Platformer Directions

1. File-New project – Platformer\_Demo or whatever you want
2. Give it time, left click on the center map/scene and press “2”
3. GameObject- new empty
4. Add component, sprite renderer, black box, make it a larger size to see
5. Add component box collider 2d
6. Add component rigidbody2d, set linear drag to 2, angular drag to 2, fixed angle = true
7. New gameobject, addcomponent sprite renderer, white box, box collider 2d
8. Play
9. New physics 2d material, name bounce, set bounciness to 0 (< 1 slows down, > 1 gains speed), friction to 0
10. Apply to ball
11. Make sure all objectes have z = 0
12. Play
13. Create new script, name ball controller
14. Open script, in update type
    * void Update () {
    * if (Input.GetKey(KeyCode.A)) {
    * this.rigidbody2D.AddForce(new Vector2(-30, 0));
    * } else if (Input.GetKey(KeyCode.D)) {
    * this.rigidbody2D.AddForce(new Vector2(30, 0));
    * } else if (Input.GetKey(KeyCode.D)) {
    * this.rigidbody2D.AddForce(new Vector2(0, -15))
    * }
15. Save, run
16. Open same script, add line
    * if (Input.GetKeyDown(KeyCode.W)) {
    * this.rigidbody2D.AddForce(new Vector2(0, 80));
    * }
17. Explain player has infinite jumps, add counter
    * void OnCollisionEnter2D(Collision2D col) {
    * if (!col.gameObject.GetComponent<BallController>()) jumpcounter = 0;
    * }
    * void OnCollisionExit2D(Collision2D col) { // to prevent two jumps if you fall off
    * if (!col.gameObject.GetComponent<BallController>()) jumpcounter++;
    * }
    * //change other
    * if (Input.GetKeyDown(KeyCode.W) && jumpcounter < 2) {
    * if (jumpcounter > 0) jumpcounter++;
    * this.rigidbody2D.velocity = new Vector2(this.rigidbody2D.velocity.x, 0);
    * this.rigidbody2D.AddForce(new Vector2(0, 300));}
18. Add methods for other players
    * 1. bool GetLeftKey(float id) {
    * if (id == 0) return Input.GetKey(KeyCode.A);
    * if (id == 1) return Input.GetKey(KeyCode.J);
    * return false;
    * }
    * bool GetRightKey(float id) {
    * if (id == 0) return Input.GetKey(KeyCode.D);
    * if (id == 1) return Input.GetKey(KeyCode.L);
    * return false;
    * }
    * bool GetJumpKey(float id) {
    * if (id == 0) return Input.GetKeyDown(KeyCode.W); //keydown for the rest
    * if (id == 1) return Input.GetKeyDown(KeyCode.I);
    * return false;
    * }
      1. bool GetDownKey(float id) {
    * if (id == 0) return Input.GetKey(KeyCode.S);
    * if (id == 1) return Input.GetKey(KeyCode.K);
    * return false;
    * }
    * bool GetAttackKey(float id) {
    * if (id == 0) return Input.GetKeyDown(KeyCode.E);
    * if (id == 1) return Input.GetKeyDown(KeyCode.O);
    * return false;
    * }
19. Replace input.getkey with the get keys
20. Add another entity, make their color red, set player id to 1
21. Play around, watch them jump around each other
22. Add attack and takedamage functions
    * public float damage;
    * Add this in keys : if (GetAttackKey(playerid)) {
    * This.SendMessage(“Attack”);
    * }
    * // create child gameobject, add circlecollider, set istrigger to true, do the same for both.
    * Add to ballcontroller:
    * public void TakeDamage(float f, Vector2 placement) {
    * this.damage += f;
    * this.rigidbody2D.AddForce(new Vector2(0, damage));
    * this.rigidbody2D.AddForce (f \* damage \* ((Vector2)this.transform.position - placement));
    * //this.rigidbody2D.AddForce (new Vector2(0, damage));
    * }
    * //create new script, called AttackZone
    * private bool inzone;
    * private GameObject focus;
    * void OnTriggerEnter2D(Collider2D col) {
    * if (col.GetComponent<BallController>()) {
    * inzone = true;
    * }
    * }
    * void OnTriggerStay2D(Collider2D col) {
    * if (col.GetComponent<BallController>()) focus = col.gameObject;
    * }
    * void OnTriggerExit2D(Collider2D col) {
    * if (col.GetComponent<BallController>()) {
    * inzone = false;
    * }
    * }
    * void Attack() {
    * if (inzone) {
    * focus.GetComponent<BallController>().TakeDamage(5, this.transform.position);
    * }
    * }
    * }
23. Add attack cooldown time
    * Two public variables, one being “attackcooldowntime” and the other “attackcooldownsofar”
    * Change attack method to:
    * this.coolDownAttack+= Time.deltaTime;
    * if (GetAttackKey(playerid) && coolDownAttack > this.damageCooldownTime) {
    * coolDownAttack = 0
    * this.transform.FindChild("TriggerZone").SendMessage("Attack");
    * }
24. Add four boxes, each trigger boxes, and add a script called “out of bounds”
    * void OnTriggerEnter2D(Collider2D col) {
    * if (col.GetComponent<BallController>()) {
    * col.transform.position = respawnPoint;
    * col.rigidbody2D.velocity = Vector2.zero;
    * col.GetComponent<BallController>().deaths++;
    * col.GetComponent<BallController>().damage = 0;
    * }
    * }
25. Add new component, add guitext, add text and set font to 10, set to scene view, look for the thingy and set it to your desired place
26. Create guitext script and add
    * void Update () {
    * string text = "";
    * GameObject[] Players = GameObject.FindGameObjectsWithTag("Player");
    * foreach (GameObject p in Players) {
    * text += "Player " + p.GetComponent<BallController>().playerid +
    * " " + p.GetComponent<BallController>().damage +
    * "% Deaths = " + p.GetComponent<BallController>().deaths + '\n';
    * }
    * this.guiText.text = text;
    * }
27. Add component to camera, label it “CameraZoomer”
    * public float dampTime = 0.15f;
    * private Vector3 velocity = Vector3.zero;
    * GameObject[] targets;
    * public float bufferX = 0, bufferY = 0;
    * void Update ()
    * {
    * targets = GameObject.FindGameObjectsWithTag("Player");
    * if (targets != null)
    * {
    * float avgX = 0;
    * float avgY = 0;
    * int i = 0;
    * foreach (GameObject g in targets) {
    * avgX += g.transform.position.x;
    * avgY += g.transform.position.y;
    * i++;
    * }
    * avgX += GameObject.Find("CenterPlatform").transform.position.x;
    * avgY += GameObject.Find("CenterPlatform").transform.position.y;
    * i++;
    * Vector3 avg = new Vector3(avgX/i, avgY/i, this.transform.position.z);
    * print(avg);
    * dampTime = 1/Vector2.Distance(this.transform.position, avg);
    * Vector3 point = camera.WorldToViewportPoint(avg);
    * Vector3 delta = avg - camera.ViewportToWorldPoint(new Vector3(.05f, .05f, point.z));
    * Vector3 destination = transform.position + delta;
    * destination.Set (destination.x + bufferX, destination.y + bufferY, destination.z);
    * transform.position = Vector3.SmoothDamp(transform.position, destination, ref velocity, dampTime);
    * }
28. There we go, your new game is fun to play. Feel free to edit it around, and enjoy yourself to add your own little things.
29. Known bugs:
    * Can double jump again if you hit the edge of the platform