All-Around Virtual Photo Walls

Boris Bernegger, Ahmet Özüdoğru, Selin Barash, Onat Vuran

Motivation & Goal

- Scan a new room with HoloLens 2, recover all planes and automatically generate customized photo walls on **suitable planar surface**
- Allow the user to **customize an automatically generated photo wall layout**

- **Input:**
  - Set of images & physical room to be scanned
- **Output:**
  - Multiple photo wall layouts, tailored to the rooms physical properties

Main Challenge: Room Scanning

Challenge	How we addressed it
The scene understanding observer always assigns a visible material to occluding real-world objects	Defined a custom rule when to apply visible materials and when to apply invisible materials

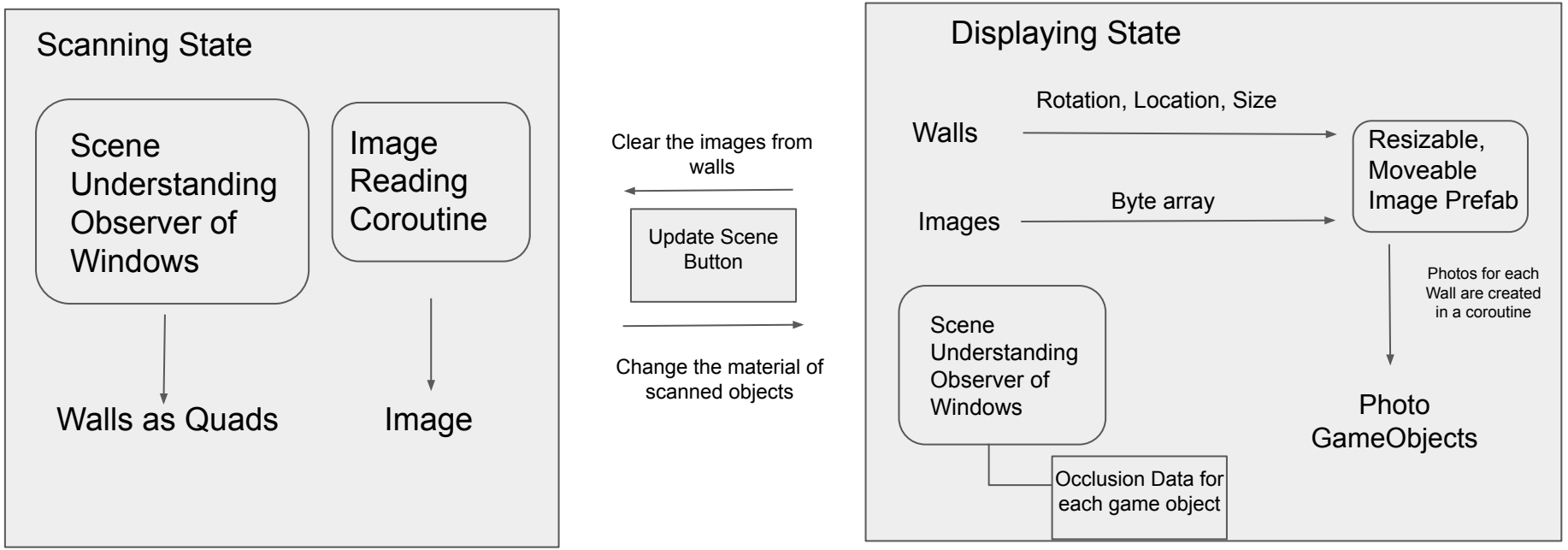
Main Challenge: Performance

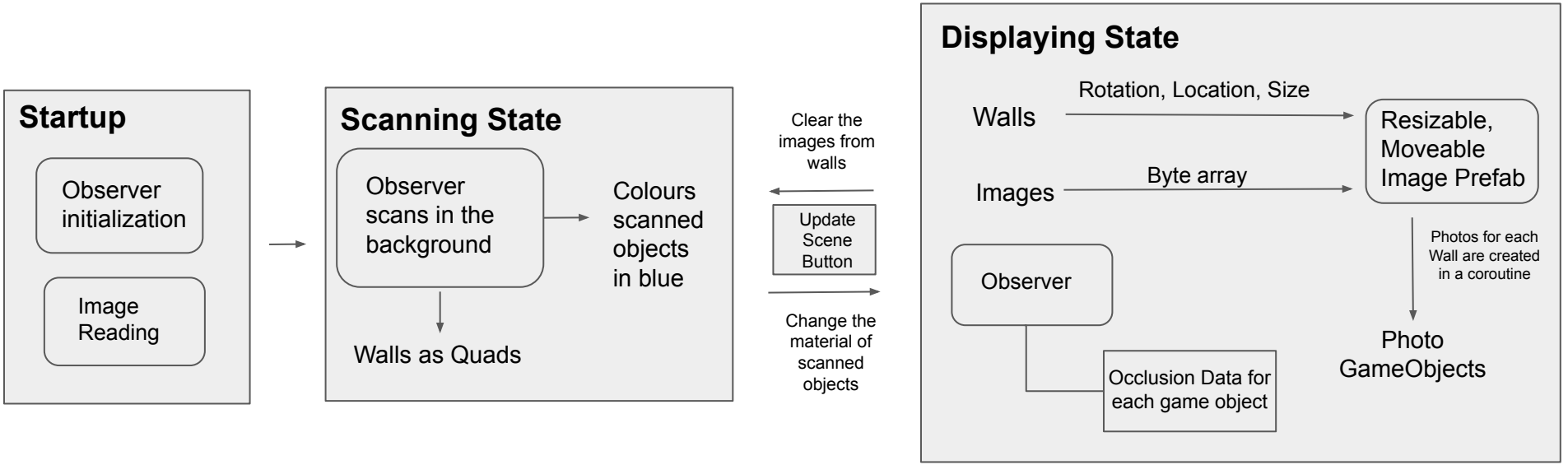
Challenge	How we addressed it
Calculating the photo wall layout and displaying potentially many images can take multiple seconds resulting in bad user experience if the application freezes	- Use prefabs for more performant displaying - Use coroutines to calculate and load photo walls asynchronously -> UI does never freeze
Heavy computations for occlusion calculations and wall recognition of big detailed rooms	- Intrinsic limitation of HoloLens 2 - Limit number of photos that are displayed

Main Challenge: UX during Interaction with Photo Walls

Challenge	How we addressed it
Providing the set of images that should be displayed	The algorithm loads photos from the Pictures folder
Providing a good UX during interaction with a photo wall as not all transformations make sense	Use ObjectManager and transformation constraints in addition to BoundsControl

App Architecture




Startup
Observer initialization
Image Reading
Scanning State
Observer scans in the background
Colours scanned objects in blue
Walls as Quads
Clear the images from walls
Update Scene Button
Change the material of scanned objects
Displaying State
Walls
Rotation, Location, Size
Resizable, Moveable Image Prefab
Images
Byte array
Photos for each Wall are created in a coroutine
Observer
Photo GameObjects
Occlusion Data for each game object

Video Demo

Evaluation

- **Round One – Most requested features**
  - Request for occlusion
    - Occlusion support added for major real world objects
  - Request for interaction with the photo wall
    - Can move and resize images
- **Round Two**
  - Strengths
    - Smooth interaction with the photo walls
    - Lag-free interaction experience
    - Photos are displayed in a stable manner
  - Weaknesses / room for improvement
    - Degrading performance in big rooms
    - No selection of different layouting algorithms
    - Some bugs and edge cases

Next Steps

- **More extensive user experiments**
  - Quantitative measurements of utility and UX metrics
- **Eliminate some edge cases**
  - Photos are sometimes displayed on walls that are too small
- **Final Report**