Scientific and Practical Cyber Security Journal (SPCSJ) 6(4): 34-39 ISSN 2587- 4667 Scientific Cyber Security Association (SCSA)

PYTHON USE IN GAMING

Nikoloz Tchitanava Student at the "Tbilisi's Archimede's School"
Luka Tchitanava, Student at the "Atitanti School"
Dachi Beridze, Student at the "Tbilisi N147 Public School"
Beqa Chxirodze, Student at the"N169 Public School"

ABSTRACT: In this article,we talk about Usage of Python programming language in gamedevelopment: advantages,disadvantages,engines working on python and comparison with the different programming languages used for game creation.
Our research shows that, python is the best choice for 2D games. It gives developers ability to create the game easily and quickly.
KEYWORDS: python, game, 2D games

Introduction:

We live in the most technologically advanced era. This helps different fields like game development.The video games industry is growing every single day. Because of corona virus, people spent lots of time at their homes. They had free time and no place to go, so majority of worlds population started spending more time on their gadgets. During this boom, not only the players, but also the people who wanted to learn game development doubled in numbers.In 2022, the number of gamers worldwide was estimated at three billion, down from the 3.2 billion global gamers during the height of COVID-19 in 2021. Despite this momentarily decline, global gaming audiences are projected to increase at a steady growth rate. Number of game developers raised by 37% in 2021,yet the games industry is experiencing a tremendous global skills shortage. Nor is it not getting easier to match the right talent with the right opportunity, especially given the additional challenges brought on by remote work and distributed teams. A change is desperately needed.So, for people who want to become game developers in the future, python based game engines might be the best first step.

First Video games

Scientist William Higginbotham is credited with developing the first video game in October 1958. It was a relatively simple tennis game that played similarly to the legendary video game Pong from the 1970s. He had a little analog computer that could show different curves on an oscilloscope, including the trajectory of a ball in motion. Higginbotham needed a few hours to develop the concept for a tennis game and a few days to put the fundamental components together. Higginbotham had no issue building the straightforward game display because he had experience designing displays for radar systems and other electrical equipment.

Drawings by Higginbotham were used to create plans. About two weeks were spent developing the device by technician Robert Dvorak. The first video game was ready for release after some debugging. Tennis for Two was the name of the game. Players could push a button to hit the ball in the direction of their opponent and spin a knob to change the ball's angle. Players couldn't miss the ball as long as they pressed the button in their court, but if they hit it at the wrong moment or tilted, the ball wouldn't cross the net. A ball would bounce like a tennis ball when it struck the ground. The players pressed a reset button to begin the following round once the ball left the court or went through the net. Tennis for Two needed more modern video games' sophisticated graphics. A side image of a tennis court with only two lines—one for the net and one for the ground—was displayed on the cathode ray tube display. The ball was merely a dot that moved back and forth. Players have to keep track of their scores as well. Many resistors, capacitors, and relays were used in the game's circuitry, but transistors were also used for the quick switching required when the ball was in play. Unfortunately, they could not develop the engines that would have allowed other developers to produce their games at the time. They were forced to start from scratch each time as a result.[1]

Game engine description

A game engine is a software architecture primarily made for video games. It typically comes with essential libraries and assistance applications. Game engines are tools developers can use to create games for computers and other gaming platforms. The primary features that a game engine often offers include rendering engines ("renderers") for 2D or 3D graphics, physics engines or collision detection (and response), sound, scripting, animation, artificial intelligence, networking, streaming, memory management, threading, localization support, scene graph, and video support for cinematics. Implementers of game engines frequently save money on the game production process by reusing/adapting, in large part, the same game engine to produce other games or to help with game porting to numerous platforms. For instance, a game for the Atari 2600 had to be created from the ground up to best use the display hardware. Today, makers of games for older systems refer to this basic display routine as the kernel. Although there was more room for maneuvering other systems,[2] memory limitations frequently prevented the data-heavy designs that an engine requires. Very little could be reused between games, not even on more forgiving platforms. Most of the code had to be thrown out later anyway because following generations of games would employ entirely different game designs that took advantage of more excellent resources due to the rapid advancement of arcade hardware, which was the leading edge of the market at the time. Because of this, most game designs from the 1980s used a fixed set of rules, a limited amount of levels, and graphics data. Since the heyday of arcade games, it has become customary for video game developers to create their game engines for use with first-party software. The fluid side-scrolling engine created by Shigeru Miyamoto's team at Nintendo for the Nintendo Entertainment System was a prominent example of an in-house game engine on home consoles in the middle of the 1980s (NES). Super Mario Bros later used the technology they had created for the side-scrolling racing game Excite bike, which was released in 1984. (1985). As a result, Mario could easily transition from a stroll to a run rather than moving at a fixed speed, like in earlier platform games. Even though third-party game engines were uncommon until the emergence of 3D computer graphics in the 1990s, several 2D game creation systems were created for independent video game development in the 1980s. Pinball Construction Set (1983), ASCII's War Game Construction Kit (1983Th), under Force Construction (1984), Adventure Construction Set (1984), Garry Kitchen's Game Maker (1985), Wargame Construction Set (1986), Shoot-'Em-Up Construction Kit (1987), Arcade Game Construction Kit (1988), and most notably ASCII's RPG Maker engines from 1998 onwards are some of them. Another classic title still accessible is Click & Play (1994).

Around the middle of the 1990s, the phrase "game engine" came into use, particularly about 3D games like first-person shooters that used a first-person shooter engine. Unreal Engine was first released in 1998 by Tim Sweeney's company, Epic Games. Since Doom and Quake by Id Program were so well-liked, other developers decided to license the core software instead of starting from scratch and creating their graphics, characters, weapons, and levels referred to as "game content" or "game assets." Teams could expand and specialize due to the separation of game-specific rules and data from fundamental ideas like collision detection and game entities. Later games, with the engine and content developed independently, used a similar strategy, including Epic Games' 1998 Unreal and id Software's Quake III Arena. A high-end commercial gaming engine license can cost anywhere between US$10,000 and millions. Then several appointments might exceed several dozen businesses, as seen with the Unreal Engine. Licensing technologies have proven to be a valuable supplemental revenue source for several game producers. At the very least, reusable engines expedite and simplify the process of creating game sequels, an essential benefit in the cutthroat video game market. Around 2000, there was intense competition between Epic and id, but since then, Unreal Engine from Epic has become far more well-liked than id Tech 4 and its successor, id Tech 5. One of the most complicated programs ever created, modern game engines frequently contain dozens of intricately tuned modules that interact to provide a tightly regulated user experience. Level design, rendering, scripting, and art have become distinct due to the ongoing development of gaming engines. For instance, an average game development team now typically has far more artists than actual programmers.[3][4][5]

Python based game engines

Game engines for Python most often take the form of Python libraries, which can be installed in a variety of ways. Most are available on PyPI and can be installed with pip. However, a few are available only on GitHub, GitLab, or other code sharing locations, and they may require other installation steps.

Pros and cons of python based game engines compared to others

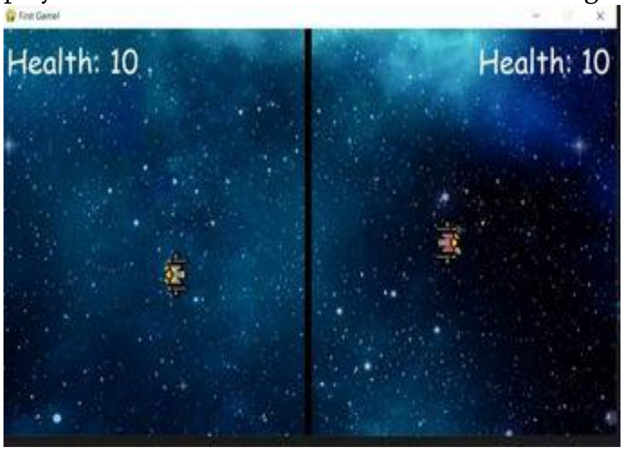
Python is a general purpose programming language, and it’s used for a variety of tasks other than writing computer games. In contrast, there are many different stand-alone game engines that are tailored specifically to writing games. Some of these include:
The Unreal Engine
Unity
Godot
These stand-alone game engines differ from Python game engines in several key aspects:
Language support: Languages like C++, C#, and JavaScript are popular for games written in stand-alone game engines, as the engines themselves are often written in these languages. Very few stand-alone engines support Python.Proprietary scripting support: In addition, many stand-alone game engines maintain and support their own scripting languages, which may not resemble Python. For example, Unity uses C# natively, while Unreal works best with C++.
Platform support: Many modern stand-alone game engines can produce games for a variety of platforms, including mobile and dedicated game systems, with very little effort. In contrast, porting a Python game across various platforms, especially mobile platforms, can be a major undertaking.
Licensing options: Games written using a stand-alone game engine may have different licensing options and restrictions, based on the engine used.
So why use Python to write games at all? In a word, Python. Using a stand-alone game engine often requires you to learn a new programming or scripting language. Python game engines leverage your existing knowledge of Python, reducing the learning curve and getting you moving forward quickly.There are many game engines available for the Python environment. The engines that you’ll learn about here all share the following criteria:They’re relatively popular engines, or they cover

aspects of gaming that aren't usually covered.They're currently maintained.They have good documentation available.[6][7]

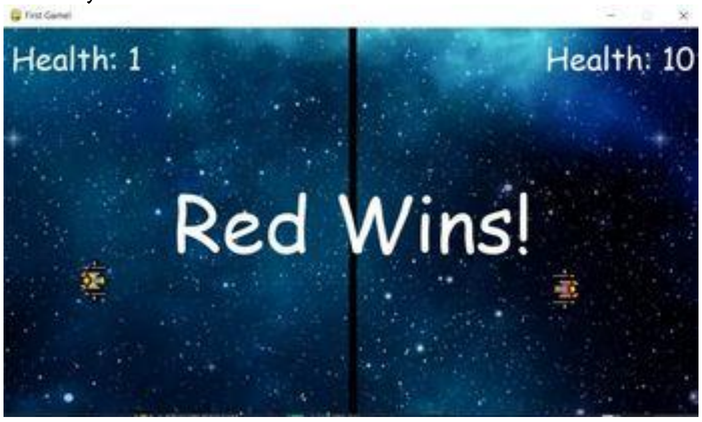
Our project
For this research, we created the game.Here is the summary of it.

Gamedescription:

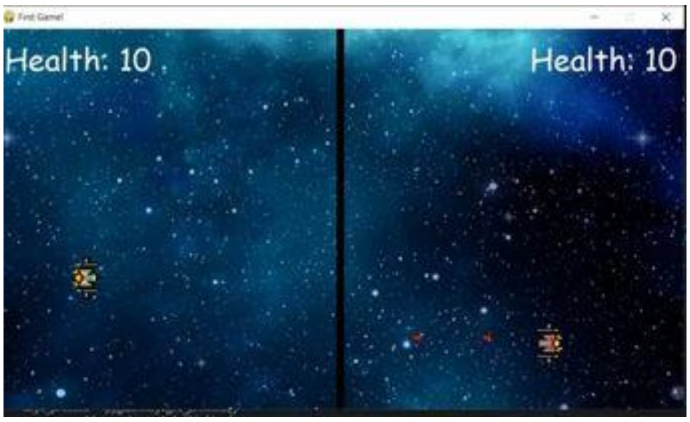
A game where two spaceships fight each other. The game has two players, one controlling a red spaceship and the other controlling a yellow spaceship. The players can move their spaceships around the screen using the WASD keys for the yellow player and the arrow keys for the red player. The players can also shoot bullets at each other using the control keys.



The game has a health bar for each player, and the player whose health reaches 0 first loses the game. The game also has a border that divides the screen in half and that the spaceships cannot pass through. The game displays the current health of each player on the screen, as well as the spaceships and any bullets that have been fired.



The game has a fixed frame rate of 60 frames per second and the spaceships and bullets move at a fixed velocity. The game has a maximum of 3 bullets that can be on the screen at any given time for each player. Also, it has an event system that can be triggered by the players getting hit by bullets, which causes the spaceship to flash and become invulnerable for a short period of time.



Conclusions:

main problem was downloading the pygame itself. Sometimes the operating system or code writing environment may give you hard time using the library. Code writing is simple, the syntaxes are easy to remember and it does not need much time to run. It takes around 5 hours to create the mini game, that you can enjoy later with your friends. It will also help you to understand the main principles of game creation, which you can also use in different engines. Unfortunately, python based game engines are suitable just for the 2D games such as platformers*. It is not for creating big projects like AAA* and AA* games. But, if you are new to game development(like us), python based game development is great, because it is much more easy than other engines such as "Unreal engine" or "Unity".

Footnotes:

AAA games-The term "AAA Games" is a classification used within the video gaming industry to signify high-budget, high-profile games that are typically produced and distributed by large, well-known publishers. These games often rank as "blockbusters" due to their extreme popularity.

AA games-"AA" (although rarely used) means budgets around just several million.

Platformers-A platform game, commonly referred to as a "platformer," is a style of video game where the player makes a character move through an environment with a series of action-based moves, like running, jumping, or swinging from ropes.

Acknowledgement: This work was supported by Shota Rustaveli National Science Foundation of Georgia [SPG-22-218]

REFERENCES

1. Alan Chodos, This Month in Physics History, APS News, 2008 (Volume 17, Number 9)
2. What is a Gaming of Game Engine, ARM Glossary https://www.arm.com/glossary/gaming-engines
3. What is a Game Engine?,Fullscale IO https://fullscale.io/blog/what-is-game-engine/

4. The Complete Game Engine Overview, Perforce https://www.perforce.com/resources/vcs/game-engine-overview
5. PyGame: A Primer on Game Programming in Python https://realpython.com/pygame-a-primer/
6. Jon Fincher, Top Python Game Engines https://realpython.com/top-python-game-engines/