

HAZIRLAYANLAR

Adı: Onur Hüseyin

Soyadı: Çantay

Öğrenci NO :152802022

Adı: Bertuğ

Soyadı: Yılmaz

Öğrenci NO : 142905009

AMAÇ

- Ekran üzerinde ki boş alanlara tıklayarak oluşturulan labirentte boş kalan 2 bölgeye fare ve peyniri yerleştirerek farenin tek tek gidebileceği yolları deneyerek hedef konum olan peynire ulaştırmaya çalışmak

GIRDILER

-Seçilen Duvarlar dışarıdan girdi olarak alanıdır kullanıcı seçtiği duvarı tekrar seçerek iptal edebilecektir.

MANTIK

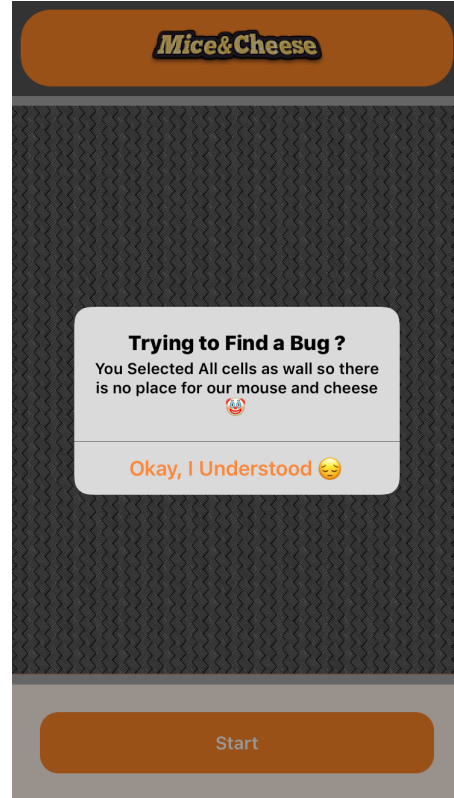
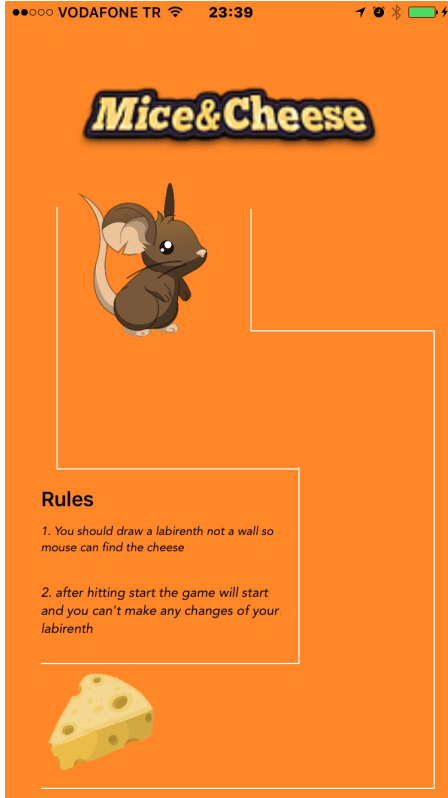
-Girilen duvar sayılarını sayarak olası hatalar engellenerek, random sayılar oluşturulur bu sayılar aynı olmadığı sürece ve duvar a denk gelmediği taktirde atama işlemi gerçekleşir ve bu sayılar Collectionview'ın Cell indisine atanır.Böylelikle Fare ve Peynirin konumu hazırlanmış olur. Daha sonra Farenin mevcut konumunda etrafında ki olası pozisyonları kontrol ederek gitmek istediği yönü doğrulayacaktır ve eğer uygunsa bu yone yönelecektir ardından. Bu işlemler fare peyniri bulana kadar tekrar edecektir.

TEST CASES

Risk Level	Yüksek
Purpose	Bu test durumunda kullanıcının seçtiği duvar miktarı test edilmektedir.
Inputs	Seçilen duvar miktarına bağlı olarak yerleştirme işlemi yapılması
Pass Criterias	Fare ve Peynirin ekranda gerekli yere yerleştirilmesi
Fail Criterias	Tüm ekranı duvarlar ile seçerek farenin ve peynirin yerleşememesi
Test Procedure	Test Kullanıcısı , Yazılımı desteklenen sistem ve cihazlarda bunlar (Apple Cihazlar-> Macbook) çalıştırarak belirtilen girdilerle testi gerçekleştirmelidir. Test işlemini tamamladıktan sonra test sonucunu Pass /Fail olarak belirterek nedenleri ile raporlamalıdır.

Risk Level	Yüksek
Purpose	Bu test durumunda kullanıcının seçtiği duvar miktarına bağlı olarak farenin etrafının çevrili olup olmadığı test edilecektir
Inputs	Seçilen duvar miktarına bağlı olarak etrafının kontrolünün yapılması
Pass Criterias	Fare ve Peynirin ekranda gerekli yere yerleştirilmesi
Fail Criterias	Farenin konumunun etrafının duvarlar ile kapatılmış olması ve hareket edememesi
Test Procedure	Test Kullanıcısı , Yazılımı desteklenen sistem ve cihazlarda bunlar (Apple Cihazlar-> Macbook) çalıştırarak belirtilen girdilerle testi gerçekleştirmelidir. Test işlemini tamamladıktan sonra test sonucunu Pass /Fail olarak belirterek nedenleri ile raporlamalıdır.

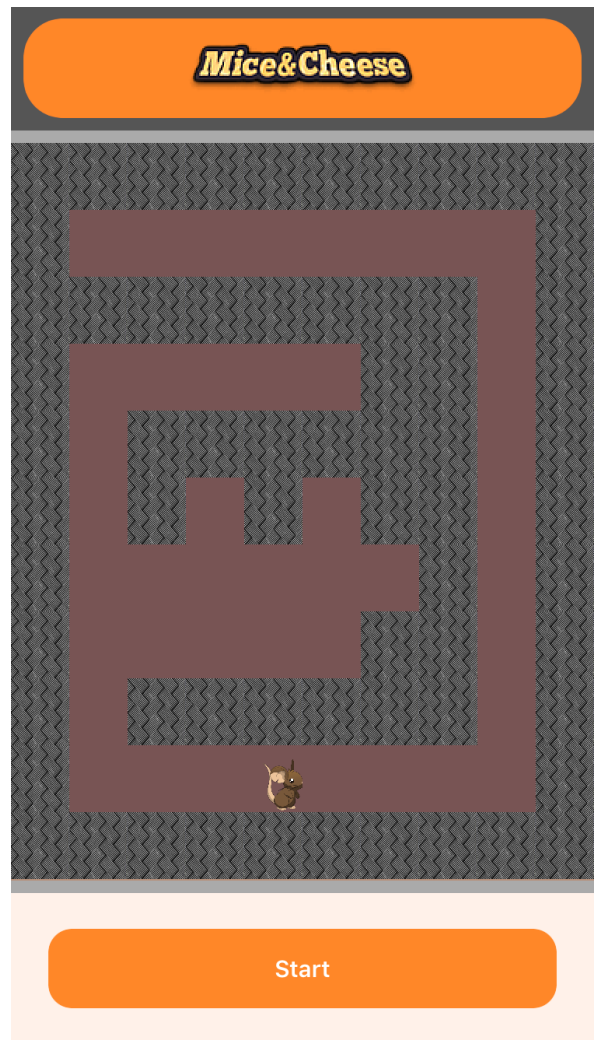
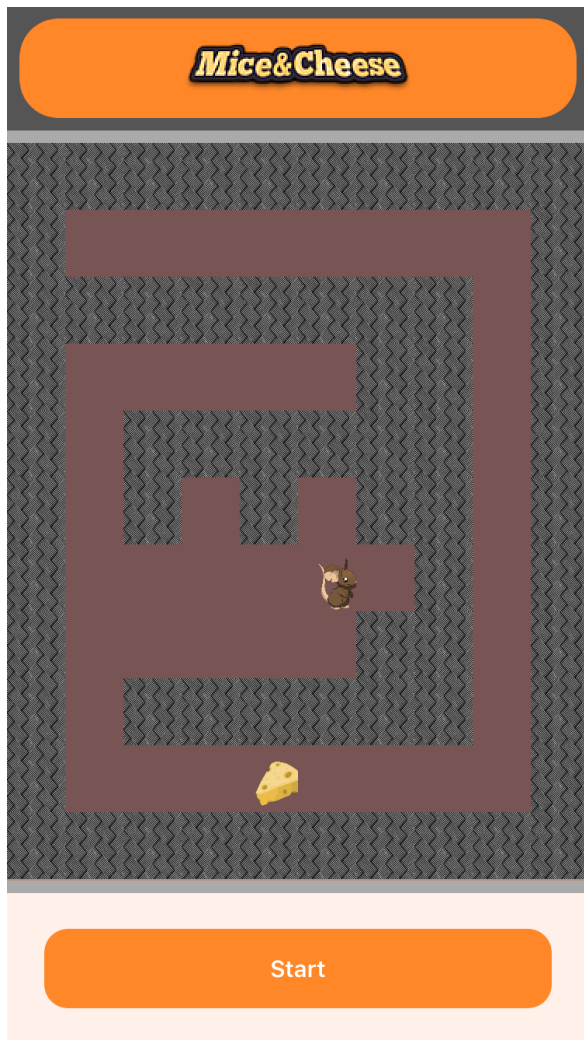
PROGRAM KODU



```

1 //
2 // ViewController.swift
3 // Mice&Cheese
4 //
5 // Created by Bertuğ YILMAZ on 20/10/2017.
6 // Copyright © 2017 Bertuğ YILMAZ. All rights reserved.
7
8 import UIKit
9
10 class MainVC: UIViewController {
11
12     var isStart: Bool = false
13     var mousePosition : Int!
14     var selectedWallCounter : Int = 0
15     var cheesePosition : Int!
16     var isValidPosition : Bool!
17     var mouseCell : CollectionCell!
18     var cheeseCell : CollectionCell!
19     var pug: [Int] = []
20     var repeatCount: [Int] = []
21
22     @IBOutlet weak var collectionView: UICollectionView!
23     @IBOutlet weak var startButton: UIButton!
24
25     override func viewDidLoad() {
26         super.viewDidLoad()
27
28         self.configureCollectionView()
29
30         self.startButton.addTarget(self, action: #selector(self.startButtonAction(_:)), for: .touchUpInside)
31
32     }
33
34     override func viewWillAppear(_ animated: Bool) {
35         super.viewWillAppear(animated)
36         self.createWall()
37     }
38
39     override var prefersStatusBarHidden: Bool {
40         return true
41     }
42
43     func startButtonAction(_ sender: UIButton) {
44         self.isStart = !isStart
45         createWall()
46         generateLocationForMouseAndCheese()
47         sender.isEnabled = false
48     }
49
50     func generateLocationForMouseAndCheese() {
51         repeat{
52             if selectedWallCounter > 100 {
53                 break
54             }
55
56             isValidPosition = false
57             mousePosition = Int(arc4random_uniform(110))
58             cheesePosition = Int(arc4random_uniform(110))
59             mouseCell = self.collectionView.cellForItem(at: IndexPath(row: mousePosition,section: 0)) as! CollectionCell
60             cheeseCell = self.collectionView.cellForItem(at: IndexPath(row: cheesePosition,section: 0)) as! CollectionCell
61
62             if mouseCell.isWall || cheeseCell.isWall{
63                 isValidPosition = false
64             }else{
65                 isValidPosition = true
66             }
67
68         }while(mousePosition == cheesePosition || isValidPosition == false)
69
70         if selectedWallCounter > 100 {
71             let alert = UIAlertController(title: "Trying to Find a Bug ?", message: "You Selected All cells as well so there is no place for our mouse and cheese 🐭", preferredStyle: UIAlertControllerStyle.alert)
72             alert.view.tintColor = UIColor(red:0.99, green:0.53, blue:0.22, alpha:1.0)
73             alert.addAction(UIAlertAction(title: "Okay, I Understood 🐭", style: UIAlertActionStyle.default, handler: nil))
74             self.present(alert, animated: true, completion: nil)
75
76         }else{
77             self.pug.append(mousePosition) // First step
78             mouseCell.imageView.image = UIImage(named: "Mouse")
79             cheeseCell.imageView.image = UIImage(named: "Cheese")
80             print("Cheese position --> \(cheesePosition)")
81             print("Mouse position --> \(mousePosition)")
82             findThatCheese()
83         }
84     }
85
86     func findThatCheese() {
87         while self.mousePosition != self.cheesePosition {
88             foundPrefferedCell()
89         }
90         onAnimation()
91     }
92
93     func foundPrefferedCell() {
94         var min: [Int] = []
95         var cells = [CollectionCell]()
96
97         if let positionRight = mousePosition + 1 as? Int {
98             if let right = collectionView.cellForItem(at: IndexPath(row: positionRight,section: 0)) as? CollectionCell {
99                 if !right.isWall {
100                     cells.append(right)
101                 }
102             }
103         }
104
105         if let positionLeft = mousePosition - 1 as? Int {
106             if let left = collectionView.cellForItem(at: IndexPath(row: positionLeft,section: 0)) as? CollectionCell {
107                 if !left.isWall {
108                     cells.append(left)
109                 }
110             }
111         }
112
113         if let positionTop = mousePosition - 10 as? Int {
114             if let top = collectionView.cellForItem(at: IndexPath(row: positionTop,section: 0)) as? CollectionCell {
115                 if !top.isWall {
116                     cells.append(top)
117                 }
118             }
119         }
120
121         if let positionBottom = mousePosition + 10 as? Int {
122             if let bottom = collectionView.cellForItem(at: IndexPath(row: positionBottom,section: 0)) as? CollectionCell {
123                 if !bottom.isWall {
124                     cells.append(bottom)
125                 }
126             }
127         }
128
129         var cell : CollectionCell!
130
131         for item in cells{
132             if let count = item?.repeatCount{
133                 min.append(count)
134             }else {
135                 break
136             }
137         }
138
139         for item in cells{
140             if item != nil{
141                 if item?.repeatCount == min.min(){
142                     cell = item
143                 }
144             }else {
145                 break
146             }
147         }
148
149         if let index = self.collectionView.indexPath(for: cell)?.row {
150             self.pug.append(index)
151             self.mousePosition = index
152             print("Mouse path --> \(index)")
153         }
154     }
155
156     // Girilen parametrelere göre cell in imageView ına ya da backgroundView ına image set eder.
157     func setImageOfCell(index: Int, imageName: String, isImageView: Bool) {
158         let dummyCell = self.collectionView.cellForItem(at: IndexPath(row: index,section: 0)) as! CollectionCell
159         if isImageView {
160             dummyCell.imageView.image = UIImage(named: imageName)
161             self.mousePosition = index
162             if index == self.cheesePosition {
163                 dummyCell.foundItCheese = true
164                 return
165             }
166         }else {
167             dummyCell.backgroundColor = UIColor(named: imageName)
168             if imageName == "wireImp"{
169                 dummyCell.isWall = true
170                 dummyCell.tag = WALL
171             }
172         }
173     }
174
175     // Index i baz alarak labirentin kenar bölgelerine image(Duvar) ekler.
176     func createWall(){
177         for index in 0...CELLCOUNT - 1{
178             let strIndex = String(index)
179             let arrIndex = Array(strIndex.characters)
180
181             if arrIndex.last == "0" || arrIndex.last == "9" {
182                 self.setImageOfCell(index: index, imageName: "wireImp", isImageView: false)
183             }
184
185             if arrIndex.count == 1 || arrIndex.count == 3 {
186                 self.setImageOfCell(index: index, imageName: "wireImp", isImageView: false)
187             }
188         }
189     }
190
191     func delay(_ delay:Double?, closure:@escaping ()->()) {
192         let when = DispatchTime.now() + delay
193         DispatchQueue.main.asyncAfter(deadline: when, execute: closure)
194     }
195
196     func onAnimation(){
197         self.delay(2) {
198
199             if self.pug.count != 0{
200                 for index in 1..self.pug.count - 1 {
201                     if let beforeCell = self.collectionView.cellForItem(at: IndexPath(row: self.pug[index],section: 0)) as? CollectionCell {
202                         if let beforeCell = self.collectionView.cellForItem(at: IndexPath(row: self.pug[index - 1],section: 0)) as? CollectionCell {
203                             UIView.transition(with: self.view, duration: 2, options: .transitionCrossDissolve, animations: {
204                                 cell.imageView.image = UIImage(named: "Mouse")
205                                 beforeCell.imageView.image = UIImage(named: "")
206                             }, completion: { (success: Bool) in
207                                 if success {
208                                     //
209                                 }
210                             })
211                         }
212                     }
213                 }
214             }
215         }
216     }
217
218     }
219
220 }
221
222
223
224
225

```



```
Cheese Position --> 94
Mouse Position --> 65
Mouse path --> 75
Mouse path --> 65
Mouse path --> 55
Mouse path --> 65
Mouse path --> 64
Mouse path --> 74
Mouse path --> 73
Mouse path --> 63
Mouse path --> 53
Mouse path --> 63
Mouse path --> 62
Mouse path --> 72
Mouse path --> 71
Mouse path --> 81
Mouse path --> 91
Mouse path --> 92
Mouse path --> 93
Mouse path --> 94
```