

This document describes how to create a new round.

A round is contained in a single .js file, which must be loaded in [screen.html](#) before loading roundList.js

In the round-defining .js file are the following statements:

var round* = {} // creates an object to hold the round information *) name must be unique

- round.**rooms** must be an array which contains room objects
- round.boss is a string describing the boss to spawn in the last room. The accepted strings are defined in [enemyManager.js](#)
- round.bgm is a string that defines the background music to be played during the round. Accepted strings can be checked in load.js.

Room objects are objects that contain several necessary variables

- room.roomBg contains a string that points to a loaded background image defined in load.js
- room.colliders is an object which itself contains all collision objects to be made into sprites within the Phaser engine in [room.js](#); **colliders** must be defined beforehand
- room.moveDirection contains a string that defines a direction where the camera will scroll next once it reaches the middle of the room. Accepted strings are 'east', 'west', 'north' and 'south'
- room.moveSpeed contains a string that defines the speed at which the camera will scroll to the next room: 'normal', 'fast' and 'stop' are valid options, 'stop' is only used in the last room of the round

Colliders are objects, which contain several number values that are used in the creation of collision sprites.

To simplify collision object creation, [util.js](#) contains helper objects to be used with collider placement:

- wallCollider (width, height, xPos, yPos, xAnchor, yAnchor)
- smallCollisionSprite (xPos, yPos, image)
- largeCollisionSprite (xPos, yPos, image)

(Placing a collision sprite on xpos= 0, ypos = 0 will place it on the top leftmost tile)

width, height, xPos and yPos are values that are base on the used tileHeight and tileWidth. The tile values used in the game are:

- maxTilesX = 24 // the rooms are this many tiles wide
- maxTilesY = 15 // the rooms are this many tiles tall
- tileWidth = 80 // a tile is this many pixels wide

- tileHeight = 72 // a tile is this many pixels tall
- image is a string that the collidable object will use (contained in load.js)
- xAnchor and yAnchor are the anchor values that Phaser uses on sprites to define their origin point location, values between 0 - 1 are within the bounds of the image. When creating the sprites, these values are set before object placement

An Example Round with two rooms, two collidable walls and two pillars:

```
var room1colliders = {};
// left wall, 1 tile wide
room1colliders.leftwall = new wallCollider (1, maxTilesY, 0, 0, 0, 0);
// a pillar
room1colliders.pillar = new smallCollisionSprite (2, 2, 'pillar');

var room2colliders = {};
// right wall, 1 tile wide
room2colliders.right = new wallCollider (1, maxTilesY, maxTilesX, 0, 1, 0);
// a pillar
room2colliders.pillar = new smallCollisionSprite (2, 2, 'pillar');

exampleRound = {};
exampleRound.rooms = [
    {"roomBg": "room1", "colliders": room1colliders, "moveDirection": "east",
    "moveSpeed": "normal"},
    {"roomBg": "room2", "colliders": room2colliders, "moveDirection": "null",
    "moveSpeed": "stop"}
];
exampleRound.boss = 'king';
exampleRound.bgm = 'bgm02'
```

After defining the round it must be loaded in [screen.html](#), and to add it into the round rotation the name of the round variable (exampleRound in this case) must be added into the array in [roundList.js](#)