

My progress so far up from Nov. 8 to Nov. 28, 2014:

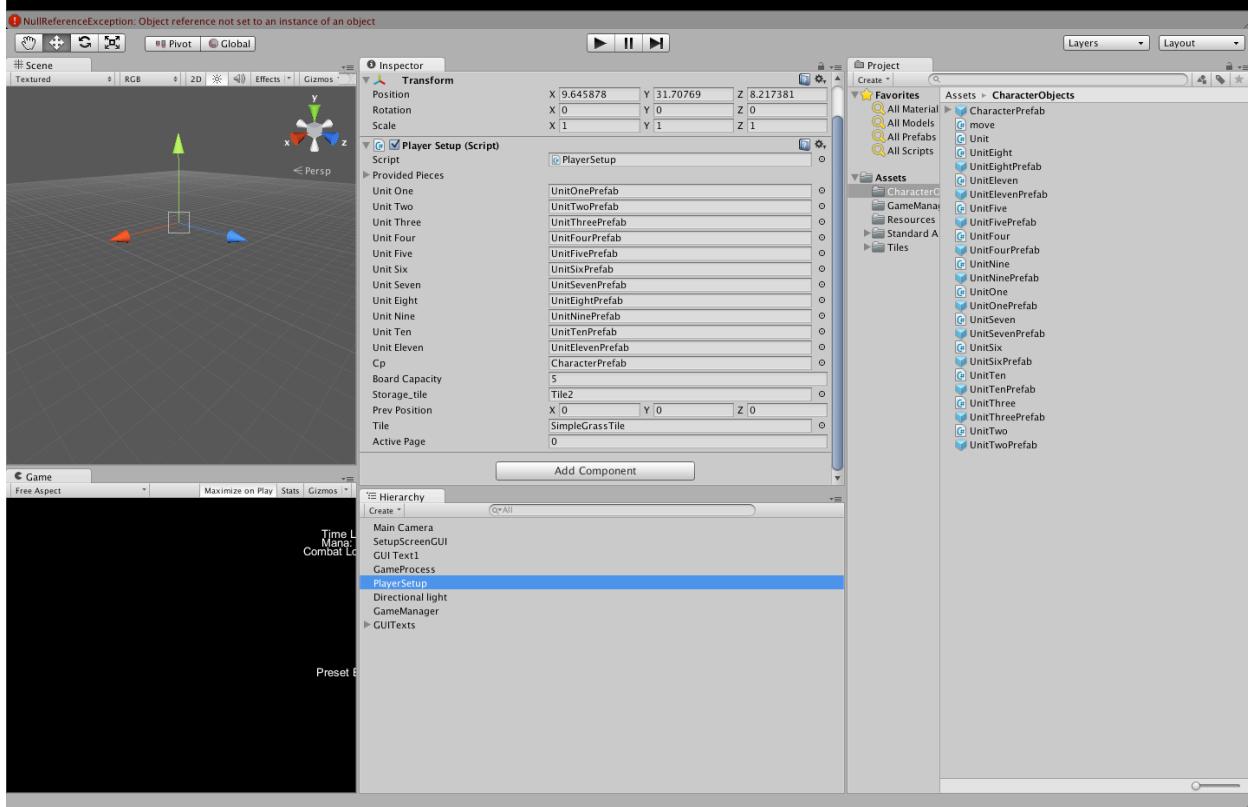
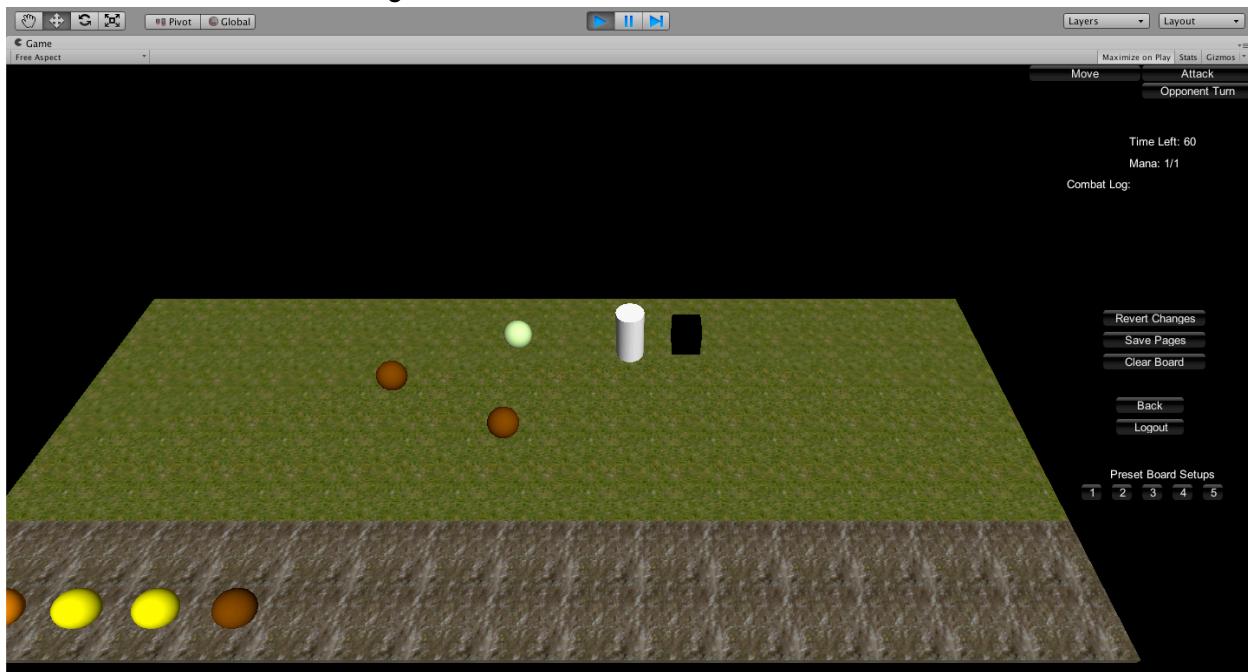
1. What I finished so far:
 - a. Completely designed and tested the design of the Unit Veterancy mechanic, wrote it in the design document, and implemented all the units' corresponding Trophy Values for unit experience earning.
 - b. Researched and integrated the final game art into the design document.
 - c. I completely changed the move.cs script to make sure that the Guardian and the SoulStone can never be removed from the board on-field.
 - d. I finished the designing and implementing the system that limits how many pieces are allowed to be placed onto the board.
 - e. I finished getting the setup screen save the player's interaction on the setup screen for usage by the server, as I have described towards the end of my previous status report. saving the arrangement in how the players move his/her pieces on the board. I worked closely with Joey to design how each page (board preset) is saved in the setup screen. I implemented this by designing how each page is represented on the 5 board presets. I implemented this design by starting with building the Page.cs data structure. I also finished designing and implementing the syntax logic needed to save each page with the following scripts:
 - i. page.cs- This was used as the basis to represent each page object
 - ii. SetupScreenGUI.cs
 - iii. move.cs
 - f. I further refined how the movement logic in the move.cs script to accommodate with the new design of saving each page.
 - g. I redefined and reimplemented how each unit is different from each other by using "unitType" to replace unitRole and unitID in unit.cs.
 - h.
2. What I did for the project that can't be seen on the workload sheet or Trello:

Paper prototyping each design I propose on my own before proposing the design to Joey. This goes for everything I described above.

3. What I am still working on:

Now that I have found and consolidated a collection of the art assets needed, I need to replace the placeholder art currently in the game with the sprites and their animations. I am also currently getting the project ready for the Unit Veterancy system for the next milestone.

Relevant screenshots showing some of what I did so far:



MonoDevelop-Unity File Edit View Search Project Build Run Version Control Tools Window Help <img alt="

The screenshot shows the MonoDevelop-Unity IDE interface. The title bar reads "Assembly-CSharp - CharacterObjects/move.cs - MonoDevelop-Unity". The left sidebar displays the "Solution" tree, which includes the "CharacterObjects" project with its files: move.cs, Unit.cs, UnitEight.cs, UnitEleven.cs, UnitFive.cs, UnitFour.cs, UnitNine.cs, UnitOne.cs, UnitSeven.cs, UnitSix.cs, UnitTen.cs, UnitThree.cs, and UnitTwo.cs. It also lists "GameManaging", "Standard Assets", and "Tiles" categories with their respective files like AudioManager.cs, CameraScript1.cs, etc. The main editor window shows the C# code for the move.cs file, specifically the OnMouseDown() method. The code handles transforming objects between board slots and off-board pieces based on player setup and tile type.

```
else
{
    transform.position = playerSetup.prevPosition;
}

else if(playerSetup.pages[playerSetup.activePage].onBoardPieces.Count < playerSetup.boardCapacity)
{
    if(slot.GetComponent<SetupTileScript>().tt == SetupTileScript.TileType.ONFIELD)
    {
        if(!playerSetup.pages[playerSetup.activePage].onBoardPieces.Contains(this.gameObject))
        {

            Debug.Log ("piece added. page " + playerSetup.activePage);
            playerSetup.pages[playerSetup.activePage].offBoardPieces.Remove(this.gameObject);
            playerSetup.pages[playerSetup.activePage].onBoardPieces.Add(this.gameObject);
        }

        playerSetup.pages[playerSetup.activePage].modified = true;
        this.transform.parent.GetComponent<SetupTileScript>().occupied = false;
        this.transform.parent = slot;
        transform.position = new Vector3(slot.transform.position.x, 5.0f, slot.position.z);
        slot.GetComponent<SetupTileScript>().occupied = true;
    }
}
else
{
    if(playerSetup.pages[playerSetup.activePage].onBoardPieces.Contains(gameObject))
    {
        playerSetup.pages[playerSetup.activePage].offBoardPieces.Add(this.gameObject);
        playerSetup.pages[playerSetup.activePage].onBoardPieces.Remove(this.gameObject);
    }

    playerSetup.pages[playerSetup.activePage].modified = true;
    this.transform.parent.GetComponent<SetupTileScript>().occupied = false;
    this.transform.parent = slot;
    transform.position = new Vector3(slot.transform.position.x, 5.0f, slot.position.z);
    slot.GetComponent<SetupTileScript>().occupied = true;
}
}
else
{
    if(slot.GetComponent<SetupTileScript>().tt == SetupTileScript.TileType.ONFIELD)
    {
        // Debug.Log (playerSetup.pages[playerSetup.activePage].onBoardPieces.Contains(gameObject));
    }
}
```

The screenshot shows the MonoDevelop-Unity IDE interface. The title bar reads "Assembly-CSharp – CharacterObjects/move.cs – MonoDevelop-Unity". The main window displays the code for the `move.cs` file, specifically the `OnMouseDown` and `OnMouseUp` events. The code handles piece movement logic, including board piece placement and tile setup. The left sidebar shows the project structure under "Solution", including "CharacterObjects" and "GameManaging" packages. The bottom status bar indicates "Unity" and the system tray shows icons for "Errors" and "Tasks".

```
        else
    {
        if(playerSetup.pages[playerSetup.activePage].onBoardPieces.Contains(gameObject))
        {
            playerSetup.pages[playerSetup.activePage].offBoardPieces.Add(this.gameObject);
            playerSetup.pages[playerSetup.activePage].onBoardPieces.Remove(this.gameObject);
        }

        playerSetup.pages[playerSetup.activePage].modified = true;
        this.transform.parent.GetComponent<SetupFileScript>().occupied = false;
        this.transform.parent = slot;
        transform.position = new Vector3(slot.transform.position.x, 5.0f, slot.position.z);
        slot.GetComponent<SetupFileScript>().occupied = true;
    }
}

else
{
    if(slot.GetComponent<SetupTileScript>().tt == SetupTileScript.TileType.ONFIELD)
    {
        // Debug.Log (playerSetup.pages[playerSetup.activePage].onBoardPieces.Contains(gameObject));
        if(playerSetup.pages[playerSetup.activePage].onBoardPieces.Contains(gameObject))
        {
            playerSetup.pages[playerSetup.activePage].modified = true;
            this.transform.parent.GetComponent<SetupFileScript>().occupied = false;
            this.transform.parent = slot;
            transform.position = new Vector3(slot.transform.position.x, 5.0f, slot.position.z);
            slot.GetComponent<SetupFileScript>().occupied = true;
        }

        else
            transform.position = playerSetup.prevPosition;
    }
}

else if(playerSetup.pages[playerSetup.activePage].onBoardPieces.Contains(gameObject))
{
    playerSetup.pages[playerSetup.activePage].offBoardPieces.Add(this.gameObject);
    playerSetup.pages[playerSetup.activePage].onBoardPieces.Remove(this.gameObject);

    playerSetup.pages[playerSetup.activePage].modified = true;
    this.transform.parent.GetComponent<SetupFileScript>().occupied = false;
    this.transform.parent = slot;
```

MonoDevelop-Unity File Edit View Search Project Build Run Version Control Tools Window Help Wed 8:15 AM Jordan Hua

MonoDevelop-Unity Assembly-CSharp - CharacterObjects/move.cs - MonoDevelop-Unity

Press '⌘.' to search

Solution

- CharacterObjects
 - move.cs
 - Unit.cs
 - UnitEight.cs
 - UnitEleven.cs
 - UnitFive.cs
 - UnitFour.cs
 - UnitNine.cs
 - UnitOne.cs
 - UnitSeven.cs
 - UnitSix.cs
 - UnitTen.cs
 - UnitThree.cs
 - UnitTwo.cs
- GameManaging
- Standard Assets
- Tiles
 - AudioManager.cs
 - CameraScript1.cs
 - CameraScript2.cs
 - challengeScript.cs
 - GameProcess.cs
 - globalChatScript.cs
 - ListOfPlayersScript.cs
 - LoginScreenGUI.cs
 - MainMenuGUI.cs
 - Page.cs
 - PlayerSetup.cs
 - PopupMenus.cs
 - SetupScreenGUI.cs
 - SetupTileScript.cs
 - Sockets.cs
 - ThreadSock.cs
- Assembly-CSharp-firstpass
- Assembly-UnityScritps-firstpass

Unit.cs move.cs

```

 99 }
100 }
101 else
102 {
103     if(slot.GetComponent<SetupTileScript>().tt == SetupTileScript.TileType.ONFIELD)
104     {
105         // Debug.Log (playerSetup.pages[playerSetup.activePage].onBoardPieces.Contains(gameObject));
106         if(playerSetup.pages[playerSetup.activePage].onBoardPieces.Contains(gameObject))
107         {
108             playerSetup.pages[playerSetup.activePage].modified = true;
109             this.transform.parent.GetComponent<SetupTileScript>().occupied = false;
110             this.transform.parent = slot;
111             transform.position = new Vector3(slot.transform.position.x, 5.0f, slot.position.z);
112             slot.GetComponent<SetupTileScript>().occupied = true;
113         }
114     }
115     else
116         transform.position = playerSetup.prevPosition;
117 }
118 }
119 }
120 else if(playerSetup.pages[playerSetup.activePage].onBoardPieces.Contains(gameObject))
121 {
122     playerSetup.pages[playerSetup.activePage].offBoardPieces.Add(this.gameObject);
123     playerSetup.pages[playerSetup.activePage].onBoardPieces.Remove(this.gameObject);
124
125     playerSetup.pages[playerSetup.activePage].modified = true;
126     this.transform.parent.GetComponent<SetupTileScript>().occupied = false;
127     this.transform.parent = slot;
128     transform.position = new Vector3(slot.transform.position.x, 5.0f, slot.position.z);
129     slot.GetComponent<SetupTileScript>().occupied = true;
130 }
131 }
132 }
133
134 //Special if dragging object to field-type tile:
135 //Debug.Log(playerSetup.activePage);
136 Debug.Log(playerSetup.pages[playerSetup.activePage].onBoardPieces.Count);
137
138
139
140
141 }
142 }
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157 Transform findNearest()
158 {
159     float nearestDistanceTile = Mathf.Infinity;
160     GameObject[] taggedGameObjects = GameObject.FindGameObjectsWithTag(tagName); //KEY compares all objects near with that tag name! Works perfectly
161     Transform nearestObj = null;
162     foreach (GameObject obj in taggedGameObjects)
163     {
164         Vector3 objectPos = obj.transform.position;
165         float distanceTile = (objectPos - transform.position).sqrMagnitude;
166
167         if (distanceTile < nearestDistanceTile)
168         {
169             nearestObj = obj.transform;
170             nearestDistanceTile = distanceTile;
171         }
172     }
173
174     return nearestObj;
175 }
```

Unity

MonoDevelop-Unity File Edit View Search Project Build Run Version Control Tools Window Help Wed 8:15 AM Jordan Hua

MonoDevelop-Unity Assembly-CSharp - CharacterObjects/move.cs - MonoDevelop-Unity

Press '⌘.' to search

Solution

- CharacterObjects
 - move.cs
 - Unit.cs
 - UnitEight.cs
 - UnitEleven.cs
 - UnitFive.cs
 - UnitFour.cs
 - UnitNine.cs
 - UnitOne.cs
 - UnitSeven.cs
 - UnitSix.cs
 - UnitTen.cs
 - UnitThree.cs
 - UnitTwo.cs
- GameManaging
- Standard Assets
- Tiles
 - AudioManager.cs
 - CameraScript1.cs
 - CameraScript2.cs
 - challengeScript.cs
 - GameProcess.cs
 - globalChatScript.cs
 - ListOfPlayersScript.cs
 - LoginScreenGUI.cs
 - MainMenuGUI.cs
 - Page.cs
 - PlayerSetup.cs
 - PopupMenus.cs
 - SetupScreenGUI.cs
 - SetupTileScript.cs
 - Sockets.cs
 - ThreadSock.cs
- Assembly-CSharp-firstpass
- Assembly-UnityScritps-firstpass

Unit.cs move.cs

```

126 this.transform.parent.GetComponent<SetupTileScript>().occupied = false;
127 this.transform.parent = slot;
128 transform.position = new Vector3(slot.transform.position.x, 5.0f, slot.position.z);
129 slot.GetComponent<SetupTileScript>().occupied = true;
130 }
131 }
132
133
134 //Special if dragging object to field-type tile:
135 //Debug.Log(playerSetup.activePage);
136 Debug.Log(playerSetup.pages[playerSetup.activePage].onBoardPieces.Count);
137
138
139
140
141 }
142 }
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157 Transform findNearest()
158 {
159     float nearestDistanceTile = Mathf.Infinity;
160     GameObject[] taggedGameObjects = GameObject.FindGameObjectsWithTag(tagName); //KEY compares all objects near with that tag name! Works perfectly
161     Transform nearestObj = null;
162     foreach (GameObject obj in taggedGameObjects)
163     {
164         Vector3 objectPos = obj.transform.position;
165         float distanceTile = (objectPos - transform.position).sqrMagnitude;
166
167         if (distanceTile < nearestDistanceTile)
168         {
169             nearestObj = obj.transform;
170             nearestDistanceTile = distanceTile;
171         }
172     }
173
174     return nearestObj;
175 }
```

Unity

MonoDevelop-Unity File Edit View Search Project Build Run Version Control Tools Window Help Wed 8:15 AM Jordan Hua

Assembly-CSharp - CharacterObjects/move.cs - MonoDevelop-Unity

Press '⌘.' to search

Solution

- CharacterObjects
 - move.cs
 - Unit.cs
 - UnitEight.cs
 - UnitEleven.cs
 - UnitFive.cs
 - UnitFour.cs
 - UnitNine.cs
 - UnitOne.cs
 - UnitSeven.cs
 - UnitSix.cs
 - UnitTen.cs
 - UnitThree.cs
 - UnitTwo.cs
- Standard Assets
- Tiles
 - AudioManager.cs
 - CameraScript1.cs
 - CameraScript2.cs
 - challengeScript.cs
 - GameProcess.cs
 - globalChatScript.cs
 - ListOfPlayersScript.cs
 - LoginScreenGUI.cs
 - MainMenuGUI.cs
 - Page.cs
 - PlayerSetup.cs
 - PopUpMenus.cs
 - SetupScreenGUI.cs
 - SetupTileScript.cs
 - Sockets.cs
 - ThreadSock.cs
- Assembly-CSharp-firstpass
- Assembly-UnityScript-firstpass

Unit.cs move.cs

```

132
133
134 //Special if dragging object to field-type tile;
135 //Debug.Log(playerSetup.activePage);
136 Debug.Log(playerSetup.pages[playerSetup.activePage].onBoardPieces.Count);
137
138
139
140
141 }
142 else
143 {
144
145     //transform.position = PlayerSetup.prevPosition;
146     transform.position = playerSetup.prevPosition;
147 }
148
149
150
151
152
153
154
155
156
157 Transform findNearest()
158 {
159     float nearestDistanceTile = Mathf.Infinity;
160     GameObject[] taggedGameObjects = GameObject.FindGameObjectsWithTag(tagName); //---KEY compares all objects near with that tag name! Works perfectly
161     Transform nearestObj = null;
162     foreach (GameObject obj in taggedGameObjects)
163     {
164         Vector3 objectPos = obj.transform.position;
165         float distanceTile = (objectPos - transform.position).sqrMagnitude;
166
167         if (distanceTile < nearestDistanceTile)
168         {
169             nearestObj = obj.transform;
170             nearestDistanceTile = distanceTile;
171         }
172     }
173
174     return nearestObj;
175 }
```

Unity

MonoDevelop-Unity File Edit View Search Project Build Run Version Control Tools Window Help Wed 8:16 AM Jordan Hua

Assembly-CSharp - SetupTileScript.cs - MonoDevelop-Unity

Press '⌘.' to search

Solution

- UnitNine.cs
- UnitOne.cs
- UnitSeven.cs
- UnitSix.cs
- UnitTen.cs
- UnitThree.cs
- UnitTwo.cs
- GameManaging
 - GameManager.cs
- Standard Assets
- Tiles
 - AudioManager.cs
 - CameraScript1.cs
 - CameraScript2.cs
 - challengeScript.cs
 - GameProcess.cs
 - globalChatScript.cs
 - ListOfPlayersScript.cs
 - LoginScreenGUI.cs
 - MainMenuGUI.cs
 - Page.cs
 - PlayerSetup.cs
 - PopUpMenus.cs
 - SetupScreenGUI.cs
 - SetupTileScript.cs
 - Sockets.cs
 - ThreadSock.cs
- Assembly-CSharp-firstpass
- Assembly-UnityScript-firstpass
- Guardians of the Arena (m)
 - Assembly-CSharp
 - Assembly-CSharp-firstpass
 - Assembly-UnityScript
 - Assembly-UnityScript-firstpass

Unit.cs move.cs GameManager.cs SetupTileScript.cs

```

1 using UnityEngine;
2 using System.Collections;
3
4 public class SetupTileScript : MonoBehaviour {
5     PlayerSetup ps;
6     public int x,y;
7     public GameObject up,down,left,right;
8
9     public bool occupied;
10
11     public enum TileType{ONFIELD, OFFFIELD};//values given to them from 0 to 1 by default, since this is an enum
12
13     public TileType tt;
14
15     void Start () {
16         ps = GameObject.Find("PlayerSetup").GetComponent<PlayerSetup>();
17
18         //This start is called after playersetup is called. So we need to check if occupied is true
19         if(occupied == true)
20         {
21             if(ps.occupied == true)
22             {
23                 Debug.Log ("occupied is true");
24             }
25             else
26             {
27                 occupied = false;
28             }
29
30         }
31
32     }
33
34
35
36     void Update () {
37
38
39
40
41     }
42
43
44 }
```

Unity

MonoDevelop-Unity File Edit View Search Project Build Run Version Control Tools Window Help Wed 8:17 AM Jordan Hua

Assembly-CSharp - Page.cs - MonoDevelop-Unity

Solution

```

UnitNine.cs
UnitOne.cs
UnitSeven.cs
UnitSix.cs
UnitTen.cs
UnitThree.cs
UnitTwo.cs
GameManaging
  GameManager.cs
Standard Assets
Tiles
  AudioManager.cs
  CameraScript1.cs
  CameraScript2.cs
  challengeScript.cs
  GameProcess.cs
  globalChatScript.cs
  ListOfPlayersScript.cs
  LoginScreenGUI.cs
  MainMenuGUI.cs
  Page.cs
  PlayerSetup.cs
  PopUpMenu.cs
  SetupScreenGUI.cs
  SetupTileScript.cs
  Sockets.cs
  ThreadSocks.cs
Assembly-CSharp-firstpass
Assembly-UnityScript-firstpass
Guardians of the Arena (m)
  Assembly-CSharp
  Assembly-CSharp-firstpass
  Assembly-UnityScript
  Assembly-UnityScript-firstpass

```

Unit.cs **move.cs** **GameManager.cs** **SetupTileScript.cs** **Page.cs**

```

using UnityEngine;
using System.Collections;
public class Page {
    public PlayerSetup playerSetup;
    public bool modified;
    public ArrayList onBoardPieces;
    public ArrayList offBoardPieces;
    // Use this for initialization
    public Page () {
        playerSetup = GameObject.Find("PlayerSetup").GetComponent<PlayerSetup>();
        modified = false;
        onBoardPieces = new ArrayList(playerSetup.boardCapacity);
        offBoardPieces = new ArrayList(playerSetup.boardCapacity);
    }
    // Update is called once per frame
    void Update () {
        //
    }
}

```

Errors **Tasks**

MonoDevelop-Unity File Edit View Search Project Build Run Version Control Tools Window Help Wed 8:23 AM Jordan Hua

Assembly-CSharp - SetupScreenGUI.cs - MonoDevelop-Unity

Solution

```

UnitNine.cs
UnitOne.cs
UnitSeven.cs
UnitSix.cs
UnitTen.cs
UnitThree.cs
UnitTwo.cs
GameManaging
  GameManager.cs
Standard Assets
Tiles
  AudioManager.cs
  CameraScript1.cs
  CameraScript2.cs
  challengeScript.cs
  GameProcess.cs
  globalChatScript.cs
  ListOfPlayersScript.cs
  LoginScreenGUI.cs
  MainMenuGUI.cs
  Page.cs
  PlayerSetup.cs
  PopUpMenu.cs
  SetupScreenGUI.cs
  SetupTileScript.cs
  Sockets.cs
  ThreadSocks.cs
Assembly-CSharp-firstpass
Assembly-UnityScript-firstpass
Guardians of the Arena (m)
  Assembly-CSharp
  Assembly-CSharp-firstpass
  Assembly-UnityScript
  Assembly-UnityScript-firstpass

```

Unit.cs **move.cs** **GameManager.cs** **SetupTileScript.cs** **Page.cs** **PlayerSetup.cs** **SetupScreenGUI.cs**

```

public PlayerSetup playerSetup;
// Use this for initialization
void Start () {
    showGUI = true;
    gp = GameObject.Find("GameProcess").GetComponent<GameProcess>();
}
playerSetup = GameObject.Find("PlayerSetup").GetComponent<PlayerSetup>();
}

void OnGUI () {
    if(showGUI) {
        if(GUI.Button(new Rect(Screen.width - 175, Screen.height / 2 - 125, 120, 20), "Revert Changes"))
        {
            switch(playerSetup.activePage)
            {
                case 1:
                    gp.returnSocket().SendTCPacket("getBoardData\\1\\" + gp.playerName);
                    playerSetup.pages[0].modified = false;
                    break;
                case 2:
                    gp.returnSocket().SendTCPacket("getBoardData\\2\\" + gp.playerName);
                    playerSetup.pages[1].modified = false;
                    break;
                case 3:
                    gp.returnSocket().SendTCPacket("getBoardData\\3\\" + gp.playerName);
                    playerSetup.pages[2].modified = false;
                    break;
                case 4:
                    gp.returnSocket().SendTCPacket("getBoardData\\4\\" + gp.playerName);
                    playerSetup.pages[3].modified = false;
                    break;
                case 5:
                    gp.returnSocket().SendTCPacket("getBoardData\\5\\" + gp.playerName);
                    playerSetup.pages[4].modified = false;
                    break;
            }
        }
    }
}

```

Errors **Tasks**

MonoDevelop-Unity File Edit View Search Project Build Run Version Control Tools Window Help Wed 8:23 AM Jordan Hua

Assembly-CSharp - SetupScreenGUI.cs - MonoDevelop-Unity

Solution Unit.cs move.cs GameManager.cs SetupTileScript.cs Page.cs PlayerSetup.cs SetupScreenGUI.cs

```

SetupScreenGUI > OnGUI()
139     Destroy(GameObject.Find ("ListOfPlayersGUIText"));
140     DontDestroyOnLoad(gp);
141     Application.LoadLevel(0);
142
143     // KILL THREAD AND SERVER CONNECTION
144     gp.returnSocket().t.Abort();
145     gp.returnSocket().endThread();
146     gp.returnSocket().Disconnect();
147 }
148
149 if ( GUI.Button( new Rect( Screen.width - 200, Screen.height / 2 + 75, 25, 20), "1" ) )
150 {
151     gp.returnSocket().SendCPPacket("getBoardData\\1\\\" + gp.playerName);
152     playerSetup.activePage = 0;
153 }
154
155 if ( GUI.Button( new Rect( Screen.width - 165, Screen.height / 2 + 75, 25, 20), "2" ) )
156 {
157     gp.returnSocket().SendCPPacket("getBoardData\\2\\\" + gp.playerName);
158     playerSetup.activePage = 1;
159 }
160
161 if ( GUI.Button( new Rect( Screen.width - 130, Screen.height / 2 + 75, 25, 20), "3" ) )
162 {
163     gp.returnSocket().SendCPPacket("getBoardData\\3\\\" + gp.playerName);
164     playerSetup.activePage = 2;
165 }
166
167 if ( GUI.Button( new Rect( Screen.width - 95, Screen.height / 2 + 75, 25, 20), "4" ) )
168 {
169     gp.returnSocket().SendCPPacket("getBoardData\\4\\\" + gp.playerName);
170     playerSetup.activePage = 3;
171 }
172
173 if ( GUI.Button( new Rect( Screen.width - 60, Screen.height / 2 + 75, 25, 20), "5" ) )
174 {
175     gp.returnSocket().SendCPPacket("getBoardData\\5\\\" + gp.playerName);
176     playerSetup.activePage = 4;
177 }
178
179
180 }
181
182
183 // Update is called once per frame
void Update () { }

```

MonoDevelop-Unity File Edit View Search Project Build Run Version Control Tools Window Help Wed 8:24 AM Jordan Hua

Assembly-CSharp - PlayerSetup.cs - MonoDevelop-Unity

Solution Unit.cs move.cs GameManager.cs SetupTileScript.cs Page.cs PlayerSetup.cs SetupScreenGUI.cs

```

PlayerSetup > No selection
318     activePage = 0;
319
320     pages = new Page[5];
321     for(int i = 0; i < 5; i++)
322     {
323         pages[i] = new Page();
324     }
325
326     pages[0].onBoardPieces.Add(providedPieces[0]);
327     pages[0].onBoardPieces.Add(providedPieces[0]);
328 }
329
330
331 void updatePageModifier()
332 {
333     switch(activePage)
334     {
335         case 1:
336             pages[0].modified = true;
337             break;
338         case 2:
339             pages[1].modified = true;
340             break;
341         case 3:
342             pages[2].modified = true;
343             break;
344         case 4:
345             pages[3].modified = true;
346             break;
347         case 5:
348             pages[4].modified = true;
349             break;
350     }
351 }
352
353
354 // Drag and drop
355 // Update is called once per frame
void Update () { }
356
357
358
359
360
361
362 }
363

```

MonoDevelop-Unity File Edit View Search Project Build Run Version Control Tools Window Help Wed 8:24 AM Jordan Hua

Assembly-CSharp - PlayerSetup.cs - MonoDevelop-Unity

Press '⌘.' to search

Solution

```

Unit.cs move.cs GameManager.cs SetupTileScript.cs Page.cs PlayerSetup.cs SetupScreenGUI.cs

```

PlayerSetup > Start 0

```

318
319
320
321
322
323
324
325
326
327
328
329
330
331
332
333
334
335
336
337
338
339
340
341
342
343
344
345
346
347
348
349
350
351
352
353
354
355
356
357
358
359
360
361
362
363

```

void Start()
{
 activePage = 0;
 pages = new Page[5];
 for(int i = 0; i < 5; i++)
 {
 pages[i] = new Page();
 }
 pages[0].onBoardPieces.Add(providedPieces[0]);
 pages[0].onBoardPieces.Add(providedPieces[1]);
}

void updatePageModifier()
{
 switch(activePage)
 {
 case 1:
 pages[0].modified = true;
 break;
 case 2:
 pages[1].modified = true;
 break;
 case 3:
 pages[2].modified = true;
 break;
 case 4:
 pages[3].modified = true;
 break;
 case 5:
 pages[4].modified = true;
 break;
 }
}

//Drag and drop
// Update is called once per frame
void Update () {
}

Errors Tasks

MonoDevelop-Unity File Edit View Search Project Build Run Version Control Tools Window Help Wed 8:24 AM Jordan Hua

Assembly-CSharp - PlayerSetup.cs - MonoDevelop-Unity

Press '⌘.' to search

Solution

```

Unit.cs move.cs GameManager.cs SetupTileScript.cs Page.cs PlayerSetup.cs SetupScreenGUI.cs

```

PlayerSetup > Start 0

```

296
297
298
299
300
301
302
303
304
305
306
307
308
309
310
311
312
313
314
315
316
317
318
319
320
321
322
323
324
325
326
327
328
329
330
331
332
333
334
335
336
337
338
339
340
341
342
343
344
345
346
347
348
349
350
351
352
353
354
355
356
357
358
359
360
361
362
363

```

//Add a move script to each, just for this scene though (this scene should be the only scene that calls this script)
//Position each of the pieces just made onto the board:
//for(int i = 0; i < providedPieces.Length - 1; i++)
//for(int i = 0; i < providedPieces.Length; i++)
for(int i = 0; i < providedPieces.Length; i++)
{
 providedPieces[i].AddComponent("move");
}

playerPieces = new ArrayList(boardCapacity);

//CHANGE the indices of the providedPieces in
//THIS PART WHENEVER YOU CHANGE THE SIZE OF THE PROVIDEDPIECES ARRAY:

playerPieces.Add (providedPieces[0]);
playerPieces.Add (providedPieces[1]);

activePage = 0;

pages = new Page[5];
for(int i = 0; i < 5; i++)
{
 pages[i] = new Page();
}

pages[0].onBoardPieces.Add(providedPieces[0]);
pages[0].onBoardPieces.Add(providedPieces[1]);
}

void updatePageModifier()
{
 switch(activePage)
 {
 case 1:
 pages[0].modified = true;
 break;
 case 2:
 pages[1].modified = true;
 break;
 }
}

Errors Tasks

MonoDevelop-Unity File Edit View Search Project Build Run Version Control Tools Window Help Wed 8:26 AM Jordan Hua

Assembly-CSharp - SetupTileScript.cs - MonoDevelop-Unity

Solution

```

Unit.cs move.cs GameManager.cs SetupTileScript.cs Page.cs PlayerSetup.cs SetupScreenGUI.cs
SetupTileScript.cs Start()
1 using UnityEngine;
2 using System.Collections;
3
4 public class SetupTileScript : MonoBehaviour {
5     PlayerSetup ps;
6     public int x,y;
7     public GameObject up,down,left,right;
8
9     public bool occupied;
10
11    public enum TileType{ONFIELD, OFFFIELD};//values given to them from 0 to 1 by default, since this is an enum
12
13    public TileType tt;
14
15    void Start () {
16        ps = GameObject.Find("PlayerSetup").GetComponent<PlayerSetup>();
17
18        //This start is called after playersetup is called. So we need to check if occupied is true
19        if(occupied == true)
20        {
21            Debug.Log ("occupied is true");
22        }
23        else
24        {
25            occupied = false;
26        }
27    }
28
29
30    void Update () {
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
}

```

Errors Tasks

Chrome File Edit View History Bookmarks Window Help

Unit Design Spreadsheet

File Edit View Insert Format Data Tools Add-ons Help All changes saved in Drive

| | B | C | D | E | F | G | H | I | J | K | L |
|----|---|------------------|---------------|---------------|---------------|-----------------------|----------------------|---------------------|--------------------|---|---|
| 1 | | | | | | | | | | TODO | TODO |
| 2 | | | | | | | | | | | |
| 3 | Unit Name | Unit Type | Copies | Health | Attack | Movement Range | Movement Cost | Attack Range | Attack Cost | First Bonus | Second Bonus |
| 4 | Knife Thrower | 1 | 1 | 25 | 18 | 4 | 2 | 4 | 2 | add 1 atk range | AoE attack on point (1 range) |
| 5 | Mystic | 2 | 2 | 30 | - | 2 | 2 | 4 | 4 | buffed ally unit gains 5 atk | paralyzed enemy unit takes 8 dmg at the |
| 6 | Templar | 3 | 3 | 38 | 13 | 3 | 2 | 1 | 3 | deals 5 bonus damage to full health units | allied units in AoE get healed for |
| 7 | Swordsman | 7 | 4 | 38 | 10 | 3 | 1 | 1 | 1 | heals for 5 health when hits enemy unit | can attack 2x per turn |
| 8 | Priest | 8 | 1 | 20 | -20 | 3 | 1 | - | 4 | heal single target to max hp | and 10 health to all nearby allies within |
| 9 | | | | | | | | | | | |
| 10 | Guardian | 10 | 1 | 45 | 23 | 2 | 2 | 1 | 1 | Cannot take more than 10 dmg per attack | executes enemies below 50% |
| 11 | Soulstone | 11 | 1 | 1 | - | - | - | - | - | | - |
| 12 | | | | | | | | | | Red is needs implementation | Green is done |
| 13 | | | | | | | | | | | |
| 14 | Shrub | | | N/A | N/A | N/A | N/A | N/A | N/A | | |
| 15 | | | | | | | | | | | |
| 16 | | | | | | | | | | | |
| 17 | | | | | | | | | | | |
| 18 | | | | | | | | | | | |
| 19 | Certain units can cast "focus" spells. The focus spell casted will persist until the unit that casted the focus spell takes damage. | | | | | | | | | | |
| 20 | Paralyzed units are incapable of moving and attacking. | | | | | | | | | | |
| 21 | | | | | | | | | | | |
| 22 | | | | | | | | | | | |
| 23 | Archer | 9 | 2 | 22 | 12 | 3 | 2 | 4 | 3 | | |
| 24 | Dragon | 4 | 1 | 28 | 10 | 4 | 1 | 1 | 2 | | |
| 25 | ... | 5 | 1 | 40 | - | - | - | 6 | 2 | | |
| 26 | | | | | | | | | | | |
| 27 | | | | | | | | | | | |
| 28 | *s and ID #'s. The Role #'s correspond to the role on the leftmost column. | | | | | | | | | | |
| 29 | unit's "name". Use this logic in the networking part. | | | | | | | | | | |

Sheet1 Sum: 17

Unit Design Spreadsheet

keep a copy of this column at this end

| Trophy Value | How XP is Gained | Unit Name | Details (Unit Abilities) |
|--|---|---------------|---|
| 36 | on enemy unit kill | Knife Thrower | Can only attack tiles in the 4 cardinal directions, and the knife will collide with the first target regardless of the targeted tile. |
| 24 | when buffed unit gets a kill or enemy dies while paralyzed | Focus Mage | Focus: Target Ally: Grant +3 Movement Range, OR Target Enemy: Paralyze that unit. |
| 24 | on enemy unit kill | Templar | Picks one of 4 adjacent tiles to attack, then damages enemy units on that tile and the 3 adjacent tiles behind it (NUMPA). |
| 18 | on enemy unit kill | Swordsman | "Attack" targets a single unit |
| 30 | If a unit was healed by the priest at any point during the game and it kills a unit | Priest | |
| 60 | on enemy unit kill | Guardian | |
| 0 | - | Soulstone | Takes damage = game over |
| 10 squares are randomly chosen on the grid to have shrubs, indicating that the selected squares prevent pieces from moving onto them. | | | |
| Selects one of 4 adjacent tiles to attack, then breathes fire on that tile and the 2 adjacent to it (sides). This unit switches places with an enemy unit within its attack range. | | | |

Sheet1 Sum: 192

Game Design Document

Leveling Up

Overview: Each time a player successfully kills an enemy piece with one of their own, all the pieces that contributed towards inflicting damage towards that enemy piece gains a fraction of the trophy value that the slain piece is worth.

- Trophy value:** "Trophy value" is a number assigned to each unit type that determines how much experience is distributed to the pieces that kills the enemy of that unit type type.
- (Trophy Value System) How the trophy value of the slain piece is shared:**
 - Cumulative Damage Value-** This value is individually given to each piece that has attacked an enemy piece. Pieces that die lose their cumulative damage value towards the enemy piece that is attacked. Cumulative damage value is calculated based on how much damage the piece inflicted times the number of attacks the piece made towards that specific enemy unit. Cumulative Damage Value determines which Effort Rank that that piece falls into for that specifically targeted enemy piece.

ii. Accumulative Trophy Value (Effort Ranks)

inflicted times the number of attacks the piece made towards that specific enemy unit. Cumulative Damage Value determines which Effort Rank that piece falls into for that specifically targeted enemy piece.

ii. **Accumulative Trophy Value (Effort Ranks)-**

Within each Effort Rank, the fraction of the Trophy Value given to that effort rank is further equally divided among all the pieces in that Effort Rank. For example, the fraction 3/6 of Trophy Value 36 is calculated as 18 for pieces that qualify in Effort Rank 1. 18 is then divided among all the pieces in Effort Rank 1. So if there are 3 pieces in Effort Rank 1 in total, each of the 3 pieces gain 6 experience.

1. **Effort Rank 1:** The pieces that inflicted the most cumulative damage towards the enemy piece (either by performing single attack or multiple attacks), shares the largest fraction (3/6) of the trophy value is shared within this effort rank, (4/6) of the trophy value if there are no pieces that qualify that accumulating the 3rd-greatest damage (Effort Rank 3), and 6/6 if all the pieces that attacked the enemy piece did equal amount of damage (No pieces fall in Effort Rank 2).

ii. **Accumulative Trophy Value (Effort Ranks)-**

Within each Effort Rank, the fraction of the Trophy Value given to that effort rank is further equally divided among all the pieces in that Effort Rank. For example, the fraction 3/6 of Trophy Value 36 is calculated as 18 for pieces that qualify in Effort Rank 1. 18 is then divided among all the pieces in Effort Rank 1. So if there are 3 pieces in Effort Rank 1 in total, each of the 3 pieces gain 6 experience.

1. **Effort Rank 1:** The pieces that inflicted the most cumulative damage towards the enemy piece (either by performing single attack or multiple attacks), shares the largest fraction (3/6) of the trophy value is shared within this effort rank, (4/6) of the trophy value if there are no pieces that qualify that accumulating the 3rd-greatest damage (Effort Rank 3), and 6/6 if all the pieces that attacked the enemy piece did equal amount of damage (No pieces fall in Effort Rank 2).
2. **Effort Rank 2:** All the pieces that do the second greatest cumulative damage share 2/6 of the trophy value amongst each other.
3. **Effort Rank 3:** All the remaining pieces that did the third greatest cumulative damage or less share 1/6 of the trophy value as experience amongst each other.

If more than 1 piece falls in inflicting the most damage, second-most damage, or third-most damage (aka: falls into Effort Rank 1, Effort Rank 2, or Effort Rank 3), all the pieces in that place (Effort Rank) array gets that specified fraction of the trophy value further divided among all the pieces that fell in that effort rank. More information about this is described in the section of this game.