Godot C#
Top Down Shooter
Tutorial

2

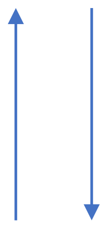
For Complete Beginners

Objectives

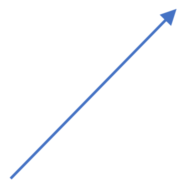
- improve player movement
- create Bullet object
- make Player shoot bullets (when mouse clicked)

Vectors

- think of them as arrows (with an x and y component)

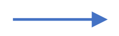

no x component

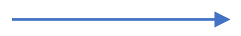
no y component

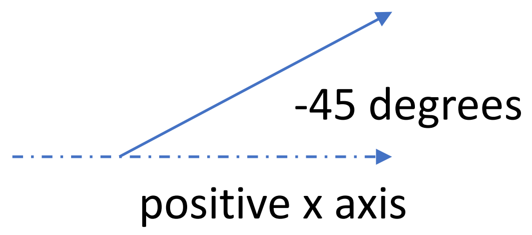

both x and y component

Vectors

- have size (aka length, aka magnitude) and direction





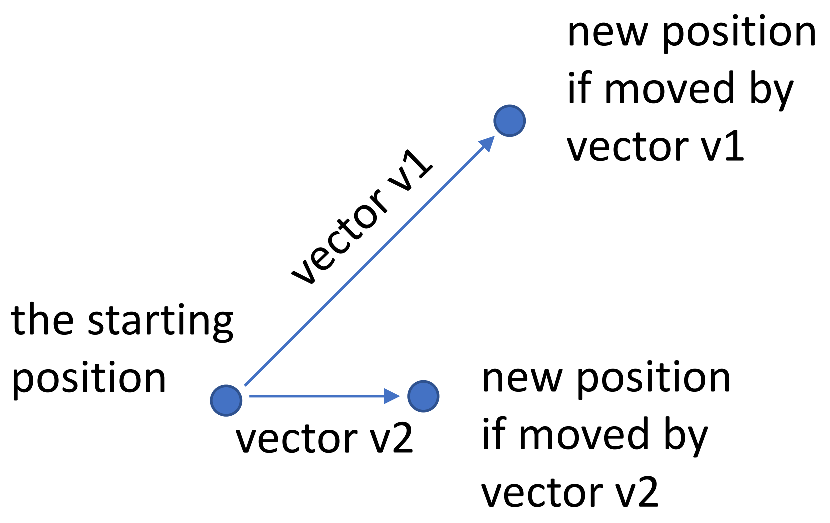


a small vector that points to the right

a vector twice as large, that still points to the right

a vector that points in -45 degrees from positive x axis (in Godot, angles are relative to positive x axis)

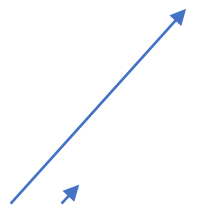
Moving a Position by a Vector



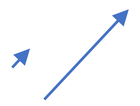
just slide the position along the vector!

More Vector Operations

- normalizing means keeping the same direction but making length 1
- multiplying a vector by a number greater than 1 means enlarging it (e.g. multiplying it by 2 means making it twice as large)
- multiplying a vector by a fraction (i.e. number smaller than 1) means shrinking it (e.g. multiplying by 0.5 means making it half as big)



right vector is normalized version of left vector (has same direction, but is very tiny, i.e. has length of only 1)



right vector is left vector multiplied by about 5 (i.e. is 5 times bigger)

said another way, left vector is right vector multiplied by 1/5 (i.e. is 5 times smaller)

signals and slots

- some nodes emit signals
- a signal is emitted by an object when something interesting happens to that object
- you can attach methods that will be called when a certain signal of a certain object is emitted
  - these callback methods are often called slots

Input

- when mouse or keyboard events occur (i.e. left mouse button clicked), all nodes are notified one by one by calling certain methods on the nodes
- each node has a chance to declare the event as “handled”, once a node declares the event as handled, the event stops getting passed to other nodes
- usually want to handle input events in your node’s UnhandledInput() method. Override this, if the event is what you were looking for, handle it by calling GetTree().SetInputAsHandled()

Bullet

- Bullet.tscn
  - Node2D <- Bullet.cs
    - Area2D (for detecting collisions)
    - Sprite (graphical representation)
- Bullet.cs
  - in _Process, keep moving forward until moved enough
  - in _Ready, connect to Area2D’s area_entered and body_entered signals
    - in callback, print a message and free (destroy) the bullet