MULTI-MARKER TABLETOP AUGMENTED REALITY FOR ANAMORPHIC ILLUSIONS TECHNICAL MANUAL

A Thesis Proposal
Presented to
the Faculty of the College of Computer Studies
De La Salle University Manila

In Partial Fulfillment
of the Requirements for the Degree of
Bachelor of Science in Computer Science

by

BUHION, Deborah Rose P.
DIZON, Michaela Nicole P.
GO, Thea Ellen T.
OAFALLAS, Kenneth Neil B.

Neil Patrick DEL GALLEGO
Adviser

May 27, 2022

Multi-Marker Tabletop Augmented Reality for Anamorphic Illusions

The system is a mobile marker-based augmented reality (AR) anamorphic puzzle game. It uses the concept of anamorphic illusions to produce levels that can be solved by moving the camera physically. It also implements the ability to track multiple AR markers, allowing players to customize their play area.

Getting Started

Prerequisites

- Unity Engine Version 2020.3.9
- Vuforia Version 10.6.3

Installation

1. Clone the repository.
2. Open the project in Unity.
3. Import Vuforia Engine into the Unity project from the menu **Assets → Import Package → Custom Package.**

Project Files

- There are a total of ten puzzles in the system. These files can be found under **Scenes/Final**.
- Scripts that are attached to the objects in the scenes can be found under the **Scripts** folder.
- The models that were used, both whole and sliced, can be found under the **Models** folder.

Running the Application

The application APK installer is available and can be downloaded from a public Google Drive link.

System Requirements

As basis for the minimum camera specifications, the oldest specified mobile device is used For better performance, the system recommends an OS of Android 8 or higher, API 24 and above, at least 4GB of RAM, at least 512MB of free storage space, and a rear camera with a resolution of at least 8MP.

System Summary

The application works stably for low-end mobile devices. The application requires installation to the mobile device, and needs the prescribed markers to enable gameplay. The markers can be found under **Assets/Markers** or downloaded here.

System Configuration

The application operates on a mobile device with Android OS 6.0 or newer. The application was developed with Unity, Vuforia and C. For smooth gameplay, it is recommended that the mobile device running the application runs Android OS 8 or newer at least 4GB of RAM, at least 512MB of free storage space, and a rear camera with a resolution of at least 8MP.

CLASS	AnamorphicTransformer
SUPERCLASS	MonoBehaviour
PROPERTIES	1. camera a. Type: GameObject b. Purpose: holds the main camera for solution checking c. Constraints: must be the Vuforia main camera 2. markerArray a. Type: GameObject Array b. Purpose: holds the prescribed marker objects c. Constraints: None 3. sliceArray a. Type: GameObject Array b. Purpose: holds the original sliced model c. Constraints: None 4. solutionPoint a. Type: GameObject b. Purpose: for referencing solution transform position c. Constraints: Deactivated in editor 5. markerOrderedArray a. Type: Marker ArrayList b. Purpose: holds ordered list of markers obtained from broadcast params c. Constraints: Only set on receiving broadcast params 6. newMode a. Type: string b. Purpose: current marker mode c. Constraints: Only from constants in Tracking_Modes 7. model a. Type: GameObject b. Purpose: holds the completed model for end screen c. Constraints: None 8. markerdist a. Type: float b. Purpose: holds the maximum x and y distances among the markers in targePosition c. Constraints: targetPosition has valid values 9. scale a. Type: float b. Purpose: holds the scale of current level based on positions in targetPosition c. Constraints: targetPosition has valid values 10. centerPosition a. Type: Vector3 b. Purpose: holds the mean of positions in targetPosition c. Constraints: targetPosition has valid values 11. targetPosition a. Type: Vector3 List

	b. Purpose: holds positions of markers currently tracked c. Constraints: None 12. tracking a. Type: bool b. Purpose: if currently tracking for if system should check solution position c. Constraints: None
METHODS	1. Update a. Description: Updated per frame, checks if camera coincides with solutionPoint b. Parameters: None c. Return Type: Void d. Constraints: None 2. OnMarkerModeChange a. Description: calls functions to reposition, perform anamorphic transformation, and tracking status when marker mode changes b. Parameters: Parameters parameters c. Return Type: Void d. Constraints: called on observed event 3. Anamorphosize a. Description: performs anamorphic transformation b. Parameters: None c. Return Type: Void d. Constraints: None 4. Reposition a. Description: Repositions and scales the scene to the center of the markers b. Parameters: None c. Return Type: Void d. Constraints: None

CLASS	ChangeScene
SUPERCLASS	MonoBehaviour
PROPERTIES	None
METHODS	1. LoadScene a. Description: Loads a scene b. Parameters: string sceneName c. Return Type: void d. Constraints: None 2. QuitGame a. Description: Closes the application b. Parameters: None

	c. Return Type: None d. Constraints: None 3. UnloadScene a. Description: Destroys all GameObjects associated with the given Scene and removes the Scene from the SceneManager when changing levels b. Parameters: string sceneName c. Return Type: Void d. Constraints: None

CLASS	Marker
SUPERCLASS	None
PROPERTIES	1. markerObject a. Type: GameObject b. Purpose: holds Vuforia target object that this Marker object represents c. Constraints: is Vuforia image target object 2. name a. Type: string b. Purpose: name of marker other modules will reference c. Constraints: name comes from global param constant list 3. status a. Type: bool b. Purpose: true if marker is currently being tracked, false otherwise c. Constraints: None
METHODS	1. Marker constructor a. Description: constructs Marker object with value setting for all properties b. Parameters: GameObject markerObject, string name, bool status c. Return Type: None d. Constraints: None 2. GetMarkerObject a. Description: returns markerObject b. Parameters: None c. Return Type: GameObject d. Constraints: GameObject is a Vuforia image target object 3. GetStatus a. Description: returns status b. Parameters: None c. Return Type: bool d. Constraints: None 4. GetName

	a. Description: returns name b. Parameters: None c. Return Type: string d. Constraints: None 5. SetStatus a. Description: sets status b. Parameters: bool status c. Return Type: Void d. Constraints: None

CLASS	MarkerDetection
SUPERCLASS	MonoBehaviour
PROPERTIES	1. markerArray a. Type: GameObject Array b. Purpose: holds default image markers c. Constraints: GameObjects are Vuforia image target objects 2. markerDict a. Type: Dictionary<string, Marker> b. Purpose: allows random referencing of names to Marker objects, instead of checking each Marker object c. Constraints: one name for each object in markerArray, only for objects in markerArray 3. markerOrderedArray a. Type: Marker ArrayList b. Purpose: holds Markers in an ordered way to broadcast to other systems c. Constraints: For 4 markers, element 0 is at top left, element 1 at bottom left, element 2 at top right, element 3 at bottom right. For 2 markers, element 0 at top, element 1 at bottom. 4. trackingMode a. Type: string b. Purpose: tracks and holds current tracking mode to broadcast to other systems c. Constraints: from the Marker_Names list
METHODS	1. Awake a. Description: sets broadcast observers and populates markerDict b. Parameters: None c. Return Type: Void d. Constraints: None 2. OnMarkerChange a. Description: Called when a marker enters or leaves tracking, and updates status of marker in markerDict b. Parameters: Parameters parameters (holds String marker

	name and bool marker status) c. Return Type: Void d. Constraints: Called only on broadcast event 3. CheckMarkerMode a. Description: Counts markers with true status in markerDict b. Parameters: None c. Return Type: string, representing new marker mode d. Constraints: None 4. SetMarkerOrderedArray a. Description: determines relative positions of markers and orders them according to position. For 4 markers, element 0 is at top left, element 1 at bottom left, element 2 at top right, element 3 at bottom right. For 2 markers, element 0 at top, element 1 at bottom. b. Parameters: None c. Return Type: Void d. Constraints: None 5. getCenterPosition a. Description: finds relative center of tracked markers b. Parameters: List<Vector3> v c. Return Type: Vector3 d. Constraints: accessory to SetMarkerOrderedArray 6. MarkerModeChangeSignal a. Description: broadcasts new marker mode b. Parameters: string mode c. Return Type: Void d. Constraints: accessory to CheckMarkerMode 7. OnDestroy a. Description: removes broadcast observers on destroy b. Parameters: None c. Return Type: Void d. Constraints: None

CLASS	MarkerSignaling
SUPERCLASS	MonoBehaviour
PROPERTIES	None
METHODS	1. MarkerFoundSIgnal a. Description: Broadcasts event with name of marker and true tracking status. Called when the parent marker is tracked. b. Parameters: string marker c. Return Type: Void d. Constraints: None 2. MarkerLostSignal

	a. Description: Broadcasts event with name of marker and false tracking status. Called when the parent marker is lost. b. Parameters: string marker c. Return Type: Void d. Constraints: None

CLASS	ResetButton
SUPERCLASS	MonoBehaviour
PROPERTIES	1. resetButton a. Type: Button b. Purpose: holds reset button object c. Constraints: None
METHODS	1. Start a. Description: adds an onclick listener to the button b. Parameters: None c. Return Type: Void d. Constraints: None 2. TaskOnClick a. Description: Disables then immediately enables the VuforiaBehavior component b. Parameters: None c. Return Type: Void d. Constraints: None

CLASS	UI
SUPERCLASS	MonoBehaviour
PROPERTIES	1. title a. Type: Image b. Purpose: background image c. Constraints: None 2. startBtn a. Type: Button b. Purpose: moves to level select sceen c. Constraints: None 3. exitBtn a. Type: Button

	b. Purpose: closes the application c. Constraints: None 4. helpBtn a. Type: Button b. Purpose: toggles instructions screen c. Constraints: None 5. instructions a. Type: Image b. Purpose: instructions screen c. Constraints: None 6. closeBtn a. Type: Button b. Purpose: closes instructions screen c. Constraints: None
METHODS	1. Start a. Description: hides instructions and close button at start b. Parameters: None c. Return Type: Void d. Constraints: None 2. showSelection a. Description: disables all buttons to show level select b. Parameters: None c. Return Type: Void d. Constraints: None 3. showInst a. Description: shows instructions b. Parameters: None c. Return Type: Void d. Constraints: None 4. hideInst a. Description: hides instructions b. Parameters: None c. Return Type: Void d. Constraints: None

CLASS	Timer
SUPERCLASS	MonoBehaviour
PROPERTIES	1. markerArray a. Type: GameObject Array b. Purpose: holds default markers c. Constraints: None 2. whole a. Type: GameObject b. Purpose: holds golden compete version of model

	c. Constraints: None 3. fireworks a. Type: GameObjects b. Purpose: animated fireworks effects for end screen c. Constraints: None 4. modal a. Type: Canvas b. Purpose: end screen modal c. Constraints: None 5. timerText a. Type: Text b. Purpose: timer in real-time c. Constraints: None 6. finalTime a. Type: Text b. Purpose: time in end screen c. Constraints: None 7. startTime a. Type: float b. Purpose: time when the level is loaded c. Constraints: None 8. totalTime a. Type: Button b. Purpose: cumulative time solving the level c. Constraints: None 9. finalTime a. Type: Button b. Purpose: time when level is solved for display at end screen c. Constraints: None
METHODS	1. Start a. Description: hides modal and fireworks, initializes start time, and creates listener for when level is solved b. Parameters: None c. Return Type: Void d. Constraints: None 2. Update a. Description: constantly updates the displayed time b. Parameters: None c. Return Type: Void d. Constraints: None 3. Completed a. Description: shows the end screen modal, activates the fireworks and shows the completed model b. Parameters: None c. Return Type: Void d. Constraints: None 4. ShowWinPopup a. Description: delays end screen by 3 seconds so that player can see completed model

	b. Parameters: None c. Return Type: IEnumerator d. Constraints: None

CLASS	GameUI
SUPERCLASS	MonoBehaviour
PROPERTIES	1. helpBtn a. Type: Button b. Purpose: toggles instructions image c. Constraints: None 2. onBtn a. Type: Button b. Purpose: activates bounding box c. Constraints: None 3. offBtn a. Type: Button b. Purpose: deactivates bounding box c. Constraints: None 4. instructions a. Type: Image b. Purpose: instructions screen c. Constraints: None 5. boundingBox a. Type: GameObject b. Purpose: shows practical limits of play area c. Constraints: None
METHODS	1. Start a. Description: hides instructions and bounding box at start b. Parameters: None c. Return Type: Void d. Constraints: None 2. showSelection a. Description: disables all buttons to show select b. Parameters: None c. Return Type: Void d. Constraints: None 3. showInst a. Description: shows instructions b. Parameters: None c. Return Type: Void d. Constraints: None 4. closeInst a. Description: hides instructions

	b. Parameters: None c. Return Type: Void d. Constraints: None 5. showBox a. Description: shows bounding box b. Parameters: None c. Return Type: Void d. Constraints: None 6. hideBox a. Description: hides bounding box b. Parameters: None c. Return Type: Void d. Constraints: None

CLASS	SFXManager
SUPERCLASS	MonoBehaviour
PROPERTIES	1. Audio a. Type: AudioSource b. Purpose: generic audio source c. Constraints: None 2. WinSFX a. Type: AudioClip b. Purpose: played when level is solved c. Constraints: None 3. sfxInstance a. Type: SFXManager b. Purpose: singleton class instance of SFXManager c. Constraints: None
METHODS	1. Awake a. Description: initializes singleton class b. Parameters: None c. Return Type: Void d. Constraints: None