*Quartz Core is a collection of API for image processing and video image manipulation.*

*use of the*[*CABasicAnimation*](https://developer.apple.com/library/mac/#documentation/GraphicsImaging/Reference/CABasicAnimation_class/Introduction/Introduction.html)*class to create basic animation.*

*The iOS Core Motion framework allows developers to get motion data from device hardware and process that data. Here the hardware includes the accelerometer, magnetometer and gyroscope. By using the*[*CMMotionManager*](http://developer.apple.com/library/ios/#documentation/CoreMotion/Reference/CMMotionManager_Class/Reference/Reference.html)*class, we can get the data detected by accelerometer at regular intervals or we can poll for them periodically. For this game we will use the later approach and get the data at regular intervals.*

*In the first two lines, we define the starting point and the end point of the animation. Then we create a CABasicAnimation object by specifying that we want to move along the y axis (animationWithKeyPath), from the origin.y point (fromValue) to the target.y point (toValue). We also set the duration of animation to 2 seconds (duration). In other word, the ghost image should come back to its original position once it has reached the target point (autoreverses). Finally, we want the animation to repeat forever, so we set the repeatCount as HUGE\_VALF, that is a fancy way to say it should repeat forever.*

*currentPoint holds the current position of the pacman*

*previousPoint holds the position from where the pacman has been moved*

*pacmanXVelocity contains the X component of the velocity (remember that velocity is a vector), and pacmanYVelocity the Y component*

*angle is the current angle of pacman, since the sprite will rotate, as we will explain later on*

*acceleration is the current acceleration measured by the accelerometer*

*motionManager is a queue that helps us to receive and process the data sent from the accelerometer*

*lastUpdateTime allows us to control how long has been since the last call from the accelerometer*

*Core Graphics provides the function*[*CGRectIntersectsRect*](https://developer.apple.com/library/mac/#documentation/graphicsimaging/reference/CGGeometry/Reference/reference.html)*to help us detect whether one view’s frames overlaps another specified view’s frame*

*CGRectIntersectsRect function is used to determine whether the frames of two views are overlapped. have to do is to get the current presentationLayer of the ghost that represents the state of the layer as it currently appears on screen. In other words, we get the current position of the ghost and use that frame to check if it overlaps with the pacman’s frame.*