

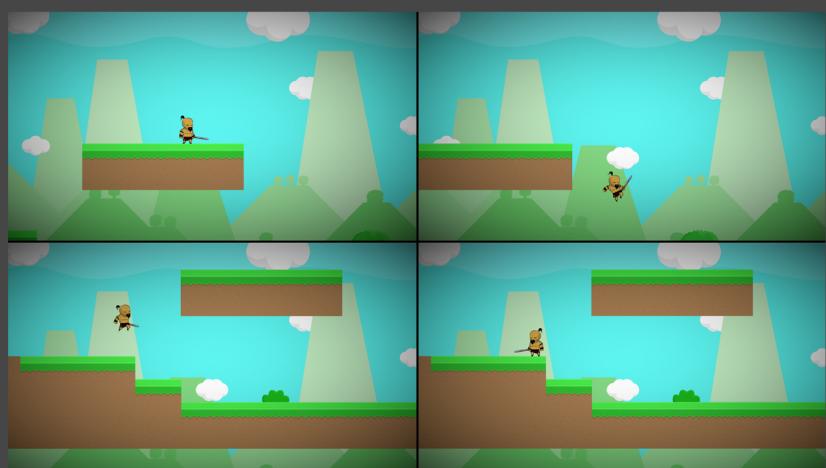
# Improving user experience through game camera implementation

Kieran Bond

## Introduction

The camera is a powerful tool for a designer to utilise and has an 'enormous influence over the final quality of the game' (Kremers, 2009) through more than just gameplay.

The investigation examined implementations of cameras in video games and what can be done to improve the user experience, through the camera. To demonstrate the research, a project was created that implements some of what was found with the Unity3D game engine.



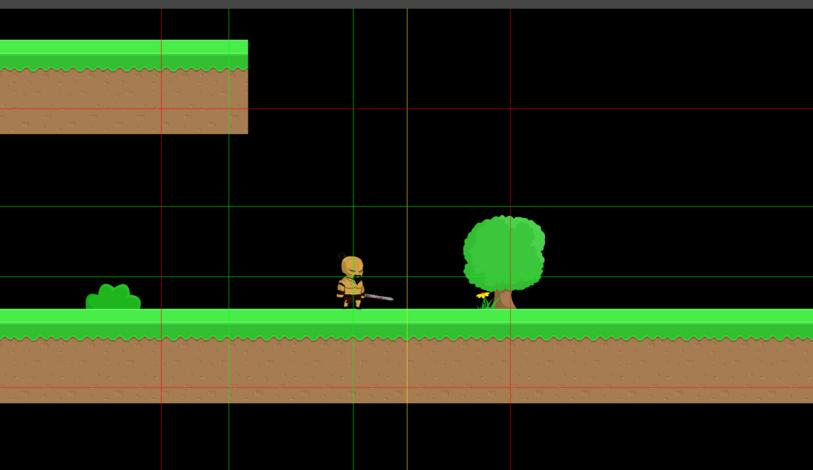
The aim of this implementation was to create a tool that aids the designer in building the kind of gameplay the team has as a goal, allowing them to create immersive and emotional gameplay by providing a powerful camera tuning system - A solution by from Meigs (2003).

The tool was developed in the Unity3D game engine, using C#. The tool design took heavy influence from the game Tiny Thor and an article released about its camera system (Heizmann, 2017), as well as influence from Keren (2015) who describes many different camera implementations.

The tool was developed with ease of use in mind, but does require a small amount of coding to implement a small interface on the target, for maximum compatibility.

```
namespace CameraDesign.Controller.API
{
    public interface ICameraTarget
    {
        Transform m_transform { get; }
        Vector2 m_velocity { get; }
        bool m_isGrounded { get; }
    }
}
```

## Features



The main features developed, are a focus zone and a danger zone.

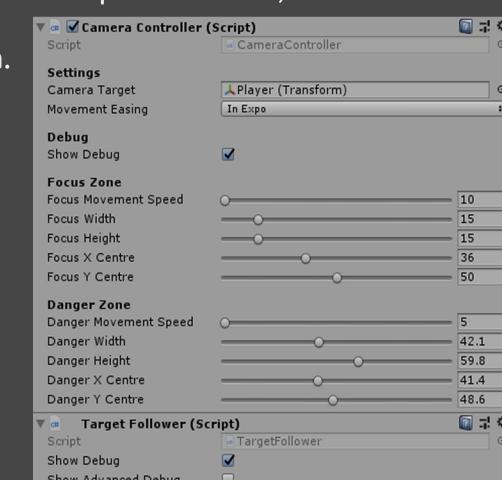
The focus zone (green lines) aims to provide a smoother experience, doing so by providing a Linearly Interpolated (Lerp) movement once the player leaves the zone, and not moving the camera whilst the player is inside the zone.

The danger zone (red lines), also known as panic lines, provides a hard border to how far the player can move from the camera before the camera moves very quickly to close the gap and avoid the player moving too far from focus.

These features work together to provide soft and hard borders for the player to explore within, whilst avoiding jarring movements and offering freedom.

These features are completely customisable and provide a solution to a common problem found in games - that the camera should never let the player out of sight, as told by Rogers (2014).

The camera also features a more complex following on the Y-axis, by only moving when past the panic lines or when touching ground. This adds more cinematic effect to falling and jumping whilst ensuring control is kept with no jarring effect from rushed movement.



## Analysis

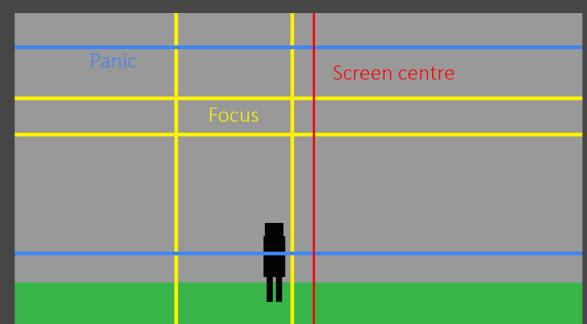
A downfall of an automated camera system is that it strips the player of control, which can lead to the breaking of immersion. Rogers (2014) believes this to be justifiable as they state 'Bad things happen when players are given control of the game camera'.

This downfall is also countered by providing the player with a smooth experience without an interruptive camera, thus decreasing frustration along with providing freedom to the designer - The settings are changeable at runtime (through scripts), alongside being straightforward to implement into an already existing game.

Although the system does not automatically make use of the rule of thirds, it can be done by the designer. This could not be included without stripping the designer of control. Haign-Hutchinson (2009) believes that cinematography is not always appropriate so by not enforcing this we leave it to the designer to decide if it will be appropriate.

## Conclusion

The implementation of the design provides a good example of the direction to take when undergoing creating a strong camera system for a game. It remains simple and easy to use, whilst being effective at providing the desired effect and getting the camera right can make all the difference.



This system demonstrates that if the camera is thoroughly designed early in their development then the notorious camera wanting to kill players situation as described by Totten (2014) is avoidable, as the difficulty in making camera AI is reduced.

## References