**Lab Prep 2**

|  |  |
| --- | --- |
| Student Name | Lillian Nose |

**1. Preparation for Augmented Reality (AR) Technology (15 pts)**

There are different types of AR technology such as Marker Based Augmented Reality, Markerless Augmented Reality, Projection Based Augmented Reality, Superimposition Based Augmented Reality, and probably more.

Please do your own research and describe each category in one paragraph. Try at least one or more example apps for each category and answer for the questions below.

Feel free to use the Android tablets in the lab. You might have to use your own accounts, but be careful not to sync your personal things from your account and make sure to log out after use it.

Please create table like below for each category:

|  |  |
| --- | --- |
| 1. Category | Marker Based Augmented Reality |
| Description | Marker based AR apps need a link to the real world, whether this be by a logo or a distinct figure. These figures are known as markers and connect to the virtual world by first understanding what its surroundings are. This is especially useful for QR codes and special logos that provide more information when scanned. Image recognition is key when creating a marker based augmented reality app and the markers are most commonly in black and white. The marker makes it easier for programmers to visualize the image beforehand and create specific scenarios for them. |

|  |  |
| --- | --- |
| 1. App name | Strata AR |
| (1) | Write **one/two** sentence(s) to describe what this app is about. |
| ­­­­­­­­­­ | Answer:  This app displays animations and leads to videos depending on what image the app is scanning. It provides a list of markers to try and play around with which all have unique actions. |
| (2) | Write **two or more** good things about this app . |
| ­­­­­­­­­­ | Answer: I like that this app isn’t super repetitive with what the markers do and provides a lot of variety of images. The instructions were also very clear and unproblematic. |
| (3) | Write **two or more** improvements to suggest. |
|  | Answer: I would provide more interesting markers with different animations that actually add to the experience. Some of the videos and pictures they covered over the marker didn’t fit the marker size and detracted from the immersion so I would try to fix that. |
| (4) | Write **one/two** sentence(s) of ideas how to apply to the goals of our stream.  I like the idea of having an interactive photo gallery, this could be utilized to maybe introduce artists with a story board of their life that comes to life. |
| ­­­­­­­­­­ | Answer: |

|  |  |
| --- | --- |
| 2. Category | Markerless Augmented Reality |
| Description | Much unlike Marker-based AR, there is no previous knowledge about surroundings being used. Users can point their camera anywhere without the worry of capturing a marker and a virtual world will be integrated with reality no matter what direction you turn. Common uses of Markerless Augmented reality are for retail purposes, it is especially useful in cases of comparing two objects: one being virtual and the other in real life. Furniture stores are using this technology to allow potential customers to test different pieces in the comfort of their own home. |

|  |  |
| --- | --- |
| 2. App name | Stack AR |
| (1) | Write **one/two** sentence(s) to describe what this app is about. |
| ­­­­­­­­­­ | Answer: The game has a simple premise; the goal is to stack blocks as high as possible without missing the target or falling off. There is no endgame, just an incentive to beat your personal high score. |
| (2) | Write **two or more** good things about this app . |
| ­­­­­­­­­­ | Answer:  As simple as the game is, it gets really addicting and the simplicity of it makes it easy to play without putting in a lot of effort. The AR aspect of this made it a bit interesting as you could change your view of the blocks by moving the phone, whereas in the original version you were restricted to one point of view. |
| (3) | Write **two or more** improvements to suggest. |
|  | Answer: When I used this app, I used a flat surface and it somehow configured in the air; this could definitely be improved upon as it ruined the illusion that it sought after. Another thing that could be improved is the fact that there is not much happening besides the blocks stacking. With such a simple game, an entertaining background would help engage users, whether this be by adding special effects or what not. |
| (4) | Write **one/two** sentence(s) of ideas how to apply to the goals of our stream. |
| ­­­­­­­­­­ | Answer:  I like the idea of combining games or animations in the scope of the real world, I feel that this personalizes experiences for users and could be useful to entertain visitors while educating them. |

|  |  |
| --- | --- |
| 3. Category | Projection Based Augmented Reality |
| Description | This type of Augmented Reality utilizes projection of lights onto the surface of the real world and sensors to interact with the projection. Projection Based AR depends on a two-dimensional plane but doesn’t require a personal device for all users and can reach a larger audience. The experience fails to be fully immersive however as it is still just a projection on a 2d surface. This experience would enhance physical settings and allow users to interact differently with their surroundings. |

|  |  |
| --- | --- |
| 3. App name | Dynamic Shader Lamps https://www.youtube.com/watch?v=u58Di91qbDw |
| (1) | Write **one/two** sentence(s) to describe what this app is about. |
| ­­­­­­­­­­ | Answer: This app allows for users to color on a blank flat surface with a  Virtual paint palette through a “paint brush” tool that could not only choose  Colors projected but could paint with the same tool. |
| (2) | Write **two or more** good things about this app . |
| ­­­­­­­­­­ | Answer:  I like the fact that this is such an easy way to have multiple people have access to AR without the need for a headset or other devices. This allows for any blank wall to be made much more interesting and doesn’t require as much hardware. |
| (3) | Write **two or more** improvements to suggest. |
|  | Answer: The distortion of the image whenever a hand passes through the projection ruins a bit of the immersive experience but it’s to be expected with a projection. I would also decorate the surface a bit more to look more impressive than simply drawing on a whiteboard. |
| (4) | Write **one/two** sentence(s) of ideas how to apply to the goals of our stream. |
| ­­­­­­­­­­ | Answer: I think that this unlocks potential for settings to be more realistic with the ability to have real physical objects appear as others with projection and interactivity. |

|  |  |
| --- | --- |
| 4. Category | Superimposition Based Augmented Reality |
| Description | Superimposition Based AR replaces the image partially or fully using object recognition. This is useful in cases where images are needed to be layered on the real world to provide more information or to use as reference. Some examples of this include doctors overlaying a patient’s X-RAY over the physical body part or for military use for targeting multiple enemies simultaneously. |

|  |  |
| --- | --- |
| 4. App name | Measure by Apple |
| (1) | Write **one/two** sentence(s) to describe what this app is about. |
| ­­­­­­­­­­ | Answer: This app utilizes AR to provide the accessibility of measuring anything without the hassle of using a ruler. It stores the measurements and makes a menial task efficient. |
| (2) | Write **two or more** good things about this app . |
| ­­­­­­­­­­ | Answer: This app has proven to be fairly accurate and much quicker than it would take to do manually. One small detail I especially like is the feature of quick connecting two segments by a vector point. |
| (3) | Write **two or more** improvements to suggest. |
|  | Answer: It would be useful if there was a better storage method of the measurements, for example a box’s length width and height could be saved. Right now, the only option is to measure one item at a time and write down the measurements or to take a screenshot of the results. Another thing would be to make the measurements customizable for users to adjust color or size of the text. |
| (4) | Write **one/two** sentence(s) of ideas how to apply to the goals of our stream. |
| ­­­­­­­­­­ | Answer: This app inspires me to value utility and efficiency when thinking about what visitors would appreciate in a museum setting. |

If you have more than one example in the same category, create more tables from the above.

**2. Building your Unity game to an Android device (15 pts)**

Please follow the “Lab 2 - 1 Building your Unity game to an Android device” document from the Canvas, and complete the work.

Submit your work both with a unity package (.unitypackage), and Android Package Kit (.apk) file as you completed. If you use other device than Android tablet in the lab, you should specify what device and what version of Android you used.

Submitting both your unity package file and apk file, describe below for your work as you completed: Control of balls by moving the mobile device, Zoom in and Zoom out, Pan the camera view, Adjustment of Camera view, etc.

|  |  |
| --- | --- |
| 1. Package Name & apk Name | Prep2\_AndroidApp\_LastName\_Firstname.unitypackage &  Prep2\_AndroidApp\_LastName\_Firstname.apk |
| (1) | Describe what you completed in your Unity package |
| ­­­­­­­­­­ | I modified my unity game to utilize the accelerometer and to pan the camera and pinch and zoom using mobile touch inputs. The accelerometer script was added to the player whereas the pinch and zoom functions were added to the camera. |
| (2) | Copy of your main part of your codes to describe here with comments |
|  | FOR THE ACCELEROMETER  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  void Update () {         //get input data from keyboard or controller         float moveHorizontal = Input.GetAxis("Horizontal");         float moveVertical = Input.GetAxis("Vertical");          Vector3 dir = Vector3.zero;  //stores the acceleration of the y component         dir.x = -Input.acceleration.y;         //stores the acceleration of the y component  dir.z = Input.acceleration.x;          if (dir.sqrMagnitude > 1)             dir.Normalize();          dir \*= Time.deltaTime;  //Multiplies Vector dir by the time passed           transform.Translate(dir \* speed);  //moves the player      }  FOR PINCH ZOOM      private static readonly float PanSpeed = 20f;     private static readonly float ZoomSpeedTouch = 0.1f;     private static readonly float ZoomSpeedMous = 0.5f;      private static readonly float[] BoundsX = new float[] { -10f, 5f };     private static readonly float[] BoundsZ = new float[] { -18f, -4f };     private static readonly float[] ZoomBounds = new float[] { 10f, 85f };      private Camera cam;      private Vector3 lastPanPosition;     private int panFingerId;      private bool wasZoomingLastFrame;     private Vector2[] lastZoomPositions;      private void Awake()     {         cam = GetComponent<Camera>();     }     // Use this for initialization     void Start()     {       }      // Update is called once per frame     void Update()     {         if(Input.touchSupported && Application.platform != RuntimePlatform.WebGLPlayer)         {             HandleTouch();          }       }      void HandleTouch()     {         switch (Input.touchCount)         {              case 1: // Panning                 wasZoomingLastFrame = false;                  // If the touch began, capture its position and its finger ID.                 // Otherwise, if the finger ID of the touch doesn't match, skip it.                 Touch touch = Input.GetTouch(0);                 if (touch.phase == TouchPhase.Began)                 {                     lastPanPosition = touch.position;                     panFingerId = touch.fingerId;                 }                 else if (touch.fingerId == panFingerId && touch.phase == TouchPhase.Moved)                 {                     PanCamera(touch.position);                 }                 break;              case 2: // Zooming                 Vector2[] newPositions = new Vector2[] { Input.GetTouch(0).position, Input.GetTouch(1).position };                 if (!wasZoomingLastFrame)                 {                     lastZoomPositions = newPositions;                     wasZoomingLastFrame = true;                 }                 else                 {                     // Zoom based on the distance between the new positions compared to the                      // distance between the previous positions.                     float newDistance = Vector2.Distance(newPositions[0], newPositions[1]);                     float oldDistance = Vector2.Distance(lastZoomPositions[0], lastZoomPositions[1]);                     float offset = newDistance - oldDistance;                      ZoomCamera(offset, ZoomSpeedTouch);                      lastZoomPositions = newPositions;                 }                 break;              default:                 wasZoomingLastFrame = false;                 break;         }     }      void PanCamera(Vector3 newPanPosition)     {         // Determine how much to move the camera         Vector3 offset = cam.ScreenToViewportPoint(lastPanPosition - newPanPosition);         Vector3 move = new Vector3(offset.x \* PanSpeed, 0, offset.y \* PanSpeed);          // Perform the movement         transform.Translate(move, Space.World);          // Ensure the camera remains within bounds.         Vector3 pos = transform.position;         pos.x = Mathf.Clamp(transform.position.x, BoundsX[0], BoundsX[1]);         pos.z = Mathf.Clamp(transform.position.z, BoundsZ[0], BoundsZ[1]);         transform.position = pos;          // Cache the position         lastPanPosition = newPanPosition;     }      void ZoomCamera(float offset, float speed)     {         if (offset == 0)         {             return;         }          cam.fieldOfView = Mathf.Clamp(cam.fieldOfView - (offset \* speed), ZoomBounds[0], ZoomBounds[1]);     } |
| (3) | Details of explanation of your implementation. Algorithms, Methods, Functions, Libraries, Variables, etc. |
| ­­­­­­­­­­ | To utilize the accelerometer to move the player I modified the script for the player in the Update() method. I created a third vector that was  Set to 0,0 and then changed the vector’s x and z to the Input’s acceleration of the y and x components respectively. If the square magnitude of the vector is greater than 1, it is normalized. Otherwise, the vector is multiplied by the time and then multiplied by the speed.  For the pinch and zoom functionalities, I added this to the camera script. I initialized different variables such as PanSpeed which would measure how fast the camera would move when swiped and ZoomSpeedTouch and ZoomSpeedmous. Another important variable was initializing the Camera in the class to modify later. A new type that I didn’t use in the previous script is bool which returns either true or false as opposed to numeric values. This type is used for declaring the variable wasZoomingLastFrame. In the Update method I used an if statement to determine if the input was touch based by calling the touchSupported bool variable. If it returns true, the method HandleTouch() is called.  HandleTouch() is another method with the responsibility of controlling how touch is interpreted and how it affects the camera. A switch statement is implemented with the touch count inputed as a parameter. If the touchcount is one (Panning): wasZoomingLastFrame is set to false and if the touch was moving as can be found with TouchPhased.Moved, the PanCamera() method was called with touch.position as an argument. If there are two touches detected, this implies a zoom and the touch positions are stored in the Vector2 array. The distances of the original positions subtracted from the new touch positions are stored in the variable float offset and is passed into the method ZoomCamera() with offset and ZoomSpeedTouch passed as arguments.  Lastly, the PanCamera() method actually moves the view by the distance in passed a parameter. ZoomCamera() uses a Mathf method called Clamp that clamps a value between a minimum and a maximum value which creates the zoom. |
| (4) | References |
|  | <https://unity3d.com/learn/tutorials/topics/mobile-touch/accelerometer-input>  <https://unity3d.com/learn/tutorials/topics/mobile-touch/pinch-zoom> |