**Lab Prep 2**

|  |  |
| --- | --- |
| Student Name | Glen Joy |

**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 | **Maker Based Augmented Reality** |
| Description | Marker Based Augmented Reality involves the use of physical shapes or “markers” placed on real world objects to enable AR systems to distinctively identify them from other objects and track digital components onto them. For example, an AR application may need have to have an animation be displayed on a particular page of a book. A shape or symbol may be placed on the page to allow the application to distinguish that page from another and track the appropriate animation onto it. |

|  |  |
| --- | --- |
| 1. App name | Genesis |
| (1) | Write **one/two** sentence(s) to describe what this app is about. |
| ­­­­­­­­­­ | Answer: Genesis is an augmented reality-based trading card game. Users purchase physical trading cards from stores which give information on different fictitious characters. Players place their cards on a flat surface and battle other characters using the mobile application. The app uses **marker-based** AR to identify the characters from the physical cards and display an animation onto the user’s phone of the characters battling. |
| (2) | Write **two or more** good things about this app . |
| ­­­­­­­­­­ | Answer: This game, unlike other trading card games, allows users a unique aspect of reality and intractability. While most card-based games operate exclusively in the physical world, this application allows players to take their imagination in the characters on their cards directly to their phones. The tracking of the app was generally very good and the character animations were very amusing. |
| (3) | Write **two or more** improvements to suggest. |
|  | Answer: As a possible improvement, there should be a greater variety of character animations. In addition, some of the animations appear to be very sporadic and glitchy so smoother animations/tracking would greatly improve the user experience. |
| (4) | Write **one/two** sentence(s) of ideas how to apply to the goals of our stream. |
| ­­­­­­­­­­ | Answer: This app, or marker-based AR in general, may provide an idea of possibly integrating paper publications of the museum with AR. For example, the museum may have a flyer advertising a new exhibition. Using marker-based AR, a museum visitor could use a mobile application to see an AR projection of the new exhibition on the flyer. |

|  |  |
| --- | --- |
| 2. | **Markerless Augmented Reality** |
|  | Markerless augmented reality, unlike marker based augmented reality, does not require the use of physical markers for the AR system to identify and project digital animations upon an object or surface. While marker based augmented reality is almost always used to identify specific real-world objects, markerless is typically used to simply generate a digital projection/animation on any simple flat surface. |

|  |  |
| --- | --- |
| 1. App name | Pokémon Go! |
| (1) | Write **one/two** sentence(s) to describe what this app is about. |
| ­­­­­­­­­­ | Answer: This AR-based mobile game has users explore their surroundings outside to find AR-projected Pokémon characters. The characters are projected on the ground using **markerless** AR. Players can collect Pokémon, complete challenges, level up their Pokémon, and engage in battles with other players. |
| (2) | Write **two or more** good things about this app . |
| ­­­­­­­­­­ | Answer: As a formerly addictive player of this app, I found this app to be very fun and engaging. Unlike most apps in general, this app allows players to go outside and interact with the world around them. What I found to be very cool was that digital locations in the app such as “Poke stops” were tied to physical locations in the real world. In addition, the app tracked Pokémon characters well amongst the world and there were little to no glitches or bugs. |
| (3) | Write **two or more** improvements to suggest. |
|  | Answer: Just like the Genesis app, I believe that this app could incorporate more character animations. The Pokémon characters in the app just simply appear to “stand” in their locations with little movement. Even when interacting with the characters they appear to replay the same animation over and over. Another improvement I would consider would be more player customization rather than the user sticking to the constructs of the game. |
| (4) | Write **one/two** sentence(s) of ideas how to apply to the goals of our stream. |
| ­­­­­­­­­­ | Answer: This game, and markerless AR in general, could provide ideas for AR apps that allow users to see a digital object through their phones projected on a museum floor or other surface in the museum. This could allow visitors to have an added experience in the museum than just be limited to the physical exhibits and artwork on display. |

|  |  |
| --- | --- |
| 3. | **Projection Based Augmented Reality** |
|  | Projection based augmented reality involves the use of projectors to project text, images, animations, or videos onto real world objects. Unlike other forms of AR, projected based AR is not viewed on the screen of a phone or other device; it is meant to be seen in person in physical light. |

|  |  |
| --- | --- |
| 1. App name | Light Guide Systems™ Projection AR |
| (1) | Write **one/two** sentence(s) to describe what this app is about. |
| ­­­­­­­­­­ | Answer: The projection AR technology from Light Guide Systems is an advanced manufacturing technology that enables manufacturing workers to see manufacturing instructions and graphics on physical materials they are working through physical sensors and projectors which project the graphics onto the materials. |
| (2) | Write **two or more** good things about this app. |
| ­­­­­­­­­­ | Answer: This application of projection AR technology allows for added guidance to help workers complete tasks successfully, consistently, and efficiently according to Light Guide Systems. In addition, this technology eliminates the previous need for workers to look back and forth between a monitor for assistance and their workspace. The application of this technology specifically from this company also includes an audit feature where every step of the manufacturing process is recorded and audited so any mistakes that were to occur can be traced and resolved. |
| (3) | Write **two or more** improvements to suggest. |
|  | Answer: Although I have not physically used this technology, one possible improvement/idea that could be added is the ability to make personal configurations to the graphics being displayed. For example, if a particular worker follows instructions better with text in a certain color or if the text is audibly read out, then the worker should have the ability to make change such settings on the application to his/her individual needs. Another idea could be the use of smaller projectors to enable portability of workspaces and possibly lesser cost. |
| (4) | Write **one/two** sentence(s) of ideas how to apply to the goals of our stream. |
| ­­­­­­­­­­ | Answer: This technology could be used to project dynamic graphics over museum artwork/exhibits. These graphics could give visitors information on the pieces and possibly allow the visitor to even interact with graphics. For example, a visitor could inquire about a certain aspect about an artwork and that graphic would appear over/next to the piece. Another possibility are for projectors to be placed over hallways of the museum facing the floor which provide graphics to help guide visitors on self-tours. |

|  |  |
| --- | --- |
| 4. | **Superimposition Based Augmented Reality** |
|  | Superimposition based augmented reality is augmented reality which replaces all or a portion of a physical object with an augmented form of the object. |

|  |  |
| --- | --- |
| 1. App name | Google Translate |
| (1) | Write **one/two** sentence(s) to describe what this app is about. |
| ­­­­­­­­­­ | Answer: The Google Translate application is most commonly known to simply translate text in one language to another. However, the mobile app version has a feature which allows a user to use their camera to point at a piece of text and have it superimposed with the text in the language you are attempting to translate to. |
| (2) | Write **two or more** good things about this app. |
| ­­­­­­­­­­ | Answer: The app is extremely useful as it contains an expansive list of available languages. The AR feature of the app is very advanced and does a good job of detecting the text, identifying the language, and replacing the text with text in the language you are translating to. The AR feature also adjusts the color of the text and its background so it appears that the text indeed belongs with the rest of the image. |
| (3) | Write **two or more** improvements to suggest. |
|  | Answer: One suggestion for the AR feature of the app is better text tracking. There are some cases where the text appears to glitch and move even when the camera is nearly still. Another suggestion is removing the necessity for the user to download an offline file in order to use real-time translation. |
| (4) | Write **one/two** sentence(s) of ideas how to apply to the goals of our stream. |
| ­­­­­­­­­­ | Answer: Superimposition based AR could be used to superimpose alternate versions or interpretations of artwork on an original piece at a museum. |

**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 |
| ­­­­­­­­­­ | In my Unity package, I was able to modify my existing project from Lab Deliverable 1 and make it compatible for an Android device and ready for testing and debugging. In addition, I was able to have the ball move with the tilt of the tablet as well as add the ability to pinch, zoom, and pan around the environment. |
| (2) | Copy of your main part of your codes to describe here with comments |
|  | **//function under PlayerControll class**  void FixedUpdate() { **//function to update ball movement**  **//variables to store x and y positions**  float moveHorizontal = Input.GetAxis("Horizontal");  float moveVertical = Input.GetAxis("Vertical");  **//setting the x and y based on the input acceleration**  moveHorizontal = Input.acceleration.x;  moveVertical = Input.acceleration.y; Vector3 movement = new Vector3(moveHorizontal, 0.0f, moveVertical);  **//adds force to the RigidBody (the ball)**  rb.AddForce(movement \* speed \* 2);  }  **//the following portion of code is from the //CameraHandling class in its HandleTouch function**  switch (Input.touchCount){   case 1: **// Case where user is attempting to pan**    zoomActive = false;  **// If the touch begins, get its position  // 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;       panActive = true;     }else if (touch.fingerId == panFingerId && touch.phase == TouchPhase.Moved) {        PanCamera(touch.position);     }    break;     case 2: **// If the user is attemptimg to zoom**     panActive = false;       Vector2[] newPositions = new Vector2[] {Input.GetTouch(0).position, Input.GetTouch(1).position};      if (!zoomActive){         lastZoomPositions = newPositions;         zoomActive = 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:      panActive = false;       zoomActive = false;       break;     }   } |
| (3) | Details of explanation of your implementation. Algorithms, Methods, Functions, Libraries, Variables, etc. |
| ­­­­­­­­­­ | Both the code for having the ball move to the tilt of the tablet as well as the code for enabling panning and zooming are all centered around their respective update functions which are called once per frame. The PlayerController and CameraHandler classes both use the same default packages/libraries for their implementation, however, the PlayerController class has the added UnityEngine.UI library to facilitate user input. The PlayerController class works overall by setting the x and y positions of the ball equal to the input acceleration which is based on the user tilt of the tablet. Those values are then placed into a vector and a force is applied to the ball to make it move a certain amount by multiplying the vector by a set speed and constant. The CameraHandler class works by first detecting whether the user is attempting to pan or zoom. If attempting to pan, the position of the camera is moved alongside the position of the finger detected. If attempting to zoom, two individual touches should be detected and the difference between the two locations of the touch pinch/shrink the field of view of the camera accordingly. |
| (4) | References |
|  | <https://answers.unity.com/questions/976817/roll-a-ball-%20how-do-i-control-the-ball-with-the-gyr.html>  <https://kylewbanks.com/blog/unity3d-panning-and-pinch-to-zoom-camera-with-touch-and-mouse-input> |