

**TRƯỜNG ĐẠI HỌC SƯ PHẠM KỸ THUẬT TP. HCM**

**KHOA CHẤT LƯỢNG CAO**

**🕯✡🕮🕮✡🕯**

**ĐỒ ÁN I**

**BUILD A INTERFACE TO RENDER THE FRUITS ON UNITY**

**SVTH: MSSV:**

**Nguyễn Quốc Khánh 16110119**

**GVHD: Huỳnh Xuân Phụng**

**Tp. Hồ Chí Minh, tháng 11 năm 2019**

**NHẬN XÉT**

*Ký tên*

**Huỳnh Xuân Phụng**

**MỤC LỤC**

[I. Đặc tả 4](#_Toc24534091)

[1. Mục đích 4](#_Toc24534092)

[2. Tính năng 4](#_Toc24534093)

[3. Giao diện 5](#_Toc24534094)

[II. Phân công công việc 6](#_Toc24534095)

[III. Thiết kế 6](#_Toc24534096)

[1.Thông tin đầu vào 6](#_Toc24534097)

[2. Thiết kế lớp 7](#_Toc24534098)

[3. Thiết kế giao diện 11](#_Toc24534099)

[Hình ảnh thu được 11](#_Toc24534100)

[Hướng dẫn sử dụng 13](#_Toc24534101)

[IV. Đánh giá 14](#_Toc24534102)

[1. Đánh giá mức độ hoàn thành: hoàn thành xong các chức năng tối thiểu cần phải có cho chương trình 14](#_Toc24534103)

[2. Đánh giá quá trình thực hiện 14](#_Toc24534104)

[3. Hạn chế của chương trình và hướng phát triểu 14](#_Toc24534105)

[V. Tài liệu tham khảo 15](#_Toc24534106)

**Mục lục về hình ảnh**

[Hình 1. Giao diện khi vừa khởi động 4](#_Toc24315064)

[Hình 2. Giao diện khi chọn xong Object. 4](#_Toc24315146)

[Một số ví dụ 11](#_Toc24315236)

**Mục lục về bảng biểu**

[Bảng 2.1 Phân công công việc 5](#_Toc24315542)

[Bảng 2.2.1 Thiết kế lớp 6](#_Toc24315657)

[Bảng 2.2.2 Đặc tả lớp cha MonoBehaviour. 6](#_Toc24315774)

[Bảng 2.2.3 Đặc tả interface fruit 7](#_Toc24316052)

[Bảng 2.2.4 Phương thức trong Class Orange 8](#_Toc24316005)

[Bảng 2.2.5 Các phương thức sử dụng trong Class Screen Texture 9](#_Toc24316098)

[Bảng 2.2.6 Các phương thức sử dụng trong Class Cameramove 10](#_Toc24316144)

[Bảng 2.3.1 Thiết kế giao diện 11](#_Toc24316375)

[Bảng hướng dẫn sử dụng 12](#_Toc24534057)

# **I. Đặc tả**

## **1. Mục đích**

Render hay rendering là một từ diễn tả quá trình xuất một project có thể tùy chỉnh thành một sản phẩm hoàn thiện mà mọi người đều có thể xem và sử dụng chúng.

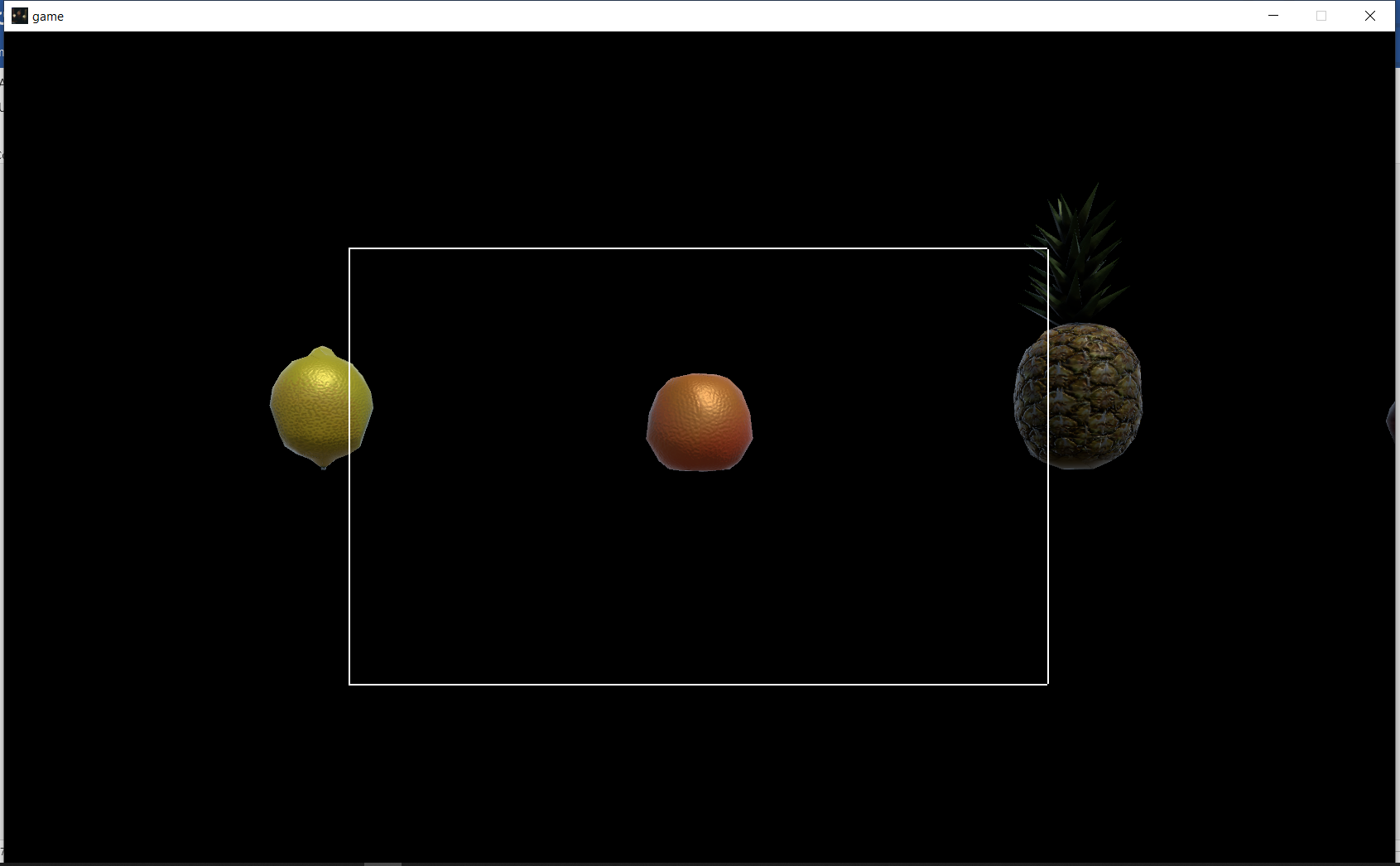
Render Fruits là phần mền phục vụ mục đích khi User muốn chụp hình một trong những object fruits 3D ở nhiều góc nhìn khác nhau và xuất thành file ảnh có định dạng \*.png được lưu với tên tương ứng để đào tạo các mô hình máy học khi không thể thu thập được đủ ảnh trong thực tế.

Dữ liệu hình ảnh thu được đồng thời dùng để kiểm tra mô hình huấn luyện trên tập dữ liệu mô phỏng có thực thi tốt trên tập dữ liệu thực hay không.

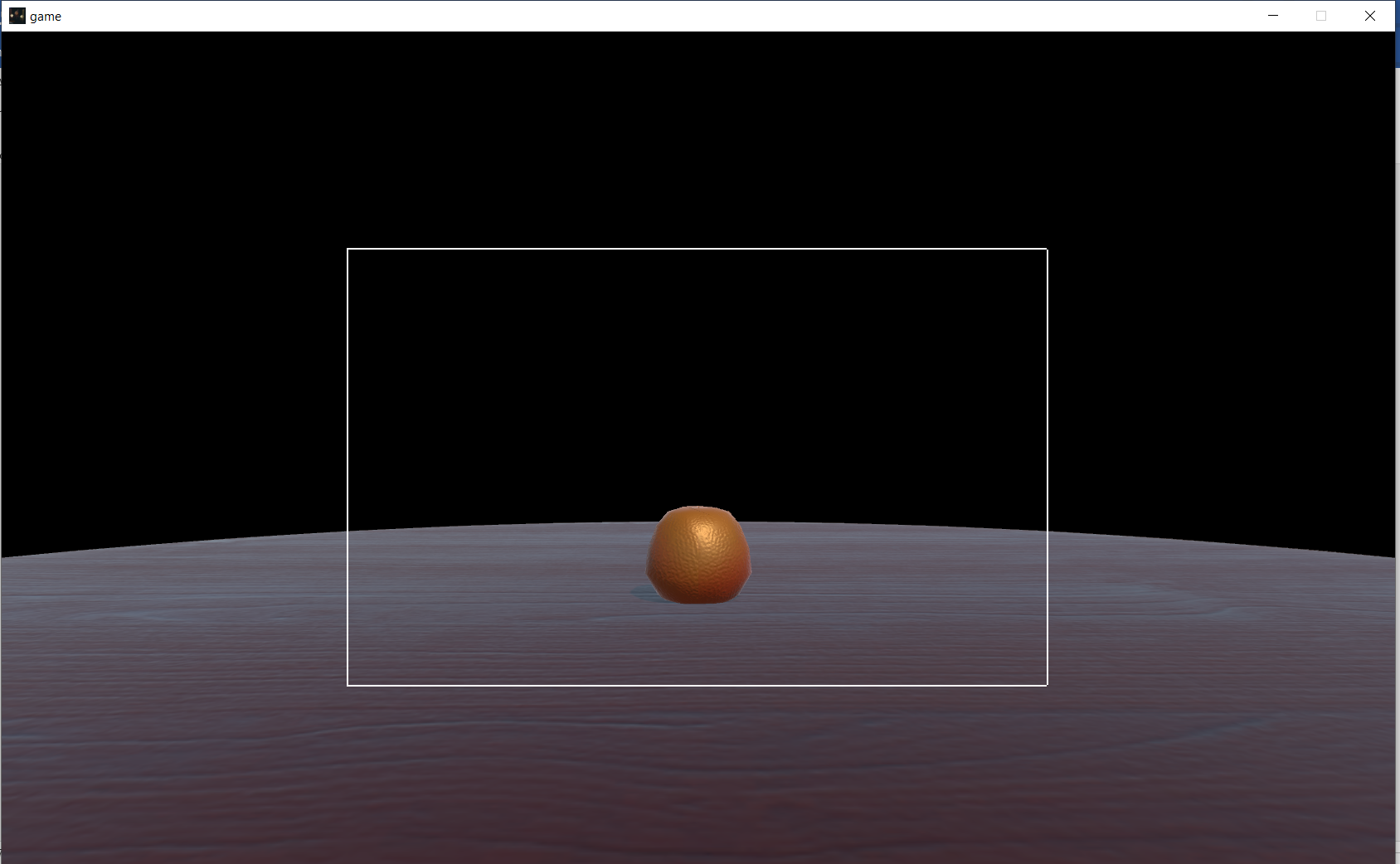
## **2. Tính năng**

Khi khởi động game sẽ hiện ra một loạt các hoa quả có sẵn trong game theo hàng ngang. Người dùng có thể chọn một hay nhiều quả khác loại với nhau để chụp ảnh và có thể di chuyển camera cũng như góc quay để có góc chụp theo ý muốn. Sau khi chụp ảnh có thể xuất ra file \*.png, file ảnh này sẽ đươc lưu trong folder có tên tương ứng trong thư mục data của game.

## **3. Giao diện**



Hình 1. Giao diện khi vừa khởi động



Hình 2. Giao diện khi chọn xong Object.

# **II. Phân công công việc**

Bảng 2.1 Phân công công việc

|  |  |  |  |
| --- | --- | --- | --- |
| TT | Tên sinh viên | Mô tả | đóng góp (%) |
| 1 | Nguyễn Quốc Khánh | Tìm kiếm Package 3D của Fruits | 100% |
| 2 | Nguyễn Quốc Khánh | Code các component của Object | 100% |

# **III. Thiết kế**

## **Thông tin đầu vào**

- Các Object trái cây muốn chụp.

- Các loại trái cây như: chanh, cam, táo, dưa hấu,… có thông số cơ bản:

+ Position.X = 0; Position.Y = 200; Position.Z = 0;

+ Rotation.X = 0; Rotation.Y = 200; Rotation.Z = 0;

+ Scale.X = 50; Scale.Y = 50; Scale.Z = 50;

- Góc quay Camera không giới hạn.

- Ánh sáng là Object: Directional Light do Unity tạo sẵn cho chúng ta với các thông số như:

+ Position.X = 15.8; Position.Y = 23; Position.Z = 0;

+ Rotation.X = 50; Rotation.Y = -30; Rotation.Z = 0;

+ Scale.X = 1; Scale.Y = 1; Scale.Z = 1;

- Đầu ra là các hình ảnh sau khi chụp trong game

## **2. Thiết kế lớp**

Bảng 2.2.1 Thiết kế lớp

|  |  |  |  |
| --- | --- | --- | --- |
| TT | Tên lớp | Lớp cha | Mục đích |
| 1 | public class banana | MonoBehaviour, fruit | Điều khiển vị trí của Object và vị trí lưu hình |
| 2 | public class cameramove | MonoBehaviour, fruit | Điều khiển vị trí của Object và vị trí lưu hình |
| 3 | public class kiwi | MonoBehaviour, fruit | Điều khiển vị trí của Object và vị trí lưu hình |
| 4 | public class lemon | MonoBehaviour, fruit | Điều khiển vị trí của Object và vị trí lưu hình |
| 5 | public class orange | MonoBehaviour, fruit | Điều khiển vị trí của Object và vị trí lưu hình |
| 6 | public class pineapple | MonoBehaviour, fruit | Điều khiển vị trí của Object và vị trí lưu hình |
| 7 | public class pomegranate | MonoBehaviour, fruit | Điều khiển vị trí của Object và vị trí lưu hình |
| 8 | public class quince | MonoBehaviour, fruit | Điều khiển vị trí của Object và vị trí lưu hình |
| 9 | public class watermelon | MonoBehaviour, fruit | Điều khiển vị trí của Object và vị trí lưu hình |
| 10 | interface fruit |  | Thể hiện tính đa hình |

Sinh viên phụ trách: Nguyễn Quốc Khánh

Lớp cha do Unity hổ trợ có nhiều phương thức hổ trợ ví dụ như:

Bảng 2.2.2 Đặc tả lớp cha MonoBehaviour.

|  |  |  |  |
| --- | --- | --- | --- |
| **TT** | **Phương thức** | **Mục đích** | **Tên file, số thứ tự dòng chứa khai báo** |
| 1 | useGUILayout **()**  Input: không có  Output: không có | Disabling this lets you skip the GUI layout phase. | MonoBehaviour.cs (27) |
| 2 | runInEditMode **()**  Input: không có  Output: không có | Allow a specific instance of a MonoBehaviour to run in edit mode | MonoBehaviour.cs (32) |
| 3 | StartCoroutine **(**string methodName**)**  Input: methodName  Output: không có | Starts a coroutine named methodName | MonoBehaviour.cs (93) |
| 4 | InvokeRepeating ()  Input: không có  Output: không có | Is any invoke pending on this MonoBehaviour? | MonoBehaviour.cs (83) |
| 5 | StopAllCoroutines()  Input: aname  Output: không có | Stops all coroutines running on this behaviour | MonoBehaviour.cs (115) |

Bảng 2.2.3 Đặc tả interface fruit

|  |  |  |  |
| --- | --- | --- | --- |
| **TT** | **Phương thức** | **Mục đích** | **Tên file, số thứ tự dòng chứa khai báo** |
| 1 | CaptureScreen(string aname)  Input: aname  Output: không có | Chụp lại hình ảnh rồi đưa ra góc trên bên trái cho người dùng xem trước hình ảnh mình vừa mới chụp | ScreenTexture.cs (57) |
| 2 | UploadPNG(string aname)  Input: aname  Output: không có | Lưu lại hình ảnh vừa mới chụp vô folder với tên tương ứng | ScreenTexture.cs (58) |

Bảng 2.2.4 Phương thức trong Class Orange

|  |  |  |  |
| --- | --- | --- | --- |
| **TT** | **Phương thức** | **Mục đích** | **Tên file, số thứ tự dòng chứa khai báo** |
| 1 | Start **()**  Input: không có  Output: không có | Khi bắt đầu sẽ khởi tạo các giá trị lúc khởi động game | orange.cs (31) |
| 2 | Update **()**  Input: không có  Output: không có | * Khi click chuột vào sẽ chuyển Object xuống bàn để chụp hình, nếu click thêm lần nữa sẽ đưa Object về vị trí cũ * Khi bấm phím KeypadEnter sẽ chụp hình Object | orange.cs (41) |
| 3 | OnMouseDrag **()**  Input: không có  Output: không có | Cho biết là đã click hay chưa click Object | orange.cs (60) |
| 4 | CaptureScreen(string aname)  Input: aname  Output: không có | Chụp lại hình ảnh rồi đưa ra góc trên bên trái cho người dùng xem trước hình ảnh mình vừa mới chụp | orange.cs (72) |
| 5 | UploadPNG(string aname)  Input: aname  Output: không có | Lưu lại hình ảnh vừa mới chụp vô folder với tên tương ứng | orange.cs (2) |

Bảng 2.2.5 Các phương thức sử dụng trong Class Screen Texture

|  |  |  |  |
| --- | --- | --- | --- |
| **TT** | **Phương thức** | **Mục đích** | **Tên file, số thứ tự dòng chứa khai báo** |
| 1 | Start **()**  Input: không có  Output: không có | Khi bắt đầu sẽ khởi tạo các giá trị lúc khởi động game | ScreenTexture.cs (25) |
| 2 | OnGUI **()**  Input: không có  Output: không có | Vẽ ra khung hình khi chụp ảnh | ScreenTexture.cs (42) |

Bảng 2.2.6 Các phương thức sử dụng trong Class Cameramove

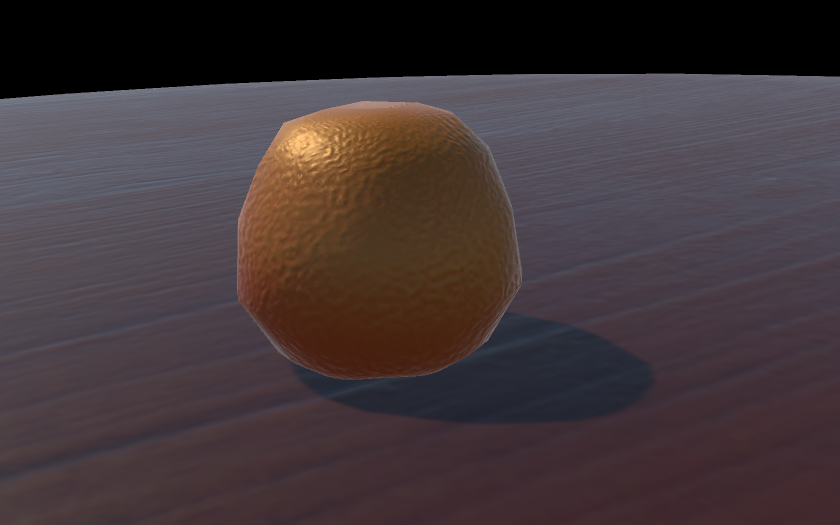
|  |  |  |  |
| --- | --- | --- | --- |
| **TT** | **Phương thức** | **Mục đích** | **Tên file, số thứ tự dòng chứa khai báo** |
| 1 | Update **()**  Input: không có  Output: không có | Di chuyển camera và xoay camera | cameramove.cs (13) |

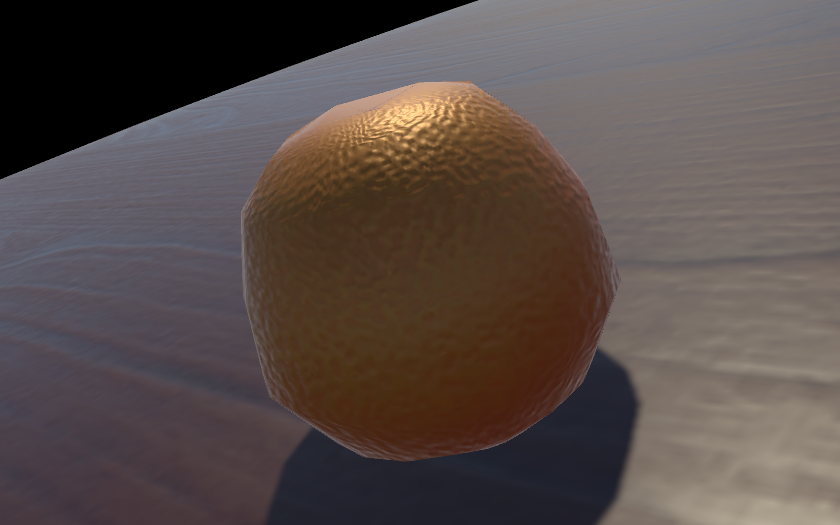
## **3. Thiết kế giao diện**

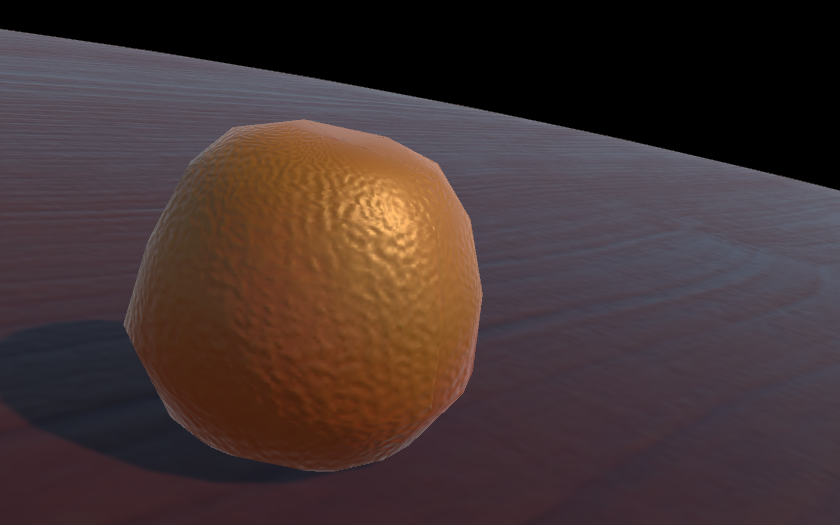
Bảng 2.3.1 Thiết kế giao diện

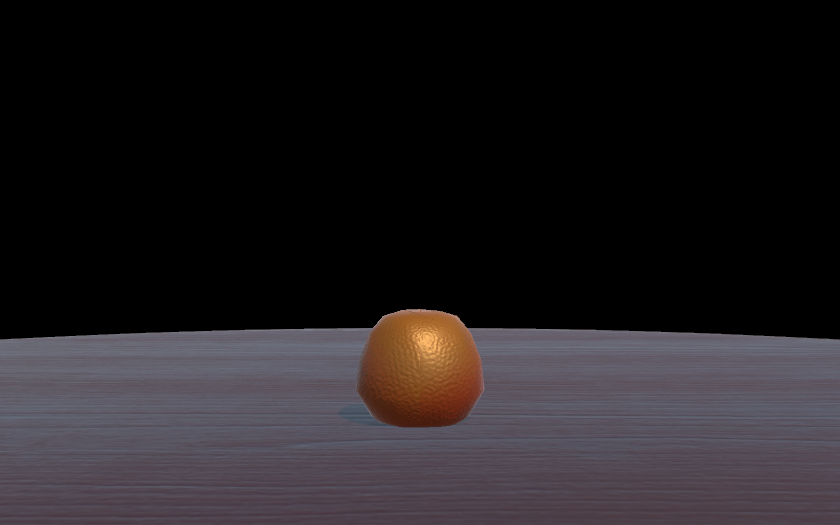
|  |  |  |  |
| --- | --- | --- | --- |
| **Tt** | **Màn hình** | **Mục đích** | **Giải thích** |
| 1 | Chọn object | Chọn object để thao tác | Mục đích thiết kế giao diện là đơn giản nhất có thể để ai cũng có thể lập tức sử dụng vì nhu cầu này cần nhanh và tiện lợi |
| 2 | Thao tác di chuyển góc nhìn và chụp ảnh | Chọn góc chụp ảnh |

# **Hình ảnh thu được**









Một số ví dụ 1

Các hình ví dụ có các góc quay khác nhau, điều kiện sáng khác nhau.

# **Hướng dẫn sử dụng**

Bảng hướng dẫn sử dụng

|  |  |
| --- | --- |
| Tên phím | Chức năng |
| W | Hướng camera lên trên |
| S | Hướng camera xuống dưới |
| A | Hướng camera sang trái |
| D | Hướng camera sang phải |
| Q | Xoay camera theo chiều ngược kim đồng hồ |
| E | Xoay camera theo chiều thuận kim đồng hồ |
| Numpad 8 | Di chuyển lùi về sau |
| Numpad 2 | Di chuyển tiến về trước |
| Numpad 4 | Di chuyển sang trái |
| Numpad 6 | Di chuyển sang phải |
| Numpad ENTER | Chụp hình |
| Numpad 5 | Di chuyển lên trên |
| Ctrl + Numpad 5 | Di chuyển xuống dưới |
| space bar | Di chuyển xuống nơi chụp hình |
| U | Di chuyển lên nơi chứa các Object |

# **IV. Đánh giá**

## **1. Đánh giá mức độ hoàn thành:** hoàn thành xong các chức năng tối thiểu cần phải có cho chương trình

* Giao diện đơn giản, dễ sử dụng.
* Thực hiện được chức năng cần thiết.
* Vẫn còn một số vấn đề thiếu sót như delay khi click Object

## **2. Đánh giá quá trình thực hiện**

Thuận lợi là có nhiều công nghệ hỗ trợ cho việc phát triển ứng dụng như Unity tạo sẵn cho ta Directional Light và Scripting API ImageConversion.EncodeToPNG giúp ta lưu lại hình ảnh chụp trong game thành file \*.png. Hơn nữa chúng ta còn có khá nhiều nguồn tài liệu phong phú trên internet như các diễn đàn nói về Unity hay trang <https://docs.unity3d.com> cũng giải thích rất rõ những tính năng cần thiết để hoàn thành Project.

Khó khăn vì còn ít kinh nghiệm trong việc thiết kế giao diện nên lúc lên ý tưởng và thiết kế game tốn nhiều thời gian, UI trong Unity hộ trợ ta khá tốt về việc này nhưng làm sao để nó trở nên đẹp, bắt mắt người dùng khá khó nên cuối cùng quyết định làm mọi thứ đơn giản nhất có thể. Và vấn đề lớn nhất là bản than chưa có nhiều kiến thức về ngôn ngữ C# script nên một số vấn đề phát sinh như lớp trừu tượng không sử dụng được mặc dù đã viết theo tài liệu hướng dẫn, kết quả phải bỏ đi làm code lặp lại nhiều mang lại cảm giác dư thừa.

## **3. Hạn chế của chương trình và hướng phát triểu**

* Vẫn còn lỗi xảy ra trong quá trình thực thi chương trình và còn sót 1 số vấn đề cần phải phát triển:

+ Người dùng có thể chọn 2 Object nhưng lúc lưu thì sẽ lưu vào đâu?

Nếu tạo 1 Folder chứ thì nếu có nhiều Object hoán vị cho nhau thì số lượng sẽ rất nhiều, còn nếu tạo 1 Folder chứa hết thì lại khó tìm kiếm

+ Độ trễ khi click chuột chọn Object vẫn còn

+ Điều khiển camera muốn mượt mà phải quen được vì chưa làm camera xoay theo mà chỉ có khung nhìn xoay theo thôi, vì thế tọa độ của camera vẫn như cũ. Lúc quay phải biết mình đang ở hướng nào.

+ Chưa giải quyết vấn đề bị đè số hình đã chụp để khi tắt/mở lại thì chụp hình mới sẽ có trường hợp xóa hình cũ và lưu lại hình mới nếu chưa di chuyển hoặc đổi tên hình trước đó.

**V. Tài liệu tham khảo**

1. <http://gamestudio.vn/tin-tuc/28-unity-3d/huong-dan-tao-texture-tu-anh-chup-man-hinh-trong-unity-472.html> (tham khảo code chụp lại màn hình trong game) (ngày truy cập: không xác định)

2. <https://docs.unity3d.com/ScriptReference/ImageConversion.EncodeToPNG.html> (tham khảo code lưu hình ảnh trong folder) (ngày truy cập: không xác định)