Gameobject Classes

public function Food()

creates reference objects for variables used for food and initializes food settings

includes:

* Which way food is facing
* Movie clip for food
* Pos x, y, z for food

Different foodtypes are defined

* foodType 0, 99, 100, 101

Defines the width of the movieclip

Function update()

Checks if food object is dead or alive and updates it. If the food is not alive, the respawn timer is checked. Once the time is greater than the respawn time the food object is reinitialized and set to alive. Otherwise food object is set to random movement and random movement of segment is established.

Function turnRandom()

The currentAim of the food object is manipulated randomly.

Effect

Function reset ()

Reference objects of common variables are initialized

Alive is set to true and timer of object is set to zero

MovieClips positions are reinitialized and rescaled depending on type of effect of object

* eType is either 0 or 1

Function update

Timer is incremented. Once greater then delay time of object and alive status is checked to continue movieClip

Otherwise movieclip is removed, alive is set to false, and timer set to 0 for object.

Game Object

Function GameObject()

Initializes object position in the game and visual factors of the object such as blur, glow, and color.

Function applyCamera()

Takes factors of game level and position z to apply scaleRatio and blur factor of object.

Changes soundvolume depending on position of player on the the x y grid.

Objects are rescaled taking into account offset from the camera. That is the x, y, and z position.

Function blurImage()

Applies blur to images

Functioin shadeImage()

Applies the primary colors red, green, and blue to objects

Function dist()

Calculates the distance between two objects and returns distance for other calculations.

How difficulty changes through player interactions:

var level:Number;

var levelSize:Number;

var zDepth:Number;

this.zDepth = level\*zLevelGap;

public function Jellyfish(

1. posX:Number,
2. posY:Number,
3. numSegs:Number,
4. maxSegs:Number,
5. randEvolve:Number,
6. segLength:Number,
7. speed:Number,
8. turnSpeed:Number,
9. panic:Boolean,
10. mc:MovieClip,
11. ID:Number,
12. prefix:String)

fish[fishLength] = new Jellyfish(

1. random(levelSize\*2)-levelSize,
2. random(levelSize\*2)-levelSize,
3. numSegs,
4. maxSegs,
5. randEvolve,
6. segLength,
7. speedMin+random(speedVar),
8. (turnSpeedMin+random(turnSpeedVar))/180\*Math.PI,
9. panic,
10. currentObj,
11. fishLength,
12. assetPrefix);

if (wounded || coward) {

currentSpeed = panicIntensity\*speed;

} else if (!wounded && !coward && panicIntensity>1) {

currentSpeed = panicIntensity\*speed;

panicIntensity \*= 1-dt\*2;

if (panicIntensity<=1) {

panicIntensity = 0;

Player attacking creature will lead their panic intensity and speed to increase making it harder to kill the creature if the current state of the creature is set to coward, or harder for the player to stay alive if the creature is wounded but not set to coward.