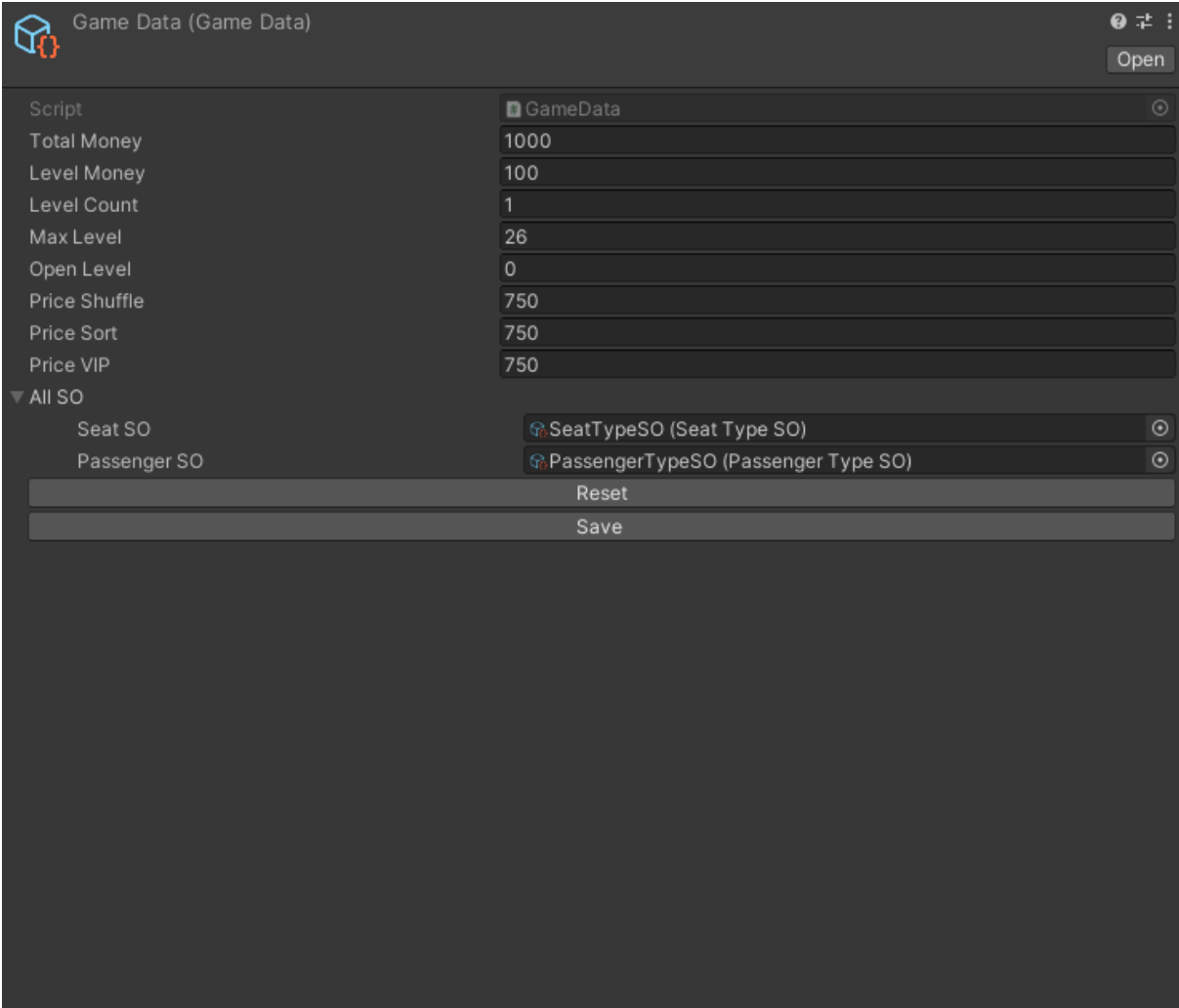


Seat Sorting Template Game

▼ Game Data

\Assets\SeatSorting_Files\Data
GameData.asset

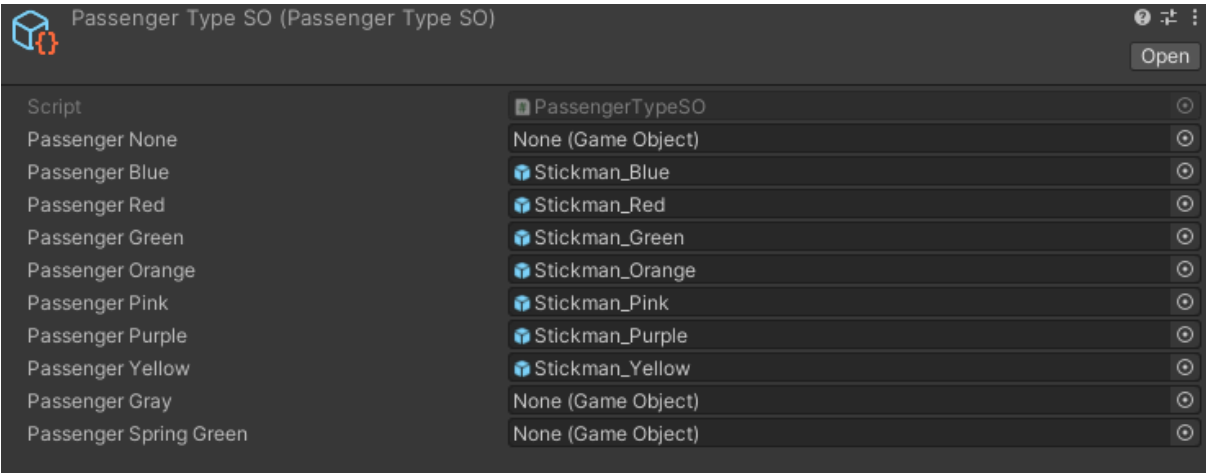


- **Total Money:** Determines the total money in the game. You can change the total money from here.
 - **Level Money:** Determines the earnings per level. You can modify the amount from here.
 - **Level Count:** Enter the level number you want to start. Then, open the corresponding scene and start the game.
 - **Max Level:** Specifies the total number of levels in the project. Don't forget to increase this value when adding a new level.
 - **Open Level:** Must be the same as Level Count. Otherwise, the level won't start properly.
-
- **Price Shuffle:** Determines the price of the Shuffle boost. You can change the price from here.
 - **Price Sort:** Determines the price of the Sort boost. You can change the price from here.
 - **Price VIP:** Determines the price of the VIP boost. You can change the price from here.

- **Seat SO:** Make sure *“SeatTypeSO”* is selected.
 - **Passenger SO:** Make sure *“PassengerTypeSO”* is selected.
-
- **Reset:** If you want to reset the Game Data, press the **“Reset”** button to reset all values.
 - **Save:** Press the **“Save”** button to save your changes.

▼ Passenger Type SO

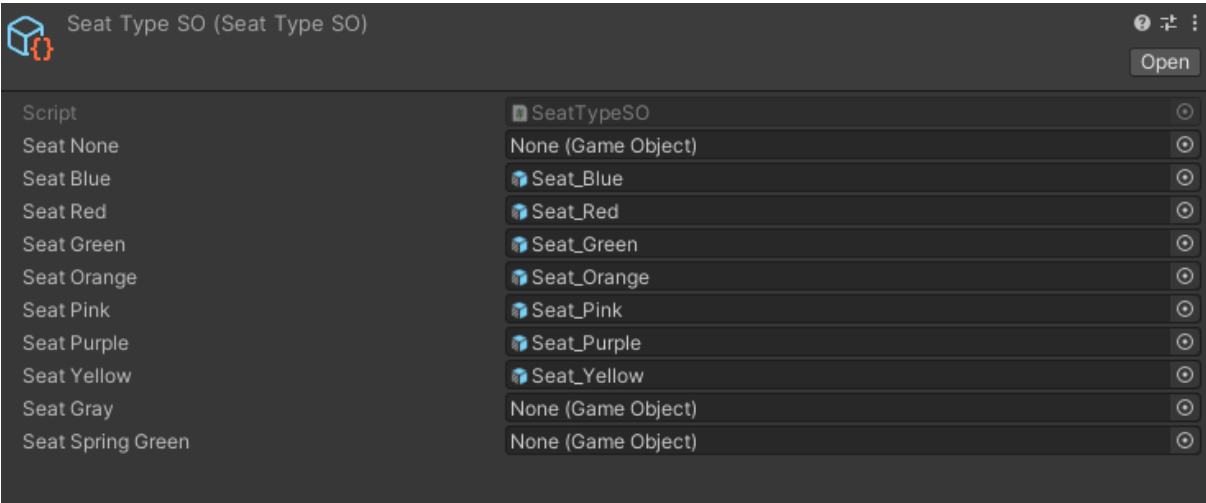
Assets\SeatSorting_Files\Data\PassengerTypeSO



The PassengerTypeSO data contains the prefabs of stickman characters. By modifying these stickman prefabs, you can add different characters to your game.

▼ Seat Type SO

Assets\SeatSorting_Files\Data\SeatTypeSO



The SeatTypeSO data contains the prefabs of seats. By modifying these seat prefabs, you can add different seat models to your game.

▼ Add Sound Effect

To add sound to the game, follow these steps:

1. **Sound File Location**

Add your sound files to the following directory in your project:

`Assets\SeatSorting_Files\Game\Resources`

2. Naming the Sound Files

Rename your sound files as follows before adding them to the project. **Do not use different file names; the sound file names must be exactly as listed below.**

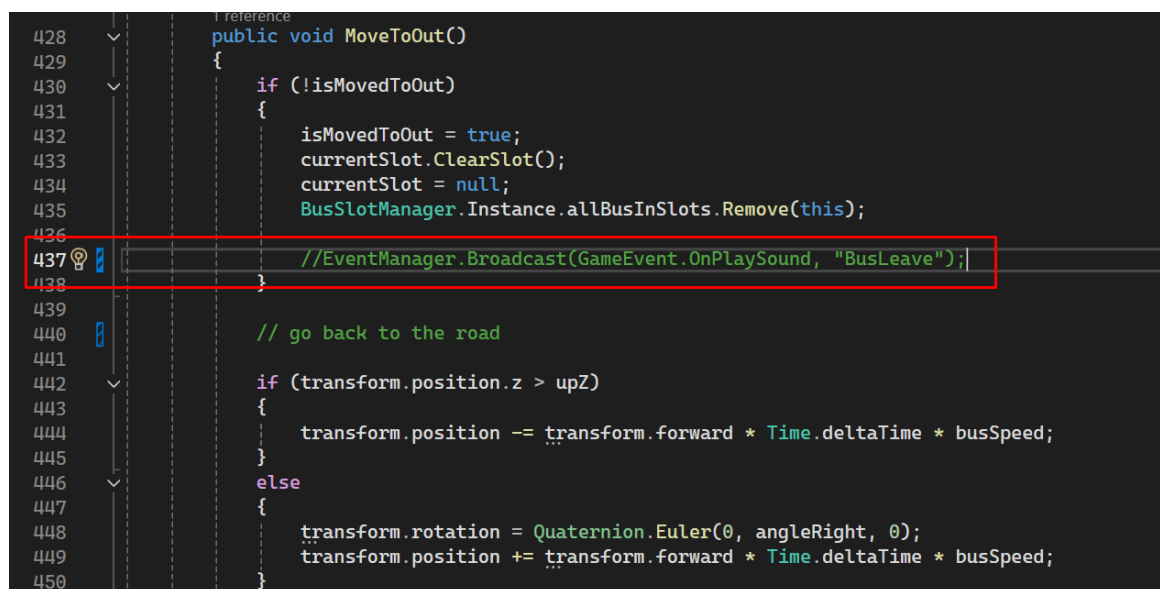
- `BusLeave.wav` – Plays when the characters have finished boarding the bus and it departs.
- `BusMove.wav` – Plays when the bus moves from the center area to the characters' area.
- `Fail.wav` – Plays on the "Fail" screen when the player loses.
- `Hit.wav` – Plays when buses collide.
- `Pop.wav` – Plays when characters board the bus.
- `VIPSlot.wav` – Plays when the VIP boost is used.
- `Win.wav` – Plays on the "Win" screen when the player succeeds.

Follow these steps to add your own sounds to the project. **For the sounds to function correctly, they must be placed in the specified folder and must not be renamed.**

IMPORTANT!

After adding the sounds to the folder, you need to remove the comment lines in the specified scripts. **Once this is done, your sounds will work seamlessly in the game.**

1. BusController.cs



```
428 public void MoveToOut()
429 {
430     if (!isMovedToOut)
431     {
432         isMovedToOut = true;
433         currentSlot.ClearSlot();
434         currentSlot = null;
435         BusSlotManager.Instance.allBusInSlots.Remove(this);
436     }
437     //EventManager.Broadcast(GameEvent.OnPlaySound, "BusLeave");
438 }
439
440 // go back to the road
441
442 if (transform.position.z > upZ)
443 {
444     transform.position -= transform.forward * Time.deltaTime * busSpeed;
445 }
446 else
447 {
448     transform.rotation = Quaternion.Euler(0, angleRight, 0);
449     transform.position += transform.forward * Time.deltaTime * busSpeed;
450 }
```

Remove the comment line in the section marked at line 437.

You can do this by deleting the `//` at the beginning of the line.

2. BusController.cs

```

144         if (isReadyToMove && targetSlot != null)
145         {
146             currentBusState = BusState.OnMovingToSlot;
147             isReadyToMove = false;
148
149             if (PlayerPrefs.GetInt("IsHapticOpen") == 1)
150                 Vibration.VibratePop();
151
152             //EventManager.Broadcast(GameEvent.OnSoundStart, "BusMove");
153         }
154     }
155
156     Unity Message | 0 references
157     private void Update()
158     {
159         if (isInTunnel)
160             return;

```

Remove the comment line in the section marked at line 152.

You can do this by deleting the `//` at the beginning of the line.

3. BusController.cs

```

233
234         EventManager.Broadcast(GameEvent.OnParticlePlay, "HitParticle", transform.position);
235
236         if (PlayerPrefs.GetInt("IsHapticOpen") == 1)
237             Vibration.VibratePop();
238
239         //EventManager.Broadcast(GameEvent.OnSoundStop);
240         //EventManager.Broadcast(GameEvent.OnPlaySound, "Hit");
241     }
242
243
244
245     Debug.DrawRay(leftRayOrigin.position, transform.forward * rayDistance, Color.red);
246     Debug.DrawRay(rightRayOrigin.position, transform.forward * rayDistance, Color.red);
247
248

```

Remove the comment line in the section marked at line 239. and 240.

You can do this by deleting the `//` at the beginning of the line.

4. InputController.cs

```

80
81         if (PlayerPrefs.GetInt("IsHapticOpen") == 1)
82             Vibration.VibratePop();
83
84         //EventManager.Broadcast(GameEvent.OnPlaySound, "Pop");
85
86         EventManager.Broadcast(GameEvent.SeatMoved);
87     }
88
89

```

Remove the comment line in the section marked at line 84.

You can do this by deleting the `//` at the beginning of the line.

5. UpgradeManager.cs

```

76         DOVirtual.DelayedCall(0.5f, buttonEvent.Invoke);
77     }
78
79
80     1 reference
81     void ShuffleAllSeats()
82     {
83         //EventManager.Broadcast(GameEvent.OnPlaySound, "Sort");
84         List<ObjectData> sortedSeatControllers = new List<ObjectData>();
85
86         foreach (var seat in manager.allseatControllers)
87     {

```

Remove the comment line in the section marked at line 82.

You can do this by deleting the `//` at the beginning of the line.

6. UpgradeManager.cs

```

156
157
158     1 reference
159     void UseVipSlot()
160     {
161         //EventManager.Broadcast(GameEvent.OnPlaySound, "VIPSlot");
162         BusSlotController vipSlot = BusSlotManager.Instance.OpenUpgradeSlot();
163         manager.isFirstSeatMoved = true;
164
165

```

Remove the comment line in the section marked at line 160.

You can do this by deleting the `//` at the beginning of the line.

7. GameManager.cs

```

116     2 references
117     void OnWin()
118     {
119         //EventManager.Broadcast(GameEvent.OnPlaySound, "Win");
120         isFinished = true;
121     }
122
123     2 references
124     void OnLose()
125     {
126         //EventManager.Broadcast(GameEvent.OnPlaySound, "Fail");
127         isFinished = true;
128     }
129

```

Remove the comment line in the section marked at line 118. and 124.

You can do this by deleting the `//` at the beginning of the line.