

# Tasks Information

Tasks in code are highlighted with a „TODO: Task X, Step Y“ - simply search in Xcode via Command + Shift + F for them

There is a tag for each solution in the git repository

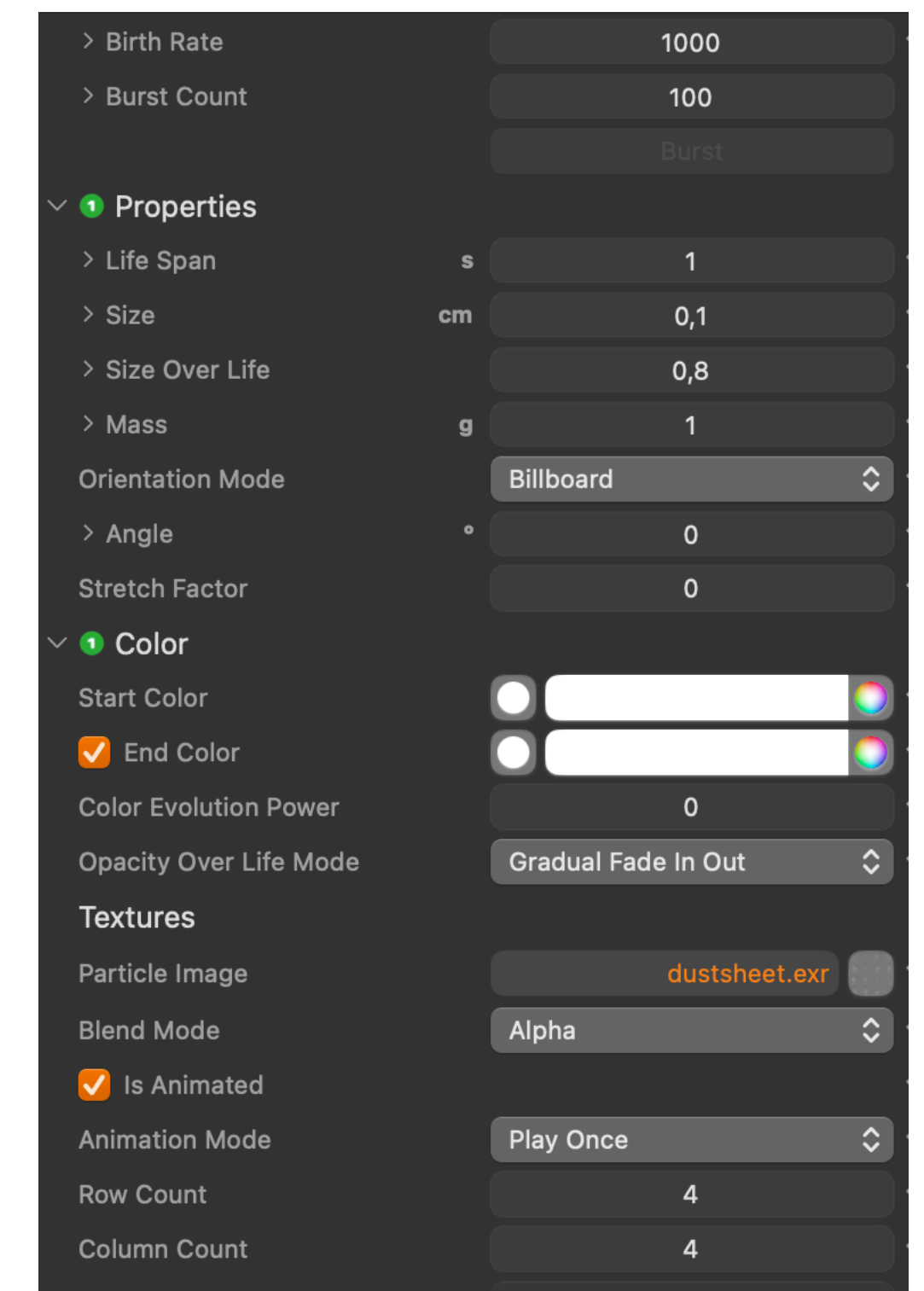
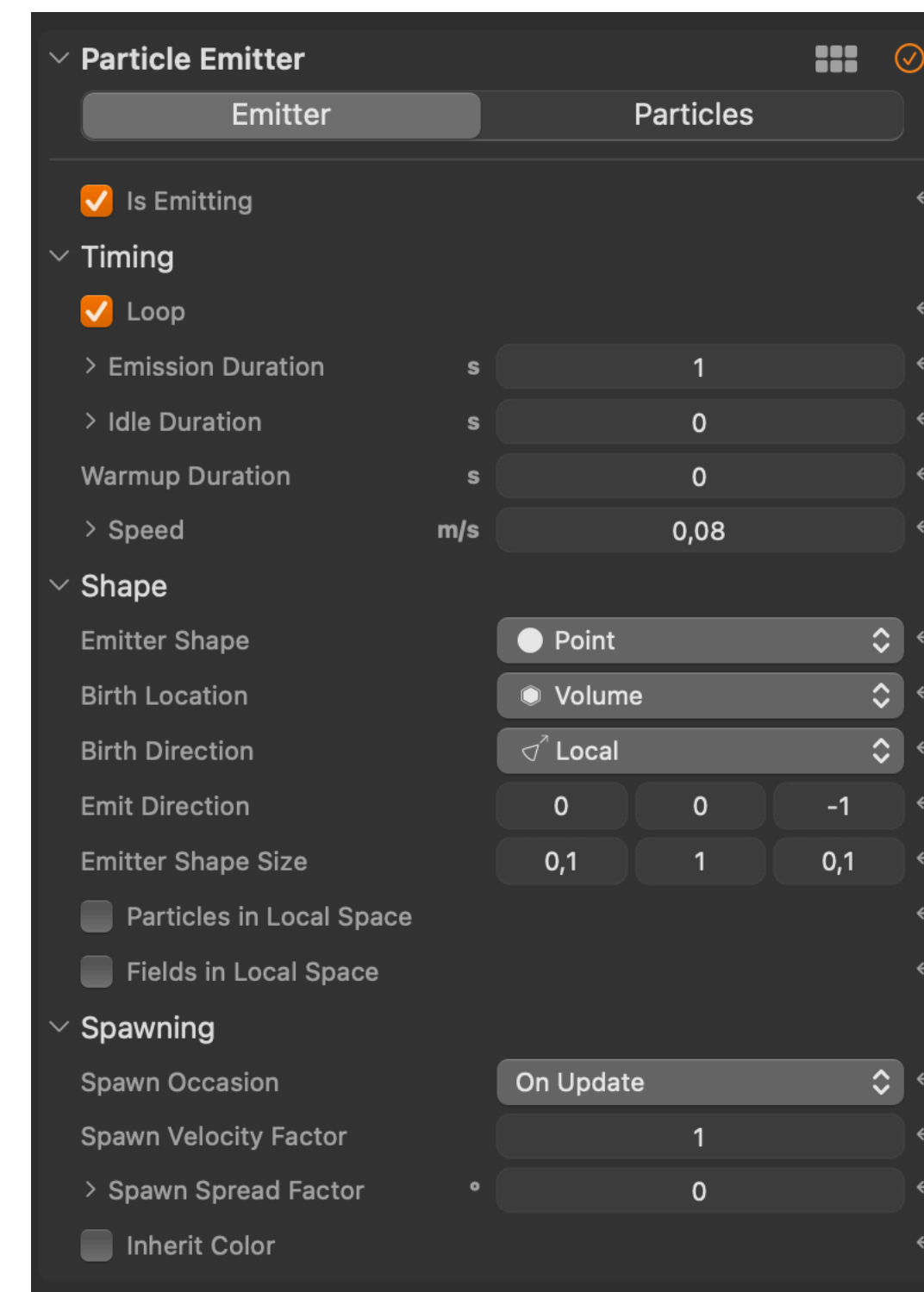
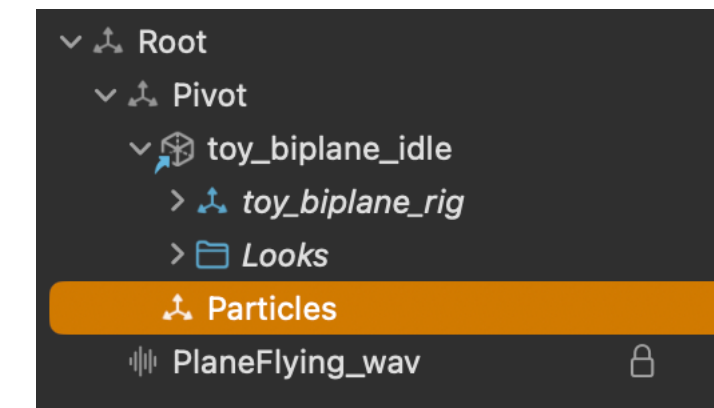
# Task 1 - Customize Scene

1 - Try adding new entities, particles, adapt materials & colors

2 - Add trail particles to enemy

3 - (Optional) Change the texture of the smiley

All tasks above can be done in Reality Composer Pro



# Task 2 - Load 3D scenes

Read yourself a bit into the code how everything works

1 - Load GameScene into app

2 - Load enemy scenes in spawn system

```
import SwiftUI
import RealityKit
import RealityKitContent

struct GameView: View {

    @Environment(ViewModel.self) var viewModel

    var body: some View {
        RealityView { content in
            if let scene = try? await Entity(named: "GameScene", in: realityKitContentBundle) {
                viewModel.root.addChild(scene)
            }

            content.add(viewModel.root)
        }
    }
}
```

Hint: Needs to be loaded from  
realityKitContentBundle

# Task 3 - Handling Components

- 1 - Decrease health of target when bullet collides with it
- 2 - Remove bullet on collision
- 3 - Adapt scale of health bar when health changes
- 4 - Remove entity when health is zero
- 5 - Add health component to Enemy in RCP
- 6 - Reduce health of enemy by 50 when it gets tapped
- 7 - Stop game when target health is zero
- 8 - Increase score by 10 when enemy health is zero

```
var hasHealth = entity.components.has(HealthComponent.self)

// -----

entity.components.set(HealthComponent())
entity.components[HealthComponent.self] = HealthComponent()

// -----

entity.components[HealthComponent.self]?.health -= 5

// ...

if let healthComponent = entity.components[HealthComponent.self] {
    // Modify healthcomponent
    // ...

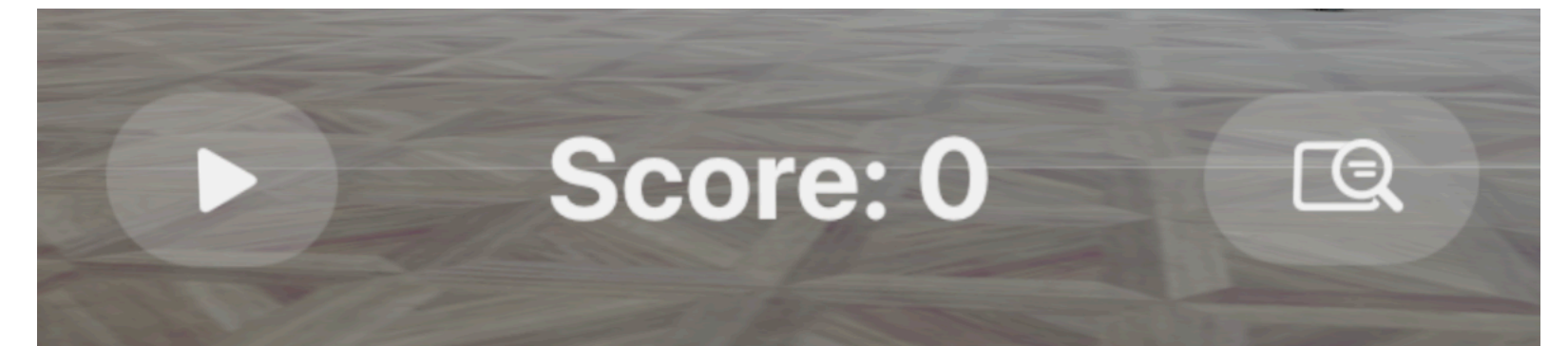
    entity.components[HealthComponent.self] = healthComponent
}

// -----

entity.components.remove(HealthComponent.self)
```

# Task 4 - Add UI

- 1 - Remove automatic game start
- 2 - Add a button to start / stop the game
- 3 - Add a text view to show the current score



Hint: Extend existing MenuView