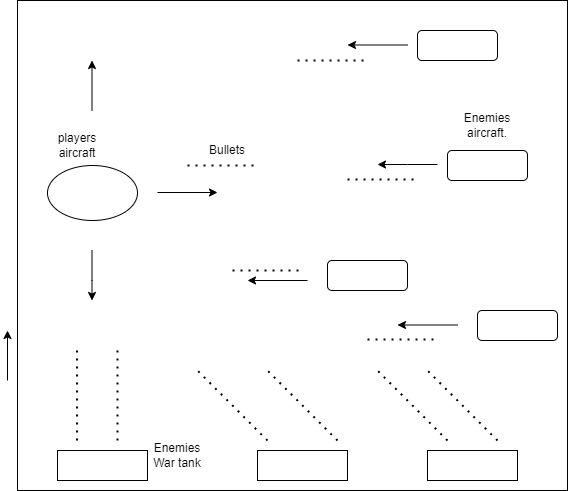
“Sky shooter” is a basic 2D arcade game where the player tries to avoid and shoot down the enemies' space ship , hover ships, and asteroids while getting as many points as possible.

The game consists of an aircraft which appears at the side of the gameplay screen. A series of enemies' airships come from the side of the screen and try to knockdown the player.



GAME STATES:

Intro-> gameplay->quit

* Intro
  + games title and about game
  + btnplay
  + btnquit
  + intro background
* Game play
  + Show all sprites
  + Timer and Hp score reset
* Quit
  + Game ends

SPRITES:

Players Spaceship:

* Player can move the spaceship right, left, forward, and backward using the corresponding keys.
* If the player shoots down an enemy's ship, add a point and reset the enemy's ship after a certain period.

Player’s bullet:

* If self exceeds boundary(screen), hide image.
* Self shoots with a speed of 50 frame per seconds
* Self is attached to the front position of the player spaceship, so it shoots out of the player’s ship.
* If it collides with enemya and the asteroids reset the object hit.

EnemyA Spaceship:

* The enemy will move towards the player trying to shoot the player down, it shoots at random intervals.
* Moves to the left (-x) with a random speed between 3 and 5 frames per second
* If enemy’s bullet collides with player subtract players HP by 5 points.
* If self position is greater than screen width reset image

EnemyB Spaceship:

* The enemy will move towards the player trying to shoot the player down.
* Moves to the left and right (-x) with a random speed between 1 and 2 frames per. second.
* If enemy’s bullet collides with player subtract players HP by 5 points.
* If self position is greater than screen width bounce to the opposite direction.

Enemy’s bullet:

* If self exceeds boundary(screen), hide image.
* Self shoots with a speed of 20 frame per seconds
* Self is attached to the front position of the enemies spaceship, so it shoots out of the enemy’s ship.
* If it collides with player’s ship reduce HP by -5 points.

Big Asteroid:

* When it hits player, reset self
* Moves to the left (-x) with a random speed between 1 and 4 frames per second
* Reset changes the position of self randomly
* A maximum of 2 of self can be on screen at once
* If self position is greater than screen width reset image

UI COMPONENTS:

Game Effects:

* Sound Effect
  + Play when player bullet collides with asteroid
  + Play when player bullet collides with enemya
  + Play when enemy’s bullet collides with player
  + Play when player space ship collides with asteroid

Asteroid sound - [bomb\_explosion\_8bit | OpenGameArt.org](https://opengameart.org/content/bombexplosion8bit)

Bullet sound - [Space Shoot Sounds | OpenGameArt.org](https://opengameart.org/content/space-shoot-sounds)

Intro background - [Space Backgrounds | OpenGameArt.org](https://opengameart.org/content/space-backgrounds-3)

The space ships, asteroids, game background and bullet - [Space Shooter art | OpenGameArt.org](https://opengameart.org/content/space-shooter-art)

ALGORITHM

Import pygame, simpleGE and random

Create a class named “Player”

Define \_\_init\_\_

Set players image

Define the size, move speed, and position.

Define process

If left arrow key is pressed add 5 to current image angle, and move by 5 frame per second.

If right arrow key is pressed add 5 to current image angle, and move by 5 frame per second.

If up arrow key is pressed and self > 0 on the y axis add move speed

If left arrow key is pressed and self < screen height add move speed

Create a class name “asteroid”

Define \_\_init\_\_

Set asteroid’s image.

Define the size, position, and move speed.

Set dx to a random number between min and max speed.

Define reset

Set random start position on the y axis when it resets.

Subtract dx from min and max random number

Define checkbounds

If it’s position is greater than screen width.

Reset image.

Create a class name “Enemya”

Define \_\_init\_\_

Set enemy’s image.

Define the size, position, and move speed.

Set dx to a random number between min and max speed.

Define process

Get random integer between 1 and 30

If random integer <= 5

Fire enemies bullet

Define reset

Set random start position on the y axis when it resets.

Subtract dx from min and max random number

Define checkbounds

If it’s position is greater than screen width.

Reset image.

Create a class name “Enemyb”

Define \_\_init\_\_

Set enemy’s image.

Define the size, position, and move speed.

Set dx to a random number between min and max speed.

Set image angle

Set bound action

Create a class name “bullet”

Define \_\_init\_\_

Set enemy’s image.

Define the size, and position.

Set image angle

Set bound action

Define process

if self collide with enemya

reset enemya

if self collides with asteroid

reset asteroid

Define fire

If self is not visible

Show bullet

Set position, move speed, image angle, and speed.

Define reset

Hide self

Create a class name “bullet1”

Define \_\_init\_\_

Set enemy’s image.

Define the size, and position.

Set bound action

Define fire

If self is not visible

Show bullet

Set position, move speed, image angle, and speed.

Define reset

Hide self

Create class for Time label

Define \_\_init\_\_

Set the Time text

Set Time label position

Create class for Durability label

Define \_\_init\_\_

Set the HP text

Set HP label position

Create class for Instruction label

Define \_\_init\_\_

Set the background

Create a multiple line label with instructions.

Set the location of this label

Create and give the location to the play button

Create and give the location to the quit button

Add all labels and buttons to the sprite list

Define process

If play button is clicked

Set the response to equal play and stop the current action

If quit button is clicked

Set the response to equal quit and stop the current action

Create a class named game

Define \_\_init\_\_

Set the background

Create the bullet sound

Create the asteroid sound

Create an instance of the player class to represent the player’s ship

Create instances of the asteroid classes to represent the asteroids in the game.

Create an instance of the enemya and enemyb classes to represent the enemy spaceships.

Set the number of bullets the player can fire at a time.

Create lists to store multiple instances for asteroid, enemya, and bullets objects.

Set the initial Hp of the player and

Set the initial time remaining in the game

Define process

Set enemy bullet to fire constantly

If Hp is <= 0

If the initial time equals 0

Stop game

If enemy bullet collides with player

Play bullet sound

Reset bullet

Subtract 5 from players Hp

If player collides with asteroid

Play bomb sound

Reset asteroid

Subtract Hp by 2

Define processEvent

If SPACE key is pressed

Add 1 more to current bullet

If current bullet is >= number of bullets

Current bullet gets 0

Fire the players bullet