*Florida International University*

*School of Computing and Information Sciences*

Software Engineering Focus

HoloLens:

Visualization of Object Variable’s

Accessor/Mutator Methods

User Story ID: #1304 (HoloLens) Visualization of Object Accessor/Mutator Methods

**Name:** *Andres Chalela*

**Team Member(s):** *Paola Jiron and Andres Chalela*

**Project:** *Augmented Reality for Computer Science Education: HoloLens*

**Product Owner(s)**: *Francisco Ortega*

**Mentor(s)**: *Francisco Ortega and Mohsen Taheri*

**Instructor**: *Masoud Sadjadi*

**Contents**

**1.1** **Visualization of Object Accessor/Mutator Methods 2**

**1.2 Use Cases 2**

**2** **Behaviour Requirements 6**

**3** **Sequence Diagrams 7-13**

**4.1** **Class Diagrams 14**

**4.2** **Software Architecture 15**

**5 Unit Test 16**

**6 Visual User Guide 17-18**

**Appendix 19**

# 

**1.1 Visualization of Object Accessor/Mutator Methods**

Description:

*As a user, I have to be able to be able to manipulate a holographic object’s color variable attribute, so that I can have a visualization of what object accessor/mutator methods are and how they function.*

Acceptance Criteria:

1. Use experience & knowledge from previous Sprint HoloLens tutorials to implement needed elements within Unity Project
2. Create color palette element within Unity that will allow user interaction and allow manipulation of a UI element’s color
3. Create needed 3D Holographic Object that will be affected by user color choice
4. Create text elements displaying relevant project information & prompting user interaction
5. Implement needed HoloLens input scripts that will allow user to interact with color palette via Gaze & Hand Gestures
6. Implement needed scripts for having 3D Object rotate and move up and down
7. Allow user to choose color using a cursor that will allow tracking of HoloLens Gaze functionality
8. Test
9. Deploy & build to Emulator/HoloLens

**1.2 Use Cases**

Use Case ID: US1304-1

1. Name: HoloLens Gaze & 3D Cursor Integration
2. Actor: Player, Developer
3. Preconditions: Actor has initialized the HoloLens App
4. Postcondition: Actor manipulates cursor with Gaze and head movements
5. Flow of events:

|  |  |
| --- | --- |
| 1. Actor initializes HoloLens application |  |
|  | 2. Actor looks around and point Gaze functionality to floating color palette UI element |
| 3. When the Gaze pointer shows up on the floating color palette UI element, Actor stops moving |  |
|  | 4. Actor sees 3D cursor on top of color palette UI Element |

Use Case ID: US1304-2

1. Name: Preview Color UI Element
2. Actor: Player, Developer
3. Preconditions: Application is running and actor is wearing the Hololens device
4. Postcondition: Actor sees the Preview Color UI Element change color as cursor moves on the surface of color palette
5. Flow of events:

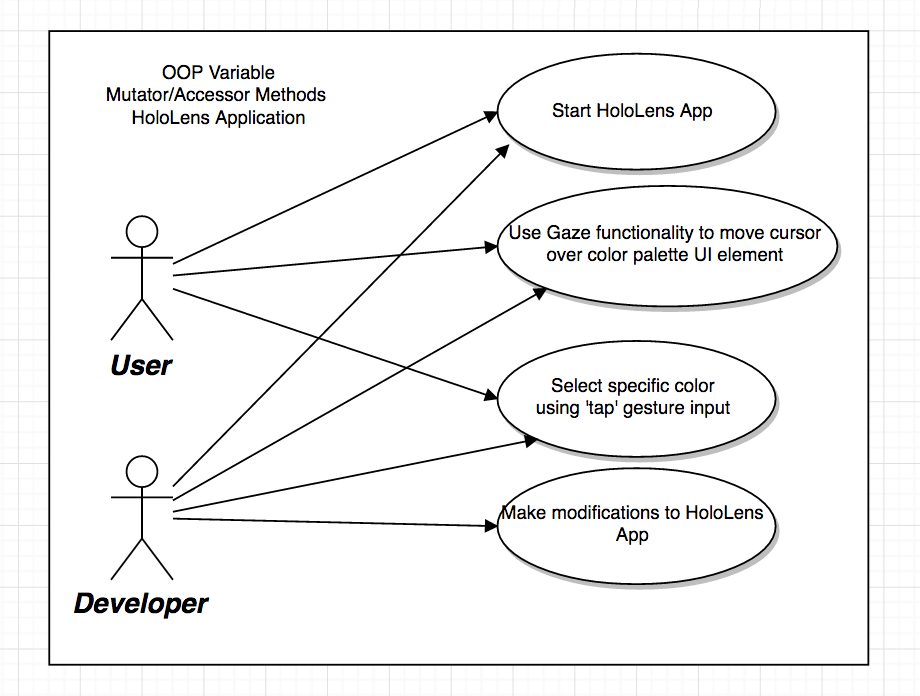
|  |  |
| --- | --- |
| 1. Actor gazes at the color palette UI element |  |
|  | 2. Actor controls cursor with head movements and moves cursor across surface of UI color palette |
| 3. Actor sees Preview Color UI element change color as cursor moves across different colors |  |

Use Case ID: US1304-3

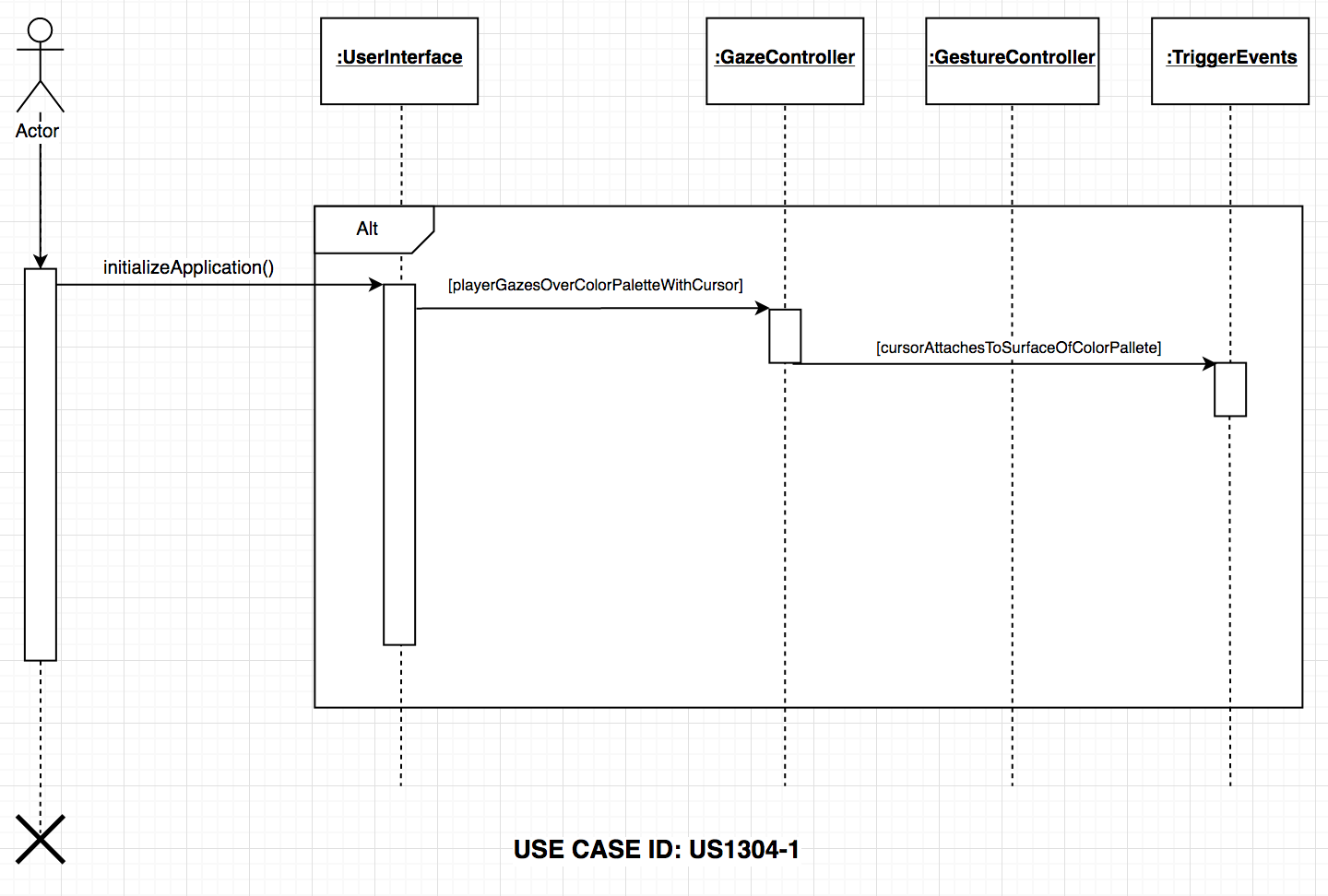
1. Name: Specific Color Selection
2. Actor: Player, Developer
3. Preconditions: Application is running and actor is wearing the Hololens device
4. Postcondition: Actor decides on a color and uses ‘tap’ HoloLens gesture to change 3D object’s color
5. Flow of events:

|  |  |
| --- | --- |
| 1. Actor gazes at the color palette UI Element |  |
|  | 2. Actor controls cursor with head movements and moves across color palette. |
| 3. Actor sees Preview Color UI element change and decides on a specific color . |  |
|  | 4. Actor ‘taps’ on color palette to select color, and floating 3D object changes to the respective color |

**2. Behaviour Requirements**



**3. Sequence Diagrams**



|  |
| --- |
| **Sequence Diagram** |

|  |  |
| --- | --- |
| X | Includes actors |
| X | Includes lifelines |
| X | Includes activations |
| X | Includes entity objects |
| X | Includes control objects |
| X | Includes method call in the receiving class for each received message |
| X | Includes method call in the sending class for each sent message |
| X | Includes messages |

|  |
| --- |
| **Standards** |

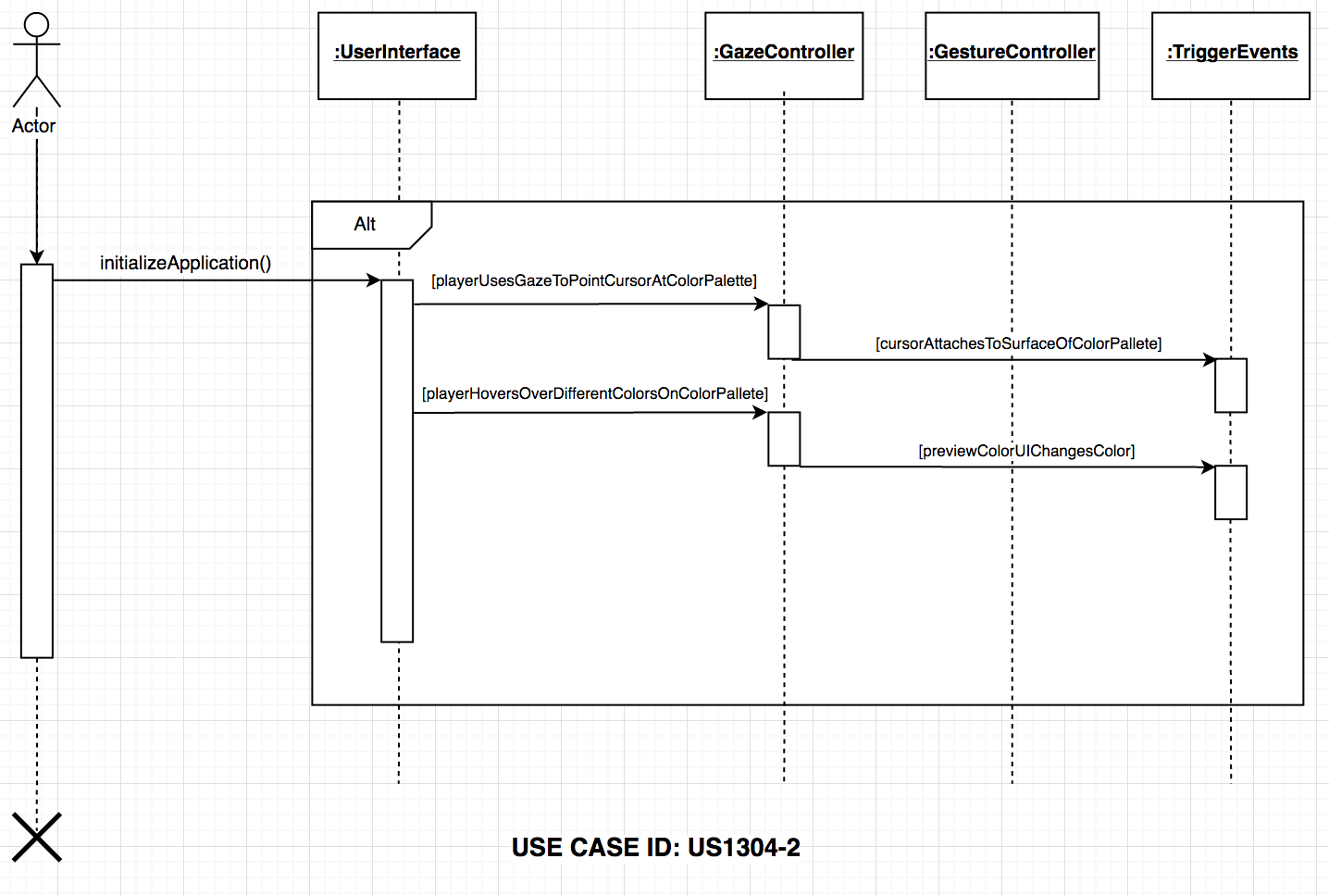
|  |  |
| --- | --- |
| X | Actors are a stick figure with full name |
| X | Message is a solid line with a filled arrow ahead |
| X | Return message are a dashed line with an open arrowhead |
| X | One outgoing action arrow per action box |

|  |
| --- |
| **Compliance of the document guidelines** |

|  |  |
| --- | --- |
| X | Is UML syntax correct? |
| X | Is there an Actor? |
| X | When there is more than one object for a class (also in other sequence diagrams) are those objects referenced with proper names? |
| X | Does the diagram cover the user case scenario? |

|  |
| --- |
| **Internal consistence of the model** |

|  |  |
| --- | --- |
| X | Is there a scenario for the sequence diagram? |
| X | Is there a documentation to which scenario belongs a sequence diagram? |
| X | Is the scenario information correctly described? |
| X | When an object of a class receives notifications, are those notifications mapped as events? |
| X | When an object of a class receives notifications, are those notifications mapped as methods defined in the class? |
| X | When an object of a class receives notifications, are those notifications mapped as actions in the State Diagrams? |
| X | Are all objects the same as shown on the class diagram? |
| X | Are there activation lifelines (rectangles) to show created and deleted objects? |
| X | Can each message be sent? |
| X | Is each class in the class diagram? |
| X | Is there a method call in the receiving class for each received message? |



|  |
| --- |
| **Sequence Diagram** |

|  |  |
| --- | --- |
| X | Includes actors |
| X | Includes lifelines |
| X | Includes activations |
| X | Includes entity objects |
| X | Includes control objects |
| X | Includes method call in the receiving class for each received message |
| X | Includes method call in the sending class for each sent message |
| X | Includes messages |

|  |
| --- |
| **Standards** |

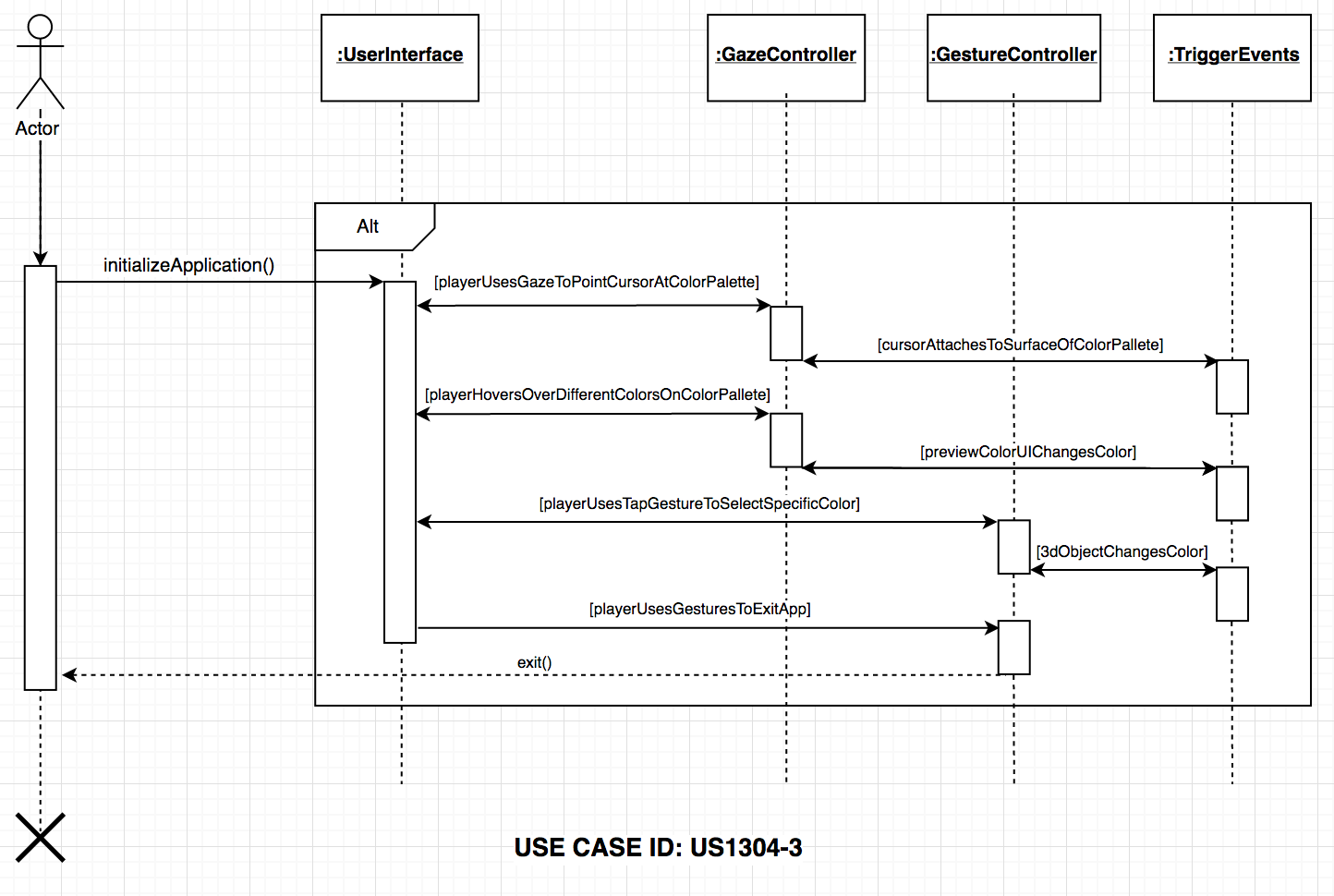
|  |  |
| --- | --- |
| X | Actors are a stick figure with full name |
| X | Message is a solid line with a filled arrow ahead |
| X | Return message are a dashed line with an open arrowhead |
| X | One outgoing action arrow per action box |

|  |
| --- |
| **Compliance of the document guidelines** |

|  |  |
| --- | --- |
| X | Is UML syntax correct? |
| X | Is there an Actor? |
| X | When there is more than one object for a class (also in other sequence diagrams) are those objects referenced with proper names? |
| X | Does the diagram cover the user case scenario? |

|  |
| --- |
| **Internal consistence of the model** |

|  |  |
| --- | --- |
| X | Is there a scenario for the sequence diagram? |
| X | Is there a documentation to which scenario belongs a sequence diagram? |
| X | Is the scenario information correctly described? |
| X | When an object of a class receives notifications, are those notifications mapped as events? |
| X | When an object of a class receives notifications, are those notifications mapped as methods defined in the class? |
| X | When an object of a class receives notifications, are those notifications mapped as actions in the State Diagrams? |
| X | Are all objects the same as shown on the class diagram? |
| X | Are there activation lifelines (rectangles) to show created and deleted objects? |
| X | Can each message be sent? |
| X | Is each class in the class diagram? |
| X | Is there a method call in the receiving class for each received message? |



|  |
| --- |
| **Sequence Diagram** |

|  |  |
| --- | --- |
| X | Includes actors |
| X | Includes lifelines |
| X | Includes activations |
| X | Includes entity objects |
| X | Includes control objects |
| X | Includes method call in the receiving class for each received message |
| X | Includes method call in the sending class for each sent message |
| X | Includes messages |

|  |
| --- |
| **Standards** |

|  |  |
| --- | --- |
| X | Actors are a stick figure with full name |
| X | Message is a solid line with a filled arrow ahead |
| X | Return message are a dashed line with an open arrowhead |
| X | One outgoing action arrow per action box |

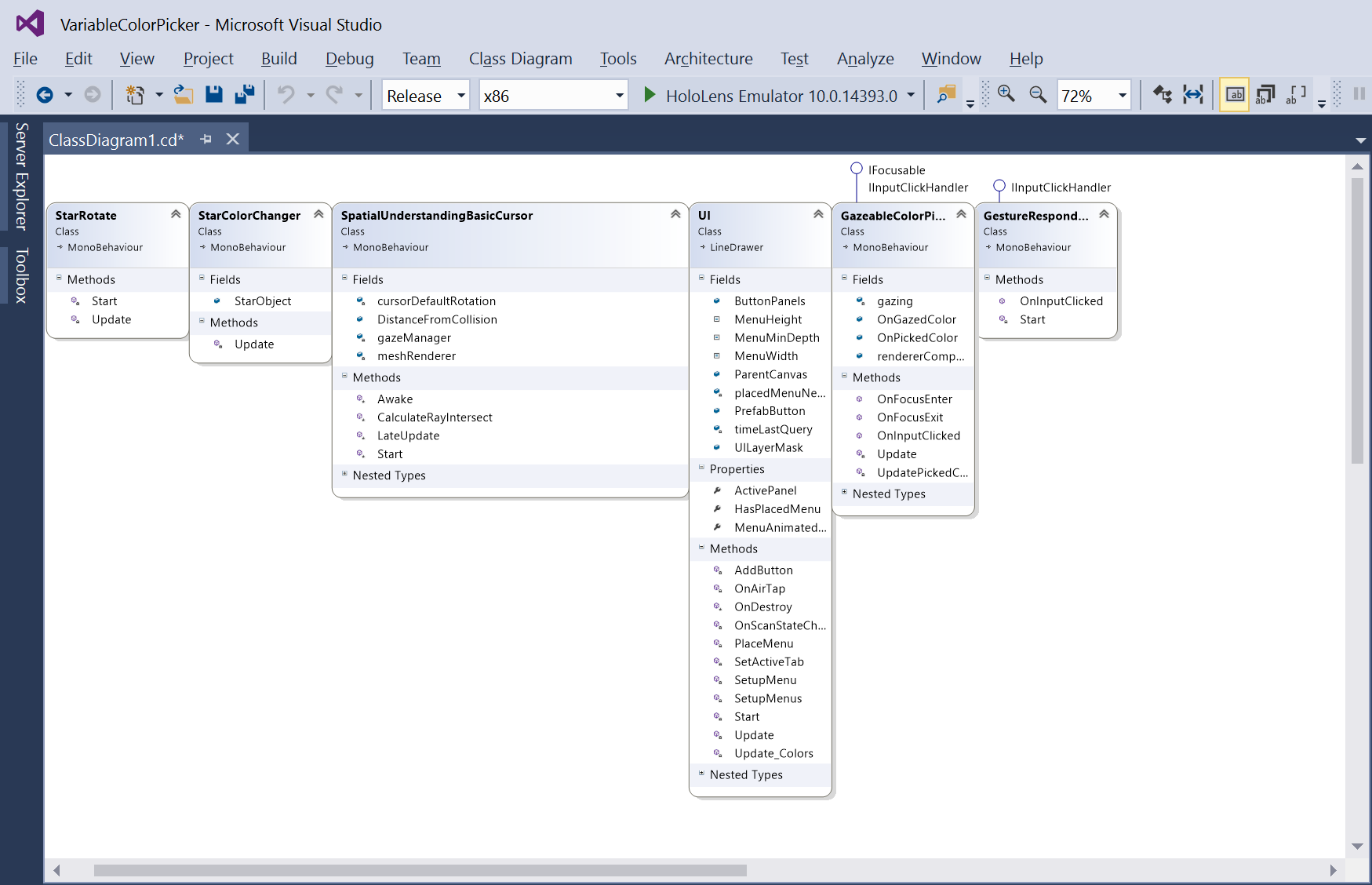
|  |
| --- |
| **Compliance of the document guidelines** |

|  |  |
| --- | --- |
| X | Is UML syntax correct? |
| X | Is there an Actor? |
| X | When there is more than one object for a class (also in other sequence diagrams) are those objects referenced with proper names? |
| X | Does the diagram cover the user case scenario? |

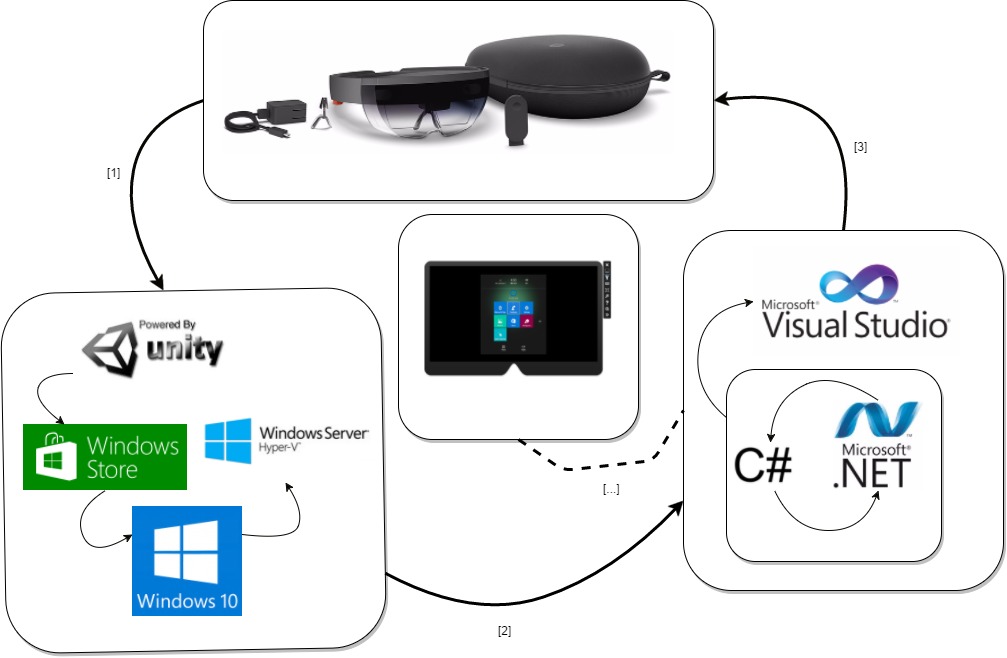
|  |
| --- |
| **Internal consistence of the model** |

|  |  |
| --- | --- |
| X | Is there a scenario for the sequence diagram? |
| X | Is there a documentation to which scenario belongs a sequence diagram? |
| X | Is the scenario information correctly described? |
| X | When an object of a class receives notifications, are those notifications mapped as events? |
| X | When an object of a class receives notifications, are those notifications mapped as methods defined in the class? |
| X | When an object of a class receives notifications, are those notifications mapped as actions in the State Diagrams? |
| X | Are all objects the same as shown on the class diagram? |
| X | Are there activation lifelines (rectangles) to show created and deleted objects? |
| X | Can each message be sent? |
| X | Is each class in the class diagram? |
| X | Is there a method call in the receiving class for each received message? |

**4.1 Class Diagram**



**4.2 Software Architecture**



**5. Unit Test**

Test case ID: US1304-1

Description/Summary of Test: User uses HoloLens Gaze functionality to look at color palette UI element. Cursor appears on color palette element once Gaze focuses on it.

Pre-condition: HoloLens app should be initialized and user must be wearing the Hololens device to be able to utilize Gaze functionality

Expected Results: User should see cursor icon on surface of color palette

Actual Result: Cursor appears on surface of color palette UI element

Status (Fail/Pass): Pass.

Test case ID: US1304-2

Description/Summary of Test: User uses HoloLens Gaze to hover over different colors within color palette using the implemented cursor. Preview Color UI element changes according to the color that cursor is focused on.

Pre-condition: HoloLens app should be initialized and user must be wearing the Hololens device to be able to utilize Gaze functionality

Expected Results: Preview Color UI element changes color as cursor hovers over different colors within color palette.

Actual Result: Preview Color UI element changes accordingly to the color that cursor is hovered over.

Status (Fail/Pass): Pass.

Test case ID: US1304-3

Description/Summary of Test: User uses Holens Gaze to hover over a specific color using the implemented cursor. Once color is decided, user can use ‘tap’ gesture to change the color of floating 3D holographic object

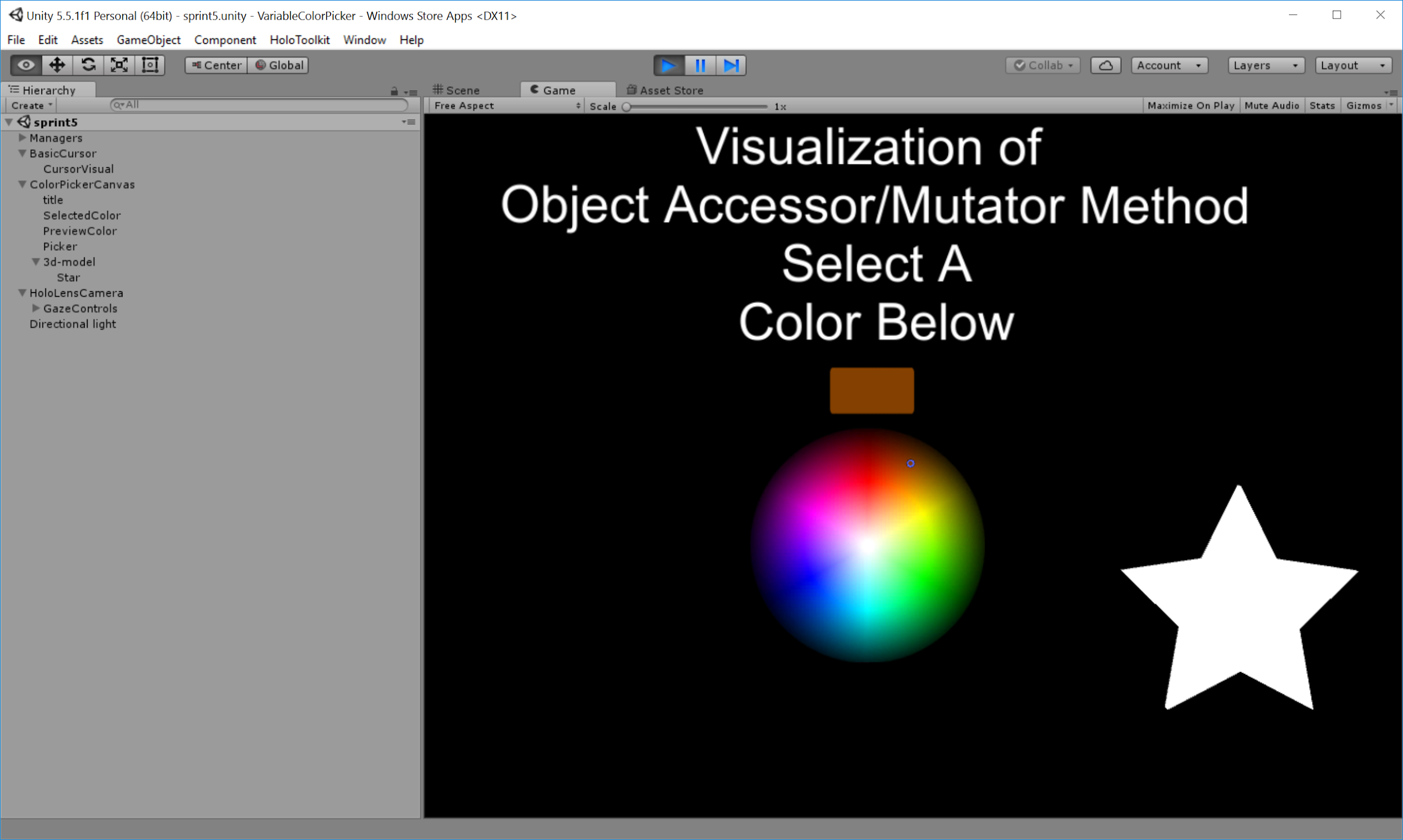
Pre-condition:

Expected Results: User sees the 3D holographic object change color once they use ‘tap’ gesture on color palette.

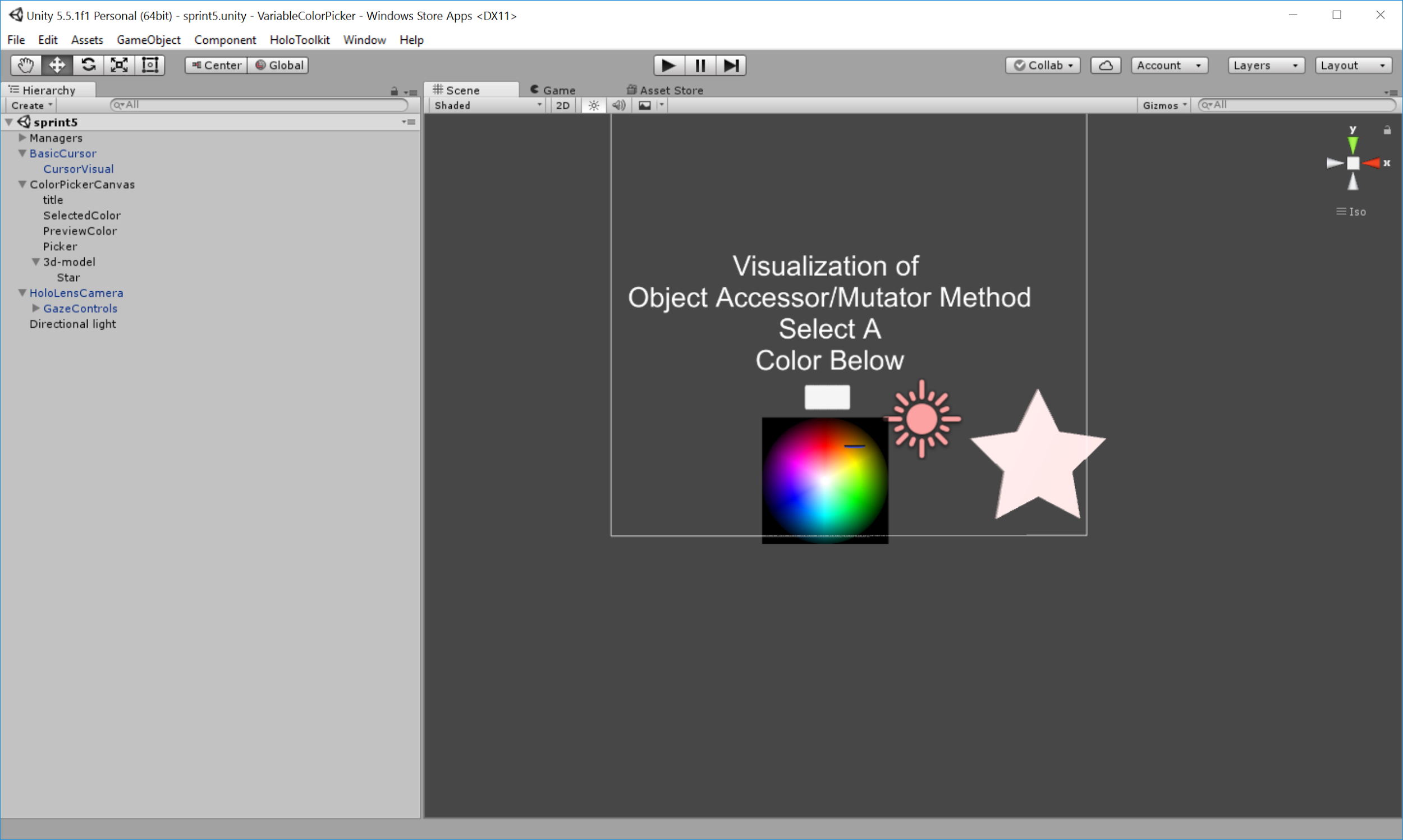
Actual Result: 3D holographic object changes to the specific color selected

Status (Fail/Pass): Fail.

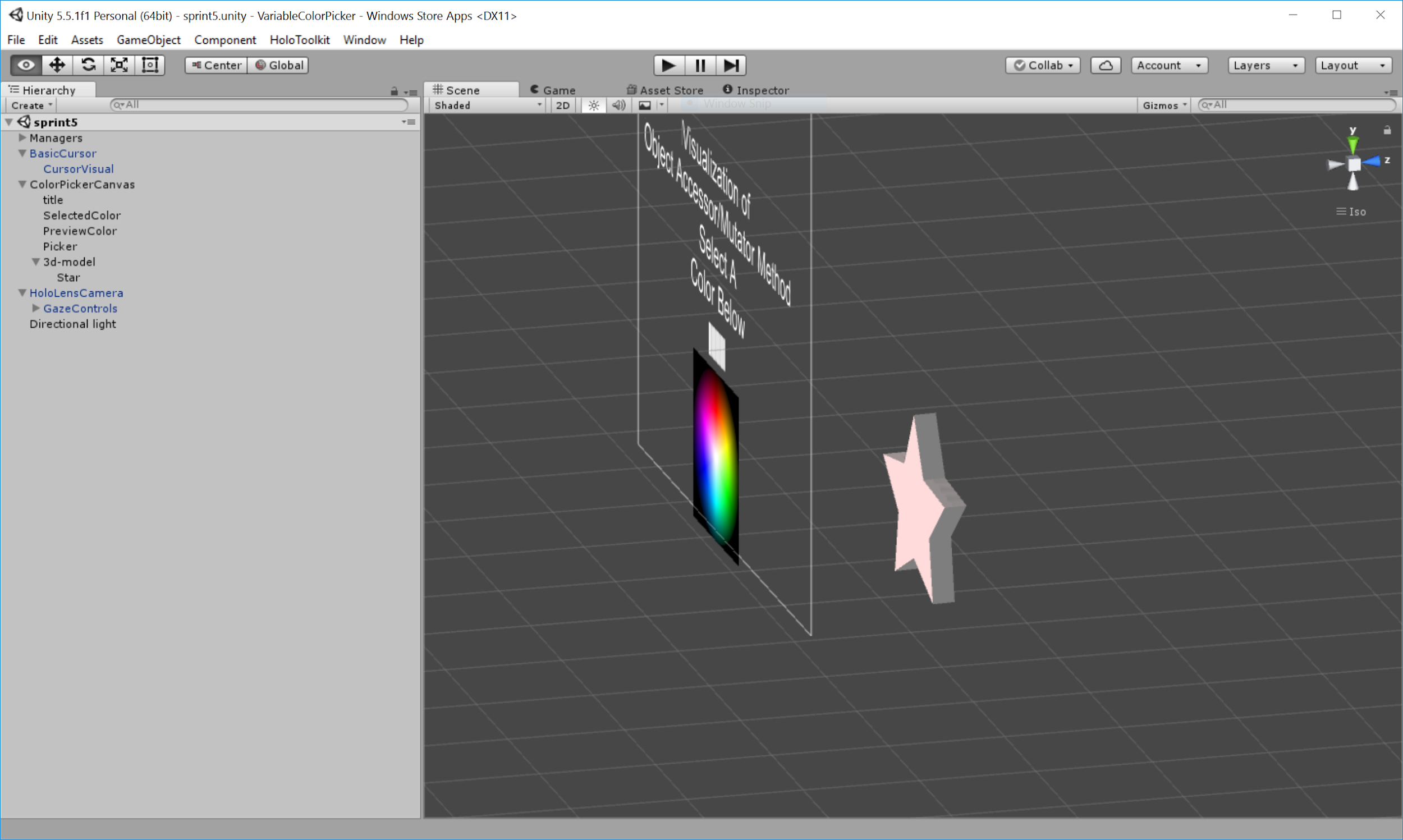
**6. Visual User Guide**



**Figure 1. Main Scene**



**Figure 2. Scene 1 [Front View]**



**Figure 3. Scene 1 [Side View]**

**APPENDIX**

*References:*

[1] "GitHub - Microsoft/HoloToolkit-Unity". Web. 30 Mar. 2017. <https://github.com/Microsoft/HoloToolkit-Unity>.