International Journal of Research in Engineering, Science and Management
Volume-2, Issue-3, March-2019
www.ijresm.com | ISSN (Online): 2581-5792

Asteroid Game using Pygame

Nidhi[1], Ruchita Srivastava[2], Manvendra Singh[3], Jayant Mishra[4]
[1,2]*Student, Department of Computer Science and Engineering, SRMGPC, Lucknow, India*
[3,4]*Assistant Professor, Department of Computer Science and Engineering, SRMGPC, Lucknow, India*

***Abstract*: This paper is a critical survey of an arcade game using Pygame. It explains the concept of how Python and Pygame can be used to a simple game. It then describes the various modules that are used in the Pygame. The paper also explains the role of various Pygame modules used in the construction of a basic and simple arcade space-shooter game. It lists the benefits of using the Pygame and Python in the development of the game. The paper also points the possible drawbacks of the game and ends with the future implications of it.**

***Keywords*: Python, Pygame, Arcade Space Shooter Game, Pygame Module, SDL Library.**

1. Introduction

The original author of Pygame is Lenard Lindstrom, Rene Duffield, Pete Shinners, Nicholas Dudfield, and Thomas Kluyver [1]. It was initially released on 28 October 2000, 17 years ago. Originally Pygame was written by Pete Shinners in order to replace PySDL after its development stopped. Pygame has been a community project since the year 2000 and it is released under the open source free software GNU Lesser General Public License. It is a free and open source Python programming language library for making multimedia applications like games built on top of the excellent SDL library. Simple Direct Media Layer is a cross- platform software development library designed to provide a hardware abstraction layer for computer multimedia hardware components [3]. Software developers use it to write high-performance computer games and other multimedia applications that can run on many operating systems such as Android, iOS, Linux, Mac OS X and Windows. SDL manages video, audio, input devices, threads, shared object loading, networking, and timers. For 3d graphics it can handle an OpenGL or Direct3d context [4]. Like Simple Direct Media Layer the Pygame is highly portable and can run on nearly every platform and operating system [1]. Also the SDL library allows various types of computer game development without using the machine level mechanics of the C programming language and its derivatives. Applications that use Pygame can easily run on Android phones and tablets with the help and use of Pygame subset for Android [3].

So the paper discusses developing or creating an Asteroid game using Pygame. Asteroids is a space shooter game where the player controls a spaceship in an asteroid field. The objective of the game is basically to shoot and destroy the asteroids while not colliding with them. The level of difficulty increases and the game becomes harder as the number of asteroids increases.

2. Basic modules of pygame

Cdrom: The Cdrom module manages the CD and DVD drives on a computer. The module is required to be initialized first before it can do anything. Each CD object that is made represents a Cdrom drive and it should be initialized separately before it can perform most things [3].

Some of the pygame modules for audio Cdrom control are as follows,

pygame.cdrom.init- initialize the Cdrom module.
pygame.cdrom.quit- uninitialize the Cdrom module.
pygame.cdrom.get._init- true if the Cdrom module is initialized.
pygame.cdrom.get_count- number of Cd drives on the system.
pygame.cdrom.CD- class to manage a Cdrom drive.

Cursors: It loads cursor images, includes standard cursors. Pygame offers control over the system hardware cursor. It only supports black and white cursors for the system. The cursor can be controlled with functions inside pygame.mouse. The cursors has function to load and decode cursor formats. These allow you to easily store the cursors in external files or directly as encoded python strings.

Display: It controls the display window or screen. It creates a visible image surface on the monitor. The surface can either cover the full screen, or be shown on platforms that support a window manager. The display surface is nothing but actually a standard surface object of pygame.

Draw: Draw simple shapes onto a surface. Many functions take a width as argument to represent the stroke's size around the shape's edge.

Some of the pygame module for drawing shapes are as follows,

pygame.draw.rect
pygame.draw.polygon
pygame.draw.ellipse
pygame.draw.arc

Event: It basically manages events and the events queue. Pygame handles all the messages of events through a queue of event. The routines in this module help to manage that event queue. The input queue is very much dependent on the display module of the pygame. The event queue won't work if the display has not been initialized and a video mode is not set [5].

Some of the pygame module for interacting with events and queues are as follows,
pygame.event.pump
pygame.event.get
pygame.event.wait
pygame.event.peek

Font: This module creates and renders true type fonts. The font module permits for creating into a new surface object the turtype fonts. This module is optional.

Some of the pygame module for loading and rendering fonts are as follows,
pygame.font.init
pygame.font.Font
pygame.font.quit
pygame.font.get_init

Image: This module saves and loads images. The image module comprises functions for loading and saving pictures and also for transferring surfaces to formats that are usable by other packages.

Some of the pygame module for image transfer are as follows,
pygame.image.load
pygame.image.save
pygame.image.fromstring
pygame.image.frombuffer

Joystick: This module manages joystick devices on a computer. The Joystick devices comprise of video-game-style gamepads, and trackballs. The module allows the use of multiple buttons. Computers may manage multiple joysticks at a time.

Some of the pygame module for interacting with joysticks, gamepads, and trackballs are as follows-
pygame.joystick.init
pygame.joystick.quit

Key: This module manages the keyboard. It contains functions for dealing with the keyboard. When the keyboard buttons are pressed and released then the event queue gets pygame.KEYDOWN and pygame.KEYUP events. Both the events have a key attribute that is an integer ID representing every key on the keyboard.

Some of the pygame module to work with the keyboard are as follows,
pygame.key.get_focused
pygame.key.get_pressed
pygame.key.name

Mouse: This module manages the mouse. The mouse functions are used to get the present or current state of the device. These functions can be used to modify cursor for the mouse.

Some of the pygame module to work with the mouse are as follows,
pygame.mouse.set_pos
pygame.mouse.get_focused
pygame.mouse.get_cursor

Sndarray: This module is used to manipulate sounds with numpy. This module is available only when the external Numpy package can be used by pygame.

Some of the pygame module for accessing sound sample data are as follows,
pygame.sndarray.array
pygame.sndarray.samples

Surfarray: This module manipulates images with numby. It functions to convert pixel data between pygame surfaces and arrays. This module is functional only when the external Numpy package can be used by pygame.

Some of the pygame module for accessing surface pixel data using array interfaces are as follows-
pygame.surfarray.array2d
pygame.surfarray.array3d

Time: This module is used to control the timings. Times in pygame are represented in milliseconds. The time resolution of most platforms is limited around 10 milliseconds.

Some of the pygame module for monitoring time are as follows,
pygame.time.get_ticks
pygame.time.wait
pygame.time.Clock

Transform: This module scale, rotate, and flip images. A surface transform is an operation of resizing or moving the pixels of the image. All these functions take a surface on which these can operate on and then a new surface with the results is returned.

Some of the pygame module to transform surfaces are as follows,
pygame.transform.flip
pygame.transform.scale

The Main Method:

The asteroid is a two-dimensional vector shooter. The playing field consists of the spaceship and the floating asteroids. The spaceship has to be controlled to avoid collisions and to blast obstacles that are the endless numbers of asteroids. So the game is very simple and comprises of the very basics of a space shooter game. The ship can move in the left and the right direction. At the start of the game the spaceship is in the middle of the screen. Heading towards it are around 4-6 large asteroids. It has to avoid collision with them as well as shoot to blast them at the same time. If the spaceship collides with a small asteroid its health declines by a certain amount but if it collides with a large asteroid, then it straightaway loses one life out of the total of the number of lives it has.

Thus after losing all of its lives, the game is over and the player has to restart. As the player progresses the number of asteroid increases as the level of difficulty increases.

Basic Game Loop:

- While game is alive
- Take input
- Shoot
- *Find collision:* Bounding volume is used to detect a

collision. These can be of any shape, such as a sphere etc. In a space-shooter game, where objects move is far away, the bounding volumes' shapes need not be precise [6].

- Put everything on the screen.

3. Conclusion

Nowadays the public's demands on the games are rising, and the cost to produce a modern game can stretch to several hundred million dollars with teams consisting of up to 500 people working on the same game [1]. One of the first popular game genres was the space-shooter genre, with games such as Space Invaders, or Asteroids [1]. The Asteroid Game takes inspiration from the gameplay from games such as Star Fox, a space-shooter game for the Super Nintendo Entertainment System console that was released in 1993, where the player follows a predefined path in the third person, shooting enemy ships in front of the player [1]. The game's concept is simple. Avoid and shoot the large asteroids. Even today Asteroids clones and ports are emerging up as java applets, or in present releases for the modern game systems [1].

4. Future Implications

Since years there have been too many home versions of asteroids. Ranging from official versions to shareware clones, the game has appeared on every video game console and computer in some form or another [2]. There are several features that can be added to the game. More content is something that would be the first thing to be added to the game since this is something that the game is lacking at the moment. For example, it would be better to have greater diversity in enemies with different behaviors and formations they attack in, as well as in their ship and weapon models. Some sort of boss or tough enemy, with a wider range of weapons and higher health than the normal enemies, would be nice to have implemented as well; difficult enemies would give the player a tougher fight and a sense of achievement when beating the stronger opponent [6]. A proper progression system, where the player receives better weapons or more health over time, can also be added. This could be implemented in a number of ways, either by collecting upgrades dropped by enemies or by receiving an upgrade when a certain number of enemies have been defeated by the player. Regardless of the specifics, this will give the player a feeling of progression and thus will inspire them to keep on playing.

Another thing that can be implemented is generated terrain. Not only would entering a planet's atmosphere and flying through the hills and valleys heighten the gameplay experience, but it would also add new good-looking environments as well as diversity to the standard space environment. Having the camera angle change during gameplay is also a feature that can be added. The idea is solid and would add a range of new possibilities in terms of gameplay.

References

[1] M. Ebner and A. Holzinger, "Successful implementation of user-centered game based learning in higher education, vol. 49, no. 3, pp. 873-890, 2007.
[2] http://www.classicgaming.cc/classics/asteroids/play-guide
[3] https://www.pygame.org
[4] L. Cohen, L. Manion and K. Morrison, "Research Methods in Education, Route ledege, Talyor & Francis group, 2007.
[5] https://pythonprogramming.net/pygame
[6] https://gupea.ub.gu.se/bitstream/2077/39605 /1/gupea_2077_39605_1.pdf