**Pertemuan Ke- 6**

**Laporan**

|  |  |  |
| --- | --- | --- |
| **LANGKAH-LANGKAH PENGERJAAN** | | |
| **1.** | Buatlah Project 3D baru pada Unity. | |
| **2.** | Masukkan paket BasicScene kedalam Assets. | |
| **3.** | Pada Project View, buka BasicScene kemudian akan muncul karakter animasi dan beberapa objek benda yang telah dibuat. | |
| **4.** | Tambahkan new Camera ke dalam scene melalui menu Create yang berada diatas Hierarchy, Pilih Create → Camera | |
| **5.** | Pilih Camera pada Hierarchy yang telah dibuat, dari Inspector rubahlah “Depth = 1” seperti gambar di bawah ini. | |
| **6.** | Pada Project View buatlah script C# baru dengan code seperti dibawah ini, dan berilah nama PictureInPicture. | |
| **7.** | Setelah menyimpan script C# tersebut, drag script tersebut menuju Camera yang ada pada Hierarchy. | |
| **8.** | Pilih Camera lalu pada Inspector hilangkan centang pada komponen Audio Listener dan rubah beberapa parameter yang ada pada Script PictureInPicture seperti gambar dibawah ini: | |
| **9.** | Silahkan jalankan program, maka pada layar tampilan akan muncul camera dari arah sudut pandang berbeda yang terletak pada kanan atas layar | |
|  | **2. Membuat dan Melakukan Perpindahan Beberapa Sudut Pandang Camera** | |
| **1.** | Buatlah Project 3D baru pada Unity 3D. | |
| **2.** | Masukkan paket BasicScene kedalam Assets | |
| **3.** | Pada Project View, buka BasicScene kemudian akan muncul karakter animasi dan beberapa objek benda yang telah dibuat. | |
| **4.** | Tambahkan 2 new Camera ke dalam scene melalui menu Create yang berada diatas Hierarchy, Pilih Create → Camera. Kemudian beri nama cam1 dan cam2. | |
| **5.** | Pada Inspector, rubahlah posisi camera pada cam1 dan cam2 seperti dibawah ini.  Cam 1 Cam 2 | |
| **6.** | Buatlah GameObject baru menggunakan menu Create pada Hierarchy. Pilih Create → Create Empty. Kemudian ganti nama menjadi Switchboard. | |
| **7.** | Dari Inspector pada cam1 dan cam2, hilangkan tanda centang pada komponen Camera dan Audio Listener. Serta atur kolom Tag menjadi MainCamera. | |
| **8.** | Kemudian buatlah script C# dengan code seperti dibawah ini kemudian berilah nama CameraSwitch | |
| **9.** | Setelah itu drag script C# tersebut menuju GameObject Switchboard. | |
| **10.** | Pada Inspector, atur ukuran Cameras dan Shortcuts dengan “sizes = 3”. Kemudian, drag dan arahkan secara urut mulai dari Main Camera (berada di Multipurpose Camera Rig → Pivot), cam1, dan cam2 menuju ke setiap slots yang ada pada Cameras. Kemudian pada Shortcuts ketikan angka 1, 2, dan 3 secara urut pada slots yang tersedia. Seperti gambar di bawah ini: | |
| **11.** | Jalankan program dan coba tekan angka 1, 2, dan 3 pada keyboard, maka akan terjadi perubahan pada sudut pandang camera utama.  Jika dipencet angka 1 pada keyboard    Jika dipencet angka 2 pada keyboard    Jika dipencet angka 3 pada keyboard | |
|  | **3. Membuat Hasil Tangkapan (Capture) Layar yang Berada di Dalam Frame** | |
| **1.** | Buatlah Project 3D baru pada Unity 3D | |
| **2.** | Masukkan paket BasicScene ke dalam Assets. | |
| **3.** | Pada Project View, buka BasicScene kemudian akan muncul karakter animasi dan beberapa objek benda yang telah dibuat. Didalam paket tersebut juga sudah terdapat fitur Canvas untuk UI elements. | |
| **4.** | Buatlah UI Image dari menu Create yang ada pada Hierarchy, Pilih Create → UI → Image. Setelah itu akan muncul Image yang berada pada Canvas kemudian ganti nama tersebut menjadi frame. | |
| **5.** | Pilih frame pada Hierarchy, kemudian pada komponen Image (Script) yang berada di Inspector atur kolom Source Image menjadi InputFieldBackground. Seperti pada gambar di bawah ini: | |
| **6.** | Pada Inspector milik Frame, Rubahlah Rect Transform sesuai dengan gambar dibawah ini: | |
| **7.** | Pada komponen Image (Script), hilangkan centang pada Fill Center. | |
| **8.** | Buatlah UI Raw Image dari menu Create yang terdapat pada Hierarchy, Pilih Create → UI → RawImage. Setelah itu akan muncul pada Canvas GameObject kemudian ganti nama menjadi Photo. | |
| **9.** | Pada Hierarchy pilih Photo, kemudian pada Inspector carilah komponen Raw Image (Script) dan atur kolom Texture menjadi None (Texture). Setelah itu hilangkan centang pada kolom Photo yang berada di atas. | |
| **10.** | Langkah selanjutnya atur Rect Transform pada Photo sesuai gambar dibawah ini: | |
| **11.** | Kemudian buatlah script C# dengan code seperti dibawah ini kemudian berilah nama ScreenTexture. | |
| **12.** | Setelah itu drag Script C# tersebut menjuju Main Camera yang berada pada Multipurpose Camera Rig -> Pivot | |
| **13.** | Pada Inspector milik Main Camera, carilah komponen Screen Texture (Script) kemudian atur kolom Photo GUI dengan memilih Photo. Sama halnya dengan kolom Frame GUI dengan memilih Frame. | |
| **14.** | Jalankan program, maka anda akan bisa meng-capture layar yang berada dalam frame dengan cara menekan klik kiri pada mouse. Hasil capture akan tampil pada pojok kiri atas layar. | |
|  | **Tugas Praktikum**   1. **Membuat tampilan Camera dari Arah Sudut Pandang lain di dalam Layar**   Buatlah sebuah project 3D yang didalamnya terdapat 3 Fitur Using Cameras pada modul diatas. | |
| **1.** | Buat projet baru untuk pengerjaan tugas | |
| **2.** | Masukkan package dari pengerjaan praktikum sebelumnya. | |
| **3.** | Kita download dulu asset freepack karakter sample, lali import ke project | |
| **4.** | Selanjutnya kita drag game object maleFree1 kedalam hirarki | |
| **5.** | Atur transform male free nya seperti berikut ini | |
| **6.** | Tambahkan beberapa komponen juga untuk menunjang pergerakan objectnya seperti pada gambar | |
| **7.** | Pada Project View buatlah script C# baru dengan code seperti dibawah ini, dan berilah nama PictureInPicture. | |
| **8.** | Setelah menyimpan script C# tersebut, drag script tersebut menuju Camera yang ada pada Hierarchy | |
| **9.** | Pilih Camera lalu pada Inspector hilangkan centang pada komponen Audio Listener dan rubah beberapa parameter yang ada pada Script PictureInPicture seperti gambar dibawah ini: | |
| **10.** | Buat material baru untuk memberi warna pada jalanan nya dengan cara klik kanan pada project > create > material | |
| **11.** | Klik pada object material dan rename menjadi warna-jalanan, lalu rubah waranya | |
| **12.** | Masuk ke tab lightning > environtmen. Drag gambar Sunny 01A ke dalam skybox material | |
| **13.** | Silahkan jalankan program, maka pada layar tampilan akan muncul camera dari arah sudut pandang berbeda yang terletak pada kanan atas layar.    Object male free1 bisa berjalan | |
|  | **2. Membuat dan Melakukan Perpindahan Beberapa Sudut Pandang Camera** | |
| **1.** | Import package dari pengerjaan project tugas sebelumnya | |
| **2.** | Tambahkan 2 new Camera ke dalam scene melalui menu Create yang berada diatas Hierarchy, Pilih Create → Camera. Kemudian beri nama camera1 dan camera2. | |
| **3.** | Pada Inspector, rubahlah posisi camera pada camera1 dan camera2 seperti dibawah ini. | |
| **4.** | Buatlah GameObject baru menggunakan menu Create pada Hierarchy. Pilih Create → Create Empty. Kemudian ganti nama menjadi Switchboard. | |
| **5.** | Dari Inspector pada camera1 dan camera2, hilangkan tanda centang pada komponen Camera dan Audio Listener. Serta atur kolom Tag menjadi MainCamera. | |
| **6.** | Kemudian buatlah script C# dengan code seperti dibawah ini kemudian berilah nama CameraSwitch. | |
| **7.** | Setelah itu drag script C# tersebut menuju GameObject Switchboard. | |
| **8.** | Pada Inspector, atur ukuran Cameras dan Shortcuts dengan “sizes = 3”. Kemudian, drag dan arahkan secara urut mulai dari Main Camera (berada di Multipurpose Camera Rig → Pivot), cam1, dan cam2 menuju ke setiap slots yang ada pada Cameras. Kemudian pada Shortcuts ketikan angka 1, 2, dan 3 secara urut pada slots yang tersedia. Seperti gambar di bawah ini: | |
| **9.** | Jalankan program dan coba tekan angka 1, 2, dan 3 pada keyboard, maka akan terjadi perubahan pada sudut pandang camera utama.  Jika dipencet angka 1    Jika dipencet angka 2    Jika dipencet angka 3 | |
|  | 3. Membuat hasil tangkapan (Capture) layar yang berada di dalam frame | |
| **1.** | Buat project baru dan import package dari project pengerjaan tugas sebelumnya | |
| **2.** | Buatlah UI Image dari menu Create yang ada pada Hierarchy, Pilih Create → UI → Image. Setelah itu akan muncul Image yang berada pada Canvas kemudian ganti nama tersebut menjadi frame. | |
| **3.** | Pilih frame pada Hierarchy, kemudian pada komponen Image (Script) yang berada di Inspector atur kolom Source Image menjadi InputFieldBackground. Seperti pada gambar di bawah ini: | |
| **4.** | Pada Inspector milik Frame, Rubahlah Rect Transform sesuai dengan gambar dibawah ini: | |
| **5.** | Pada komponen Image (Script), hilangkan centang pada Fill Center. | |
| **6.** | Buatlah UI Raw Image dari menu Create yang terdapat pada Hierarchy, Pilih Create → UI → RawImage. Setelah itu akan muncul pada Canvas GameObject kemudian ganti nama menjadi Photo. | |
| **7.** | Pada Hierarchy pilih Photo, kemudian pada Inspector carilah komponen Raw Image (Script) dan atur kolom Texture menjadi None (Texture). Setelah itu hilangkan centang pada kolom Photo yang berada di atas | |
| **8.** | Langkah selanjutnya atur Rect Transform pada Photo sesuai gambar dibawah ini: | |
| **9.** | Kemudian buatlah script C# dengan code seperti dibawah ini kemudian berilah nama ScreenTexture. | |
| **10.** | Setelah itu drag script C# tersebut menuju Main camera yang berada pada multipurpose Camera Rig -> pivot | |
| **11.** | Pada Inspector milik Main Camera, carilah komponen Screen Texture (Script) kemudian atur kolom Photo GUI dengan memilih Photo. Sama halnya dengan kolom Frame GUI dengan memilih Frame. | |
| **12.** | Jalankan program, maka anda akan bisa meng-capture layar yang berada dalam frame dengan cara menekan klik kiri pada mouse. Hasil capture akan tampil pada pojok kiri atas layar. | |
| **KODE PEMROGRAMAN** | | |
| **1.** | **Camera Switch**  using System.Collections;  using System.Collections.Generic;  using UnityEngine;  public class CameraSwitch : MonoBehaviour  {  public GameObject[] cameras;  public string[] shortcuts;  public bool changeAudioListener = true;    void Update()  {  if (Input.anyKeyDown)  {  for (int i = 0; i < cameras.Length; i++)  {  if (Input.GetKeyDown(shortcuts[i]))  SwitchCamera(i);  }  }  }  void SwitchCamera(int index)  {  for (int i = 0; i < cameras.Length; i++)  {  if (i != index)  {  cameras[i].GetComponent<Camera>().enabled = false;  if (changeAudioListener)  cameras[i].GetComponent<AudioListener>().enabled = false;  }  else  {  cameras[i].GetComponent<Camera>().enabled = true;  if (changeAudioListener)  cameras[i].GetComponent<AudioListener>().enabled = true;  }  }  }  } | |
| **2.** | **PictureInPicture**  using System.Collections;  using System.Collections.Generic;  using UnityEngine;  public class PictureInPicture : MonoBehaviour  {  public enum hAlignment { left, center, right };  public enum vAlignment { top, middle, bottom };  public hAlignment horAlign = hAlignment.left;  public vAlignment verAlign = vAlignment.top;  public enum UnitsIn { pixels, screen\_percentage };  public UnitsIn unit = UnitsIn.pixels;  public int width = 50;  public int height = 50;  public int xOffset = 0;  public int yOffset = 0;  public bool update = true;  private int hsize, vsize, hloc, vloc;  void Start()  {  AdjustCamera();  }  void Update()  {  if (update)  AdjustCamera();  }  void AdjustCamera()  {  int sw = Screen.width;  int sh = Screen.height;  float swPercent = sw \* 0.01f;  float shPercent = sh \* 0.01f;  float xOffPercent = xOffset \* swPercent;  float yOffPercent = yOffset \* shPercent;  int xOff;  int yOff;  if (unit == UnitsIn.screen\_percentage)  {  hsize = width \* (int)swPercent;  vsize = height \* (int)shPercent;  xOff = (int)xOffPercent;  yOff = (int)yOffPercent;  }  else  {  hsize = width;  vsize = height;  xOff = xOffset;  yOff = yOffset;  }  switch (horAlign)  {  case hAlignment.left:  hloc = xOff;  break;  case hAlignment.right:  int justifiedRight = (sw - hsize);  hloc = (justifiedRight - xOff);  break;  case hAlignment.center:  float justifiedCenter = (sw \* 0.5f) - (hsize \* 0.5f);  hloc = (int)(justifiedCenter - xOff);  break;  }  switch (verAlign)  {  case vAlignment.top:  int justifiedTop = sh - vsize;  vloc = (justifiedTop - (yOff));  break;  case vAlignment.bottom:  vloc = yOff;  break;  case vAlignment.middle:  float justifiedMiddle = (sh \* 0.5f) - (vsize \* 0.5f);  vloc = (int)(justifiedMiddle - yOff);  break;  }  GetComponent<Camera>().pixelRect = new Rect(hloc, vloc, hsize, vsize);  }  } | |
| **3.** | using UnityEngine;  using UnityEngine.UI;  using System.Collections;  /\* ----------------------------------------  \* class to demonstrate how to take snapshots  \* of the screen and use it as a GUI texture  \*/  public class ScreenTexture : MonoBehaviour  {  // Gameobject variable for the GUI object where to display the texture  public GameObject photoGUI;  // Gameobject variable for the GUI object to be used as frame  public GameObject frameGUI;  // Float variable for the ratio between size of the snapshot and displayed texture  public float ratio = 0.25f;  /\* ----------------------------------------  \* During Update, detect if the left mouse button was pressed,  \* starting the CaptureScreen() coroutine, if so.  \*/  void Update()  {  if (Input.GetKeyUp(KeyCode.Mouse0))  // IF the left mouse button was pressed, THEN start the CaptureScreen coroutine  StartCoroutine(CaptureScreen());  }  /\* ----------------------------------------  \* A function to calculate the dimension and location of the snapshot,  \* capture it and apply it to its respective GUI element  \*/  IEnumerator CaptureScreen()  {  // Disable GUI element for the last snapshot taken (otherwise it will be superposed to the next snapshot)  photoGUI.SetActive(false);  // A shorthand for the screen's width  int sw = Screen.width;  // A shorthand for the screen's height  int sh = Screen.height;  // A shorthand for the Rect Transform settings of the GUI element for the framing  RectTransform frameTransform = frameGUI.GetComponent<RectTransform>();  // Rect for the snapshot area, initially based on the GUI frame's the Rect Transform  Rect framing = frameTransform.rect;  // A shorthand for the coordinates of the GUI frame's pivot  Vector2 pivot = frameTransform.pivot;  // A 2D vector for the Anchor Min (defines horizontal and vertical origin of the frame)  Vector2 origin = frameTransform.anchorMin;  // Convert X coordinate of origin point to pixels by multiplying it by screen's width  origin.x \*= sw;  // Convert Y coordinate of origin point to pixels by multiplying it by screen's height  origin.y \*= sh;  // float var for horizontal offset of the frame, obtained by multiplying horizontal pivot point by frame width  float xOffset = pivot.x \* framing.width;  // Add horizontal offset to frame horizontal origin  origin.x += xOffset;  // float var for vertical offset of the frame, obtained by multiplying vertical pivot point by frame height  float yOffset = pivot.y \* framing.height;  // Add vertical offset to frame vertical origin  origin.y += yOffset;  // Offset framing horizontal location  framing.x += origin.x;  // Offset framing vertical location  framing.y += origin.y;  // Int variable for texture width, based on framing width  int textWidth = (int)framing.width;  // Int variable for texture height, based on framing height  int textHeight = (int)framing.height;  // Create a new Texture measuring textWidth x textHeight  Texture2D texture = new Texture2D(textWidth, textHeight);  // Wait for the EndOfFrame before capturing snapshot  yield return new WaitForEndOfFrame();  //Read Pixels from screen  texture.ReadPixels(framing, 0, 0);  // Apply captured pixels onto texture  texture.Apply();  // Re-activate GUI element for displaying snapshot  photoGUI.SetActive(true);  // 3D Vector for the new snapshot dimension (based on framing dimension multplied by selected ratio)  Vector3 photoScale = new Vector3(framing.width \* ratio, framing.height \* ratio, 1);  // Resize GUI texture display to specified dimensions  photoGUI.GetComponent<RectTransform>().localScale = photoScale;  // Set captured texture as GUI display's texture  photoGUI.GetComponent<RawImage>().texture = texture;  }  } | |
| **KESIMPULAN** | | |
| **Saya sudah berhasil membuat game, yang dimana object atau karakter tersebut bisa bergerak atau berjalan. Dan di dalam game tersebut saya sudah bisa meng capture, mengatur posisi sudut (View) dari kamera dan berhasil menampilkan view camera dari arah sudut pandang berbeda yang terletak pada kanan atas layar** | | |
| **SWAFOTO MAHASISWA+PROJECT** | |  |
|  | | |

Format laporan:

|  |  |  |
| --- | --- | --- |
| NIM | : | 1841720167 |
| Nama | : | Raihan Rachmadani |
| Kelas | : | TI – 3H |
| Swafoto | : |  |
| Deskripsi Game   1. Nama Game 2. Alur Game (print screen dan penjelasan) 3. Komponen materi yang dipakai (disertakan screenshoot) 4. Asset yang dipakai (sertakan screenshoot, jika terdapat asset yang digunakan berasal dari internet atau sumber lain, cantumkan link) | :  :  :  :  : | Sudah, ada di atas  Game berjalan  Untuk alurnya disini hanya berjalan mengelilingi bola- bola putih    Disini menggunakan banyak komponen berjumlah 17 komponen    Disini saya selain menggunakan asset yang diberikan oleh dosen pada praktikum, saya juga mendowload asset dari internet bernama Fantasi Skybox Free dan Supercyan Character Pack Free  Link Download assets ([Tutorial Livander Gamedev (mediafire.com)](http://www.mediafire.com/file/poy49yqleqf2wwr/file) ) |