Isle of Plum Blossom

General Description

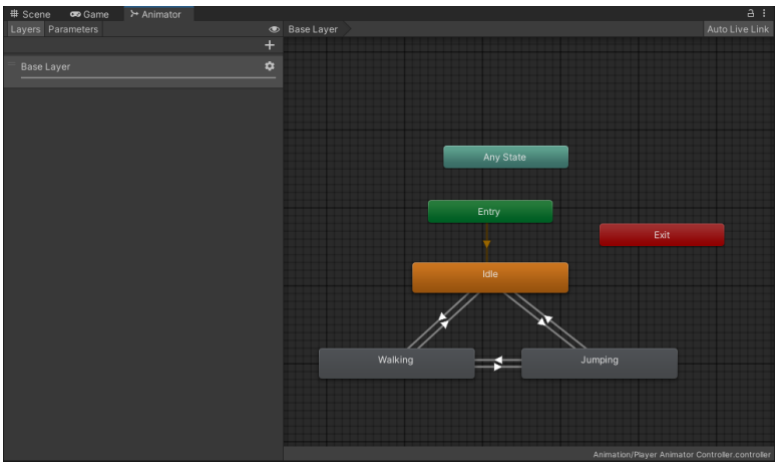
Drawing inspiration from Chinese classical culinary records, wellness philosophies, and paintings, the game weaves culinary traditions and aesthetics in a captivating kitchen setting. Players craft authentic dishes, desserts, and beverages using traditional Chinese ingredients and techniques while unlocking recipes steeped in cultural history. By managing time-sensitive orders, players embark on a flavorful journey through a beautifully designed world of culinary excellence.

Game Loop

Players are required to collect ingredients from resource spots scattered across the map, utilizing multiple cookware to create dishes according to the culinary book and serve them to the crane in the bottom left corner.

```
178     private void ReleaseObject() {
179         if (grabbedObject != null) {
180             if(collidingObject != null && collidingObject.CompareTag("Cookware")){
181                 cookware = collidingObject.GetComponent<Cookware>();
182                 if(!cookware.isCooking){
183                     cookware.PushIngredient(grabbedObject.GetComponent<Ingredient>());
184                     // disable the ingredient
185                     grabbedObject.transform.position = cookware.transform.position;
186                     grabbedObject.Release();
187                     grabbedObject.gameObject.SetActive(false);
188                     grabbedObject = null;
189                 }
190             }
191             else if(collidingObject != null && collidingObject.CompareTag("Garbage Bin")){
192                 Destroy(grabbedObject.gameObject);
193                 grabbedObject = null;
194             }
195             else if(collidingObject != null && collidingObject.CompareTag("ServingArea")){
196                 ServingArea servingArea = collidingObject.GetComponent<ServingArea>();
197                 if(servingArea.ServeOrder(grabbedObject.GetComponent<Ingredient>())){
198                     Destroy(grabbedObject.gameObject);
199                     grabbedObject = null;
200                 }
201                 else{
202                     Debug.Log("can not serve");
203                     grabbedObject.Release();
204                     grabbedObject = null;
205                 }
206             }
207             else{
208                 grabbedObject.Release();
209                 grabbedObject = null;
210             }
211         }
212     }
```

Code snippet of Player Controller



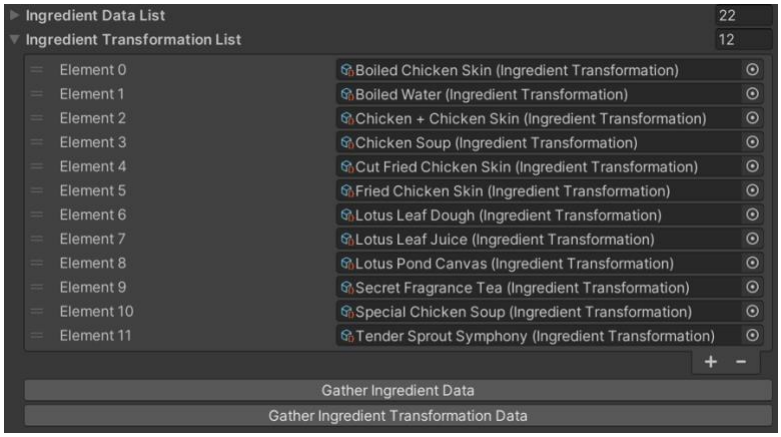
Animation System

```
98      IEnumerator Cook(List<IngredientData> result){
99          isCooking = true;
100         Debug.Log("Cooking ingredients...");
101
102         // play sound effect
103         if(soundEffect != null) soundEffect.Play();
104
105         float cookingProgress = 0f;
106         progressBarCanvas.gameObject.SetActive(true);
107
108         while (cookingProgress < cookingTime) {
109             cookingProgress += Time.deltaTime;
110             progressBar.SetProgress(cookingProgress / cookingTime);
111             yield return null;
112         }
113
114         progressBar.SetProgress(1f);
115         progressBarCanvas.gameObject.SetActive(false);
116
117         // generate the result ingredient
Controller.cs
119         DestroyIngredients();
120         InstantiateResultIngredient(result);
121
122         if(soundEffect != null) soundEffect.Stop();
123         isCooking = false;
124     }
```

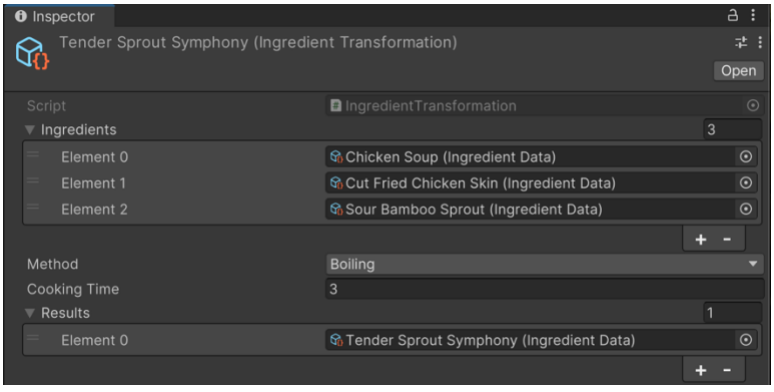
Code snippet of Cookware

```
181     [ContextMenu("Gather Ingredient Transformation Data")]
        1 reference
182     public void GatherIngredientTransformationData()
183     {
184         #if UNITY_EDITOR
185         // Clear the existing list
186         ingredientTransformationList = new List<IngredientTransformation>();
187
188         // Find all assets of type IngredientTransformation in the specified folder
189         string[] guids = AssetDatabase.FindAssets("t:IngredientTransformation", new[] { "Assets/Data/Transformations" });
190
191         foreach (string guid in guids)
192         {
193             // Convert GUID to Asset path
194             string path = AssetDatabase.GUIDToAssetPath(guid);
195
196             // Load the asset at the path
197             IngredientTransformation ingredientTransformation = AssetDatabase.LoadAssetAtPath<IngredientTransformation>(path);
198
199             if (ingredientTransformation != null)
200             {
201                 ingredientTransformationList.Add(ingredientTransformation);
202                 Debug.Log($"Gathered Ingredient Transformation: {ingredientTransformation.name}");
203             }
204         }
205
206         Debug.Log($"Total Ingredient Transformations Gathered: {ingredientTransformationList.Count}");
207         #endif
208     }
```

Code snippet of Game Manager



Reusable ScriptableObject database



Ingredient Transformation Example

Personal Contribution

As the sole developer of the game, I designed and implemented every aspect, from core functions to game art. Key contributions include:

- Implemented player movement and animation controllers for smooth and dynamic character actions.

- Designed and developed a modular cookware management system integrated with a centralized GameManager, enabling fluid interactions across game mechanics.

- Built a robust recipe workflow and a reusable ScriptableObject database for ingredient transformations, enabling efficient, dynamic interactions between ingredients.

- Implemented a multitasking gameplay loop, managing timers, player interactions, and the recipe book.

- Implemented a scene management system to oversee scene transitions between game levels and maintain a seamless player experience.

- Curated the game's aesthetics, music, and sound effects to immerse players in a vibrant world that reflects traditional Chinese culture.

Lessons Learned

This project marked my first experience using Unity, where I gained a deeper understanding of implementing complex game systems, such as modular management structures, reusable databases with ScriptableObjects, and multitasking gameplay loops. Additionally, I explored ways to incorporate traditional Chinese culture into the game. This experience strengthened my project management skills, allowing me to balance technical challenges with creative expression.