiosidad es un excelente punto de partida para ver todos (o la mayoría de) los detalles que tiene un videojuego, sutiles pero importantes, que en conjunto nos hacen sentir de la manera como nos hacen sentir.

Puede que quieras ganar dinero con los videojuegos. Es una meta aspiracional muy común vivir de hacer videojuegos, pero se requiere mucha preparación y mucha práctica. Como en un buen espectáculo, existe una gran cantidad de detalles que pasan desapercibidos en la consciencia del público, pero que pesan en cómo perciben el show. Los videojuegos no son una carrera de hits, aunque lo parezca. Es una carrera maratónica, de resistencia.

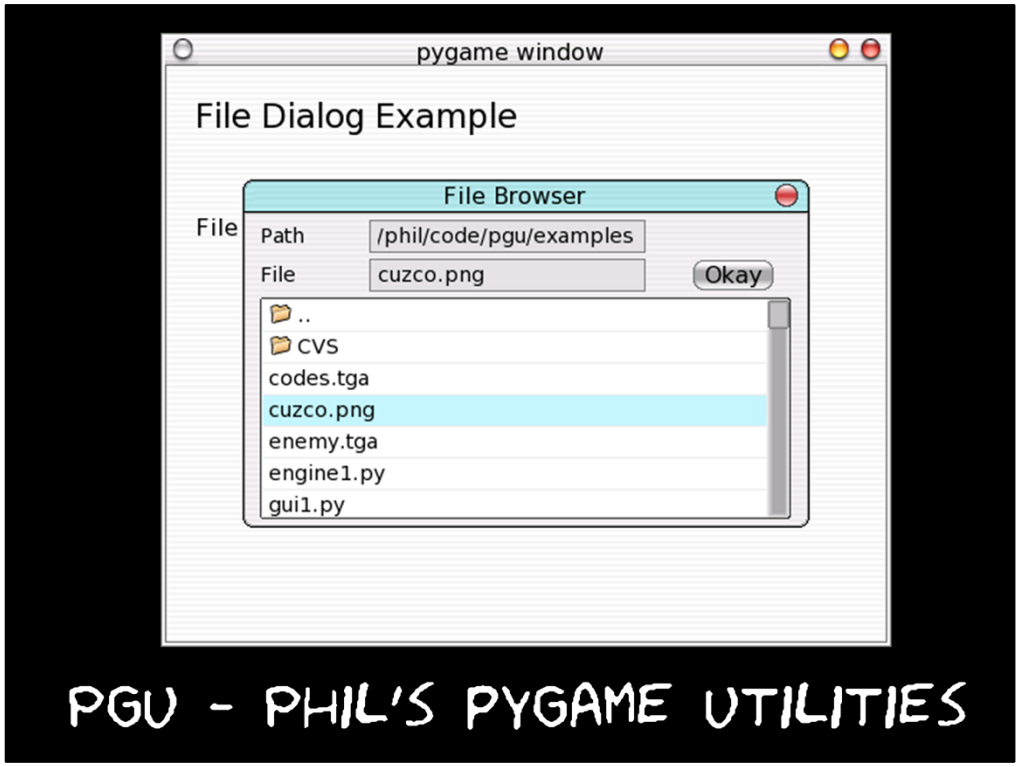
Finalmente, puede ser que quieras hacer videojuegos con el fin de expresar algo que sientes. En esas filas están desarrolladores como Peter Molyneux, Auntie Pixelante, Jonathan Blow, y otros. Los videojuegos son un instrumento para comunicar sentimientos de una manera que la literatura o el cine no pueden captar por su diseño. Es un campo todavía virgen.

Esta pregunta es importante, porque muchas veces Python es ignorado en el desarrollo de

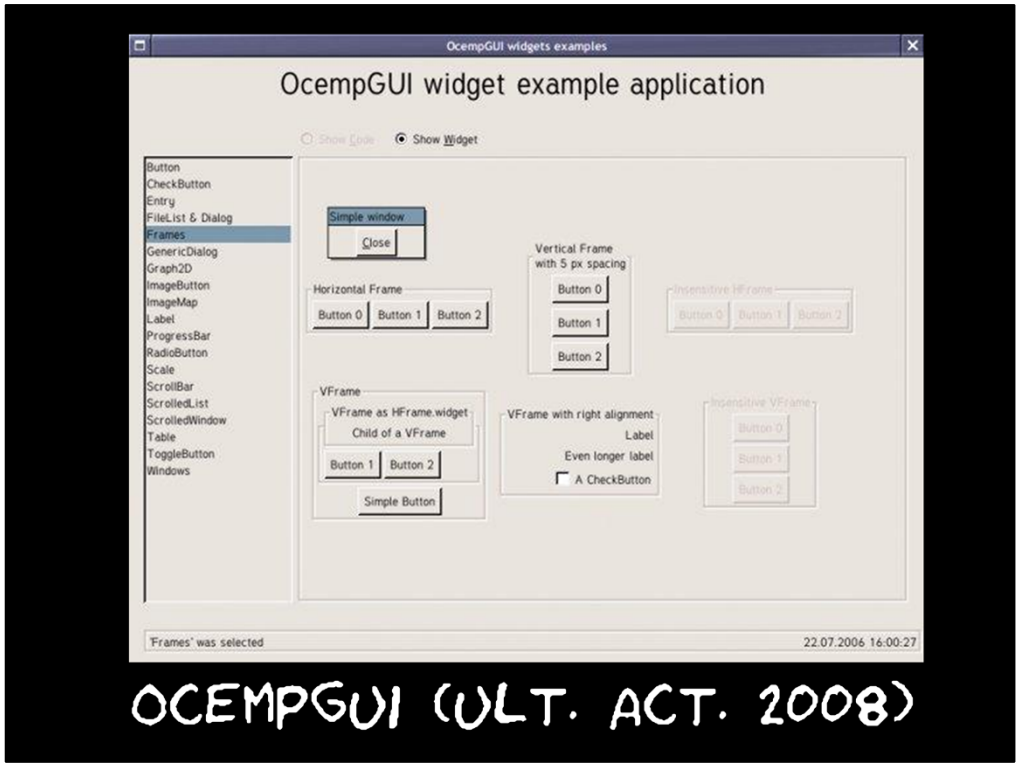
videojuegos, en favor de plataformas más comerciales. El desarrollo en Python ofrece un camino con un buen aprendizaje de muchos temas, sin tener que preocuparse en extremo por el funcionamiento de la computadora a bajo nivel. Python es un buen punto de partida para tu viaje.



Pygame es una de las librerías más conocidas en el desarrollo de videojugeos de 2 dimensiones. Es un binding de Python de la superconocida librería de desarrollo SDL, basada en C. Pygame te ofrece todas las rutinas de bajo nivel que necesites para controlar lo que ves en pantalla, reproducción de sonido, lectura de dispositivos de entrada y demás. Lo único es que en las propias estructuras que necesitas para desarollar las reglas de un videojuego se queda un poco corto, por lo que hay que trabajar bastante para obtener algunas cosas que damos por sentado que son sencillas (por ejemplol, “pasar” de una pantalla a otra).

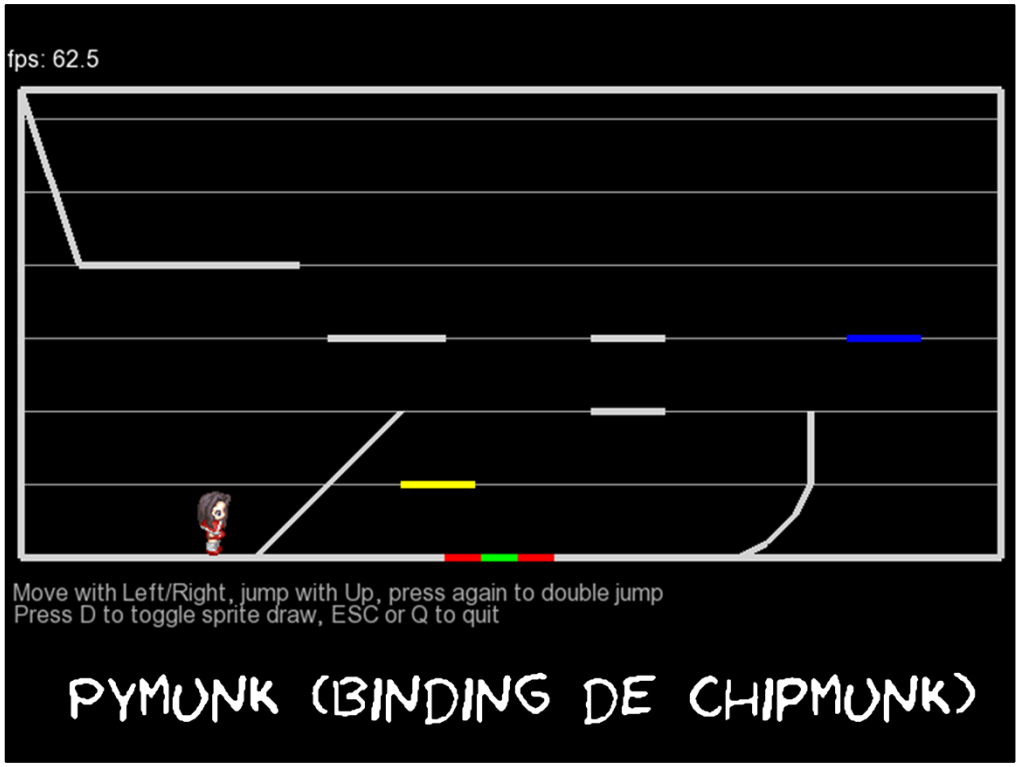


Una de esas cosas que damos por sentado son las librerías de interfaces gráficas. Pygame como tal no tiene una, asi que existen al menos dos que han sido contribuidas por usuarios de Pygame. La primera de ellas es PGU, Phil's Game Utilities, que he utilizado con bastante grado de éxito. Provee una cantidad de componentes gráficos como botones, ventanas y selectores de archivos. Como está basado en Pygame, lo puedes utilizar en conjunto con tus rutinas de dibujo.

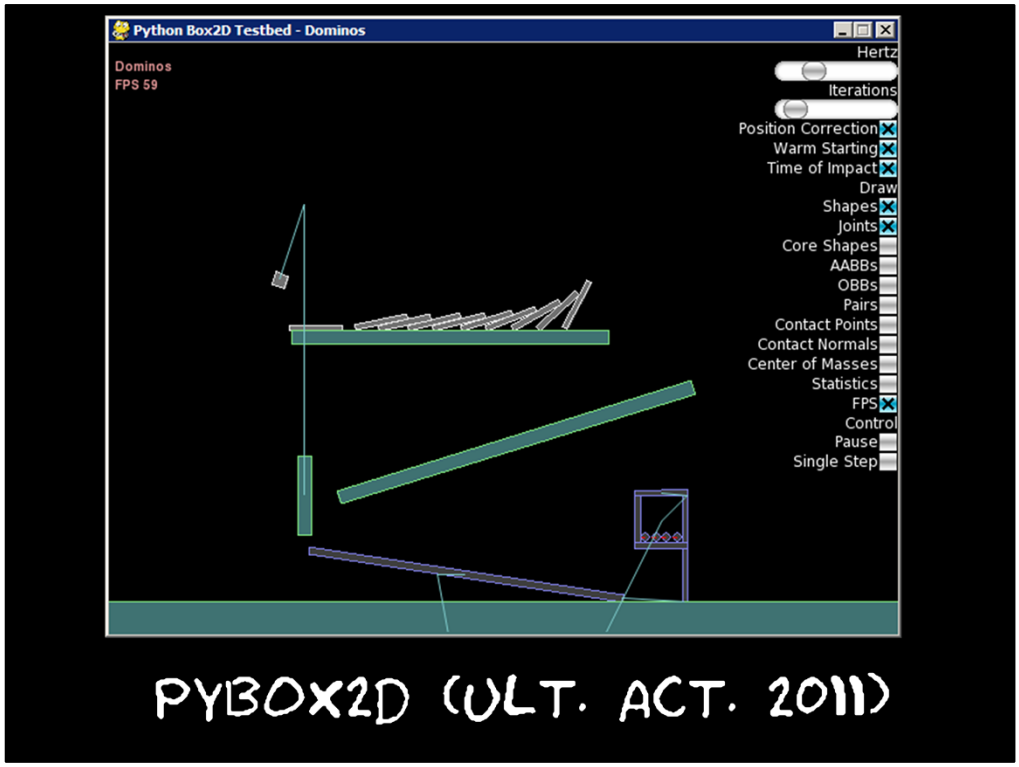


Por otro lado, esta OcempGUI, que es producto de un videojuego que nunca salió, pero cuya librería de interfaces se lanzó y es empleada por algunos proyectos. Lamentablemente, esta librería se actualizó por última vez en 2008, y no ha visto más progreso.

Es de notar que es importante hacer la selección de estas librerías temprano en el desarrollo de tu proyecto, ya que estas librerias no son sencillas de integrar en proyectos avanzados.



Otro componente cuya importancia se ha relucido con los dispositivos móviles es el simulador de cuerpos rígidos, o simulador de física. Estos componentes traen una serie de definiciones que uno emplea para crear cuerpos rígidos, insertarlos en un mundo 2D, y ponerlos a interactuar para obtener juegos cuyo comportamiento se asemeje a los del mundo real. Pymunk es una librería muy popular. Es un binding de Chipmunk (escrito en C), por lo que tiene la facilidad de utilizarse en Python con el poder de desempeño de C.



Otra librería muy popular es Box2D, creada por Eric Catto, y la librería que impulsa Angry Birds. Lamentablemente, el binding de Python se actualizó por última vez en 2011.

Una alternativa a escribir tu juego desde cero es utilizar un motor de videojuegos. Estos motores ofrecen editores para simplificar la producción del juego, con el costo de sacrificar la flexibilidad, y tener que adaptar el juego al estilo que ofrece el motor.

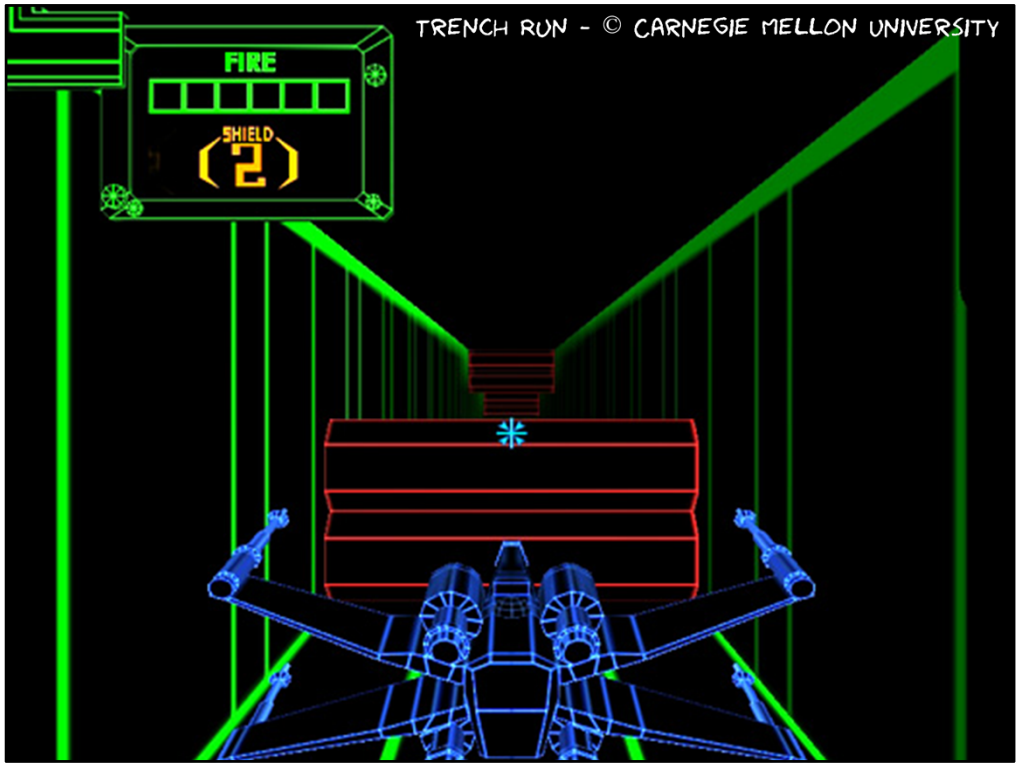
Un motor muy popular de videojuegos en Python es Ren'Py. Este motor facilita la producción de videojuegos al estilo dating sim japonés.

Un ejemplo de juego hecho con este motor es Tony T, un videojuego hecho por Christopher Ortiz, y a ser lanzado a finales de noviembre de 2012. Aquí vemos una captura de pantalla del juego.

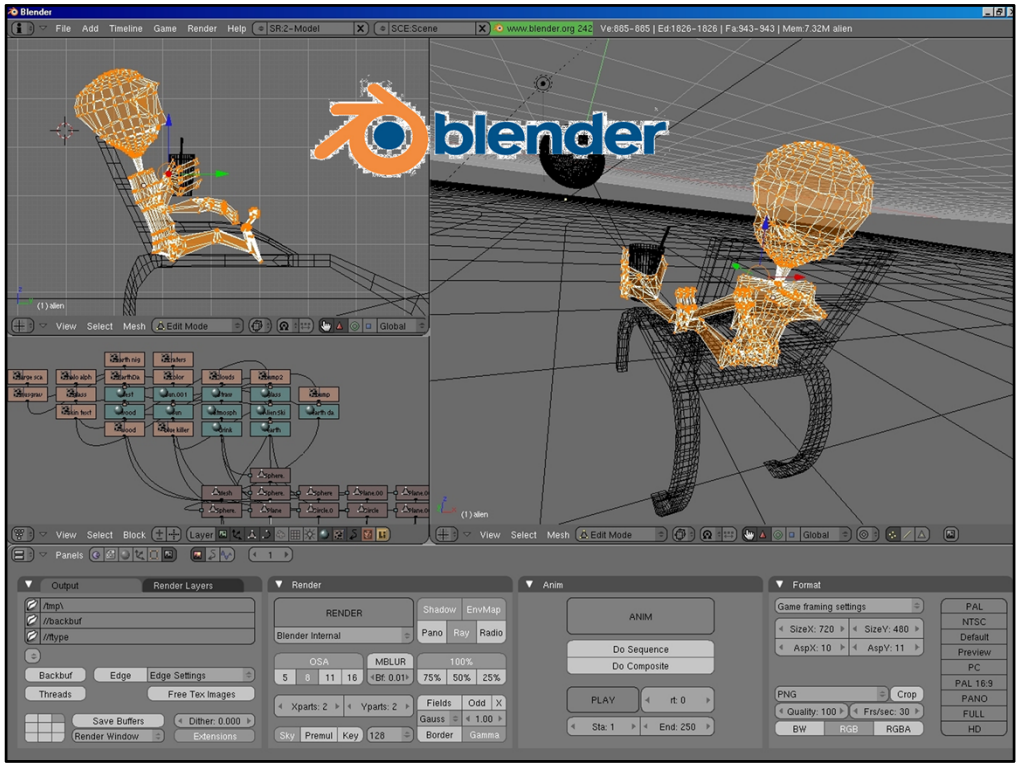
Si lo que deseas es desarrollar juegos 3D, hay soluciones más completas que Pygame. Panda3D es una librería muy madura que permite hacer esta clase de juegos. Es una librería surgida de los laboratorios de realidad virtual de Disney, y empleada por esta misma empresa para la producción de videojuegos. Actualmente su difusión está a cargo de la institución educativa Entertainment Technology Center, parte de la Universidad Carnegie-Mellon.\

Panda3D ofrece rutinas para carga de modelos, establecimiento de puntos de iluminación, y hasta detección de colisiones entre modelos.

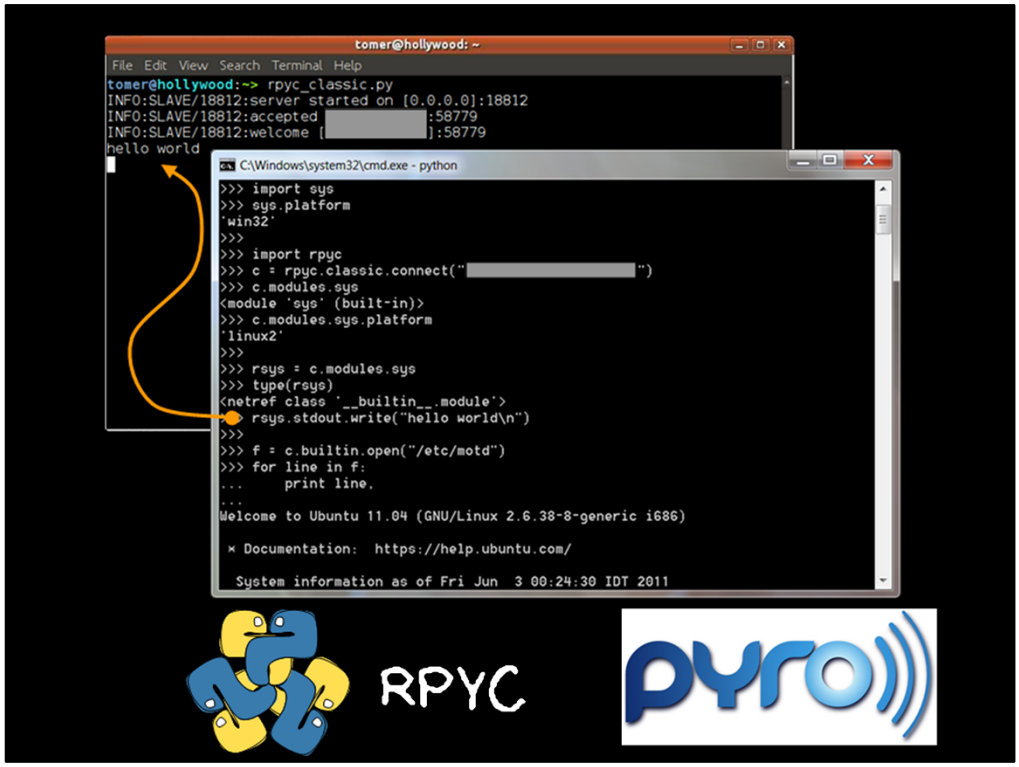
Un ejemplo de juego desarrollado con Panda3D es Pirates of the Caribbean, por Disney Interactive Studios



Otro ejemplo es este juego, Trech Run, hecho como parte de la materia Building Virtual Worlds, en la Universidad Carnegie-Mellon.

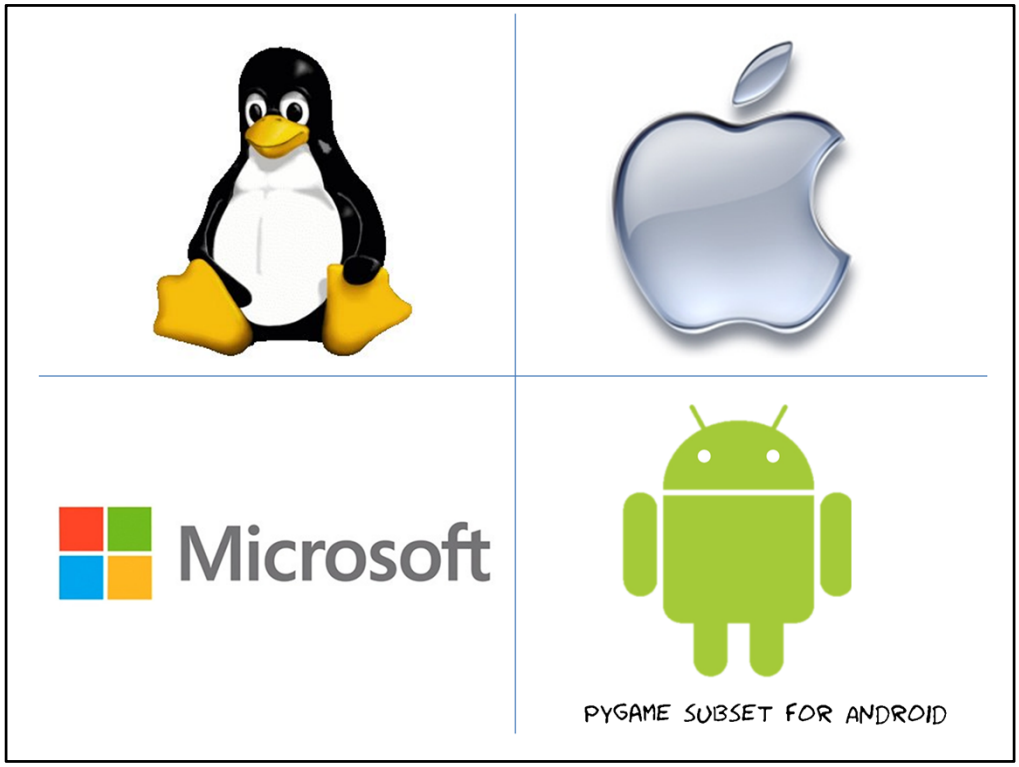


Si lo que deseas es hacer modelos 3D de objetos con Python, Blender es una excelente solución, a la par de los modeladores 3D comerciales.



Para la comunicación entre computadoras a través de las redes, presentamos dos alternativas. Una de ellas se llama RPyC, y su objetivo es ser un RPC (Remote Procedure Call) en Python. Esto te permite desarrollar tus llamadas en código Python, y que la librería se encargue de los detalles de bajo nivel, como convertir los parámetros del lenguaje a datos que se pueden transmitir a través de la red y reinterpretarlos.

Pyro es también otra alternativa. La desconozco, pero colegas la han utilizado con buenos resultados.

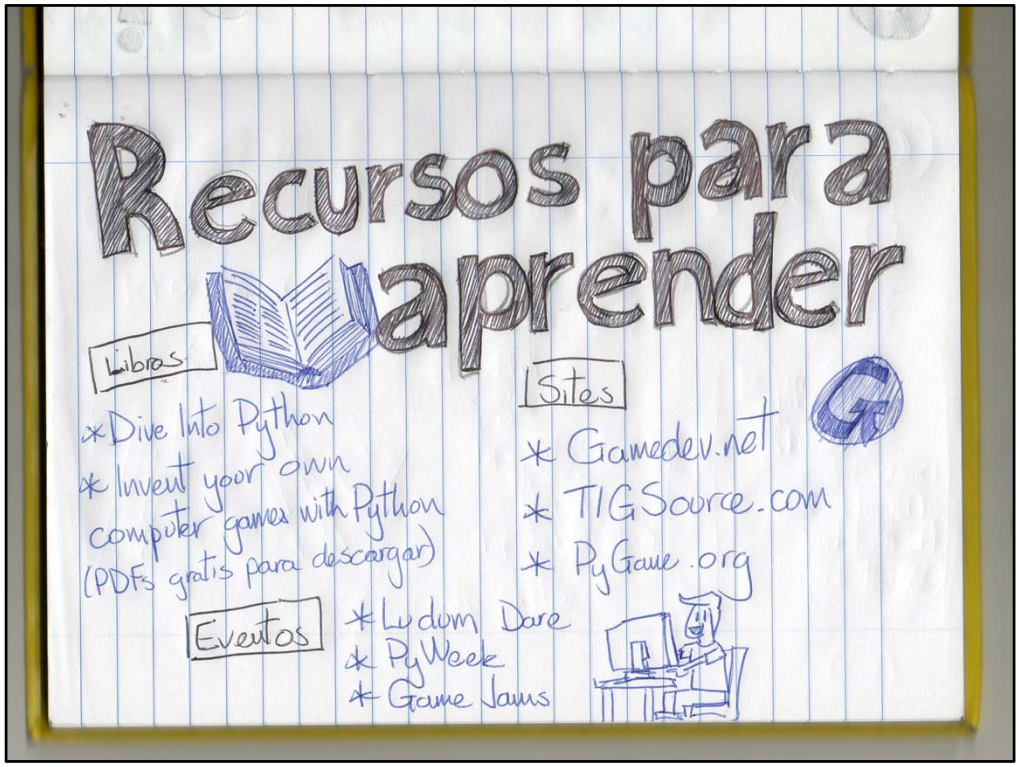


¿Asi que cómo puedes distribuir tu juego? Esta también es una pregunta importante que te debes hacer antes de escribir la primera línea de código, pues determina parte importante de tu flujo de trabajo. Con Python puedes distribuir con seguridad a:

- Linux – Linux es el hogar de Python y el que posee la manera más sencilla de instalar las librerías que necesites.
- Apple – Mac OSX tiene una versión de Python instalada, aunque ha presentado algunos problemas para algunos desarrolladores.
- Windows – Windows no viene con Python instalado, lo que dificulta convencer a alguien de jugar tu juego teniendo que instalar previamente un intérprete. Así que existen empaquetadores que meten el intérprete y tu juego en un solo compilado y asi poder facilitar la distribución.

Sin tanta seguridad, pero existe definitivamente la posibilidad, es a la plataforma Android. El desarrollador de Ren'Py ha hecho un Pygame Subset for Android, que es una librería de rutinas para usar Pygame dentro de Android. Hasta ahora desconozco la manera de poder empaquetar un juego hecho en Python a una aplicación que puedas distribuir mediante el Play Store, pero supongo que alguien está trabajando en ello.

Otro software que vas a necesitar son los paquetes gráficos (aquí muestro GIMP y GraphicsGale, uno software libre y el otro comercial, respectivamente). Paquetes de oficina para poder comunicarte con otras personas (una habilidad invaluable para trabajar en equipo). Y control de versiones. Ningún programador que se precie de ser profesional escribe código sin usar este software para mantener estados en el tiempo del codigo que ha escrito hasta el momento. Existen páginas que almacenan codigo y se acceden a través de control de versiones, como Bitbucket y Github.

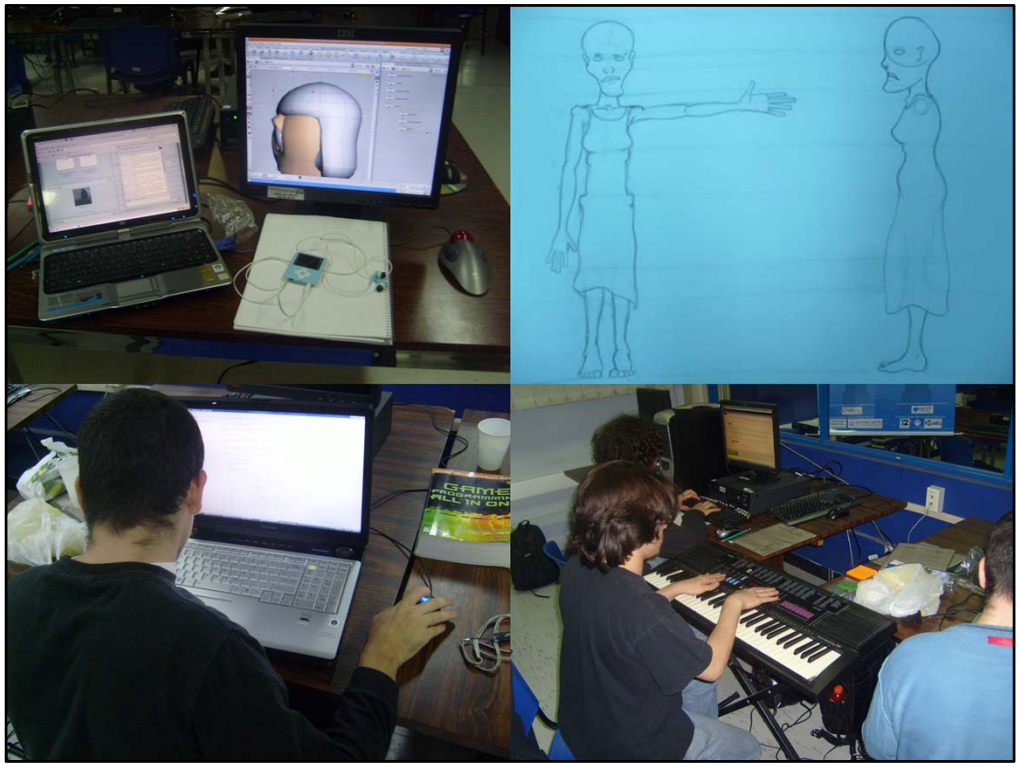


Otros recursos para aprender están en Internet.

En libros tenemos Dive into Python (disponible también en español), Invent your own computer games with Python, y Making Games with Python & Pygame, ambos del mismo autor, y dirigidos específicamente al tema de esta charla. Todos son PDF gratuitos descargables.

Sitios interesantes: Gamedev.net (aquí aprendí yo), TIGSource.com (comunidad de gente my entusiasmada en hacer jugeos), y Pygame.org. La propia página de Pygame posee un listado de juegos que puedes descargar par ajugar y estudiar su código.

Eventos importantes donde puedes participar para divertirte y aprender a hacer juegos: Ludum Dare (en internet), Pyweek (alrededor de abril-mayo, en Internet), y los game jams (¡el Caracas Game Jam 2013 se hará en Macaracuay, en enero!).



Un Game Jam no es solo para computistas. Hacer un juego requiere de múltiples habilidades y una sola persona no puede tenerlas todas. Se requiere ilustrar, se requiere hacer texturas, se requiere modelar, se requiere hacer música y sonidos. Son muchas cosas aparte de programar. Ten esto en cuenta para invitar a personas que no programan pero que sí hacen otras cosas importantes para los juegos.

Así se ve un game jam por dentro. Gente colaborando para hacer juegos.

Les agradezco su atención. Cualquier pregunta que quieran hacer la pueden enviar por los canales de comunicación señalados. ¡Gracias!