**Chung**

* Điều khiển:
  + Con lăn để zoom in/out.
  + Đè con lăn để move around.
  + Alt + left-click để xoay xung quanh center.
* Kéo 1 hồi layout của unity bị loạn -> Góc phải trên cùng có cái pill -> Chọn Reset all layout
* Focus vào object trong tab Scene: Click vào nó bên cột Hierarchy -> nhấn F
* Trục tọa độ Oxyz: y là chiều cao.
* **Mọi thay đổi mình làm bên Inspector khi đang play game sẽ dc reset hết khi stop**

**Camera**

* *Nắm cái trục cho dễ*, có 3 mặt phẳng, tùy vào mình đang chọn và drag mặt nào thì điều chỉnh dc góc nhìn theo 2 trục của mặt đó:
  + Vd: Chọn Oxz thì kéo kiểu nào cũng k đổi dc chiều cao (chỉ đổi view dọc – ngang dc thôi), chọn Oxy thì dc.
* Thay vì background bầu trời thì Clear Flags -> Solid Color rồi chọn màu.

**Phím tắt**

* Graphical user interface

  Description automatically generated Từ trái qua W E R T, hay xài Move (W) và Rotation (E) nhất.\
* Ctr + D để duplicate object
* Asset store: ctrl + 9
* Phóng to cửa sổ: Shift + Space (nhớ click/hover vào cửa sổ đó trước)

**Inspector - Các component của 1 object**

* Transform (Có thể đè vào X, Y, Z rồi nắm kéo để dễ căn chỉnh hơn là nhập số)
  + Vị trí
  + Góc xoay
  + Kích thước
* Mesh Renderer: Hiện object ra màn hình.
* Box Collider: Sự va chạm giữa các physic object, nếu bỏ tick thì object đó nó sẽ đi xuyên qua các object khác.
* Mesh Filter: Chọn shape cho object (Hình khối, hình tròn, hình trụ..)

**Lỗi**

* The reference assemblies for .NETFramework,Version=v4.7.1 were not found
* Cài .NET SDK -> Cài đúng cái .NET devpack 4.7.1 cho nó, **PHẢI LÀ 4.7.1**. Nếu chưa dc thì cài thêm mono.
* Nó hiện lỗi liên quan tới quyền or connection này nọ thì cứ Run as admin..

<https://www.stdio.vn/unity-game/cac-thanh-phan-co-ban-trong-unity-Cf3Uw1>

Tạo Scene dưới mục Asset để có để đặt tên cho scene.

Link học <https://www.youtube.com/watch?v=IlKaB1etrik&list=PLPV2KyIb3jR5QFsefuO2RlAgWEz6EvVi6&index=2>

**Coding**

**Unity**

* Click vào object -> Nhìn qua Inspector -> Add component -> New script -> Double click vào script để nó vở visual studio lên -> Code -> Ctr + S -> Quay lại Unity cho nó đồng bộ hóa cái code vừa save -> Nhấn play bên Unity.
* Sau khi khai báo biến -> Vào Unity -> Bên Inspector kéo component (Hoặc kéo object bên Hierarchy vào cũng dc luôn) vừa khai báo trong code vào chỗ tên biến trong phần Script (Biến phải public)
* Syntax suggestion trong visual studio
* A screenshot of a computer

  Description automatically generated with medium confidence

**Chung**

* Biến kiểu float khi gán phải thêm chữ f vào (kẻo Unity complains). Vd: float forwardForce = 2000f.
* Debug.Log(“Hellooooo”): in ra console của Unity.
* Input.GetKey(“d”): trả về true/false.

**Hàm**

* Start(): Được gọi đúng 1 lần khi start game (Start is called before the first frame update)
* Update(): Mỗi frame hình gọi 1 lần (Update is called once per frame), **tùy vào tốc độ render frame của PC mà tốc độ gọi hàm này cũng thay đổi theo, không cố định**.
* Dùng Time.deltaTime (the amount of seconds since the computer dew the last frame) = 1/số frame được render trong 1s
  + Vd: rg.AddForce(0, 0, 2000 \* Time.deltaTime);
    - PC render được 10 frames/s -> Time.deltaTime = 0.1 -> Mỗi 0.1s add thêm 200 force -> Sau 1s add tổng cộng 2000 force.
    - PC render được 20 frames/s -> Time.deltaTime = 0.05 -> Mỗi 0.05s add thêm 100 force -> Sau 1s add tổng cộng 2000 force.
* FixedUpdate(): Mọi code liên quan đến physics nên nằm trong hàm này (Unity likes this a lot)

**Rigidbody**

* rg.useGravity = false/true;
* rg.AddForce(value on x, value on y, value on z);
* Trong Unity, Rigidbody -> Constraints -> Có thể tick vào các ô tương ứng để khóa chuyển động (tick rồi chạy thử là biết).
* Khi tác động lực, object bị nảy lên rồi lệch hướng dần là do ma sát -> Vd muốn nó lăn thẳng đều hoài thì chọn 1 trong 2 cách:
  + Set constraint trong Rigidbody
  + Thêm Physics Material vào cho object Ground rồi sửa Dynamic friction, Static friction về 0.

**Camera**

* Muốn camera nó lăn lộn, nghiêng ngả theo movement của object thì kéo cho Main Camera thành con của object đó.
* Muốn chỉnh góc nhìn từ sau lưng object, trong hàm Update():
  + transform.position = player.position + offset;
    - player có kiểu Transform
    - offset có kiểu Vector3 (dùng để camera nó k bị lọt vào bên trong object -> biến thành góc nhìn thứ nhất).