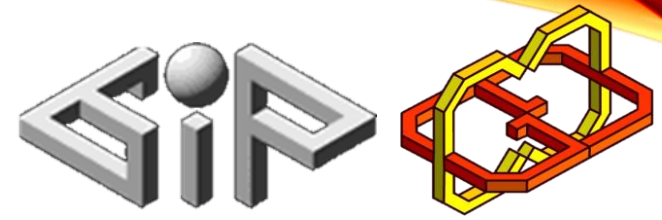


SHADOW GAMES

Bringing Yu-Gi-Oh! cards game to life using AR

By: **Or-Eli Pilosof, Yael Tsafir, Rafi Cohen**

Supervisors: **Boaz Sterenfeld, Yaron Honen**



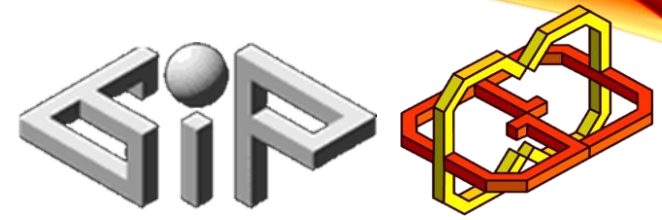
PROJECT GOALS

- Create a real-life version of a cards game
- The game will be played using the actual cards
- Apply in-game shadows according to user's input

<https://youtu.be/2ot9eV9Dybl?t=605>

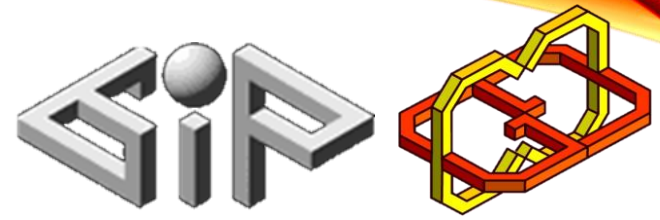


Photo 1



PROJECT OBJECTIVES

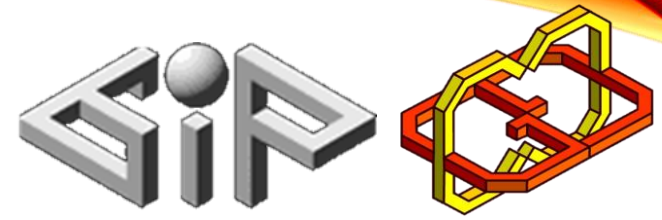
- Creating AR objects when looking at real cards
- Distinguish between different cards
- Enabling the user to adjust the AR shadows manually
- Providing interactive AR game objects
- Create a game experience where the user plays against the computer



CHALLENGES

1. Identifying unique cards via HoloLens camera

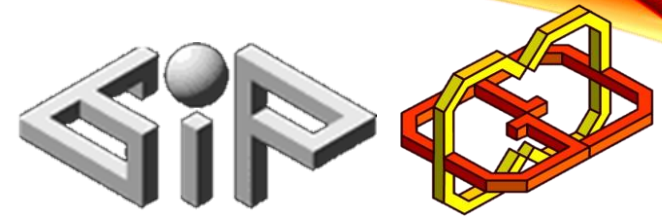
- ✓ We integrated the Vuforia SDK.
 - ✓ We create a database of pictures and upload it to Vuforia. Vuforia then creates a unique image target for each picture, which can be added to our scene and be captured by Vuforia's AR camera.
 - ✓ Each image target captured by the AR camera triggers an event.
 - ✓ We created our own event handler to handle what happens with each picture the AR camera captures.
- **Using Vuforia created another challenge for us** – It doesn't integrate well with the HoloLens when using newer versions of Unity.



CHALLENGES

2. Realtime shadow adjustment by the user

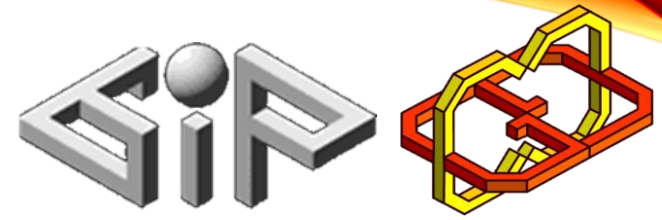
- ✓ In Unity the light is produced by an object of 'directional light'.
- ✓ When we change the rotation of the directional light in the X axis & Y axis – we change the shadows casted by the directional light.
- ✓ We created a scene for shadows adjustment, where the user can press on 4 buttons, left, right, up, down, and can see how the shadows change accordingly.
- ✓ These settings are then saved and passed on to the game scene.



CHALLENGES

3. Making Interactable AR objects

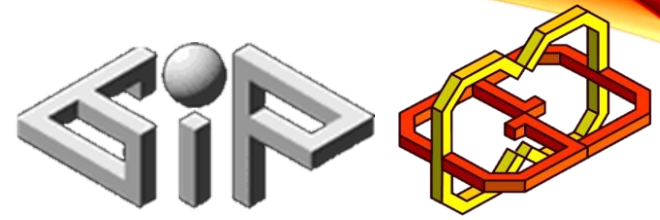
- In the real-life game, the user physically moves the cards. We wished to allow a similar experience, where the user can choose any of his monsters and set their next action.
- Each monster has a different shape, size. We cannot handle them all as one.
- ✓ The 'Compound Button' script provided by Microsoft HoloToolkit enables us to extend button-like properties to any game object, that is the identification of clicks and other inputs from the user.
- ✓ We added that script and a collider to each monster in the game, each was fitted to match exactly that monster.
- ✓ We made a vast use of Unity event system to enable different actions on the monsters.



CHALLENGES

4. Working in parallel

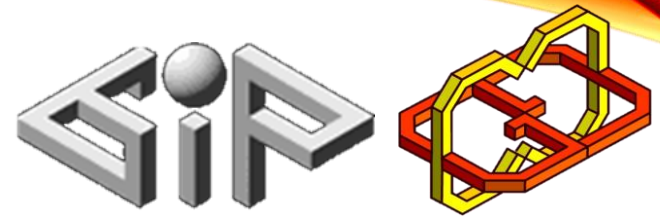
- We managed our project in Git.
- Working in parallel on the same scene and objects in Unity may cause many merge conflicts.
- ✓ We started dividing the project into smaller parts, so that each one of us can work on a different element without fear of conflicts.
- ✓ After we combined all our parts to one main project, we had to start coordinating before each merge, and try to avoid working on the same scene in parallel.
- ✓ Because most of the project is composed of smaller parts, working in parallel was mostly fine.



CHALLENGES

5. Creating a real-like game experience

- We wanted to create a real game experience – where the user plays against the computer.
- ✓ We designed a game manager script which manages the flow of the game, and calls for each player's move at his turn.
- ✓ We planned a basic AI for the computer player.
- ✓ We made vast use of Unity events system to manage the user game input.
- ✓ We designed an informative and easy-to-use UI to navigate the game phases.



SOURCES

1. Many Unity models: Unity Assets Store
<https://assetstore.unity.com>
2. Swordswoman model: <https://www.mixamo.com/>
3. Yu-Gi-Oh and some monster models:
<https://www.models-resource.com/>
4. Photo 1: <https://www.malavida.com/en/soft/yu-gi-oh-dueling-androdisc/android/#gref>



ALWAYS BELIEVE IN THE
HEART OF THE CARDS

