# Game Design Document For RWM P2 2016-17

# 

[**Game Design Document For RWM P2 2016-17**](#_c8f0jmntbuec) **1**

[**https://github.com/itcgames/RWM\_1617\_P3\_Stephen\_Ennis**](#_fa0hgy75vwy) **2**

[**2D Camera system**](#_b1k58e9rxc5q) **2**

[Pan [76]](#_vaf18ch9suwi) 2

[Conditions of satisfaction](#_ozulw6j5wrsj) 2

[Zoom [77]](#_tvozkyy4ec3o) 3

[Conditions of satisfaction](#_h4l9lgtl9c7d) 3

[Zoom to fit [78]](#_etbttbtnm7c4) 3

[Conditions of satisfaction](#_gmg7cwkvaj8e) 4

[Parallax [79]](#_om0mokpi7nlq) 4

[Conditions of satisfaction](#_76vyjudezgqm) 4

[Constraints](#_gsekl8ex29xu) 5

[Conditions of satisfaction](#_x7766ijbmrqs) 5

[Camera shake [80]](#_teqxsw5ufqyw) 5

[Conditions of satisfaction](#_b4gf9fx5adhl) 5

[Attractors/Repulsors [82, 83]](#_7i85jvp2vgex) 6

[Conditions of satisfaction](#_z8pzyqyq18yq) 6

[**User Guide:**](#_ed4taxawbxhg) **6**

# 

# 

# <https://github.com/itcgames/RWM_1617_P3_Stephen_Ennis>

# 

# 2D Camera system

The component being created is a 2D Camera system in SDL + Box2D compatible. This will consist of a camera of specified width and height being created. Only objects within the camera bounds will be visible. The camera will fit the window of the application. It will be capable of a number of basic transformations (pan, zoom) and more advanced functionality (parallax). It should be easy for a user to integrate into any game whether it’s side on such as in a platformer or if it’s top down.

The following features all modify the zoom/position of the camera. These values are used to compute a screen point/rectangle from a given “world” point/rectangle. This transformation allows the new rectangle to be drawn in such a way that it appears as if there is a camera. (In reality, everything else is being displaced.)

## **Pan [76]**

Pan is how the camera is capable of moving around the screen. This can be achieved by using the directional arrow keys. The camera will accelerate slowly up to a maximum velocity once it is moving and similarly decelerate quickly when the user input stops . Holding two keys at once is possible to moving diagonally. The values for max velocity and acceleration rate can be changed along with the drag. This could result in a varying amount of possible camera motions. The centre of the camera can be restricted to a certain rectangle. The cameras velocity and acceleration on contact with this bounds will be zeroed in order for the camera to move in the other direction cleanly.

**Manual Test**

- use right arrow to move right

- notice acceleration and increasing speed until max velocity is reached

- press up arrow to start curving up

- release keys and notice slow down and eventual stop

**Video**

[video](https://drive.google.com/open?id=0B9RF991BhZlzcElXZjZOeDdxU3M)

### **Conditions of satisfaction**

* Camera can accelerate up to max velocity
* Camera will slow down due to drag no input making it move
* Camera comes to rest when no forces acting on it

## **Zoom [77]**

The camera will be capable of zooming in and out around the centre of the screen. Zoom speed will be constant and can change direction quickly. There will be a minimum and maximum zoom threshold. Rather than user input only being the cause of a zoom, a zoom to feature is possible. Here a zoom value can set as the target to zoom to. If the target is past the min/max thresholds it will stop at them or 0 if no max zoom. Zoom thresholds for the min and max zoom and be set to a given value or unlimited. The user can also directly set the current zoom value which will apply instantly.

**Manual Test**

- Click a few times to create some boxes in order to easily notice the effects of zoom

- zoom in by scrolling forward

- zoom out by scrolling backward

- press left bracket and notice how you don't need to provide input for it to zoom out

-press right bracket and notice how you don't need to provide input for it to zoom in

- notice how it zooms very far due to unlimited max zoom threshold

- press v to set threshold

- notice it is automatically zoomed out to this threshold

**Video**

[video](https://drive.google.com/open?id=0B9RF991BhZlzZFk5S2NCVVFCejA)

### **Conditions of satisfaction**

* Can zoom in and out
* Can zoom to a given value
* Can set zoom thresholds

## **Zoom to fit [78]**

A zoom to fit is where the camera will zoom to encompass all objects specified. The objects may either be a vector of points or rectangles. The camera will move and change its zoom in order to encapsulate all of the objects while preserving its aspect ratio. Another option is to not maintain the aspect ratio. The result is a camera view which perfectly encloses the minimum space required to ensure those objects remain fully visible. If centre is being restricted then movement will stop at the restrict bounds. Zoom to fit will not work if the user is currently moving the camera. A third parameter is the time of the zoom to fit operation to complete. If set to 0 it will be instant. -1 is the default speed.

**Manual Test**

- right click a number of times to create points

- press x to zoom to fit these points

- notice how all are visible

- press c to zoom again but not to preserve aspect ratio

- notice the stretched view

- zoom out and notice the aspect ratio being reset

- left click a number of times to create rectangles

- press z to zoom and fit these

- notice how all rectangles are within the view

**Video**

[video](https://drive.google.com/open?id=0B9RF991BhZlzVW55R3gtSkU0Njg)

### **Conditions of satisfaction**

* Zoom to fit the centre of a group of points with all points visible
* Zoom to fit the centre of a group of rectangles with the entirety of all rectangles visible
* Zoom to fit the centre of a group of points/rectangles with all entities visible while also changing the aspect ratio
* Capable of resetting aspect ratio

## **Parallax [79]**

Parallax scrolling will be possible with the camera. This will allow multiple layers to scroll at different speeds to create the illusion of depth. The foreground layers move quicker in a parallax effect. A layer will be a looping set of images that move as the camera does in the world. Horizontal and vertical scrolling are both possible. The images used will be scaled to the window size. The effect is made up of at least 2 layers. A layer can use many images or copies of the 1 image.

**Manual Test**

-Press p to start parallax

- press again to stop

- notice it stop

- press once more to reenable it

- move left to see the layers scroll at different speeds

-press o to increase the scroll speed of the mountains

-press k to decrease it

-press k enough and notice how the mountains go in reverse

**Video**

[video](https://drive.google.com/open?id=0B9RF991BhZlzTFJGLUdybGtjbDA)

### **Conditions of satisfaction**

* Can enable and disable parallax scrolling
* Can have multiple different parallax effects stored
* Parallax scrolling also takes place while screen is shaking or zooming to fit
* Parallax scroll while moving
* Can scroll on x axis OR y axis
* Can set scroll speeds for each layer
* Can change these speeds in the middle of the effect
* Can use multiple images for a layer

## **Constraints**

Certain constraints can be placed on the camera motion. The possible constraints are preventing movement on the x/y axis and preventing movement altogether.

**Manual Test**

- move left and notice speed

- move down at the same time

- press 1 to constrain horizontal motion but vertical speed prevails

- press 1 to re enable horizontal motion

- repeat but press 2 to for vertical motion

- repeat but press 3 to lock all motion

**Video**

[video](https://drive.google.com/open?id=0B9RF991BhZlzOTlHUlZWZHZvZWs)

### **Conditions of satisfaction**

* Stop horizontal motion and zero out any existing velocities/accelerations
* Stop vertical motion and zero out any existing velocities/accelerations
* Stop both horizontal and vertical motion (lock) and zero out any existing velocities/accelerations
* Enable all of the above again

## **Camera shake [80]**

The camera is able to shake. This causes it to move short distances in random directions in uniform manner. This is perfect to implement using perlin noise as its concept of smooth random appears more “normal” for a camera shake than completely random changes. This effect can be started at any time and it’s magnitude and duration can be changed.

**Manual Test**

-press S and notice shake

-press S to toggle off

- once again press S to toggle back on

- press i to increase magnitude

- notice intensity of shake increases

- press j to decrease and notice weaker shake

**Video**

[video](https://drive.google.com/open?id=0B9RF991BhZlzUk1Yai1WLVMtbWM)

### **Conditions of satisfaction**

* Can enable and disable shaking
* Can have multiple different shaking effects stored
* Can change parameters (magnitude and duration) at creation and runtime
* Implemented using perlin noise in order to closer mirror the real life behaviour of camera shaking

## **Attractors/Repulsors [82, 83]**

An attractor or repulsor can be used to influence a camera’s position. An attractor will try to draw the camera towards it to highlight something of importance. A repulsor is the opposite and will push the camera away from it.

**Manual Test**

- press right arrow to move right

- move mouse to the left of the centre and press a to create attractor

- move towards it and notice it pulls you into the centre

- keep moving to get out

- repeat but press r to create repulsor and notice it pushes you away from the centre

**Video**

[video](https://drive.google.com/open?id=0B9RF991BhZlzOFVyM0tReFpLZms)

### **Conditions of satisfaction**

* While within the area of an repulsor, the camera position is pushed away from the centre of the repulsor
* Closer to the centre means a strong force
* While within the area of an attractor, the camera position is pulled towards the centre of the attractor, depending on at what velocity it has been set to stop at, the position may orbit the centre before coming to rest.
* Strength can be changed for attractor/repulsors

# User Guide:

<https://docs.google.com/document/d/1sqh2Kxz-3_OvuQSfxEgbbOuWODR64V2G_1FzD0sQNaE/edit?usp=sharing>