Tails of iron code

Initial part of the code

* Declare ArrayList for bullets.
* Declare array for enemies.
* Declare variables.
* Call classes player, and timer.
* Used inventory skills (10,38,34(initialize the PVector, Array and ArrayList))

Void setup()

* Initialize enemy array.
* Create an object for player.
* Spawn enemies
* Set background colour.
* Set canva size.
* Set flag random vector.
* Inventory skill 4: is used by using the setup() function
* Inventory skill 5: is used by drawing the background.
* inventory skill 29: keyword “new” is used to create object for player.
* Used inventory skills (4,5,29)

Boolean CheckStart()

* Paints screen all black until player presses B key.
* If player press b key return variable true.
* Inventory skill 13: is used by checking the input by the player to see if the b key is pressed.
* Inventory skill 21: is used as the function returns a type Boolean.
* Inventory skill 14: Checks if player has pressed lowercase b or uppercase B
* Inventory skills used(13,14,21)

Void draw()

* If player has pressed B key and received variable as true
  + Start game
* If player had not pressed b and variable is false
  + Draw giant black rectangle on screen and display text telling player to press B key
* Draw scenario
* Draw player
* Draw flag by calling flag()
* Draw enemy
* Assing PVector to player location.
* Loop trough bullet array
  + To run trough bullet array a for loop can be used
* Draw bullets
* Make bullet travel trough screen
* Initialize enemy array
* Run trough enemy array
  + To run trough enemy array a for loop can be used.
* Spawn enemies (populate enemy array)
* Check if enemies are dead (manage array, the enemies are the objects of the array )
  + If enemies are dead spawn more
* Check if all enemies have been spawned ( use break() to stop spawning loop)
  + If so stop spawning
* Inventory skill 16: will be used to run trough the array and arrayList
* Inventory skill 17: as the draw function is already a loop having a for loop inside it creates a nested loop
* Inventory skill 18: will use the function break() to stop spawning enemies:
* Inventory skill 20: scenario function, that draws the scenario is called, which does not return anything.
* Inventory skill 21: function checkstart is called, which returns a Boolean.
* Inventory skill 33: enemies are spawned, populating the array.
* Used inventory skills (16,17,18,20,33)

Void flag()

* Draw flag
* Check if local variable is 0.
  + If so, set local variable to 1 and draw flag.
* Initialize vector.
* Set vector values to random.
* Draw flag.
* Inventory skill 3: corner modes are used to draw the flag.
* Inventory skill 9: local variable is created and used.
* Inventory skill 41: the location vector of the flag is randomized.
* Used inventory skills (3,9,41)

Void scenario()

* Draws platforms.
* Inventory skill 1: plataforms are drawn using rect functions.
* Inventory skill 2: colors of plataforms are decided using fill() function.
* Used inventory skills (1,2)

Void keypressed()

* check keys to make player walk
  + use of if, else if to check if player has pressed the keys.
  + If player presses the key set variable in player class as true to make player move
* call function to shoot.
* inventory skill 7: keypressed is used.
* Inventory skill 12: if else statements are used to check keys that are pressed.
* Inventory skill 13: Boolean functions are used for player controller.
* Inventory skill 14: if statements used ||.
* used inventory skills (7,1213,14)

void shooting

* make switch case to check which arrow keys the player pressed to shoot
  + if the case is met populate the arrayList by one and create the object
* used inventory skills(15,29,34,38)
* Inventory skill 15: switch statement used to check player input instead of if, else functions
* Inventory skill 29: the keyword new is used to create new vector.
* Inventory skill 34: bullets ArrayList is populated by shooting.
* Inventory skill 38: PVector is used to find direction in where the player is shooting.

Keyreleased()

* Use if and else if statement to check if player has released the keys
  + If so set variable from player class as false and the player stops moving in that direction

**Class bullets**

* create variables

Bullets ( constructor)

* instantiate pvector
* sets initial location to player.
* Uses PVector direction to find where to shoot the bullets.
* make velocity.
* Inventory skill 28: constructor class for bullets is created.
* Inventory skill 30: constructor need parameters.
* used skills (23,24,28,30)

model

* draw bullets.

Travel

* make bullet travel across the screen by adding velocity to location PVector.
* Inventory skill 39: velocity is applied to bullet.
* used inventory skills (39)

collision

* check if bullets are off the screen, if so delete them from ArrayList.
* Inventory skill 36: remove bullets that are out of the screen.
* used inventory skills (36)
* check collision with enemy
  + if collided destroy bullet and deduct one health point from enemy.

**Class player**

* write variables

Create constructor

* instantiate pvector for location and velocity.
* create constructor for class player.
  + does not accept any parameters.
* Inventory skills used: 16, player class does not accept any parameters.
* Inventory skills used: 28, player class constructor is created.

Void collision

* check if player is out of the screen
  + if so teleport player to coordinates in the screen, giving the effect of the player walking towards a wall.
* check if player is colliding with the ground
  + if so make variable that player is touching the ground true;

Void applyGravity

* Add gravity value to the y axis of the velocity PVector.
* Add velocity to the location PVector to simulate gravity.
* Inventory skill 8: increments gravity in the y axis velocity PVector
* used inventory skill (8)

void move(float speedModifier)

* function takes float as speed modifier to easily change speed if necessary, during play.
* check if Booleans to see if player is on the ground
  + if so allow player to jump
* Check Boolean to see if player is pressing key to walk
  + If so increment the velocity vector with the speed times the speed modifier.
* print player location to help with coding.
* Inventory skill 11: prints player location to help with debugging.
* Inventory skill 23: float speed modifier is being passed by copy
* used inventory skills (11,23)

void model()

* draw player.
* use rectMode(CENTER) to draw player in center of vector
* Inventory skill 3: Uses rectMode(Center) to draw player in center of vector
* used inventory skills (3)

void health

* check if player has 0 health.
  + If so draw black rectangle on the screen with text showing game over and how many points the player got
  + Check if player pressed P
    - If so restart game

**class enemy**

* declare variables

void constructor

* Assign random location
* uses constrain to make sure the enemy does not get out of the screen.
* Inventory skill 6: constrains enemy location.
* used inventory skills (6).

void health

* check if enemy has 0 health.
  + if so, increase the variable of dead enemies.

void model

* draw enemy model.

void teleport

* create local float variable for distance.
* Make function take PVector which should be the player’s location as argument.
* Calculate distance between player vector and enemy vector.
* check if distance is small enough.
  + if so, teleport to a random position on screen
* Inventory skill 9: local variable to assign the float distance between player and enemy.
* Inventory skill 40: calculates location between enemy and player.
* used inventory skills (9,40)

void attack

* check if enemy is close enough to player
  + if so attack (spawn EBullets)

**Class EBullets**

* create variables
* Inventory skill 30: Constructor for Ebullets requires parameters.

used inventory skills (30)

Bullets constructor(PVector location)

* instantiate pvector based on enemy location

void Shoot(PVector player)

* lerp enemy location to player location
* Inventory skill 40: find direction of enemy and player
* Inventory skill 43: use of function .lerp on PVector.
* used skills (40,43);

model

* draw bullets

collision

* check if bullets are off the screen
  + if so delete them from ArrayList.
* Check if bullets collide with player
  + If so delete bullet and deduct one health point from player.
* Inventory skill 36: delete bullets from ArrayList.
* used inventory skills(36)

**Class timer**

* create variable time.

contructor Timer (float set)

* set variable for time as the same as variable set.

float getTime()

* return variable time.

Void setTime( float set)

* set variable for time as the same as variable set.

Void countUp()

* increase time variable each 60 frames.

void countdown()

* decrease time variable each 60 frames.