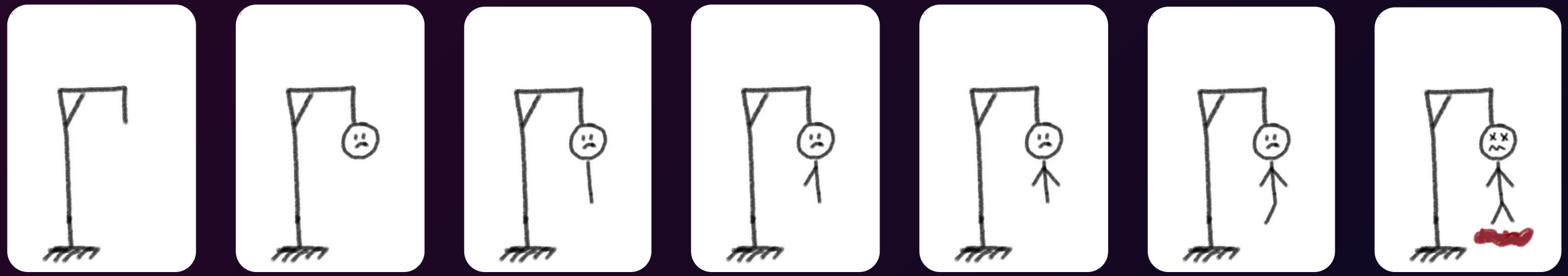
Hangman
Adam Asmaca

Hilal Dedek - 1211602069

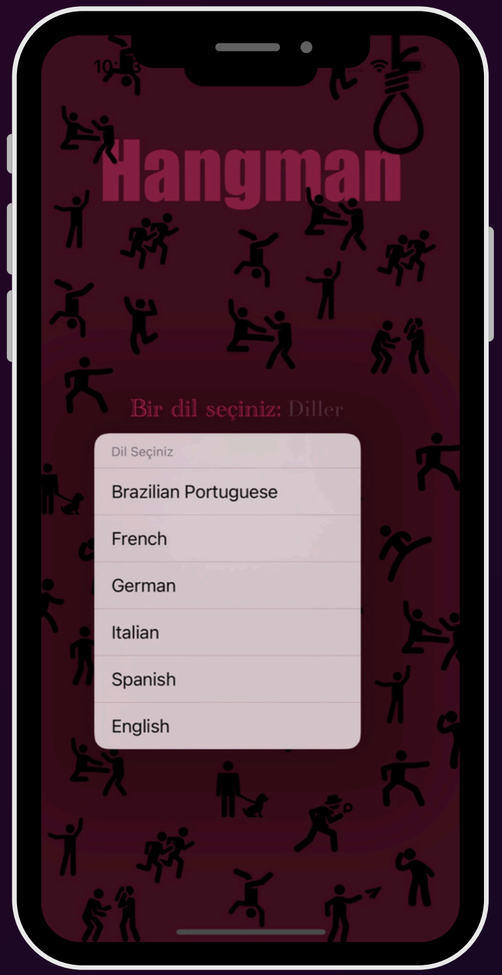


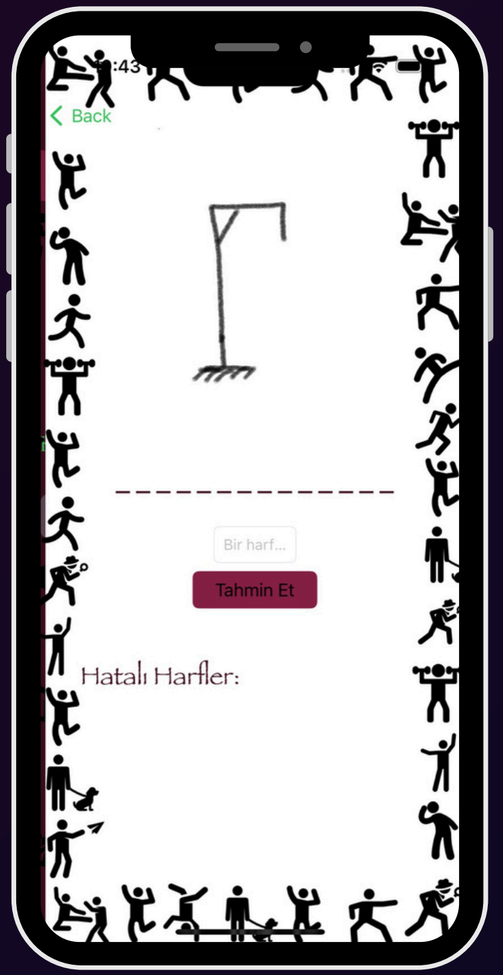
01 Uygulama açıldığında kullanıcıların gördüğü ilk ekran

02 Kullanıcılar “Diller” yazan açılır menüden bir dil seçerler.

03 Dil seçildi. “Oyuna Başla” buttonuna tıklanıldı.

04 İkinci sayfaya geçildi. Seçilen dilde random bir kelime ekrana geldi.



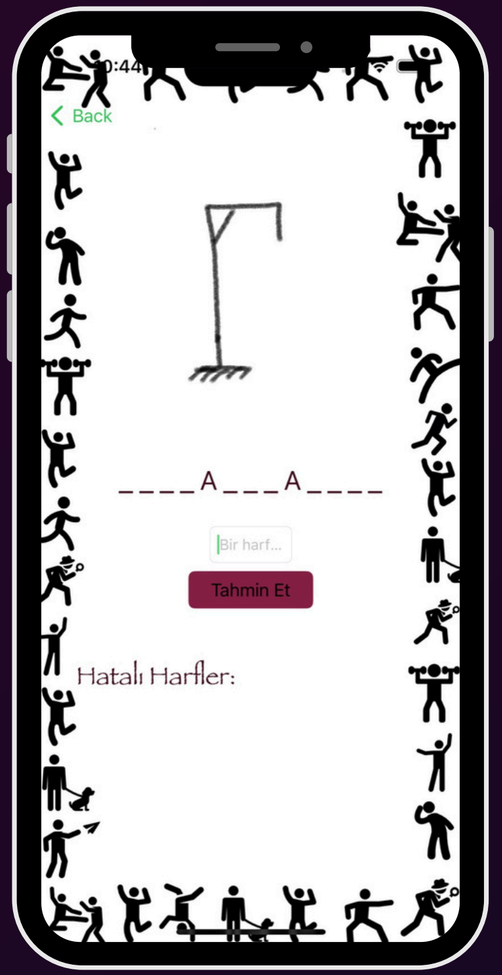


05

Tahmin edilen harf inputa girildi. Girilen harf “A”

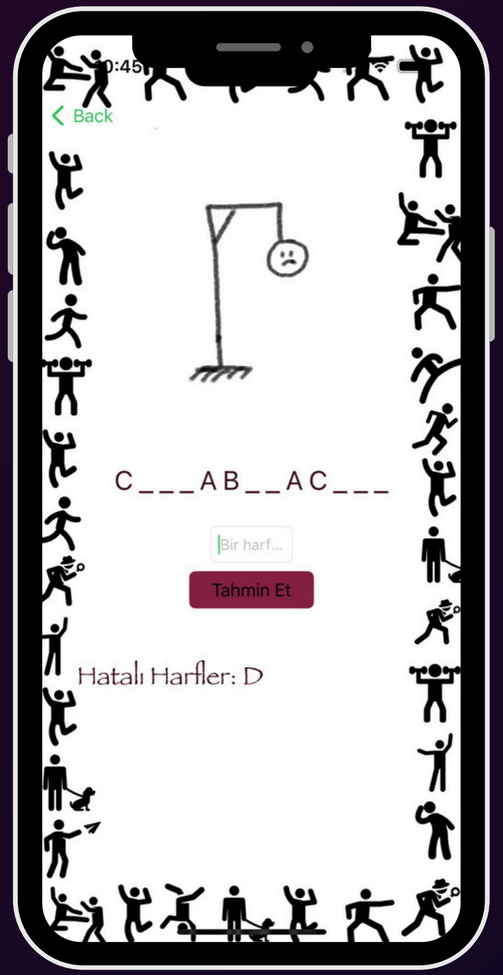
06

“A” harfi kelimede bulunduğu için ilgili yerlere “A” harfi yazıldı.



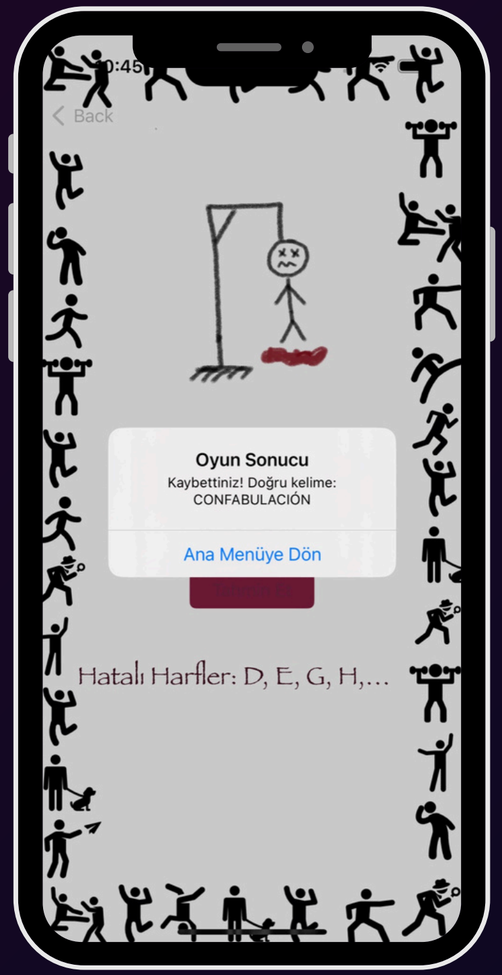
07

“D” harfi kelimede bulunmadığı için Hatalı harflere yazıldı. Ve adamın kafası asıldı.



08

Kullanıcı 5 hata hakkının hepsini kullandı. 6. hatada adam asıldı. Oyun kaybedildi.




Main Menu View Controller
View Controller
Title
Navigation Controller
HANGMAN
Bir dil seçiniz: Diller
Seçilen Dil:
Oyuna Başla
Back
Harf
Tahmin Et
Hatalı Harfler:

Storyboard

MainMenuViewController (ilk sayfa)

```swift
func fetchRandomWord(withLanguageCode languageCode: String?, completion: @escaping (String?) -> Void) {
    var apiUrlString = "https://random-word-api.herokuapp.com/word"
    if let code = languageCode, !code.isEmpty {
        apiUrlString += "?lang=\(code)"
    }
    if let url = URL(string: apiUrlString) {
        URLSession.shared.dataTask(with: url) { data, response, error in
            if let error = error {
                print("API isteği hatası: \(error)")
                completion(nil)
                return
            }
            guard let httpResponse = response as? HTTPURLResponse, (200...299).contains(httpResponse.statusCode)
                else {
                print("Geçersiz HTTP yanıtı")
                completion(nil)
                return
            }
            if let data = data {
                do {
                    if let jsonResult = try JSONSerialization.jsonObject(with: data, options: []) as? [String],
                        let randomWord = jsonResult.first?.uppercased() {
                        completion(randomWord)
                        return
                    }
                    print("Beklenmeyen JSON formatı")
                    completion(nil)
                } catch {
                    print("JSON ayrıştırma hatası: \(error)")
                    completion(nil)
                }
```

Seçilen dile göre random kelimenin api’den çekildiği kod parçası

MainMenuViewController (ilk sayfa)

```swift
@IBAction func startGameButtonTapped(_ sender: UIButton) {
    guard let selectedLanguageCode = self.selectedLanguageCode else {
        languageInstructionLabel.text = "Lütfen önce bir dil seçin."
        languageInstructionLabel.textColor = UIColor.systemRed
        return
    }

    languageInstructionLabel.text = "Kelime yükleniyor..."
    startGameButton.isEnabled = false
    updateStartButtonAppearance()
    fetchRandomWord(withLanguageCode: selectedLanguageCode) { randomWord in
        DispatchQueue.main.async {
            if let word = randomWord {
                self.performSegue(withIdentifier: "goToGameScene", sender: word)
            } else {
                self.languageInstructionLabel.text = "Kelime yüklenirken hata oluştu."
                self.startGameButton.isEnabled = true
                self.updateStartButtonAppearance()
            }
        }
    }
}
```

“Oyuna Başla” butonuna tıklanıldığında tetiklenen fonksiyon

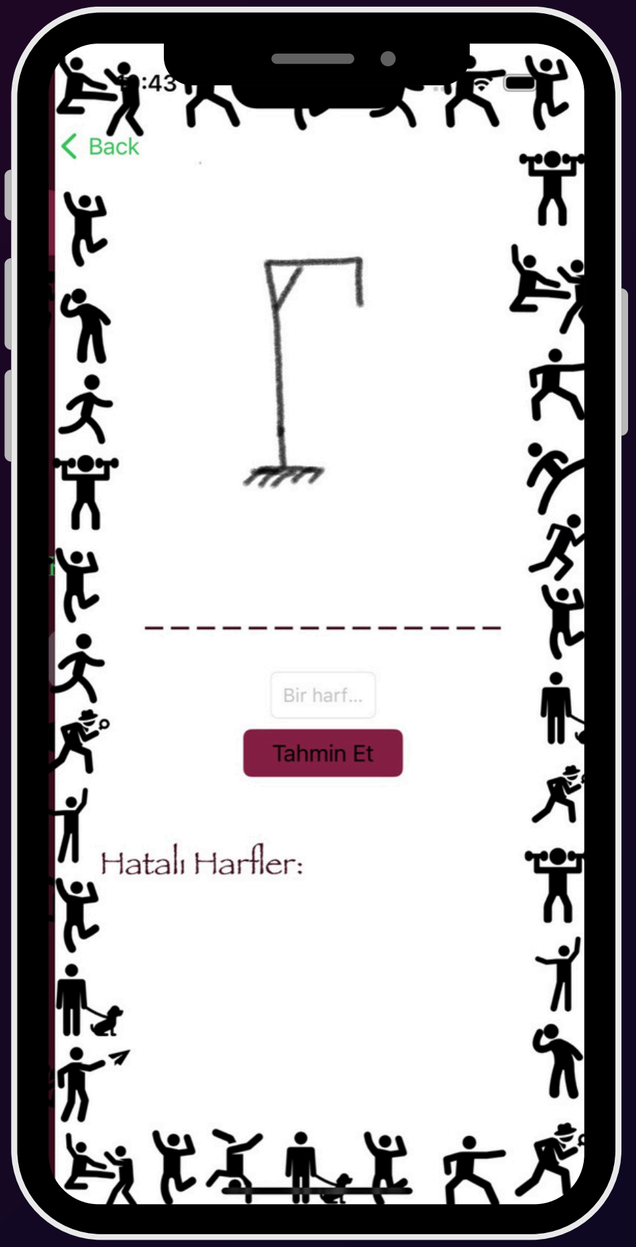
```swift
// MARK: - Navigation

override func prepare(for segue: UIStoryboardSegue, sender: Any?) {
    if segue.identifier == "goToGameScene" {
        if let gameVC = segue.destination as? ViewController, let word = sender as? String {
            gameVC.currentLanguage = selectedLanguage
            gameVC.currentWordSet = word
            print("MainMenuViewController -> ViewController: Kelime gönderildi: \(word), Dil: \(String(describing:
                selectedLanguage))")
        } else {
            print("HATA: ViewController'a geçiş başarısız!")
        }
    }
}
```

Sayfalar arası geçişi sağlayan fonksiyon

ViewController (ikinci sayfa)

```swift
@IBAction func guessButtonTapped(_ sender: UIButton) {
    guard let guessedText = guessTextField.text?.uppercased(),
          !guessedText.isEmpty,
          guessedText.count == 1,
          let guessedLetter = guessedText.first,
          guessedLetter.isLetter else {
        guessTextField.text = ""
        return
    }
    processGuessedLetter(guessedLetter)
    guessTextField.text = ""
}
```

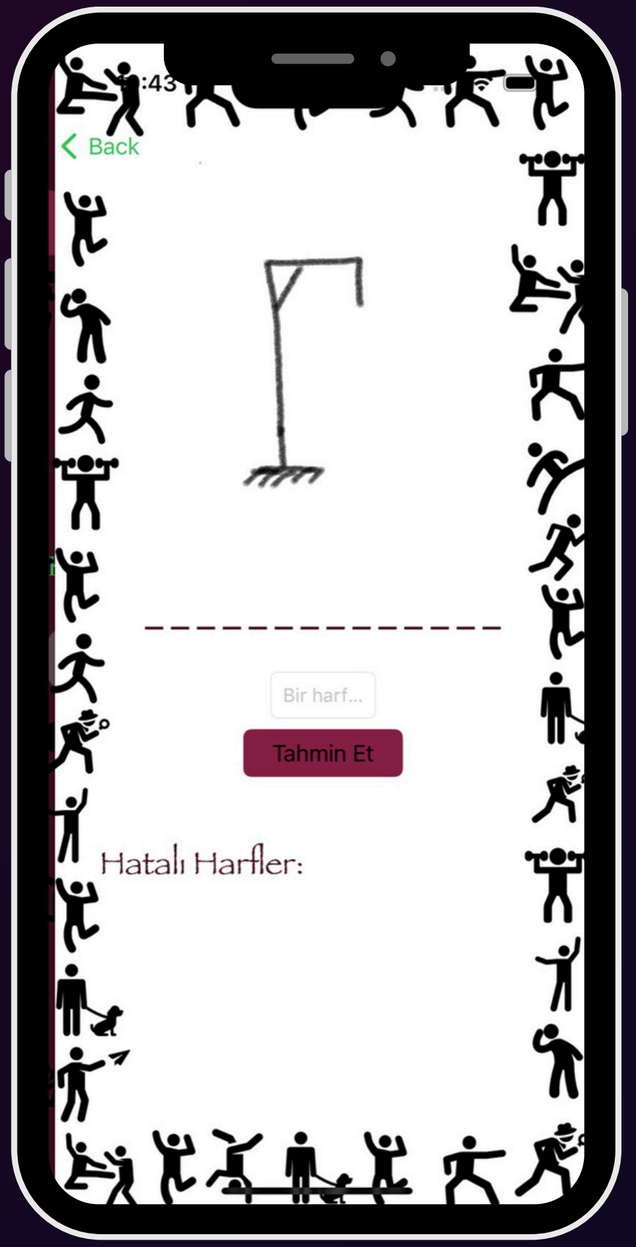
“Tahmin Et” butonuna tıklanıldığında tetiklenecek fonksiyon



ViewController (ikinci sayfa)

```swift
func processGuessedLetter(_ guessedLetter: Character) {
    if currentWord.contains(guessedLetter) {
        for (index, letter) in currentWord.enumerated() {
            if letter == guessedLetter {
                guessedLetters[index] = guessedLetter
            }
        }
        updateWordLabel()
        if !guessedLetters.contains("_") {
            showAlert(message: "Tebrikler! Kelimeyi buldunuz: \(currentWord)")
            guessButton.isEnabled = false
        }
    } else {
        if !wrongLetters.contains(guessedLetter) {
            wrongGuessesRemaining -= 1
            wrongLetters.append(guessedLetter)
            updateHangmanImage()
            updateWrongLettersDisplay()
            if wrongGuessesRemaining == 0 {
                showAlert(message: "Kaybettiniz! Doğru kelime: \(currentWord)")
                guessButton.isEnabled = false
            }
        }
    }
}
```

Tahmin edilen harf kelimede var mı? kontrolü



Beni dinlediğiniz için teşekkür ederim.