Documentation: the How-To for the entire project

Git Repository (first time)

Adding a player w/ basic movement

Adding boundaries to the map

Initiating level change

**Git Repository (first time)**

Download GitHub desktop. Select clone repository. Copy link provided on the [still empty] online repository page by clicking “clone or download”

Enter link as prompted and select a preferred path folder wherein to save your local progress.

When starting new Unity project, change the save location path to match the local repository path.

Ensure that File>Options>Editor Settings has Version Control set to “visible meta files”

Ensure that a .gitignore file exists in the projects root directory before first commit

See “.gitignore”

**Adding a player w/ basic movement**

From Unity user-space, Assets>Create>Sprites>Square. Drag Square Sprite from Assets into the Scene.

Square and Background colors can be changed by selecting each object from the hierarchy and editing the respective swatches in the Inspector. Sprite can be scaled from the Inspector or using the scale tool.

Add a C# script component to the existing player sprite. This “monobehavior” script will interpret arrow keys, when pressed by the user, into vertical and horizontal translations.

GetKey allows for consistent input whereas GetKeyDown interprets one keypress per update

See “Script PlayerMove”

**Adding boundaries to the map**

Create a new C# script component to add to the existing player sprite. This one will be called bounds or something similar and will prevent the player from moving outside of the screen space you can see. The code snippet from the online example uses different values for the second and third parameters to get the same result. These parameters are highlighted in blue under “Script bounds”

**.gitignore**

/[Ll]ibrary/

/[Tt]emp/

/[Oo]bj/

/[Bb]uild/

/[Bb]uilds/

/Assets/AssetStoreTools\*

/[Pp]ackages/

/[Pp]rojectSettings/

# Visual Studio 2015 cache directory

/.vs/

# Autogenerated VS/MD/Consulo solution and project files

ExportedObj/

.consulo/

\*.csproj

\*.unityproj

\*.sln

\*.suo

\*.tmp

\*.user

\*.userprefs

\*.pidb

\*.booproj

\*.svd

\*.pdb

# Unity3D generated meta files

\*.pidb.meta

# Unity3D Generated File On Crash Reports

sysinfo.txt

# Builds

\*.apk

\*.unitypackage

**Script PlayerMove**

using System.Collections;

using System.Collections.Generic;

using UnityEngine;

public class playermove : MonoBehaviour

{

[SerializeField]

private float speed;

private Vector2 direction;

// Update is called once per frame

void Update()

{

GetInput();

Move();

}

private void Move()

{

transform.Translate(direction\*speed);

}

private void GetInput()

{

direction = Vector2.zero;

if (Input.GetKeyDown(KeyCode.UpArrow))

{

direction += Vector2.up;

}

else if (Input.GetKeyDown(KeyCode.DownArrow))

{

direction += Vector2.down;

}

else if (Input.GetKeyDown(KeyCode.LeftArrow))

{

direction += Vector2.left;

}

else if (Input.GetKeyDown(KeyCode.RightArrow))

{

direction += Vector2.right;

}

}

}

**Script bounds**

using System.Collections;

using System.Collections.Generic;

using UnityEngine;

public class bounds : MonoBehaviour

{

public Camera MainCamera; //be sure to assign this in the inspector to your main camera

private Vector2 screenBounds;

private float objectWidth;

private float objectHeight;

// Use this for initialization

void Start()

{

screenBounds = MainCamera.ScreenToWorldPoint(new Vector3(Screen.width, Screen.height, MainCamera.transform.position.z));

objectWidth = transform.GetComponent<SpriteRenderer>().bounds.extents.x; //extents = size of width / 2

objectHeight = transform.GetComponent<SpriteRenderer>().bounds.extents.y; //extents = size of height / 2

}

// Update is called once per frame

void LateUpdate()

{

Vector3 viewPos = transform.position;

viewPos.x = Mathf.Clamp(viewPos.x, -screenBounds.x + objectWidth, screenBounds.x - objectWidth);

viewPos.y = Mathf.Clamp(viewPos.y, -screenBounds.y + objectHeight, screenBounds.y - objectHeight);

transform.position = viewPos;

}

}