

# Project 2

## Modifying Stranded Deep

Conner Batson December 2024

dnSpy v6.1.8 (64-bit, .NET)

File Edit View Debug Window Help | C# | Start | Search

Assembly Explorer Constructing\_STILL

```
79     Singleton<AtmosphereStorm>.Instance.WeatherStarted += this.AtmosphereStorm_WeatherStarted;
80     Singleton<AtmosphereStorm>.Instance.WeatherStopped += this.AtmosphereStorm_WeatherStopped;
81     if (Singleton<AtmosphereStorm>.Instance.CurrentWeatherEvent != null)
82     {
83         this.AtmosphereStorm_WeatherStarted
84             (Singleton<AtmosphereStorm>.Instance.CurrentWeatherEvent);
85     }
86     if (Singleton<GameTime>.Instance)
87     {
88         Singleton<GameTime>.Instance.HourChanged += this.GameTime_HourChanged;
89     }
90 }
91
92 // Token: 0x060037B8 RID: 14264 RVA: 0x00098D04 File Offset: 0x00096F04
93 private void GameTime_HourChanged(float hours)
94 {
95     this.Still_Collected(hours);
96 }
97
98 // Token: 0x060037B9 RID: 14265 RVA: 0x00098D10 File Offset: 0x00096F10
99 private void OnDestroy()
100 {
101     Timing.KillCoroutines(this._collectingSequence);
102     this.StartedCollecting.RemoveAllListeners();
103     this.StoppedCollecting.RemoveAllListeners();
104     this.Collected.RemoveAllListeners();
105     this.Refilled.RemoveAllListeners();
106     this.StartedCollecting = null;
107     this.StoppedCollecting = null;
108     this.Collected = null;
109     this.Refilled = null;
110     if (Singleton<AtmosphereStorm>.Instance != null)
111     {
112         Singleton<AtmosphereStorm>.Instance.WeatherStarted -= this.AtmosphereStorm_WeatherStarted;
113         Singleton<AtmosphereStorm>.Instance.WeatherStopped -= this.AtmosphereStorm_WeatherStopped;
114     }
115     if (Singleton<GameTime>.Instance)
116     {
117         Singleton<GameTime>.Instance.HourChanged -= this.GameTime_HourChanged;
118     }
119 }
```

```
1  using System;
2  using System.Runtime.CompilerServices;
3  using Beam.Crafting;
4  using Beam.Serialization.Json;
5  using Funlabs;
6  using UnityEngine;
7  using UnityEngine.Events;
8
9  namespace Beam
10 {
11     // Token: 0x0200096D RID: 2413
12     public partial class Constructing_STILL_COLLECTOR : ReplicableMonoBehaviour, IInteractable, IBase, I
13     {
14         // Token: 0x06003E6B RID: 15979 RVA: 0x000BD8E4 File Offset: 0x000BBAE4
15         private void Drink(IPlayer player)
16         {
17             this.RemoveWater(0); // Changed method from this.RemoveWater(1) to this.RemoveWater(0)
18             int num = 100;
19             player.Statistics.Eat(InteractiveType.CRAFTING_WATER_STILL, MeatProvenance.Other, 0, 0, 0, 0
20             UnityEvent drank = this.Drank;
21             if (drank == null)
22             {
23                 return;
24             }
25             drank.Invoke();
26         }
27     }
28 }
29 }
```

Code	Description	File	Line
CS0683	'Constructing_STILL_COLLECTOR.get_gameObject()' explicit method implementation cannot implement 'IBase.gameObject.get' because it is an accessor	main.g.cs	227
CS0683	'Constructing_STILL_COLLECTOR.get_transform()' explicit method implementation cannot implement 'IBase.transform.get' because it is an accessor	main.g.cs	233
CS0169	The field 'Constructing_STILL_COLLECTOR._selecting' is never used	main.g.cs	251
CS0160	The field 'Constructing_STILL_COLLECTOR._referencold' is never used	main.cs	246

```
3  using Beam.Crafting;
4  using Beam.Serialization.Json;
5  using Funlabs;
6  using UnityEngine;
7  using UnityEngine.Events;
8
9  namespace Beam
10 {
11     // Token: 0x0200096D RID: 2413
12     public partial class Constructing_STILL_COLLECTOR : ReplicableMonoBehaviour, IInte
13     {
14         // Token: 0x06003E6A RID: 15978 RVA: 0x0001FD50 File Offset: 0x0001DF50
15         private void RemoveWater(int qty)
16         {
17             //this._water -= qty; // alter code to not remove water
18             UnityEvent waterRemoved = this.WaterRemoved;
19             if (waterRemoved == null)
20             {
21                 return;
22             }
23             waterRemoved.Invoke();
24         }
25     }
26 }
27 }
```

```

198         return this.interact(prayer);
199     }
200     if (interactiveObject_FOOD.Servings < interactiveObject_FOOD.OriginalServings)
201     {
202         InteractiveObject_FOOD interactiveObject_FOOD2 = interactiveObject_FOOD;
203         int servings = interactiveObject_FOOD2.Servings;
204         interactiveObject_FOOD2.Servings = servings + 1;
205         interactiveObject_FOOD.Servings = Mathf.Clamp(interactiveObject_FOOD.Servings, 0, interactiveObject_FOOD.OriginalServings);
206         this.RemoveWater(1);
207         return true;
208     }
209     LocalizedNotification.Post(player, NotificationPriority.Low, 4f, "WATER_STILL_COLLECTOR_NTF_FULL_DESC");
210     return false;
211 }
212
213 // Token: 0x06003E6A RID: 15978 RVA: 0x0001FD50 File Offset: 0x0001DF50
214 private void RemoveWater(int qty)
215 {
216     this._water = qty;
217     UnityEvent waterRemoved = this.WaterRemoved;
218     if (waterRemoved == null)
219     {
220         return;
221     }
222     waterRemoved.Invoke();
223 }
224
225 // Token: 0x06003E6B RID: 15979 RVA: 0x000BD8E4 File Offset: 0x000BBAE4
226 private void Drink(IPlayer player)
227 {
228     this.RemoveWater(1);
229     int num = 100;
230     player.Statistics.Eat(InteractiveType.CRAFTING_WATER_STILL, MeatProvenance.Other, 0, 0, 0, 0, (float)num, 0f, false);
231     UnityEvent drank = this.Drank;
232     if (drank == null)
233     {
234         return;

```

100 %

Code	Description	File	Line
CS0683	'Constructing_STILL_COLLECTOR.get_gameObject()' explicit method implementation cannot implement 'IBase.gameObject.get' because it is an accessor	main.cs	277
CS0683	'Constructing_STILL_COLLECTOR.get_transform()' explicit method implementation cannot implement 'IBase.transform.get' because it is an accessor	main.cs	283
CS0649	Field 'Constructing_STILL_COLLECTOR._selecting' is never assigned to, and will always have its default value null	main.cs	301
CS0640	Field 'Constructing_STILL_COLLECTOR._constructingStill' is never assigned to, and will always have its default value null	main.cs	302

```
198             return;
199         }
200         if (!this._isRaining)
201     {
202             int fibre = this._fibre;
203             int num = this._collector.MaxWater - this._collector.Water;
204             int num2 = Mathf.Min(new int[]
205         {
206             fibre,
207             num,
208             elapsedIntervals
209         });
210             if (num2 > 0)
211         {
212                 this._fibre -= num2;
213                 this._fibre = Mathf.Clamp(this._fibre, 0, 4);
214                 if (this._fibre == 0)
215                 {
216                     this.StopCollecting();
217                     this._fibre += 1;
218                 }
219                 FloatUnityEvent collected = this.Collect;
220                 if (collected != null)
221                 {
222                     collected.Invoke((float)num2);
223                 }
224                 this.Reposition_Fibres();
225                 return;
226             }
227         }
228         FloatUnityEvent collected2 = this.Collect;
229         if (collected2 == null)
230         {
231             return;
232         }
233         collected2.Invoke(1f);
234     }
```

100 % ▾

Code Description

## Assembly Explorer

Plot @02000AFF

- ▶ Base Type and Interfaces
- ▶ Derived Types
- ↳ Plot(): void @06005608
- ↳ AtmosphereStorm\_WeatherStarted(WeatherEvent): void
- ↳ AtmosphereStorm\_WeatherStopped(): void @060055F
- ↳ Awake(): void @060055F4
- ↳ CanWaterCrop(): bool @06005600
- ↳ ClearCrop(): bool @06005602
- ↳ CreateCrop(PlantType): void @060055FB
- \* FixedUpdate(): void @06005604
- ↳ GetNormalizedWater(): float @060055F0
- ↳ HasPlants(): bool @060055ED
- ↳ Initialize(): void @060055FC
- ↳ KillCrop(): void @06005603
- ↳ Load(JObject): void @06005607
- ↳ OnCropCreated(PlantType): void @060055F3
- ↳ OnDestroy(): void @060055F6
- ↳ Place(): void @060055F9
- ↳ Plant(Plantable): bool @060055FA
- ↳ PollPlant\_InvokeRepeating(): void @060055FE
- ↳ PollRain\_InvokeRepeating(): void @060055FD
- ↳ PollWater\_InvokeRepeating(): void @060055FF
- ↳ ReplicatedKillCrop(): void @06005605
- ↳ Save(): JObject @06005606
- ↳ Start(): void @060055F5
- ↳ WaterCrop(int): bool @06005601
- ▶ Growing: bool @170017EC
- ▶ MultiplayerState: IPlotState @170017ED
- ▶ ⚡ CropCreated: Action<PlantType> @14000112
- DEDELETE\_INTERVAL\_HOURS: float @04003484
- ↳ \_growing: bool @04003489
- ↳ \_isRaining: bool @0400348A
- ↳ \_maxWater: int @04003485
- ↳ \_plant: Plant @04003488
- ↳ \_plantFactory: IPlantFactory @04003487
- ↳ \_plantPosition: Transform @04003486
- ↳ \_water: int @0400348B
- ▶ ReplicateKillCrop @02000E95
- ▶ ReplicatePoll @02000E96
- ▶ ProjectileHit @02000AF1
- ▶ ProjectileObject @02000AF3
- ▶ QuickAccessSlotUnlocker @02000AF4
- ▶ Raycaster @02000B17
- ▶ SceneGameAction @02000B16

## Plot X

```

318     JObject field = data.GetField("Water");
319     if (field)
320     {
321         this._water = field.GetValue<int>();
322     }
323     PlantType plantType = JSerializer.Deserialize<PlantType>(data.GetField("PlantType"));
324     if (plantType == PlantType.NONE)
325     {
326         return;
327     }
328     this.CreateCrop(plantType);
329     if (this._water <= 0)
330     {
331         this.ReplicatedKillCrop();
332         return;
333     }
334     JObject field2 = data.GetField("Plants");
335     JObject data2 = ((field2 != null) ? field2.Children.FirstOrDefault<JObject>() : null) ?? data.GetField("[0]Plant");
336     if (SavingUtilities.Load(this._plant, data2))
337     {
338         JObject field3 = data.GetField("LastPollTime");
339         if (field3 != null && !field3.IsNull())
340         {
341             int num = Mathf.RoundToInt((float)((int)GameTime.GetHoursSince(JSerializer.Deserialize<DateTime>(field3)) / 24f));
342             this._water -= num;
343             if (this._water <= 0)
344             {
345                 this.ReplicatedKillCrop();
346                 return;
347             }
348         }
349         this.Initialize();
350         return;
351     }
352     this.ClearCrop();
353 }

// Token: 0x04003484 RID: 13444
private const float DEDELETE_INTERVAL_HOURS = 24f;

// Token: 0x04003485 RID: 13445
[SerializeField]
private int _maxWater = 4;

// Token: 0x04003486 RID: 13446
[SerializeField]
private Transform _plantPosition;

// Token: 0x04003487 RID: 13447
private IPlantFactory _plantFactory;

```

```
323     PlantType plantType = JSerializer.Deserialize<PlantType>(data.GetField( PlantType ));  
324     if (plantType == PlantType.NONE)  
325     {  
326         return;  
327     }  
328     this.CreateCrop(plantType);  
329     if (this._water <= 0)  
330     {  
331         this.ReplicatedKillCrop();  
332         return;  
333     }  
334     JObject field2 = data.GetField("Plants");  
335     JObject data2 = ((field2 != null) ? field2.Children.FirstOrDefault< JObject>() : null) ?? data.GetField("[0]Plant");  
336     if (SavingUtilities.Load(this._plant, data2))  
337     {  
338         JObject field3 = data.GetField("LastPollTime");  
339         if (field3 != null && !field3.IsNull())  
340         {  
341             int num = Mathf.RoundToInt((float)((int)GameTime.GetHoursSince(JSerializer.Deserialize< DateTime>(field3))) / 240f); // Changed 24f to 240f  
342             this._water -= num;  
343             if (this._water <= 0)  
344             {  
345                 this.ReplicatedKillCrop();  
346                 return;  
347             }  
348         }  
349         this.Initialize();  
350         return;  
351     }  
352     this.ClearCrop();  
353 }  
354 // Token: 0x06005608 RID: 22024 RVA: 0x000F2665 File Offset: 0x000F0865  
355 public Plot()  
356 {  
357 }  
358 }  
359 }
```

100 %

Code Description

File Line

**RESUME**

**OPTIONS**

**SAVE**

YOU NEED TO SAVE THE GAME AT A SHELTER!

**QUIT**

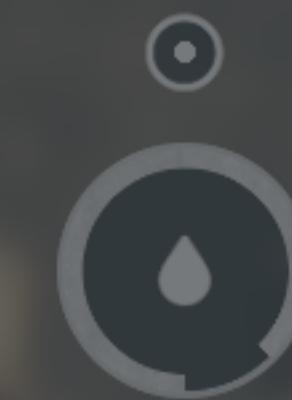
**QUIT TO DESKTOP**

cjbatson2

DAY 34 | 0d 13h 09m

## YUCCA - PLANK FARMING PLOT

HOLD E TO WATER



**RESUME**

**OPTIONS**

**SAVE**

YOU NEED TO SAVE THE GAME AT A SHELTER!

**QUIT**

**QUIT TO DESKTOP**

cjbatson2

DAY 40 | 0d 15h 17m

YUCCA - PLANK FARMING PLOT



**RESUME**  
**OPTIONS**  
**SAVE**  
YOU NEED TO SAVE THE GAME AT A SHELTER!  
**QUIT**  
**QUIT TO DESKTOP**

cjbatson2

DAY 42 | 0d 16h 09m

### YUCCA - PLANK FARMING PLOT



```
// Token: 0x060060D7 RID: 24791
public void UpdateStage()
{
    if (this._dead && !this._deadStage.activeSelf)
    {
        for (int i = 0; i < this._plantStages.Length; i++)
        {
            this._plantStages[i].SetActive(false);
        }
        this._deadStage.SetActive(true);
    }
    else if (!this._dead && this._deadStage.activeSelf)
    {
        this._deadStage.SetActive(false);
    }
    if (!this._dead)
    {
        float hoursSince = GameTime.GetHoursSince(this._startedGrowingTime);
        this._stage = Mathf.Clamp((int)(hoursSince / this._growthTime * 10f), 0, 4);
        for (int j = 0; j < this._plantStages.Length; j++)
        {
            this._plantStages[j].SetActive(this._stage == j);
        }
    }
}
```

```
9     // Token: 0x060060D7 RID: 24791 RVA: 0x0010B6C8 File Offset: 0x001098C8
10    public void UpdateStage()
11    {
12        if (this._dead && !this._deadStage.activeSelf)
13        {
14            for (int i = 0; i < this._plantStages.Length; i++)
15            {
16                this._plantStages[i].SetActive(false);
17            }
18            this._deadStage.SetActive(true);
19        }
20        else if (!this._dead && this._deadStage.activeSelf)
21        {
22            this._deadStage.SetActive(false);
23        }
24        if (!this._dead)
25        {
26            float hoursSince = GameTime.GetHoursSince(this._startedGrowingTime);
27            this._stage = Mathf.Clamp((int)(hoursSince / this._growthTime * 0.1f), 0, 4);
28            for (int j = 0; j < this._plantStages.Length; j++)
29            {
30                this._plantStages[j].SetActive(this._stage == j);
31            }
32        }
33    }
34 }
35 }
36 }
```

100 %

Code Description

File Line



```
// Token: 0x060060D7 RID: 24791
public void UpdateStage()
{
    if (this._dead && !this._deadStage.activeSelf)
    {
        for (int i = 0; i < this._plantStages.Length; i++)
        {
            this._plantStages[i].SetActive(false);
        }
        this._deadStage.SetActive(true);
    }
    else if (!this._dead && this._deadStage.activeSelf)
    {
        this._deadStage.SetActive(false);
    }
    if (!this._dead)
    {
        float hoursSince = GameTime.GetHoursSince(this._startedGrowingTime);
        this._stage = Mathf.Clamp((int)(hoursSince / this._growthTime * 10f), 0, 4);
        for (int j = 0; j < this._plantStages.Length; j++)
        {
            this._plantStages[j].SetActive(this._stage == j);
        }
    }
}
```



```
9         // Token: 0x060060D7 RID: 24791 RVA: 0x0010B6C8 File Offset: 0x001098C8
10        public void UpdateStage()
11        {
12            if (this._dead && !this._deadStage.activeSelf)
13            {
14                for (int i = 0; i < this._plantStages.Length; i++)
15                {
16                    this._plantStages[i].SetActive(false);
17                }
18                this._deadStage.SetActive(true);
19            }
20            else if (!this._dead && this._deadStage.activeSelf)
21            {
22                this._deadStage.SetActive(false);
23            }
24            if (!this._dead)
25            {
26                float hoursSince = GameTime.GetHoursSince(this._startedGrowingTime);
27                this._stage = Mathf.Clamp((int)(hoursSince * 10f / this._growthTime * 4f), 0, 4);
28                for (int j = 0; j < this._plantStages.Length; j++)
29                {
30                    this._plantStages[j].SetActive(this._stage == j);
31                }
32            }
33        }
34    }
35}
36}
```

100 %

Code Description

File Line



```
e) : void X  
fting.Farming_PLOT  
x060060BD RID: 24765  
d Plot_CropCreated(PlantType type)  
  
text = type.GetAttribute<DescriptionAttribute>()  
splayNamePrefixes.AddOrIgnore(text + " = ", 0);
```

ALOE = PLANK FARMING PLOT

