Peculiar Conquering

Let's build a game called Peculiar Conquering!

Which Language do you have developed in Peculiar Conquering?

Python



- INTRODUCTION:-
- Python is general purpose and powerful programming languages.

Who created Python?

Developed by Guido Van Rossum

How python got it's name?

 The Name python is inspired from Guido's favorite comedy Tv show called "Monty Python Flying Circus"

Why was python Created?

• Guido started python development as a hobby in 1989.

<u>Which IDE(Integrated Development Environment) do you</u> <u>have developed in game?</u>





- INTRODUCTION
- PyCharm is a software application that provides comprehensive facilities to computer programmer for software development.

Installing Pygame?

- Firstly we have to install the Pygame module in IDE PyCharm. To Install Pygame module in PyCharm.
- To install "pip install pygame".
- The module "pip install pygame" has successfully installed.





Creating a Pygame Window and Responding to User Input

• In this phase of project development we have to make an empty window with (1200, 800)



- But we should make an empty window with (800, 600) for more user friendly.
- Which will show



Setting the Background Color

- For setting project background color with (230,230,230) which genuinely shows grey background color.
- That is showing below:-



 But for more attractive and showing beautiful we should set the project background color with (0,255,0) which shows green color



The game that name was Alien Invasion

👹 Alien Invasion

×

From my point of view the game's name should be Peculiar Conquering

👙 Peculiar Conquering

Adding the Ship Image



- The name of ship will be "Water Craft".
- Now we are gonna to add image of ship in our empty colored windows at the position of center bottom.



- Now the turn of controlling the ship in our colored window that means we are going to add the functionality of the ship which continuously moves left to right after pressing the "space bar" key.
- We can also control the speed of the ship like from 1.5 replacing by 2.0
- Then we will add the bullets that are appear in left to right with speed of 1.0(we can also change it by 1.5) on our window full display after pressing the space bar key







- The ship is releasing the bullets continuously in left to right direction after pressing the space bar key.
- If we want to quit the functionality of the ship that is releasing bullets we have to press the 'Q' or 'q' key.
- During first time of releasing bullets after pressing the space bar key only and only 3 bullets can release.



Now we are going to create or add a stranger image like:



Placing one stranger on the screen is like placing a ship on the screen.



In the first row of our window colored display the peculiar will be only sive.



We have 5 peculiar in one row but we have to work in 4 rows that means we will have be 20 peculiar which are appearing in our window display with a ship:





- Now let's make the fleet of aliens move to the right across the screen until it hits the edge, and then make it drop a set amount and move in the other direction. We'll continue this movement until all aliens have been shot down, one collides with the ship, or one reaches the bottom of the screen. Let's begin by making the fleet move to the right.
- Now we'll create the settings that will make the fleet move down the screen and to the left when it hits the right edge of the screen.

- When an alien reaches the edge, the entire fleet needs to drop down and change direction.
- In game programming, collisions happen when game elements overlap.
- We want to know right away when a bullet hits an alien so we can make an alien disappear as soon as it's hit. To do this, we'll look for collisions immediately after updating the position of all the bullets.












































- To make a new fleet of aliens appear after a fleet has been destroyed, we first check whether the aliens group is empty.
- Once all rows peculiar conquering has destroyed then again the new peculiar conquering will be appeared and these will move a bit faster than previous peculiar conquering moving speeds and the bullets will be released from the ship and shoot them.
- If an alien reaches the bottom of the screen, we'll have the game respond the same way it does when an alien hits the ship.

Peculiar conquering feels more complete now, but the game never ends.



- we'll add a play button to start a game on demand or to restart a game once it ends.
- The play button that appears before a game begins and reappears when the game ends so the player can play again.
- The play button that has added in our game window display, was white color but we have changed it by replacing blue color that makes attractive gaming.



- To start the game when the player clicks play button on the display then game will be played. The play button works only for first time when player clicks play button and will not work after first game ends.
- But the one issue with play button is during running game the play button is visible on the screen. So now we will make the play button invisible after first time clicking play button.

- Because if the play button will be appeared on the screen and by mistake if the play button clicks, the game will be restarted.
- When the player will play the game the scoring of the game will be started from zero(o).



- Now we are going to update the live score of the game on the screen whenever a peculiar will be hitted.
- In one hits of peculiar the score will be increasing by 50 but we should increase the score 10 at first hits of peculiar.





- Now we are going to focus on scoreboard for setting current score and high score when player plays the game because every player wants to beat a game,s high score.
- The current score will be along with playing the game and the high score will be previously played game's high . But if the current game's score is big than high then after ending game and playing again game in that window the scoreboard will be shown the high score that was current score in previous game which was bigger than that high score.









• Then now we want to show the level of the players who plays the game and level depends upon if the first time playing the game all peculiar will be destroyed in all rows then the player's level will be one and continuously works so on....









 In the final stage of the game "The current score" will be along with playing the game and the high score will be previously played game's high. But if the current game's score is big than high then after ending game and playing again game in that current windows the scoreboard will be shown the score that was current score in previous game which was bigger than that high score and also level of the players will be shown on the top of the right below the current scoreboard which wil be increased by hitting or destroying the all peculiar that appear in all five rows.







Python Code

• Peculiar_Conquering.py

ig.py /		
1	import sys	📒 👬 Peculiar_Conquering.py 🗵
2	from time import sleep	
3	import pygame	
4	from sets import Sets	
5	from khel_stats import KhelStats	
6	from score_board import ScoreBoard	
7	from push_button import PushButton	
8	from water_craft import WaterCraft	
9	from pellet import Pellet	
10	figm stranger import Stranger	
11	Class PeculiarConquering:	
12	definit(self):	
13	pygame.init()	
14	<pre>self.sets = Sets()</pre>	
15	<pre>self.screen = pygame.display.set_mode((0, 0), pygame.FULLSCREEN)</pre>	
16	<pre>self.sets.screen_width = self.screen.get_rect().width</pre>	
17	<pre>self.sets.screen_height = self.screen.get_rect().height</pre>	
18		
19	<pre>pygame.display.set_caption("Peculiar Conquering")</pre>	
20		
21	# Create an instance to store game statistics,	
22	# and create a score_board.	
23	<pre>self.stats = KhelStats(self)</pre>	
24	<pre>self.sb = ScoreBoard(self)</pre>	
25		
26	<pre>self.water_craft = WaterCraft(self)</pre>	
27	<pre>self.pellets = pygame.sprite.Group()</pre>	
28	<pre>self.strangers = pygame.sprite.Group()</pre>	
29	<pre>selfcreate_fleet()</pre>	
30		
31	# Make the Play push button.	
32	<pre>self.play_push_button = PushButton(self, "Play")</pre>	
33		1 IDE and Plugin Updates
	PeculiarConquering	By Charm is ready to undate
-		Pychaim is ready to update.
on Con	isole	C creating

11:26 CRLF 🕈 UTF-8 🕈 🍗 🜐

g.py 〉				
31			# Make the Play push button.	Peculiar Conquering.pv ×
32			<pre>self.play_push_button = PushButton(self, "Play")</pre>	
33				
34		def	run_game(self):	
35			"""Start the main loop for the game."""	
36			while True:	
37			<pre>selfcheck_events()</pre>	
38			<pre>if self.stats.game_active:</pre>	
39			<pre>self.water_craft.update()</pre>	
40			<pre>selfupdate_pellets()</pre>	
41			<pre>selfupdate_strangers()</pre>	
42			<pre>selfupdate_screen()</pre>	
43				
44			# Make the most recently drawn screen visible.	
45			<pre>pygame.display.flip()</pre>	
46				
47		def	_check_play_push_button(self, mouse_pos):	
48			"""Start a new game when the player clicks Play."""	
49			<pre>push_button_clicked = self.play_push_button.rect.collidepoint(mouse_pos)</pre>	
50			if push_button_clicked and not self.stats.game_active:	
51			# Reset the game stes.	
52			<pre>self.sets.initialize_dynamic_sets()</pre>	
53			# Reset the game statistics.	
54			<pre>self.stats.reset_stats()</pre>	
55			<pre>self.stats.game_active = True</pre>	
56			self.sb.prep_score()	
57			<pre>self.sb.prep_level()</pre>	
58			<pre>self.sb.prep_water_crafts()</pre>	
59				
60			# Get rid of any remaining strangers and pellets.	
61			self.strangers.empty()	
62			sell.pellets.empty()	
63				1 IDE and Plugin Updates
	F	Peculiar	Conquering	PyCharm is ready to update.
				. ,



ід.ру/			
91		<pre>elif event.key == pygame.K_LEFT:</pre>	Peculiar Conquering.py ×
92		<pre>self.water_craft.moving_left = True</pre>	
93		<pre>elif event.key == pygame.K_q:</pre>	
94		sys.exit()	
95		<pre>elif event.key == pygame.K_SPACE:</pre>	
96		<pre>selffire_pellet()</pre>	
97			
98		<pre>def _check_keyup_events(self, event):</pre>	
99		"""Respond to key releases."""	
100		if event.key == pygame.K_RIGHT:	
101		<pre>self.water_craft.moving_right = False</pre>	
102		<pre>elif event.key == pygame.K_LEFT:</pre>	
103		<pre>self.water_craft.moving_left = False</pre>	
104			
105		<pre>def _fire_pellet(self):</pre>	
106		"""Create a new pellet and add it to the pellets group."""	
107		<pre>if len(self.pellets) < self.sets.pellets_allowed:</pre>	
108		<pre>new_pellet = Pellet(self)</pre>	
109		<pre>self.pellets.add(new_pellet)</pre>	
110			
111		<pre>def _update_strangers(self):</pre>	
112			
113		Check if the fleet is at an edge,	
114		then update the positions of all strangers in the fleet.	
115			
116		sell_cneck_leet_eages()	
117		sell.strangers.update()	
118		A Tack for shares when such a list	
119		# LOOK FOR Stranger-Vater_craft Collisions.	
120		<pre>ii pygame.sprite.spritecollideany(sell.water_crait, sell.strangers):</pre>	
121		<pre>seitwater_orait_nit()</pre>	
122		# Look for strangers hitting the bottom of the screen.	
100	D	. seen see beaugest instang the bottom of the bottom	IDE and Plugin Updates
	P	ecunarconquering	PyCharm is ready to update.

🖿 A_Ship_That_Fires_Bullets 🔪 襣 🖡	Peculiar_Conquering.py		
a 🔲 Project 🔹 🖸 🤅	→ ÷ ☆ - ¹²¹	<pre>selfwater_craft_hit()</pre>	Peculiar_Conquering.py ×
A_Ship_That_Fires_Bullets	C:\Users\Dell\Pyc	# Look for strangers bitting the bottom of the screen	_
-i > 🖿 image	124	<pre>self check strangers hottom()</pre>	
khel stats.pv	125	Settoncok_solargets_solow()	
Peculiar Conquering.pv	, 126 🖯	def update pellets(self):	
nellet nv	127	"""Update position of pellets and get rid of old pellets."""	
a nucle button nu	128	<pre>self.pellets.update()</pre>	
pusn_button.py	129	# Get rid of pellets that have disappeared.	
score_board.py	130 🖯	<pre>for pellet in self.pellets.copy():</pre>	
sets.py	131	if pellet.rect.bottom <= 0:	
👰 settings.py	132	<pre>self.pellets.remove(pellet)</pre>	
🛃 stranger.py	133 🖂	<pre>selfcheck_pellet_stranger_collisions()</pre>	
> 🖿 venv library root	134		
water_craft.py	135 -	def _check_pellet_stranger_collisions(sell):	
> IIII External Libraries	130	# Resource any peliete and stranger collided	
Scratches and Consoles	137	* remove any perfects and solved in the contract. collisions - purame sprite anomycollide (self pallate self strangers True True)	
	139	controlons - pygumetaprice.groupcontac(sett.perfects, sett.setangers, riac, riac)	
	140 🖯	if collisions:	
	141	for strangers in collisions.values():	
a	142	<pre>self.stats.score += self.sets.stranger points * len(strangers)</pre>	
	143	self.sb.prep_score()	
2.	144 🖨	<pre>self.sb.check_high_score()</pre>	
1 miles	145		
1	146 🤤	if not self.strangers:	
<u>े</u>	147	# Destroy existing pellets and create new fleet.	
	148	<pre>self.pellets.empty()</pre>	
	149	self_create_fleet()	
9	150	<pre>self.sets.increase_speed()</pre>	
104	151	f Ingresse lettel	
0 2	153	self state level += 1	
SI F	700	() IDEa	nd Plugin Updates
<u></u>		recullar.conquering PyCh	arm is ready to update.
▶ <u>4</u> : Run 🖽 <u>6</u> : TODO 🗷 T	erminal 🛛 🖶 Python Console		

ng.py 〉			
151			Peculiar_Conquering.py ×
152		# Increase level.	
153		self.stats.level += 1	
154	- P	<pre>self.sb.prep_level()</pre>	
155			
156		<pre>def _update_screen(self):</pre>	
157		"""Update images on the screen, and flip to the new screen."""	
158		<pre>self.screen.fill(self.sets.bg_color)</pre>	
159		<pre>self.water_craft.blitme()</pre>	
160		<pre>for pellet in self.pellets.sprites():</pre>	
161		pellet.draw_pellet()	
162		<pre>self.strangers.draw(self.screen)</pre>	
163		# Draw the score information.	
164		<pre>self.sb.show_score()</pre>	
165		# Draw the play push button if the game is inactive.	
166		<pre>if not self.stats.game_active:</pre>	
167		<pre>self.play_push_button.draw_push_button()</pre>	
168			
169		pygame.display.flip()	
170			
171		<pre>def _create_fleet(self):</pre>	
172		"""Create the fleet of strangers."""	
173		# Create an stranger and find the number of strangers in a row.	
174		# Spacing between each stranger is equal to one stranger width.	
175		<pre>stranger = Stranger(self)</pre>	_
176		<pre>stranger_width, stranger_height = stranger.rect.size</pre>	
177		<pre>available_space_x = self.sets.screen_width - (2 * stranger_width)</pre>	
178		<pre>number_strangers_x = available_space_x // (2 * stranger_width)</pre>	
179			
180		# Determine the number of rows of strangers that fit on the screen.	
181		<pre>water_craft_height = self.water_craft.rect.height</pre>	
182		<pre>available_space_y = (self.sets.screen_height - (3 * stranger_height) - water_craft_height)</pre>	
183		number_rows = available_space_y // (2 * stranger_height)	Plugin Updates
	Pe	eculiarConquering PvCharr	n is ready to update.
hon Cons	sole	· juni	

hon Console

📄 A_Ship_That_Fires_Bullets 🔪 🚧 Peculiar	_Conquering.py				
+ Project • 🕀 👻	û — 181			<pre>water_craft_height = self.water_craft.rect.height</pre>	Peculiar_Conquering.py ×
A Chin That Fires Pullets Collins	182			<pre>available_space_y = (self.sets.screen_height - (3 * stranger_height) - water_craft_h</pre>	eight)
A_Ship_mat_Fires_Builets C:(Use	183 183			<pre>number_rows = available_space_y // (2 * stranger_height)</pre>	
> image	184				
khel_stats.py	185			# Create the full fleet of strangers.	
Peculiar_Conquering.py	186			<pre>for row_number in range(number_rows):</pre>	
🛃 pellet.py	187			for stranger_number in range(number_strangers_x):	
push_button.py	188			<pre>selfcreate_stranger(stranger_number, row_number)</pre>	
score board.pv	189		4-5	shark flast share/self).	
sets ny	190		der	_cneck_fileet_edges(self):	
antin no mu	191			for stranger in solf strangers are reached an edge.	
settings.py	192			if stranger these addes():	
stranger.py	194			self, change fleet direction()	
venv library root	195			break	
🗛 water_craft.py	196				
> IIII External Libraries	197		def	<pre>change fleet direction(self):</pre>	
Scratches and Consoles	198			"""Drop the entire fleet and change the fleet's direction."""	
	199			<pre>for stranger in self.strangers.sprites():</pre>	
	200			<pre>stranger.rect.y += self.sets.fleet_drop_speed</pre>	
	201			<pre>self.sets.fleet_direction *= -1</pre>	
	202				
	203		def	<pre>_create_stranger(self, stranger_number, row_number):</pre>	
	204			"""Create an stranger and place it in the row."""	
- FR	205			<pre>stranger = Stranger(self)</pre>	
1 tro	206			<pre>stranger_width, stranger_height = stranger.rect.size</pre>	
<u>1:5</u>	207			<pre>stranger.x = stranger_width + 2 * stranger_width * stranger_number</pre>	
	208			<pre>stranger.rect.x = stranger.x</pre>	
	209			<pre>stranger.rect.y = stranger.rect.height + 2 * stranger.rect.height * row_number</pre>	
e e	210			<pre>setf.strangers.add(stranger)</pre>	
vori	211		de F	unter anoft hit/aclf).	
and a state of the	212		aer	_water_trate_int(set);	
	213				1 IDE and Plugin Updates
		Peo	culiar	onquering	PyCharm is ready to update.
▶ <u>4</u> : Run 🗮 <u>6</u> : TODO 🕨 Terminal	🔷 Python Cons	sole			Charles

11.20 CRIE + UTE 0 + 0 0

g∙bλ			
211			Peculiar Conquering.py ×
212		<pre>def _water_craft_hit(self):</pre>	
213		"""Respond to the water_craft being hit by an stranger."""	
214		<pre>if self.stats.water_crafts_left > 0:</pre>	
215		# Decrement water_crafts_left, and update score_board.	
216		<pre>self.stats.water_crafts_left -= 1</pre>	
217		<pre>self.sb.prep_water_crafts()</pre>	
218			
219		# Get rid of any remaining strangers and pellets.	
220		<pre>self.strangers.empty()</pre>	
221		<pre>self.pellets.empty()</pre>	
222			
223		# Create a new fleet and center the water_craft.	
224		<pre>selfcreate_fleet()</pre>	
225		<pre>self.water_craft.center_water_craft()</pre>	
226			
227		# Pause.	
228		sleep(0.5)	
229		else:	
230		<pre>self.stats.game_active = False</pre>	
231		<pre>pygame.mouse.set_visible(True)</pre>	
232			
233		<pre>def _check_strangers_bottom(self):</pre>	
234		"""Check if any strangers have reached the bottom of the screen."""	
235		<pre>screen_rect = self.screen.get_rect()</pre>	
236		<pre>for stranger in self.strangers.sprites():</pre>	
237		<pre>if stranger.rect.bottom >= screen_rect.bottom:</pre>	
238		# Treat this the same as if the water_craft got hit.	
239		<pre>selfwater_craft_hit()</pre>	
240		break	
241			
242			
243 I	▶ jif	name == 'main':	IDE and Plugin Lindates
	Pe	culiarConquering	
			PyCharm is ready to update.
on Con	sole		C Litera Log
			11:26 CRIE 🗧 UTE-8 🗧 🕀





<pre>private sprite import Sprite class Fellet(Sprite): """Actass to manage pellets fired from the vater_craft""" def _init(leif, ai_game.): """Create a pellet object at the vater_craft's current position.""" super()init() self.sets = ai_game.sets self.sets.pellet_color # Create a pellet rect at (0, 0) and then set correct position. self.rect = prygame.Rect(), 0, 0, 0, exists.pellet_height) self.rect.rmidtop = ai_game.vater_craft.rect.midtop # Store the pellet's position as a decimal value. self.rect = prygame.Rect(), 0, 0, exists.pellet_height) self.rect.pridtop = ai_game.vater_craft.rect.midtop # Store the pellet's position as a decimal value. self.y = sloat(self.rect.y) def update(self): """Totow the pellet up the acreen. """ # Update the decimal position of the pellet. self.y = sli.sets.pellet_spred # Update the acreen. """ # Update the acreen. """ # Update the acreen. """ # Update the acreen. self.rect) def draw_pellet(self): """Draw the pellet to the acreen. """ pyymm.draw.rect(self.screen, self.color, self.rect) # Pellet 'init() Pellet 'init() Pellet 'init() Pellet 'init() Pellet 'init() Pellet 'init() Pellet 'init() Pellet 'init() Pellet 'init() Pellet 'init ready to update. Pellet '_init ready to update.</pre>	1		4					
<pre>class Pellet(Sprite): """% class to manage pellets fired from the vater_craft""" definit(self, ai_game): """% class to manage pellets fired from the vater_craft's current position.""" super()init_() self.sets = ai_game.sets self.color = self.sets.pellet_color self.rect = pygame.Rect(0, 0, self.sets.pellet_vidth, self.sets.pellet_height) self.rect.midtop = ai_game.varer_craft.rect.midtop f Store the pellet's position as a decimal value. self.y = float(self.rect.y) def update (self): """Move the pellet up the screen.""" f Update the decimal position of the pellet. self.y = self.sets.pellet_gaped f Update the rect position. self.rect.y = self.sets.pellet_to the screen.""" f Update the pellet to the screen.""" f update the pellet to the screen.""" pygame.draw.rect(self.screen, self.color, self.rect) f DE and Plugin Updates Pellet >init_() Pellet >init_() DE and Plugin Updates PyCharm is ready to update. rectored</pre>	2	from purgame envite import Sprite						
<pre>class Pellet (Sprite): """" class to manage pellets fired from the water_craft""" definit_() definit_() definit_() def trav_pellet (self, si_game): """Create a pellet object at the water_craft's current position.""" super()init_() self.sorten = ai_game.sorten self.sorten = ai_game.sorten self.sort = pygame.ketcl(), 0, self.sets.pellet_height) self.rect.midtop = ai_game.water_craft.rect.midtop forcet = pygame.ketcl(), 0, self.sets.pellet_height) self.rect.midtop = ai_game.water_craft.rect.midtop forcet = pygame.ketcl(), 0, self.sets.pellet_height) self.y = float(self.rect.y) for update(self): """Draw the pellet up the screen.""" f Update the decimal position of the pellet. self.y = self.sets.pellet_speed f (bpdate the rect position. self.y = self.sets.pellet_speed f (bpdate the rect position. self.rect.y = self.sets.pellet_speed f (bpdate (self): """Draw the pellet to the screen.""" pygame.draw.rect(self.screen, self.color, self.rect) forcet publet /_init_() Pellet /_init_() forcet = screen.""" pycharm is ready to update. forcet = screen.""" pycharm is ready to update. forcet = screen."" pycharm is ready to update. forcet = screen."" pycharm is ready to update. forcet = screen."" pycharm is ready to update. </pre>	3	The pygunctoprice import oprice						
<pre>"""% class to manage pellets fired from the vater_craft""" definit(self, ai_game): """% class to be applete to be vater_craft's current position.""" super()init() self.sets = ai_game.sets self.sets = ai_game.sets self.sets = pellet to tat (0, 0) and then set correct position. self.rect = pygame.Rect(0, 0, self.sets.pellet_beight) self.rect.midtop = ai_game.vater_craft.rect.midtop f Store the pellet's position as a decimal value. self.y = float(self.set.yellet_speed</pre>	4	class Pellet(Sprite):						
<pre>definit(self, ai_game): """Create a pellet object at the water_craft's current position.""" super()init() self.sercen = ai_game.sercen self.serts = ai_game.sercen self.serts = ai_game.sercen self.serts = ai_game.water_craft.rect.position. self.rect.ret = pygame.Rect(), 0, 0 and then set correct position. self.rect.ret = pygame.Rect(), 0, 0 and then set correct position. self.rect.ret = pygame.Rect(), 0, 0 and then set correct position. self.rect.ret = pygame.Rect(), 0, 0 and then set correct position. self.rect.ret = pygame.Rect(), 0, 0 and then set correct position. self.rect.ret = pygame.Rect(), 0, 0 and then set correct position. self.rect.ret = pygame.Rect(), 0, 0 and then set correct position. self.rect.ret = pygame.Rect(), 0, 0 and then set correct position. self.rect.ret = pygame.Rect(), 0, 0 and then set correct position. self.rect.ret = pygame.Rect(), 0, 0 and then set correct position. self.rect.ret = pygame.Rect(), 0, 0 and then set correct position. self.rect.ry def update(self): """Move the pellet up the screen.""" f Update the decimal position of the pellet. self.rect.ry = self.g def draw.pellet(self): """Draw the pellet to the screen.""" pygame.draw.rect(self.screen, self.color, self.rect) Pellet > _init_() Pellet > _init() Pellet > _in</pre>	5	"""A class to manage pellets fired from the water craft"""						
Pellet >init0 Pellet >init0 Pellet >init0	6	G def init (self. aj game):						
<pre>super()init() super()init() super().</pre>	7	"""Crate a pellet object at the water craft's current position."""						
<pre>self.soren = ai_game.screen self.soren = ai_game.screen self.soren = ai_game.screen self.sets = ai_game.scres self.sets = ai_game.scres self.color = self.sets.pellet_color # Create a pellet rect at (0, 0) and then set correct position. self.rect = pygame.Rect(0, 0, self.sets.pellet_height) self.rect.midtop = ai_game.Vater_creat.rect.midtop f # Store the pellet's position as a decimal value. self.y = float(self.rect.y) def update(self): """Move the pellet up the screen.""" # Update the decimal position of the pellet. self.y = self.sets.pellet_speed # Update the rect position. self.rect.y = self.y def draw_pellet(self): """"Move the pellet to the screen.""" pygame.draw.rect(self.screen, self.color, self.rect) Pellet >init_0 DE and Plugin Updates PyCham is ready to update.</pre>	8	super(), init ()						
<pre>self.sets = ai_game.sets self.color = self.sets.pellet_color f Create a pellet rect at (0, 0) and then set correct position. self.rect.midtop = ai_game.water_craft.rect.midtop f Create a pellet 's position as a decimal value. self.rect.midtop = ai_game.water_craft.rect.midtop f Store the pellet's position as a decimal value. self.y = float(self.rect.y) def update (self): """Move the pellet up the screen.""" f Update the rect position. self.rect.y = self.sets.pellet_speed f Update the rect position. self.rect.y = self.y def draw_pellet(self): """Draw the pellet to the screen.""" pygame.draw.rect(self.screen, self.color, self.rect) </pre>	9	self.screen = ai game.screen						
<pre>11 self.color = self.sets.pellet_color 2 f Create a pellet rect at (0, 0) and then set correct position. 2 self.rect = pygame.Rect(0, 0), self.sets.pellet_width, self.sets.pellet_height) 3 self.rect.midtop = ai_game.water_craft.rect.midtop 4 f Store the pellet's position as a decimal value. 3 self.y = float(self.rect.y) 4 def update(self); 7 ""Wove the pellet up the screen. """ 4 f Dpdate the decimal position of the pellet. 3 self.y -= self.sets.pellet_speed 4 f f Dpdate the rect position. 5 self.rect.y = self.y 6 def draw_pellet(self): 7 def draw_pellet(self): 7 muture the pellet to the screen. """ 9 pygame.draw.rect(self.screen, self.color, self.rect) 8 of 9 def update. 9 def draw_pellet to the screen. """ 9 pygame.draw.rect(self.screen, self.color, self.rect) 9 def draw_pellet. 9 def draw_pellet. 9 def draw_pellet to the screen. """ 9 pygame.draw.rect(self.screen, self.color, self.rect) 9 def draw_pellet. 9 def draw_pellet.to the screen. """ 9 pygame.draw.rect(self.screen, self.color, self.rect) 9 def draw_pellet.to the screen. """ 9 pygame.draw.rect(self.screen, self.color, self.rect) 9 def draw_pellet.to the screen.""" 9 def draw_pellet.to the screen.""" 9 pygame.draw.rect(self.screen, self.color, self.rect) 9 def draw_pellet.to the screen.""" 9 def draw_pellet.to the screen.to the screen.to the screen.to the screen.to the screen.t</pre>	10	self.sets = ai game.sets						
<pre>12</pre>	11	self.color = self.sets.pellet color						
<pre>self.rect = pygame.Rect(0, 0, self.sets.pellet_width, self.sets.pellet_height) self.rect.midtop = ai_game.water_craft.rect.midtop # Store the pellet's position as a decimal value. self.y = float(self.rect.y) def update(self): """Move the pellet up the screen.""" # Update the decimal position of the pellet. self.y = self.sets.pellet_speed # Update the rect position. self.rect.y = self.y def draw_pellet(self): """Draw the pellet to the screen.""" pygame.draw.rect(self.screen, self.color, self.rect) </pre> Pellet >init0 DE and Plugin Updates PyCharm is ready to update.	12	# Create a pellet rect at $(0, 0)$ and then set correct position.						
<pre>14 self.rect.midtop = ai_game.water_craft.rect.midtop 15 16 17 18 19 19 19 10 10 19 10 10 10 10 10 10 10 10 10 10 10 10 10</pre>	13	<pre>self.rect = pygame.Rect(0, 0, self.sets.pellet width, self.sets.pellet height)</pre>						
<pre>15 16 17 18 19 19 19 19 19 19 19 19 19 19 19 19 19</pre>	14	self.rect.midtop = ai_game.water_craft.rect.midtop						
<pre> # Store the pellet's position as a decimal value. self.y = float(self.rect.y) def update(self): """Move the pellet up the screen.""" # Update the decimal position of the pellet. self.y -= self.sets.pellet_speed # Update the rect position. self.rect.y = self.y def draw_pellet(self): """Draw the pellet to the screen.""" pygame.draw.rect(self.screen, self.color, self.rect) Pellet >init0 DE and Plugin Updates PyCharm is ready to update. </pre>	15							
<pre>17 self.y = float(self.rect.y) 18 19 def update(self): 11</pre>	16	# Store the pellet's position as a decimal value.						
<pre>18 19 of control of the pellet up the screen.""" 20 # Update (self): 21 # Update the decimal position of the pellet. 22 ***********************************</pre>	17	<pre>self.y = float(self.rect.y)</pre>						
<pre>ls of def update (self): """Move the pellet up the screen.""" # Update the decimal position of the pellet. self.y -= self.sets.pellet_speed # Update the rect position. self.rect.y = self.y def draw_pellet(self): """Draw the pellet to the screen.""" pygame.draw.rect(self.screen, self.color, self.rect) """Draw the pellet to the screen.""" pygame.draw.rect(self.screen, self.color, self.rect) """Dotate the rect updates Pellet ></pre>	18							
<pre>20 """Move the pellet up the screen.""" 21 # Update the decimal position of the pellet. 22 23 self.y -= self.sets.pellet_speed 24 # Update the rect position. 25 self.rect.y = self.y 26 27 def draw_pellet(self): 28 """Draw the pellet to the screen.""" 29 pygame.draw.rect(self.screen, self.color, self.rect) 30 31 Pellet >init_() Pellet >init() Pellet >init() Pellet >init() Pellet >init() Pellet >init() Pellet >init(</pre>	19 🎯	def update(self):						
<pre>21 # Update the decimal position of the pellet. 22 23 self.y -= self.sets.pellet_speed 24 # Update the rect position. 25 self.rect.y = self.y 26 27 def draw_pellet(self): 28 """Drav the pellet to the screen.""" 29 pygame.draw.rect(self.screen, self.color, self.rect) 30 self.rect.y = self.y 4</pre>	20	"""Move the pellet up the screen."""						
<pre>22 23 23 24 24 25 25 25 25 25 25 25 25 25 25 25 25 25</pre>	21	# Update the decimal position of the pellet.						
<pre>23 self.y -= self.sets.pellet_speed 24</pre>	22							
<pre>24</pre>	23	<pre>self.y -= self.sets.pellet_speed</pre>						
<pre>25 self.rect.y = self.y 26 27 def draw_pellet(self): 28</pre>	24	# Update the rect position.						
<pre>26 27 26 27 26 27 26 27 28 29 29 29 29 29 29 29 29 29 29 29 29 29</pre>	25	self.rect.y = self.y						
<pre>27 def draw_pellet(self): 28</pre>	26							
28 """Draw the pellet to the screen. """ 29 pygame.draw.rect(self.screen, self.color, self.rect) 30 31 Pellet >init0 an Console	27	def draw_pellet(self):						
29 pygame.draw.rect(self.screen, self.color, self.rect) 30 31 Pellet >init() an Console	28	"""Draw the pellet to the screen."""						
30 31 Pellet >init() an Console PyCharm is ready to update.	29	<pre>pygame.draw.rect(self.screen, self.color, self.rect)</pre>						
31 ● Pellet >init0 ● on Console ●	30							
Pellet → _init_() Pellet → _init_() PyCharm is ready to update. PyCharm is ready to update.	31							
Pellet >init() IDE and Plugin Updates pyCharm is ready to update. PyCharm is ready to update.								
Pellet > _init_() PyCharm is ready to update.		1 IDE and Plugin Up	dates					
nn Console		Pellet >init()	to undate					
	on Conce	ola	io apaater					



• khel_stats.py
1 (class KhelStats:		khel_stats.py ×
2	"""Track statistics for Peculiar Conquering."""		
3			
4 (<pre>definit(self, ai_game):</pre>		
5	"""Initialize statistics."""		
6	<pre>self.sets = ai_game.sets</pre>		
7	<pre>self.reset_stats()</pre>		
8			
9	# Start khel in an inactive state.		
10	self.game_active = False		
11			
12	# High score should never be reset.		
13 (self.high_score = 0		
14			
15	def reset_stats(self):		
16	"""Initialize statistics that can change during the game. """		
17	<pre>self.water_craits_left = self.sets.water_crait_limit</pre>		-
18	self.score = 0		-
19 (Sell.level = 1		-
20			
		IDE and Plugin Updates	
		PyCharm is ready to update.	
on Consol			



• push_button.py

	Junionipy
a class DuckButton	
3 Class Fusibucton.	
4	
5 definit(self, ai_game, msg):	
6 self.screen = ai_game.screen	
<pre>7 self.screen_rect = self.screen.get_rect()</pre>	
8	
9 self.width, self.height = 145, 45	
10 self.push_button_color = (0, 0, 255)	
11 self.text_color = (255, 255, 255)	
<pre>12 self.font = pygame.font.SysFont(None, 48)</pre>	
13	
14 self.rect = pygame.Rect(0, 0, self.width, self.height)	
15 self.rect.center = self.screen_rect.center	
16	
17 self.prep_msg(msg)	
18	
19 det prep_msg(self, msg):	
<pre>20 seif.msg_image = seif.font.render(msg, True, seif.text_color, seif.push_putton_color)</pre>	
<pre>21 seil.msg_image_rect = seil.msg_image.get_rect() 21</pre>	
22 Self.msg_mage_rect.center = self.rect.center	
43 def dear much button (self):	
as def draw_usi_button(set).	
<pre>25 setFischeran bilt(setFiguar_marg_self marg_imarg_sert)</pre>	
27	
28	
29	
30	
PushButton	
PyCharm is ready to update.	

ion Console



score_board.py

1	import pygame.font	📕 🛃 score_board.py 🗵
2	from pygame.sprite import Group	
3	G from water_craft import WaterCraft	
4	Class ScoreBoard:	
5	"""A class to report scoring information."""	
6	<pre>definit(self, ai_game):</pre>	
7	"""Initialize scorekeeping attributes."""	
8	<pre>self.ai_game = ai_game</pre>	
9	self.screen = ai_game.screen	
10	<pre>self.screen_rect = self.screen.get_rect()</pre>	
11	<pre>self.sets = ai_game.sets</pre>	
12	self.stats = ai_game.stats	
13	# Font sets for scoring information.	
14	self.text_color = (30, 30, 30)	
15	<pre>self.font = pygame.font.SysFont(None, 48)</pre>	
16		
17	# Prepare the initial score images.	
18	<pre>self.prep_score()</pre>	
19	<pre>self.prep_high_score()</pre>	
20	<pre>self.prep_level()</pre>	_
21	<pre>self.prep_water_crafts()</pre>	
22		_
23	def prep_score(self):	
24	"""Turn the score into a rendered image."""	
25	rounded_score = round(self.stats.score, -1)	
26	<pre>score_str = "{:,}".format(rounded_score)</pre>	
27	self.score_image = self.font.render(score_str, True, self.text_color, self.sets.bg color)	
28		
29	# Display the score at the top right of the screen.	
30	<pre>self.score rect = self.score image.get rect()</pre>	
31	self.score rect.right = self.screen rect.right - 20	
32	self.score rect.top = 20	
33		U. d. d
	ScoreBoard >init() PyCharm is read	Updates dy to update.
on Conso	ole	_ manage

hon Console

31			<pre>self.score_rect.right = self.screen_rect.right - 20</pre>	score board.pv ×
32			<pre>self.score_rect.top = 20</pre>	
C 33				_
34			def show_score(self):	
35			"""Draw scores, level, and water_crafts to the screen."""	
36			<pre>self.screen.blit(self.score_image, self.score_rect)</pre>	-
37			<pre>self.screen.blit(self.high_score_image, self.high_score_rect)</pre>	
38			<pre>self.screen.blit(self.level_image, self.level_rect)</pre>	
39			<pre>self.water_crafts.draw(self.screen)</pre>	
40				
41			<pre>def prep_high_score (self):</pre>	
42			"""Turn the high score into a rendered image."""	
43			high_score = round(self.stats.high_score, -1)	
44			high_score_str = "{:,}".format(high_score)	
45			<pre>self.high_score_image = self.font.render(high_score_str, True, self.text_color, self.sets.)</pre>	bg_color)
46				
47			# Center the high score at the top of the screen.	
48			<pre>self.high_score_rect = self.high_score_image.get_rect()</pre>	
49			<pre>self.high_score_rect.centerx = self.screen_rect.centerx</pre>	
50			<pre>self.high_score_rect.top = self.score_rect.top</pre>	
51				
52			<pre>def check_high_score(self):</pre>	-
53			"""Check to see if there's a new high score."""	
54			<pre>if self.stats.score > self.stats.high_score:</pre>	-
55			<pre>self.stats.high_score = self.stats.score</pre>	-
56			<pre>self.prep_high_score()</pre>	
57				
58			<pre>def prep_level(self):</pre>	_
59			"""Turn the level into a rendered image. """	
60			<pre>level_str = str(self.stats.level)</pre>	
61			<pre>self.level_image = self.font.render(level_str, True, self.text_color, self.sets.bg_color)</pre>	
62			# Position the level below the score.	
63			<pre>self.level_rect = self.level_image.get_rect() </pre>	IDE and Plugin Updates
		Sco	oreBoard > _init_()	PyCharm is ready to update.
hon (Conco	le.		. ,

te. (today 15:26)

12:35 CRLF 🕈 UTF-8 🕈 🍗 😁

	51					🛃 score_board.py 🛛
2	52 (def	check_high_score(self):		
	53			"""Check to see if there's a new high score."""		
	54			<pre>if self.stats.score > self.stats.high_score:</pre>		
	55			<pre>self.stats.high_score = self.stats.score</pre>		
	56			self.prep_high_score()		
	57					
	58 (def	prep_level(self):		
1	59			"""Turn the level into a rendered image."""		
٦	60			<pre>level_str = str(self.stats.level)</pre>		
	61			<pre>self.level_image = self.font.render(level_str, True, self.text_color, self.sets.bg_color)</pre>	_	
	62			# Position the level below the score.	_	
	63			<pre>self.level_rect = self.level_image.get_rect()</pre>		
	64			<pre>self.level_rect.right = self.score_rect.right</pre>		
	65 (seir.level_rect.top = seir.score_rect.bottom + 10		
	66	L	4-5			
	67 (der	prep water_craits(sell):		
	68			Show now many valer craits are left.	_	
	69	L .		self.water_crafts = Group()	_	
	70 (for water_craft number in range(self.stats.water_crafts_left):	_	
	71			water_crait = watercrait(self.al_game)		
	72			water_crait.rect.x = 10 + water_crait_number * water_crait.rect.width		
	73			water_crait.rect.y = 10		
	74 0			sell.water_craits.add(water_crait)	- 11	
	75					
	70					
	78					
		Scor	reBoa	IDE and Plugin Updates IDE and Plugin Updates		
4		3001	coua	PyCharm is ready to update		
h	on Consol	le				C cremeroy

te. (today 15:26)



1 6	class Sets:		sets nv 2
2 (definit(self):		- secopy
3	self.screen_width = 800		
4	<pre>self.screen_height = 600</pre>		
5	<pre>self.bg_color = (0, 255, 0)</pre>		
6			
7	# water_craft sets		
8	<pre>self.water_craft_limit = 3</pre>		
9			
.0	# Pellet sets		-
.1	<pre>self.pellet_width = 3</pre>		
.2	<pre>self.pellet_height = 15</pre>		
.3	<pre>self.pellet_color = (255, 0, 0)</pre>		
L4	#self.pellet_color = (255, 0, 0)		
.5	<pre>self.pellets_allowed = 3</pre>		
16			
.7	# stranger sets		
18	<pre>self.fleet_drop_speed = 10</pre>		
.9	# How quickly the game speeds up		
0	<pre>self.speedup_scale = 1.1</pre>		-
1			
2	<pre>self.score_scale = 1.5</pre>		
3			
4 6	<pre>self.initialize_dynamic_sets()</pre>		
5			
6 5	<pre>def initialize_dynamic_sets(self):</pre>		
7	<pre>self.water_craft_speed = 2</pre>		
8	<pre>self.pellet_speed = 3.0</pre>	_	-
9	<pre>self.stranger_speed = 1.5</pre>		
80	<pre># fleet_direction of 1 represents right; -1 represents left.</pre>		
31	<pre>self.fleet_direction = 1</pre>		
32			
3	# Scoring	1 IDE and Plugin Updates	
	Sets	DuCharmin ready to undate	

- (4- J-.. 15.76)

16					a sets	5.DV ×
17			# stranger sets			17
18			<pre>self.fleet_drop_speed = 10</pre>			
19			# How quickly the game speeds up			
20			<pre>self.speedup_scale = 1.1</pre>			
21			-			
22			<pre>self.score_scale = 1.5</pre>			
23						
24			<pre>self.initialize_dynamic_sets()</pre>			
25						
26		def	f initialize_dynamic_sets(self):			
27			<pre>self.water_craft_speed = 2</pre>			
28			<pre>self.pellet_speed = 3.0</pre>			
29			<pre>self.stranger_speed = 1.5</pre>			
30			<pre># fleet_direction of 1 represents right; -1 represents left.</pre>			
31			self.fleet_direction = 1			
32						
33			# Scoring			
34			<pre>self.stranger_points = 10</pre>		1	
35						
36		def	f increase_speed(self):			
37			"""Increase speed sets and stranger point values."""			
38			<pre>self.water_craft_speed *= self.speedup_scale</pre>			
39			<pre>self.pellet_speed *= self.speedup_scale</pre>			
40			<pre>self.stranger_speed *= self.speedup_scale</pre>			
41			<pre>self.stranger_points = int(self.stranger_points * self.score_scale)</pre>			
42			<pre>print(self.stranger_points)</pre>			
43					- 18 M	
				IDE and Plugin Updates		
	Se	ts		PyCharm is ready to update.		
hon Co	nsole			,, ., ., ., ., ., ., ., ., ., ., .		E.



1	🖯 imj	port pygame	📒 🛃 stranger.p
2	efr	om pygame.sprite import Sprite	
3	⊖ cl a	ass Stranger(Sprite):	
4		"""A class to represent a single stranger in the fleet."""	
5		<pre>definit(self, ai_game):</pre>	
6		"""Initialize the stranger and set its starting position."""	
7		<pre>super()init()</pre>	
8		<pre>self.screen = ai_game.screen</pre>	
9		<pre>self.sets = ai_game.sets</pre>	
.0		# Load the stranger image and set its rect attribute.	
.1		<pre>self.image = pygame.image.load('image/stranger.bmp')</pre>	
.2		<pre>self.rect = self.image.get_rect()</pre>	
.3		# Start each new stranger near the top left of the screen.	
.4		<pre>self.rect.x = self.rect.width</pre>	
.5		<pre>self.rect.y = self.rect.height</pre>	
6		# Store the stranger's exact horizontal position.	
7		<pre>self.x = float(self.rect.x)</pre>	
.8		<pre>def check_edges(self):</pre>	
.9		"""Return True if stranger is at edge of screen."""	
0		<pre>screen_rect = self.screen.get_rect()</pre>	
1			
2		<pre>if self.rect.right >= screen_rect.right or self.rect.left <= 0:</pre>	
3		return True	
4			
5 🎯		<pre>def update(self):</pre>	
6		"""Move the stranger right or left."""	
7		<pre>self.x += (self.sets.stranger_speed * self.sets.fleet_direction)</pre>	
8		<pre>self.rect.x = self.x</pre>	
9			
0			
			IDF and Plugin Undates
	Str	ranger >init()	DuChamp is seed, to us date
<i>C</i>			PyCharm is ready to update.
Conse	Die		C man by



• water_craft.py

1 (import pygame	water craft.pv ×
2	from pygame.sprite import Sprite	
3 [class WaterCraft(Sprite):	-
4	"""A class to manage the water_craft."""	
5 (<pre>definit(self, ai_game):</pre>	
6	"""Initialize the water_craft and set its starting position."""	
7	<pre>super()init()</pre>	
8	<pre>self.screen = ai_game.screen</pre>	
9	<pre>self.sets = ai_game.sets</pre>	
10	<pre>self.screen_rect = ai_game.screen.get_rect()</pre>	
11	# Load the water_craft image and get its rect.	
12	<pre>self.image = pygame.image.load('image/waterCraft.bmp')</pre>	
13	<pre>self.rect = self.image.get_rect()</pre>	
14	# Start each new water_craft at the bottom center of the screen.	
15	<pre>self.rect.midbottom = self.screen_rect.midbottom</pre>	
16	# Store a decimal value for the water_craft's horizontal position.	
17	<pre>self.x = float(self.rect.x)</pre>	
18	# Movement flag	
19	<pre>self.moving_right = False</pre>	
20 (self.moving_left = False	
21		
22 🔍 🛛	def update(self):	
23	"""Opdate the water_craft's position based on movement flags."""	
24	# Update the water_craft's x value, not the rect.	
25	<pre>if self.moving_right and self.rect.right < self.screen_rect.right:</pre>	
26	<pre>self.x += self.sets.water_craft_speed</pre>	
27		
28	<pre>if self.moving_left and self.rect.left > 0:</pre>	
29	<pre>self.x -= self.sets.water_craft_speed</pre>	
30		
31	# Update rect object from self.x.	
32 (<pre>self.rect.x = self.x</pre>	
33		IDF and Plugin Undates
	WaterCraft	
		PyCharm is ready to update.