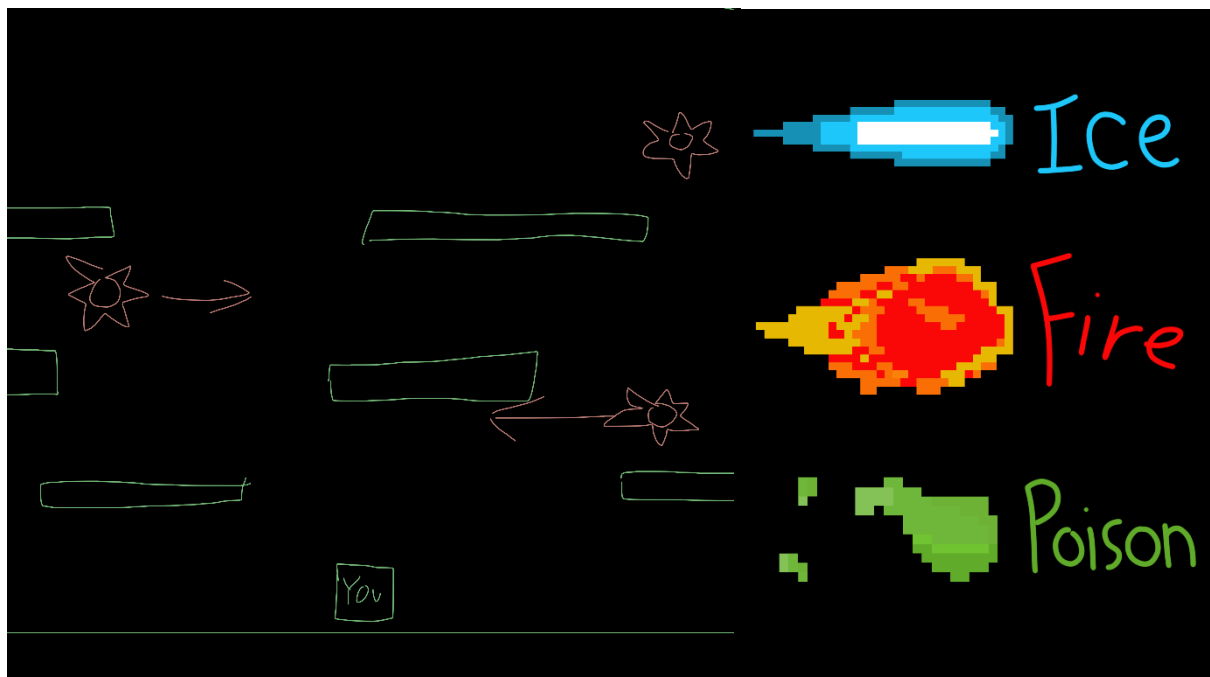


GAME ENGINES: TASK 1

By Roberto Eminyan & Stephen Cini

We will be creating a hyper-casual, platformer, jumper pixel game using the game engine Unity. The goal of the game is for the player to avoid colliding with platforms and projectiles that are unsafe for him.

The player will be portrayed as a young knight, avoiding the dangerous elements of the three-headed Hydra that will be in the background shooting three different elemental projectiles. The knight will have to maneuver through elemental platforms while avoiding falling into the lava pit below. The three elements of the hydra will be **ice**, **fire** and **poison**.



The game's only rule is not to die and hit the highest possible score. By jumping around on various platforms, which will also be element based. The player must escape the hydra's projectiles (**ice**, **fire** and **poison**). Both platforms and projectiles in the game will be RNG (Random Number Generated), which will make each gameplay different and unique, making it less of a pattern and more of a new game each round.

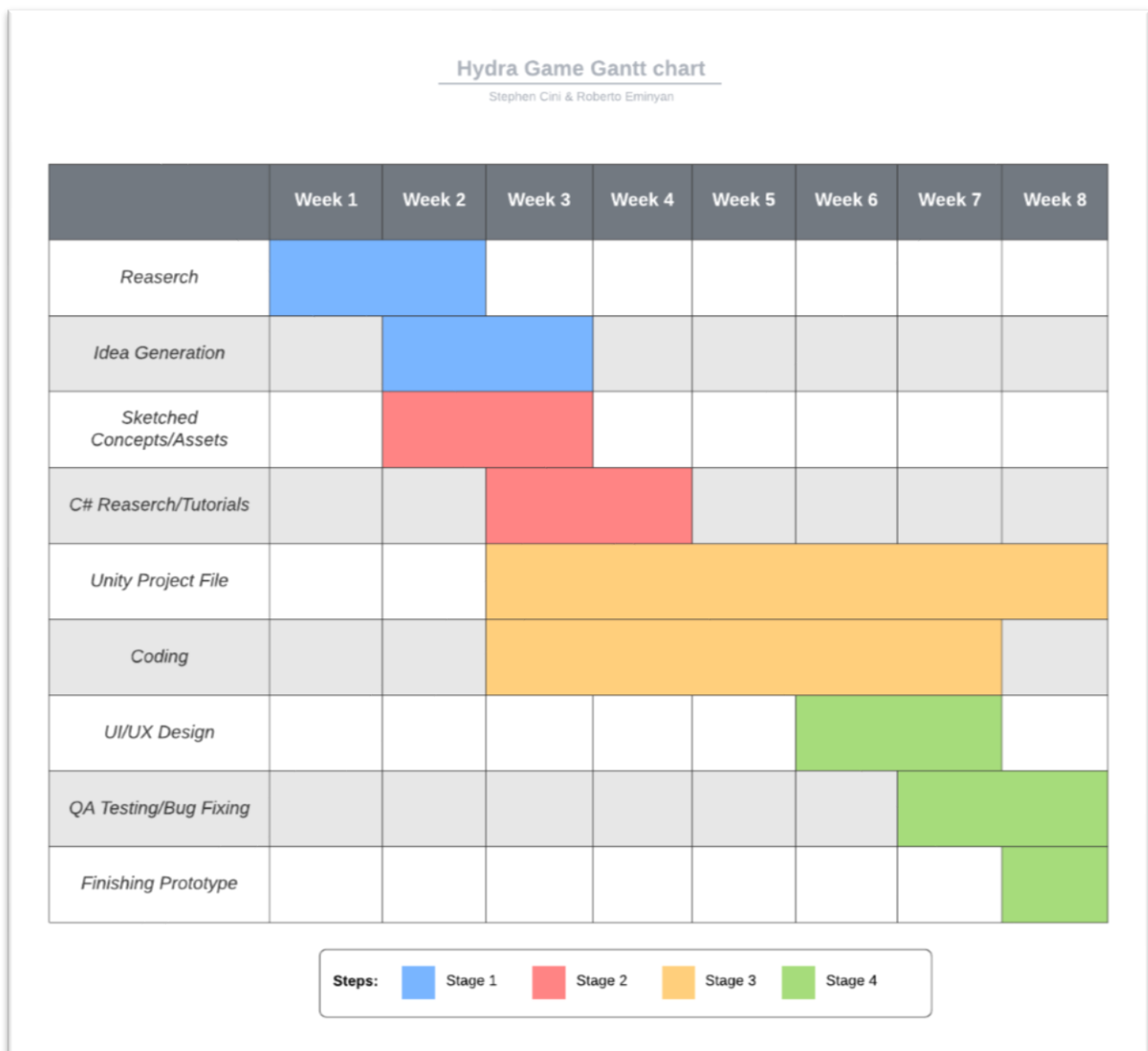


A mechanic in the game is that the armor of the knight switches its element only after a period of time has passed, which means that the player will be immune to the given current element.

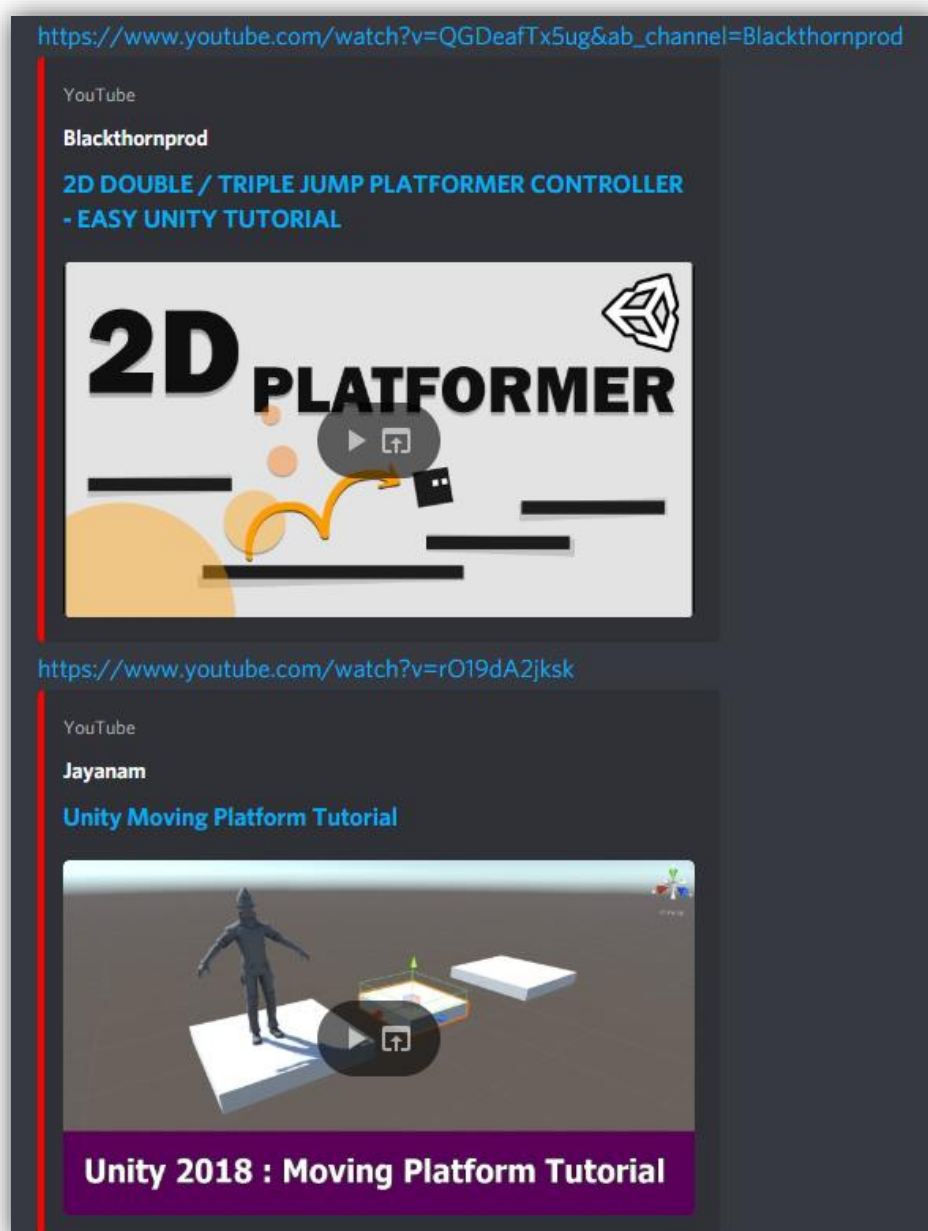
For example, if the player is currently in the **ice** element, only **fire** or **poison** platforms and projectiles can harm him. The player must only land on ice platforms, or else he's going to get injured and lose a life, leaving him with only a few seconds to move and not lose the game.



We constructed this Gantt chart in order to use as a tool and to help us guide our project throughout the stages of development.



Using YouTube tutorials and online sources we gathered information that can help us throughout our game project. We researched how we can code mechanics such as player movement, platform parenting, smooth physics, object warping, collisions and projectile shooting.



Parenting player to moving platform;

Transform.parent

[SWITCH TO MANUAL](#)

public [Transform](#) parent;

Description

The parent of the transform.

Changing the parent will modify the parent-relative position, scale and rotation but keep the world space position, rotation and scale the same.

See Also: [SetParent](#).

```
using UnityEngine;
using System.Collections;

public class ExampleClass : MonoBehaviour
{
    public GameObject player;

    //Invoked when a button is pressed.
    public void SetParent(GameObject newParent)
    {
        //Makes the GameObject "newParent" the parent of the GameObject "player".
        player.transform.parent = newParent.transform;

        //Display the parent's name in the console.
        Debug.Log("Player's Parent: " + player.transform.parent.name);

        // Check if the new parent has a parent GameObject.
        if (newParent.transform.parent != null)
        {
            //Display the name of the grand parent of the player.
            Debug.Log("Player's Grand parent: " + player.transform.parent.parent.name);
        }
    }
}
```

Using bool to check if player isOnMovingPlatform = true;

Description


Does the object exist?

The two examples below give the same result.

```
using UnityEngine;

public class Example : MonoBehaviour
{
    // check if there is a rigidbody attached to this transform
    void Start()
    {
        if (GetComponent<Rigidbody>())
        {
            Debug.Log("Rigidbody attached to this transform");
        }
    }
}
```

In order to spawn different projectiles and different material platforms we need to create a random integer generator;



The image shows a YouTube video player interface. The video title is "C# Lesson! How To Generate a Random Number or a Random String...". The video thumbnail features a purple background with a white die and a C# logo. The text "HOW TO GET A RANDOM NUMBER OR STRING" is prominently displayed in white. The Unity Tutorials logo is visible in the bottom left of the thumbnail. Below the video player, the title "How To Generate a Random Number or a Random String" is repeated. The author is listed as "by Oxmond" and the date is "August 20, 2019". The video is categorized under "TUTORIAL, UNITY" and has 2 comments. A description at the bottom states: "How to work with date and time. In this tutorial we'll work with the system datetime. How to get it, and how to format it."

C# Lesson! How To Generate a Random Number or a Random String...

Watch later Share

**HOW TO GET A
RANDOM
NUMBER
OR
STRING**

 **UNITY
TUTORIALS**

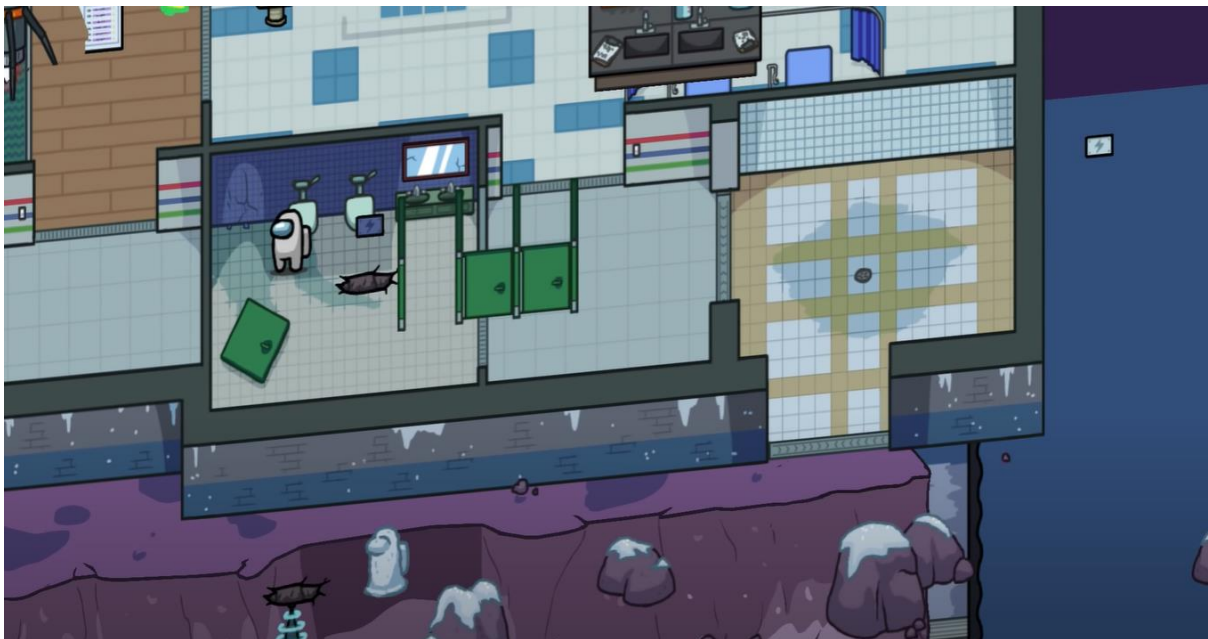
How To Generate a Random Number or a Random String

by Oxmond August 20, 2019

TUTORIAL, UNITY 2

How to work with date and time. In this tutorial we'll work with the system datetime. How to get it, and how to format it.

We took inspiration of one of our favorite games Among us which is a 2.5D PC videogame. As we researched, we discovered many new possibilities that could be included in our prototype. <https://youtu.be/N795ZNruIDM>



CRC Cards

Player
+ Player Movement + Rigidbody + 3 lives + Dmg invincibility + Time changing material
+ Platform + Projectiles + Timer ,lives, score

Platforms
+ Movement + Random generator + Warping + Damage collider + Collision
+ Player + Timer ,lives, score

Projectiles
+ Random generator + Movement + Speed + Damage collider
+ Player + Timer, lives, score

Timer, lives, score
+ Time tracking + Random generator + Scaling for slider + life count + live score count
+ Player + Projectiles + Platforms