

CASSE-BRIQUE

PROJET D'APPLICATION MOBILE EN SWIFT

RAPPEL DU BRIEF

Concevoir une petite application avec au moins deux vues et une navigation entre ces vues. L'une des vues au moins doit être associée à un contrôleur reliée à la vue par des actions et outlets, et programmée en Swift. Créez également une icône personnalisée pour votre application.

OBJECTIF DE DÉPART

Réalisation d'un mini jeu de casse-brique avec la présence de plusieurs niveaux en y intégrant des effets sonores ainsi que des bonus/malus tombant de certaines briques qui, une fois attrapés par la raquette du joueur, viendraient affecter des éléments du niveau.

FONCTIONNALITÉS

- Mémorisation des niveaux débloqués.
- Possibilité de rejouer les niveaux débloqués.
- Mise en pause d'un niveau en cours.
- Compteur de vies.
- Compteur de briques touchées.
- Musique à l'écran de sélection des niveaux.

DIFFICULTÉS RENCONTRÉES

Gestion du comportement de la balle au rebond contre une brique (inverser la valeur de X de la balle si cette dernière touche la brique par la gauche ou la droite, inverser la valeur de Y de la balle si cette dernière touche la brique par le haut ou le bas).

```
1  if balle_haut <= brique_bas && balle_bas >= brique_haut{
2      if balle_gauche <= brique_droite && balle_droite >= brique_gauche{
3          //print("Zone de la brique")
4
5
6          // BAS
7          if balle_haut < brique_bas && balle.center.y > brique_bas {
8
9              print("Brique touchée par le BAS")
10             v.y = -v.y
11
12             // HAUT
13             }
14             else if balle_bas > brique_haut && balle.center.y < brique_haut {
15
16                 print("Brique touchée par le HAUT")
17                 v.y = -v.y
18             }
19
20
21             // DROITE
22             else if balle_gauche < brique_droite && balle.center.x > brique_droite {
23
24                 print("Brique touchée par la DROITE")
25                 v.x = -v.x
26
27                 // GAUCHE
28                 }
29                 else if balle_droite > brique_gauche && balle.center.x < brique_gauche {
30
31                     print("Brique touchée par la GAUCHE")
32                     v.x = -v.x
33                 }
34             }
35             evenement_apres_brique_touchee()
36         }
37     }
```

DIFFICULTÉS RENCONTRÉES

Dissimulation visuelle et physique d'une brique touchée par la balle.

```
1 func evenement_apres_brique_touchee() {
2     print("La brique est touché par la balle")
3     bruitage_brique_touchee!.stop()
4     bruitage_brique_touchee!.currentTime = 0
5     bruitage_brique_touchee!.play()
6
7
8     brique.isHidden = true
9     brique.frame = CGRect(x: 0, y: 0, width: 0, height: 0); view.addSubview(brique)
10    compteur_briques_touchees += 1
11    indicateur_nombre_briques_detruites.text = "\\(compteur_briques_touchees)/\\(briques.count)"
12    print("\\(compteur_briques_touchees)")
13    if compteur_briques_touchees == longueur_tableau_briques {
14        bruitage_niveau_gagne!.play()
15        niveau_reussi = 2
16        print("Toutes les briques ont été touché")
17        fin_du_niveau()
18        print("Fin du niveau")
19    }
20 }
```

DIFFICULTÉS RENCONTRÉES

Mettre correctement fin à une partie afin que le jeu ne continue pas de jouer en fond si l'utilisateur revient sur le menu des niveaux.

```
1 func stopTimer() {
2     print("Fin du timer permettant le fonctionnement de la boucle")
3     t.invalidate()
4 }
5
6 func retirer_balle() {
7     balle.isHidden = true
8     balle.frame = CGRect(x: 100, y: 100, width: 0, height: 0); view.addSubview(balle)
9     print("Balle retirée de l'écran et masquée")
10 }
11
12 func retirer_raquette() {
13     raquette.isHidden = true
14     raquette.frame = CGRect(x: 0, y: 0, width: 0, height: 0); view.addSubview(balle)
15     print("Raquette retirée de l'écran et masquée")
16 }
17
18 func retirer_toutes_les_briques() {
19     for brique in briques {
20         brique.isHidden = true
21         brique.frame = CGRect(x: 0, y: 0, width: 0, height: 0); view.addSubview(brique)
22     }
23 }
24
25 func fin_du_niveau() {
26     bouton_pause.isHidden = true
27     stopTimer()
28     retirer_balle()
29     retirer_raquette()
30     bouton_rejouer.isHidden = false
31 }
```

DIFFICULTÉS RENCONTRÉES

Transmission de données d'une ViewController vers une ViewController antérieure.

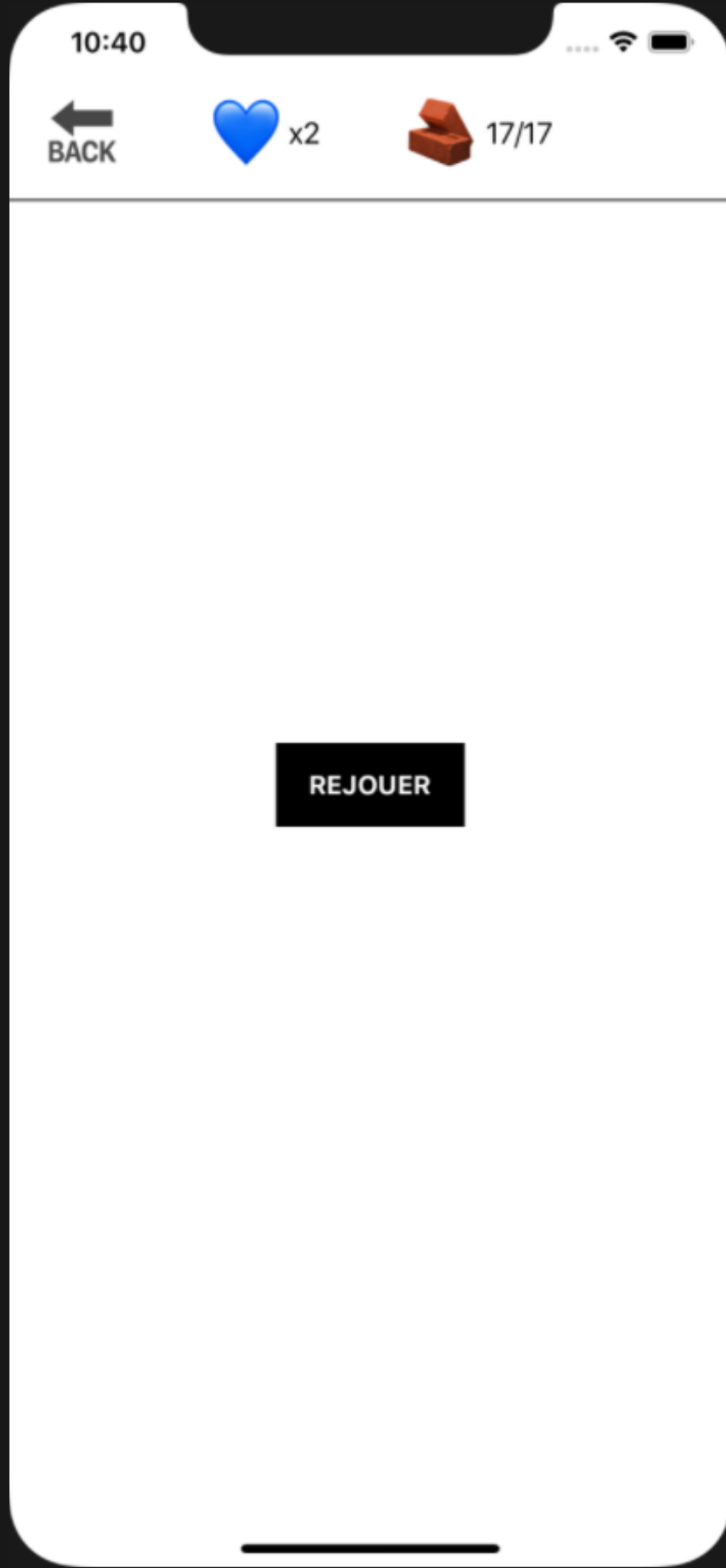
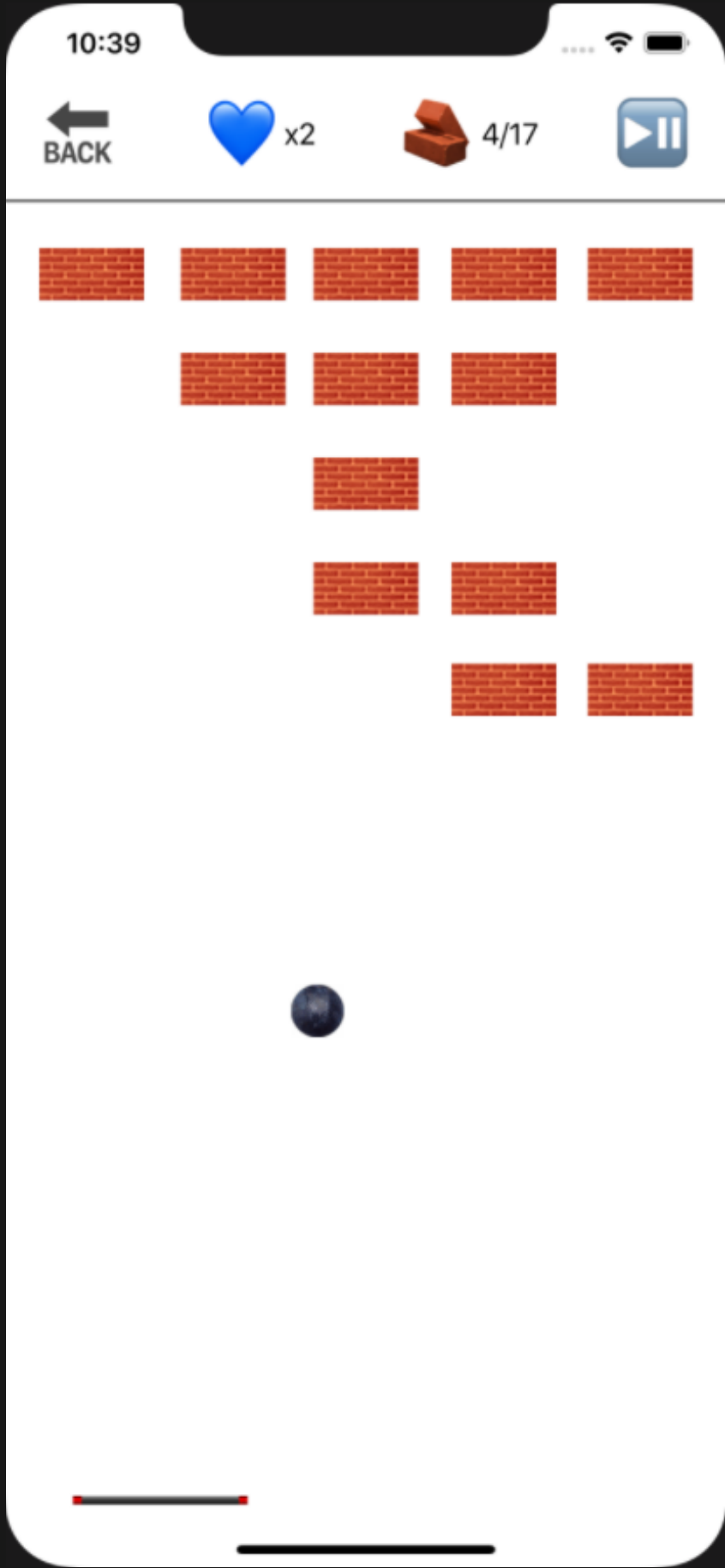
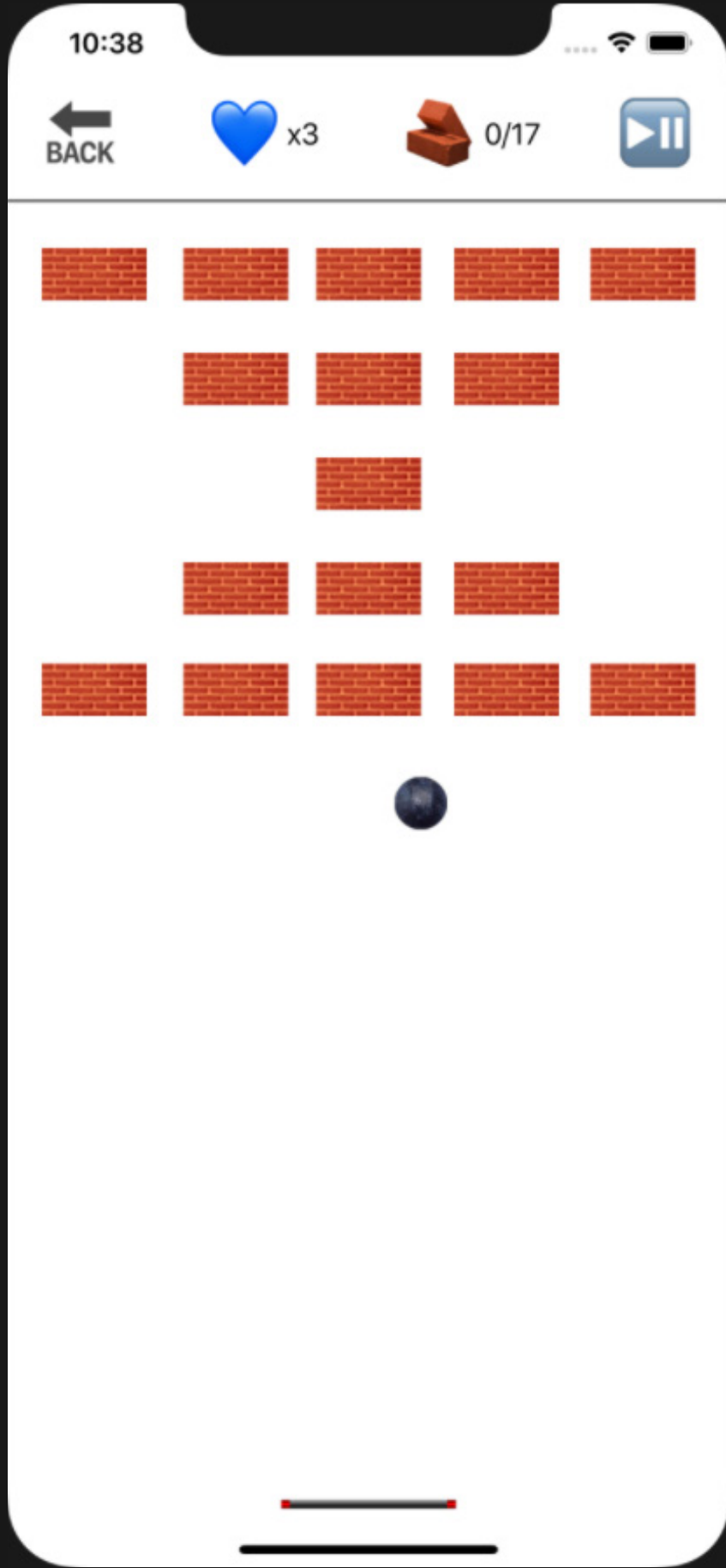
```
1 import AVFoundation
2 import AVKit
3
4 protocol CanRecieve {
5     func passDataBack(data: Int)
6 }
```

```
1 var delegate:CanRecieve?
2
3
4 // revenir au menu
5 @IBAction func revenir_accueil(_ sender: Any) {
6     delegate?.passDataBack(data: niveau_reussi)
7 }
```

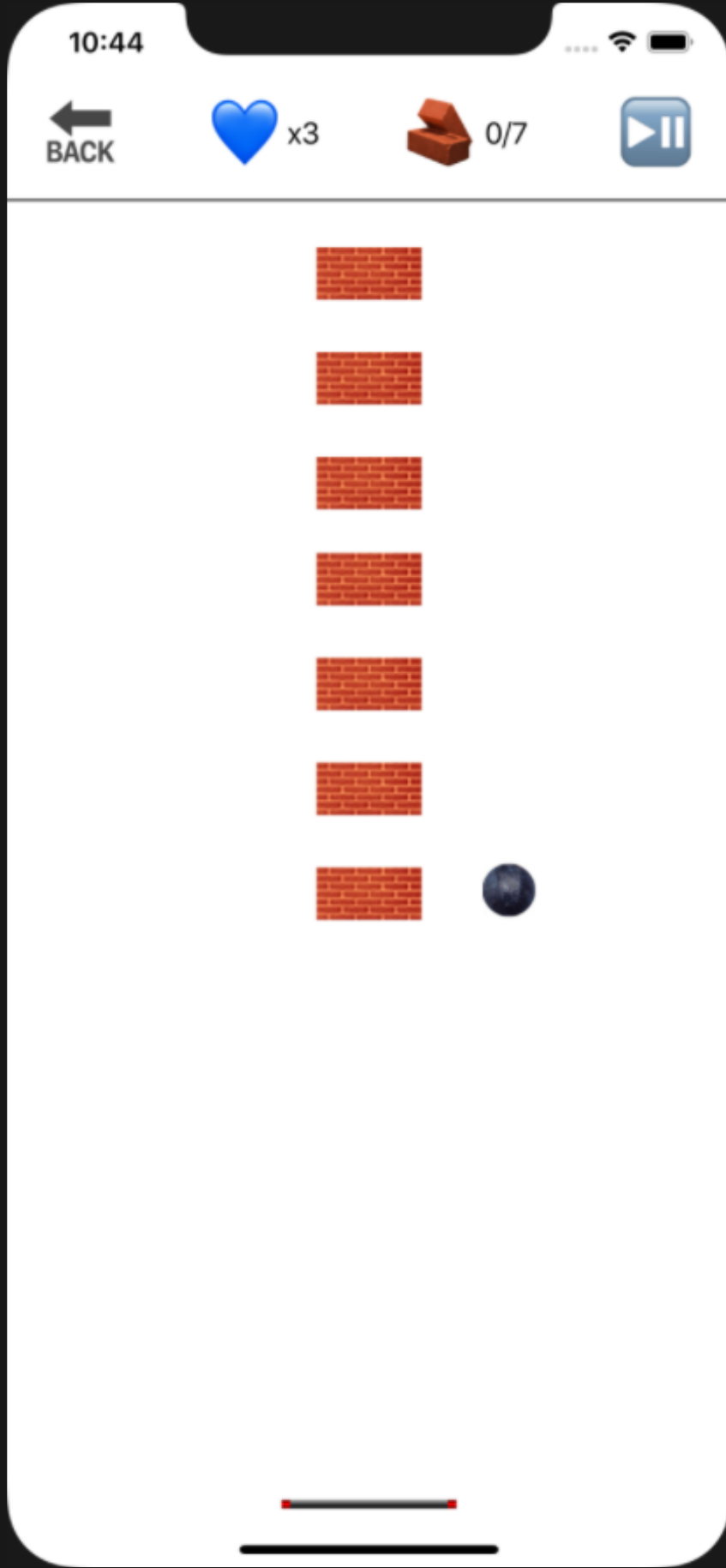
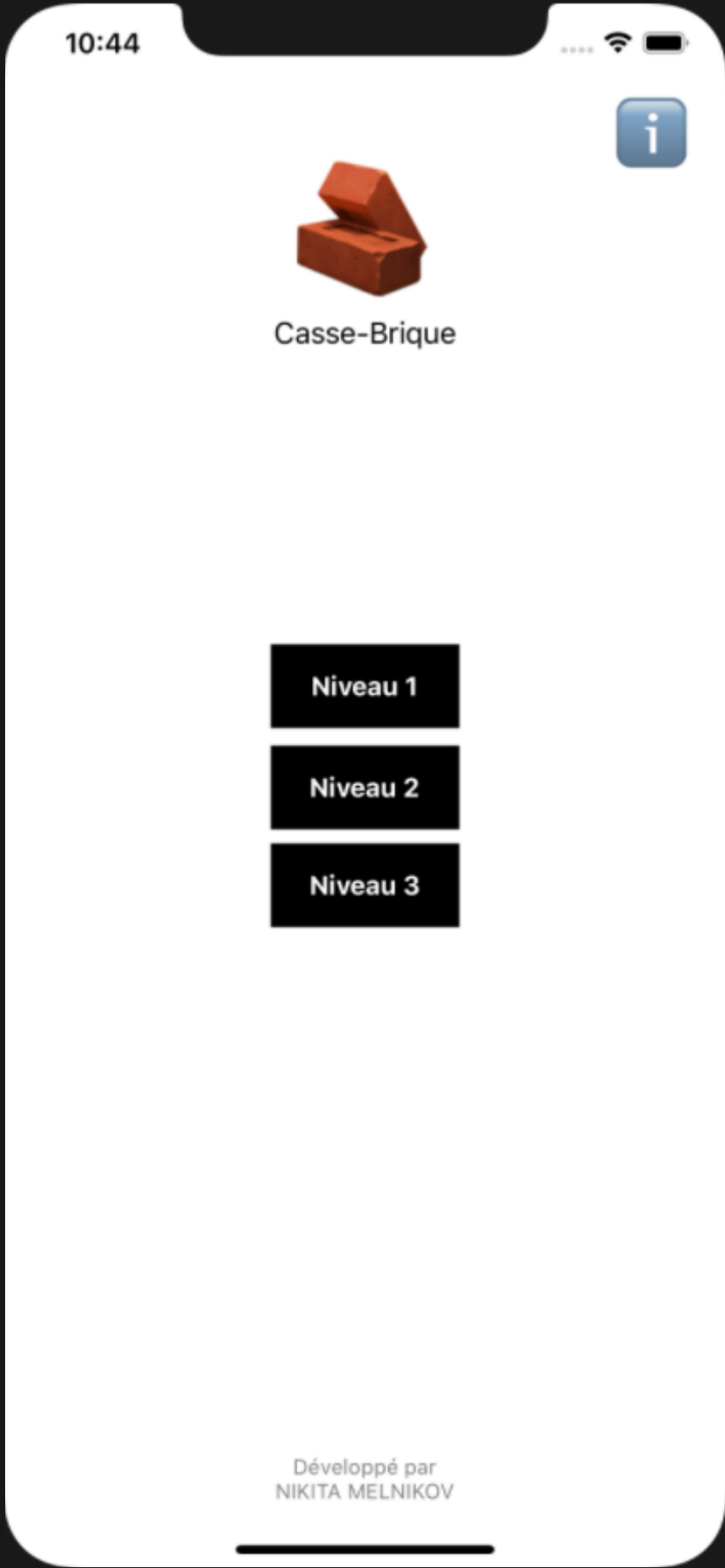
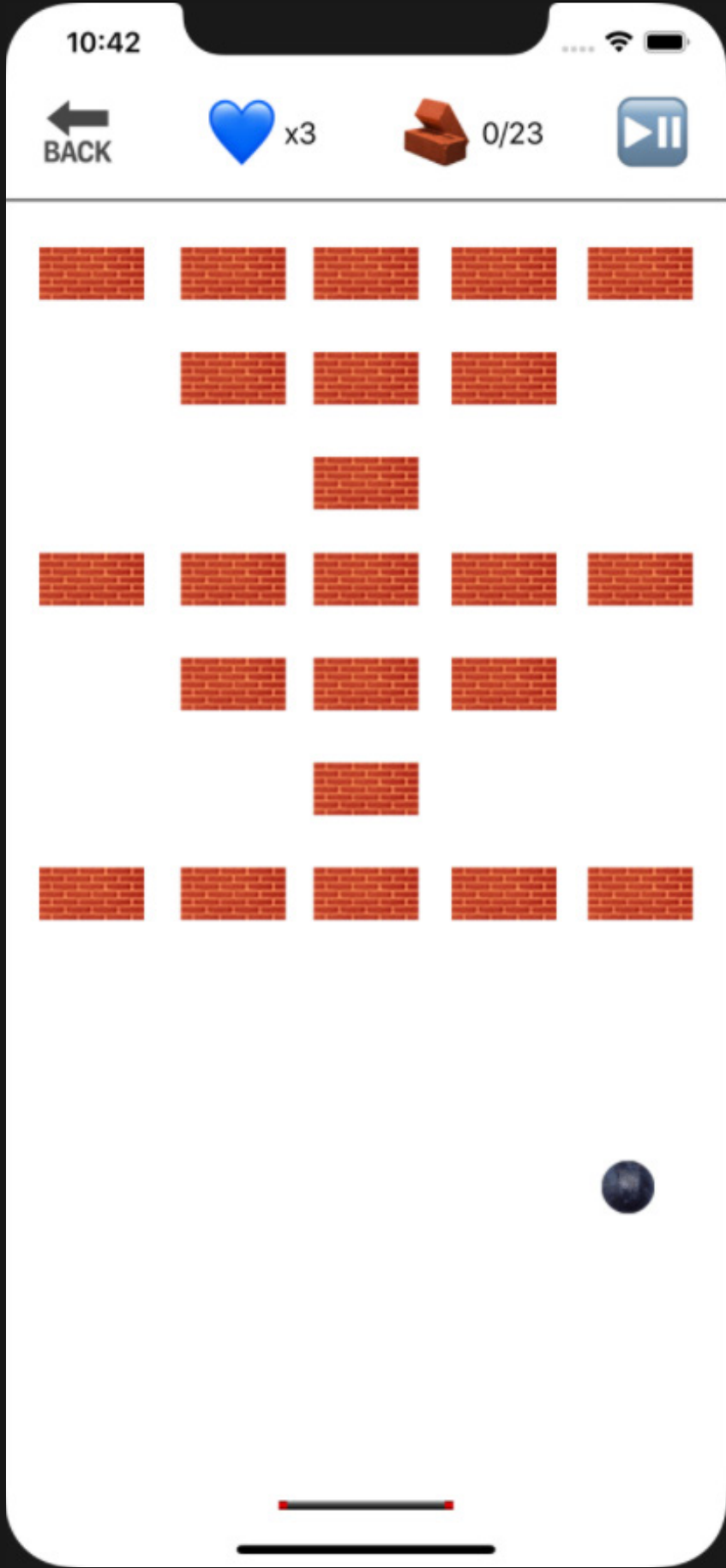
```
1 class NiveauxViewController: UIViewController,CanRecieve, CanRecieve2, CanRecieve3 {
2
3     func passDataBack(data: Int) {
4
5         print(data)
6
7         for bouton_niveau in boutons_niveaux {
8             print(bouton_niveau.tag)
9             if bouton_niveau.tag <= data {
10                 bouton_niveau.setTitle("Niveau \ \(bouton_niveau.tag)", for: .normal)
11                 bouton_niveau.isEnabled = true
12             }
13         }
14     }
```

```
1 override func prepare(for segue: UIStoryboardSegue, sender: Any?) {
2     if segue.identifier == "niveau1_to_niveaux" {
3         let secondVC = segue.destination as! Niveau1ViewController
4         secondVC.delegate = self
5     }
6     if segue.identifier == "niveau2_to_niveaux" {
7         let secondVC = segue.destination as! Niveau2ViewController
8         secondVC.delegate = self
9     }
10    if segue.identifier == "niveau3_to_niveaux" {
11        let secondVC = segue.destination as! Niveau3ViewController
12        secondVC.delegate = self
13    }
14 }
```


COPIES D'ÉCRAN



COPIES D'ÉCRAN



AMÉLIORATIONS POSSIBLES

- Ajout de bonus / malus
- Système de personnalisation de la balle et de la raquette
- Ajout de niveaux supplémentaires
- Modifier la trajectoire de la balle en fonction de sa position sur la raquette
- Diversification des types de briques
- Adapter l'application à d'autres formats que celui de l'iPhone 11 Pro Max

