// design the game

use nine pictures // other methods exists

imageView ⇒ attribute ⇒ kenny

copy and paste 9 times

time label score highest score

|-| ⇒ reset to recommend constrin

// the code

Timer

Alert

Gesture Recognizer

User defaults

```
import UIKit
lass ViewController: UIViewController{
    // variables
    var score = 0
    var timer = Timer()
    var counter = 0
    var kennyArray = [UIImageView]()
    var hideTimer = Timer()
    var highScore = 0
    // views
    @IBOutlet weak var timeLabel: UILabel!
    ↓ // for all nine pics and other labels

    override func viewDidLoad(){
        super.viewDidLoad()

        scoreLabel.text = "Score: \(score)"
        // Highscore check
        let storedHighScore = UserDefaults.standard.object(forKey: "highscore")
        if storedHighScore == nil{
            highScore = 0
            highScoreLabel.text = "Highscore: \(highScore)"
        }
        if let newScore = storedHighScore as? Int{
            highScore = newScore
            highScoreLabel.text = "Highscore: \(highScore)"
        }
        // images
        kenny1.isUserInteractionEnabled = true
        ↓ // nine similar ones
        let recognizer1 = UITapGestureRecognizer(target: self, action: #selector(increaseScore))
        ↓ // nine similar ones
        kenny1.addGestureRecognizer(recognizer1)
        ↓ // nine similar ones
        kennyArray = [kenny1, kenny2, kenny3 ..... kenny9]
        // Timers
        counter = 10
        timeLabel.text = "\(counter)" // or ⇒ String(counter)
        timer = Timer.scheduledTimer(timeInterval: 1, target: self, selector: #selector(countDown), userInfo: nil, repeats: true)
        hideTimer = Timer.scheduledTimer(timeInterval: 0.5, target: self, selector: #selector(hideKenny), userInfo: nil, repeats: true)
        hideKenny()
    }
    @objc func hideKenny(){

        for kenny in kennyArray{
            kenny.isHidden = true
        }
        let random = Int(arc4random_uniform(UInt32(kennyArray.count - 1)))
        kennyArray[random].isHidden = false
    }
```

```
@objc func increaseScore(){
    score += 1
    scoreLabel.text = "Score: \(score)"
}
@objc func countDown(){
    counter -= 1
    timeLaber = String(counter)

    if counter == 0{
    timer.invalidate()
    for kenny in kennyArray{
      kenny.isHidden = true
    }
    //HighScore
    if self.score > self.highScore{
        self.highscore = self.score
        highscoreLabel.text = "High score: \(self.highscore)"
        UserDefaults.standard.set(self.highScore, forKey: "highscore")
    }
    //Alert
    let alert = UIAlertController(title: "Time's Up", message: "Do you want to play again?", preferedStyle: UIAlertController.Style.alert)
    let okButton = UIAlertAction(title: "OK", style: UIAlertAction.Style.cancel, handler: nil)
    let replayButton = UIAlertAction(title: "Replay", style: UIAlertAction.Style.default){
        (UIAlertAction) in
        //replay function
        self.score = 0
        self.scoreLabel.text = "Score: \(self.score)"
        self.counter = 10
        self.timeLabel.text = String(self.counter)
        self.timer = Timer.scheduledTimer(timeInterval:1, target: self, selector: #selector(self.countDown), userInfo: nil, repeats: true)
        self.hideTimer = Timer.scheduledTimer(timeInterval: 0.5, target: self, selector: #selector(self.hideKenny),
                                              userInfo: nil, repeats: true)

        }
        alert.addAction(okButton)
        alert.addAction(replayButton)
        self.present(alert, animated: true, completion: nil)
        }
    }
}
```