

Topalian Game Engine 019

by
Christopher Andrew Topalian

All Rights Reserved
Copyright 2000-2024

Dedicated
to
God the Father

Topalian Game Engine

file:///D:/_1Code/0_JS_Published/Topalian_Game_Engine/Topalian_Game_Engine/Topalian_Game_Engine.html

Instructions

Periodic Table of Ele

1 H HYDROGEN 1.00794		floor1									
3 Li LITHIUM 6.941	4 Be BERYLLIUM 9.012182										
11 Na SODIUM 22.9903	12 Mg MAGNESIUM 24.305										
19 K POTASSIUM 39.098	20 Ca CALCIUM 40.078	21 Sc SCANDIUM 44.955912	22 Ti TITANIUM 47.867	23 V VANADIUM 50.9415	24 Cr CHROMIUM 51.9961	25 Mn MANGANESE 54.938045	IRON 26 Fe	27 Co COBALT 58.933195	28 Ni NICKEL 58.6934	COPPER 29 Cu	Zn
37 Rb RUBIDIUM 85.46	38 Sr STRONTIUM 87.62	39 Y YTTRIUM 88.90585	40 Zr ZIRCONIUM 91.24	41 Nb NIOBIUM 92.90638	42 Mo MOLYBDENUM 95.94	43 Tc TECHNETIUM 97.90	Ru	45 Rh RHODIUM 102.9055	46 Pd PALLADIUM 106.42	SILVER 47 Ag	Cd

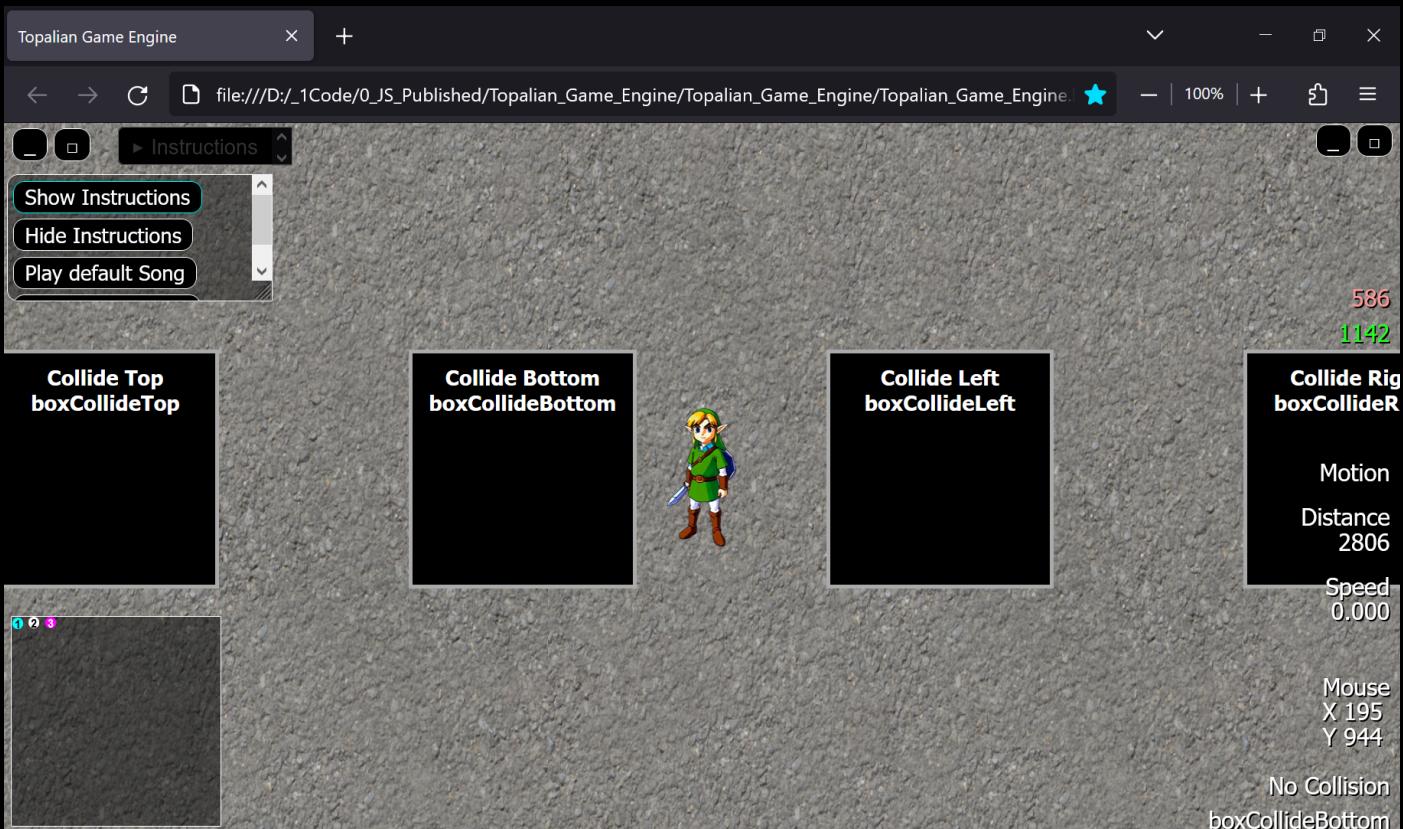
Legend:

- alkali metals
- alkaline metals
- transition metals
- other metals
- metalloids
- nonmetals
- halogens
- noble gases
- lanthanoids
- actinoids

Motion Distance Speed

The screenshot shows a game engine interface with a periodic table as the central feature. The table includes element symbols, atomic numbers, and masses. A character from the Legend of Zelda is standing on the Gold (Au) square. Various UI elements like tabs, dropdowns, and buttons are visible around the table.

floor2		transition metals										noble gases		BORON 10.811	CARBON 12.0107	NITROGEN 14.0067	OXYGEN 15.9994						
		other metals					lanthanoids																
		metalloids					actinoids																
23	V VANADIUM 50.9415	24	Cr CHROMIUM 51.9961	25	Mn MANGANESE 54.938045	IRON 26 Fe	27	Co COBALT 58.933195	28	Ni NICKEL 58.6934	COPPER 29 Cu	30	Zn ZINC 65.39	31	Ga GALLIUM 69.723	32	Ge GERMANIUM 72.64	33	As ARSENIC 74.92160	34	Se SELENIUM 78.96		
41	Nb NIOBIUM 92.90638	42	Mo MOLYBDENUM 95.94	43	Tc TECHNETIUM 97.9072	Ru RUTHENIUM 101.07	45	Rh RHODIUM 102.90550	46	P PALLADIUM 106.4	SILVER 47 Ag	48	Cd CADMIUM 112.411	49	In INDIUM 114.818	50	Sn TIN 118.710	51	Sb ANTIMONY 121.760	52	Te MOTION TELLURIUM 127.66		
M																							
73	Ta TANTALUM 180.94788	74	W TUNGSTEN 183.84	75	Re RHENIUM 186.207	76	Os OSMIUM 190.23	77	Ir IRIDIUM 192.217	78	Pt PLATINUM 195.084	GOLD 79 Au	80	Hg MERCURY 200.59	81	Tl THALLIUM 204.3833	82	Ti LEAD 207.2	83	Pb BISMUTH 208.98040	84	Bi PO Speed 0.000 208.9824	
I																							
105	Db DUBNIUM 262	106	Sg SEABORGIUM 266	107	Bh BOHRIUM 264	108	Hs HASSIUM 277	109	Mt MEITNERIUM 268	110	Ds DARMSTADTIUM 271	111	Rg ROENTGENIUM 272	112	Cn COPERNICIUM 285	113	Nh NIHONIUM 284	114	Fl FLEROVIUM 289	115	Mc MOSCOWIUM 288	116	Ly LAWRENCE MILLER 1054 253
U																							
158	159	160	161	162	163	164	165	166	167	168	169	170	171	172	173	174	175	176	177				





Topalian Game Engine

dic Instructions by Paul Sherman for wpclipart.com Graphic is Public Domain

Dedicated to God the Father
TOPALIAN GAME ENGINE
<https://github.com/ChristopherTopalian>

How many objects 1804 142

Motion

Distance 3104

Speed 0.000

Mouse X 1953 Y 208

No Collision undefined

The screenshot shows a 3D game environment with a stone floor and walls. A character in a green tunic and blue pants stands in the center. To the left is a periodic table. On the right, there are four boxes labeled "boxCarbonTrigger", "Carbon 6 C Link", "boxNitrogenTrigger", and "Nitrogen 7 N Link". Above the boxes are game statistics: Motion (1804), Distance (3104), Speed (0.000), Mouse X (1953), and Y (208). Below the boxes is a message about no collision.

Element	Symbol	Name	Atomic Number	Atomic Mass
He	He	HELIUM	2	4.002602
F	F	FLUORINE	9	18.9984032
Ne	Ne	NEON	10	20.1797
Cl	Cl	CHLORINE	17	35.453
Ar	Ar	ARGON	18	39.948
Br	Br	BROMINE	35	79.904
Kr	Kr	KRYPTON	36	83.798
I	I	IODINE	53	126.904
Xe	Xe	XENON	54	131.300

Topalian Game Engine PLAYING

file:///D:/_1Code/0_JS_Published/Topalian_Game_Engine/Topalian_Game_Engine/Topalian_Game_Engine.html

Instructions **Riodic Table Explained** **Watch later** **Share**

Atomic Number Atomic Number

up elements

Metals Metalloids Non-metals Noble gases

2894 892

IIA

Transition elements

Motion Distance Speed

6197 0.000

No Collision floor2

Watch on YouTube

Topalian Game Engine

file:///D:/_1Code/0_JS_Published/Topalian_Game_Engine/Topalian_Game_Engine/Topalian_Game_Engine.html

TOPALIAN GAME ENGINE

Hydrogen 1 Link

Helium 2 Link

Lithium 3 Link

Beryllium 4 Link

Boron 5 Link

Carbon 6 Link

Nitrogen 7 Link

Oxygen 8 Link

Fluorine 9 Link

The Periodic Table Explained

Atomic Number Atomic Mass And Mass Number

Chemical Reactions And Equations

Faraday's First Law of Electrolysis

Motion
Distance 733
Speed 0.000
Mouse X 3446
Y 1460
No Collision

Topalian Game Engine PLAYING

file:///D:/_1Code/0_JS_Published/Topalian_Game_Engine/Topalian_Game_Engine/Topalian_Game_Engine.html

Instructions Watch later Share

Atomic Number Atomic Mass And Mass Number

Watch later Share

TutorVista.com Definitions

Atomic Number, Atomic Mass and Mass Number

Close

Neon 10 P 10 N

Lithium 3 P 3 N

Sodium 11 P 12 N

A single atom consists of a nucleus made from protons and neutrons, and an electron cloud that orbits around the nucleus.

edurite

PAUSE REPLAY BACK NEXT

Vista Chemical Reacti...sta.com

ed Chemical Equations

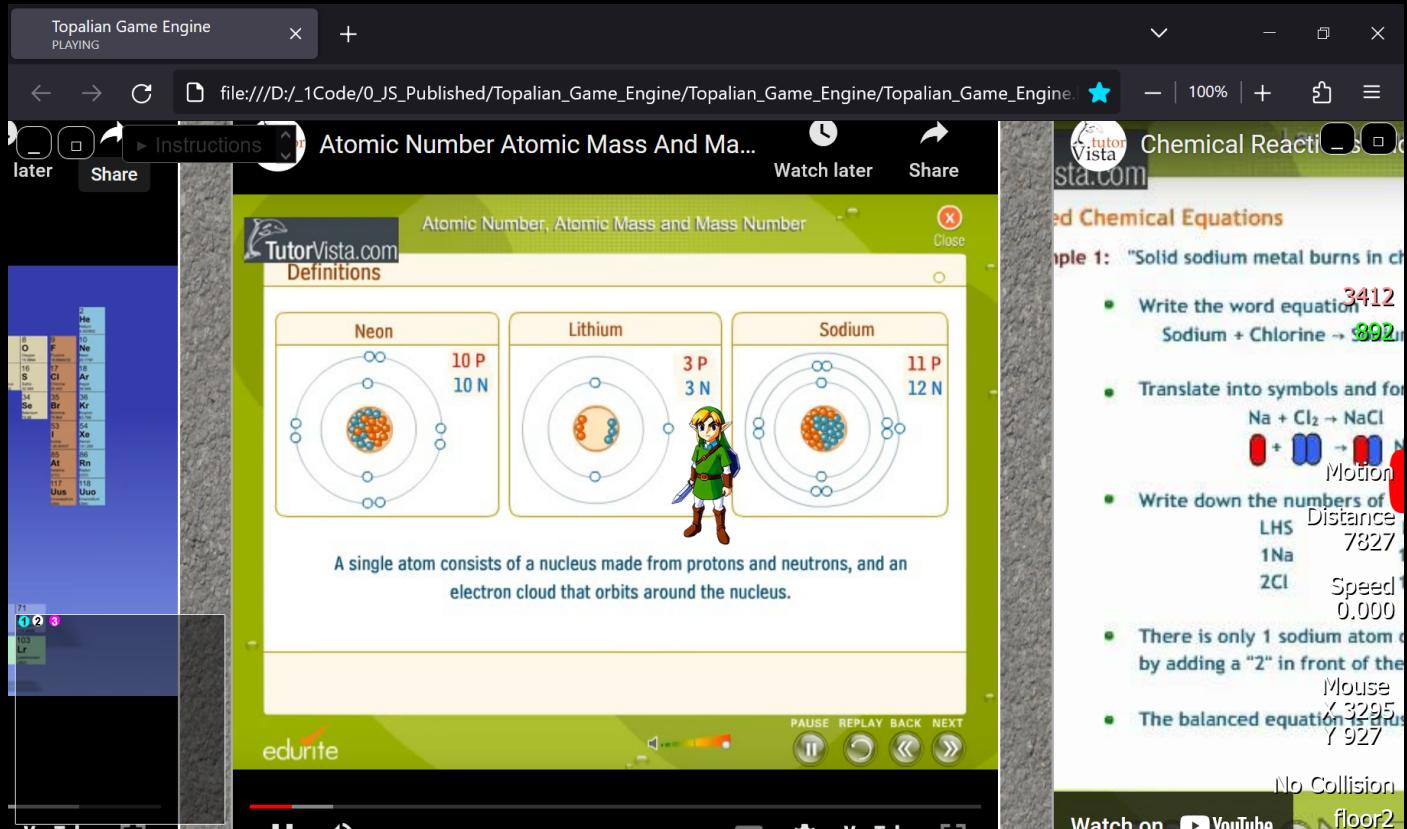
Example 1: "Solid sodium metal burns in ch...

- Write the word equation
Sodium + Chlorine → **392**
- Translate into symbols and formula
 $\text{Na} + \text{Cl}_2 \rightarrow \text{NaCl}$
Motion
- Write down the numbers of atoms
LHS Distance 7827
1Na 1Na
2Cl Speed 0.000
- There is only 1 sodium atom in the balanced equation by adding a "2" in front of the chlorine molecule
Mouse X 3295 Y 927
- The balanced equation is $2\text{Na} + \text{Cl}_2 \rightarrow 2\text{NaCl}$

No Collision

Watch on YouTube

floor2





Topalian Game Engine

Instructions

Hydrogen 1 [Link](#)
Helium 2 [Link](#)
Lithium 3 [Link](#)
Beryllium 4 [Link](#)
Boron 5 [Link](#)
Carbon 6 [Link](#)
Nitrogen 7 [Link](#)
Oxygen 8 [Link](#)
Fluorine 9 [Link](#)
Neon 10 [Link](#)
Sodium 11 [Link](#)
Magnesium 12 [Link](#)
Aluminum 13 [Link](#)
Silicon 14 [Link](#)
Phosphorus 15 [Link](#)
Sulfur 16 [Link](#)
Chlorine 17 [Link](#)
Argon 18 [Link](#)
Potassium 19 [Link](#)

600
3000

Motion

Distance 17712

Speed 0.000

Mouse X 748 Y 3136

No Collision floor2

Topalian Game Engine

Instructions

1028
5768

Motion

Distance 22283

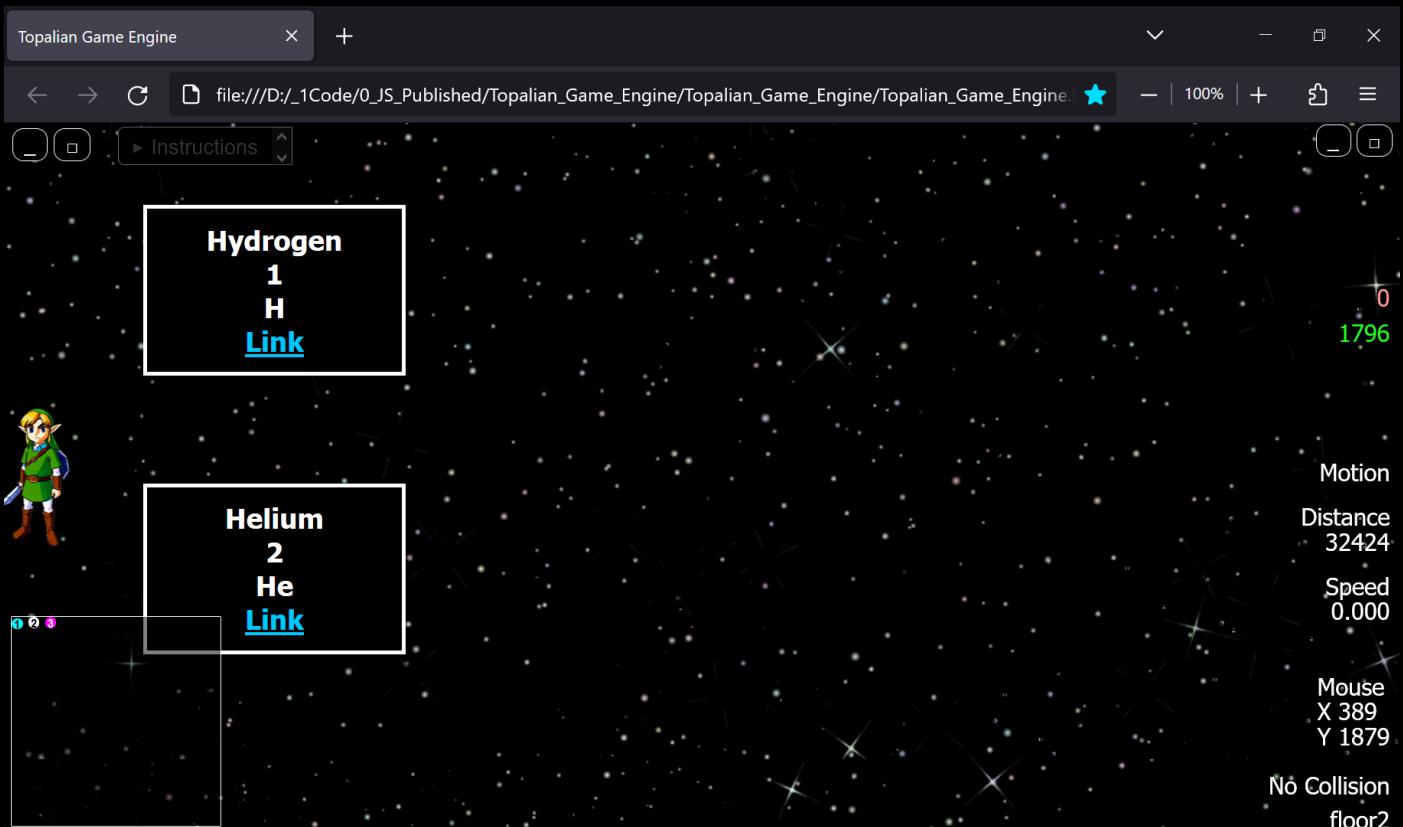
Speed 0.000

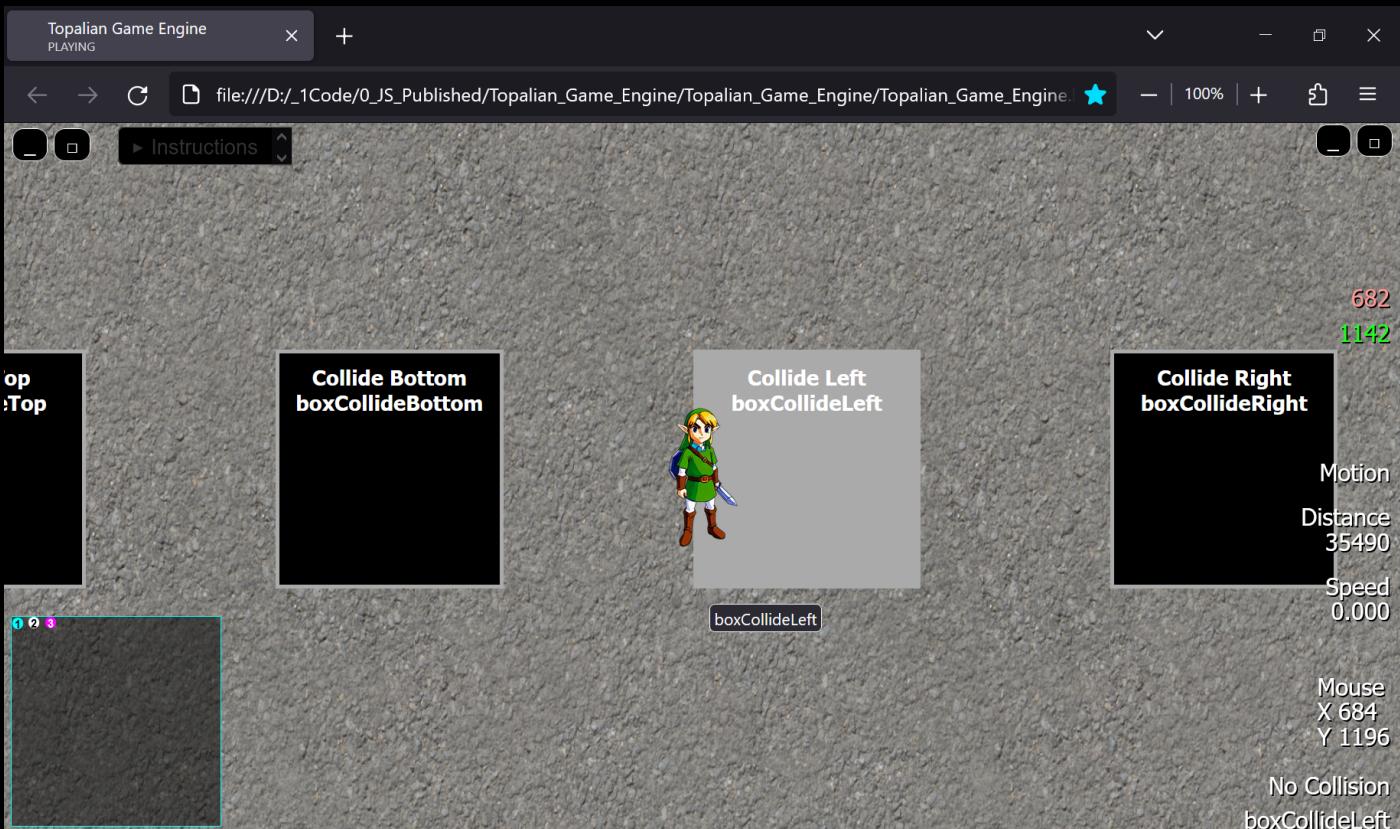
Mouse X 1219 Y 5807

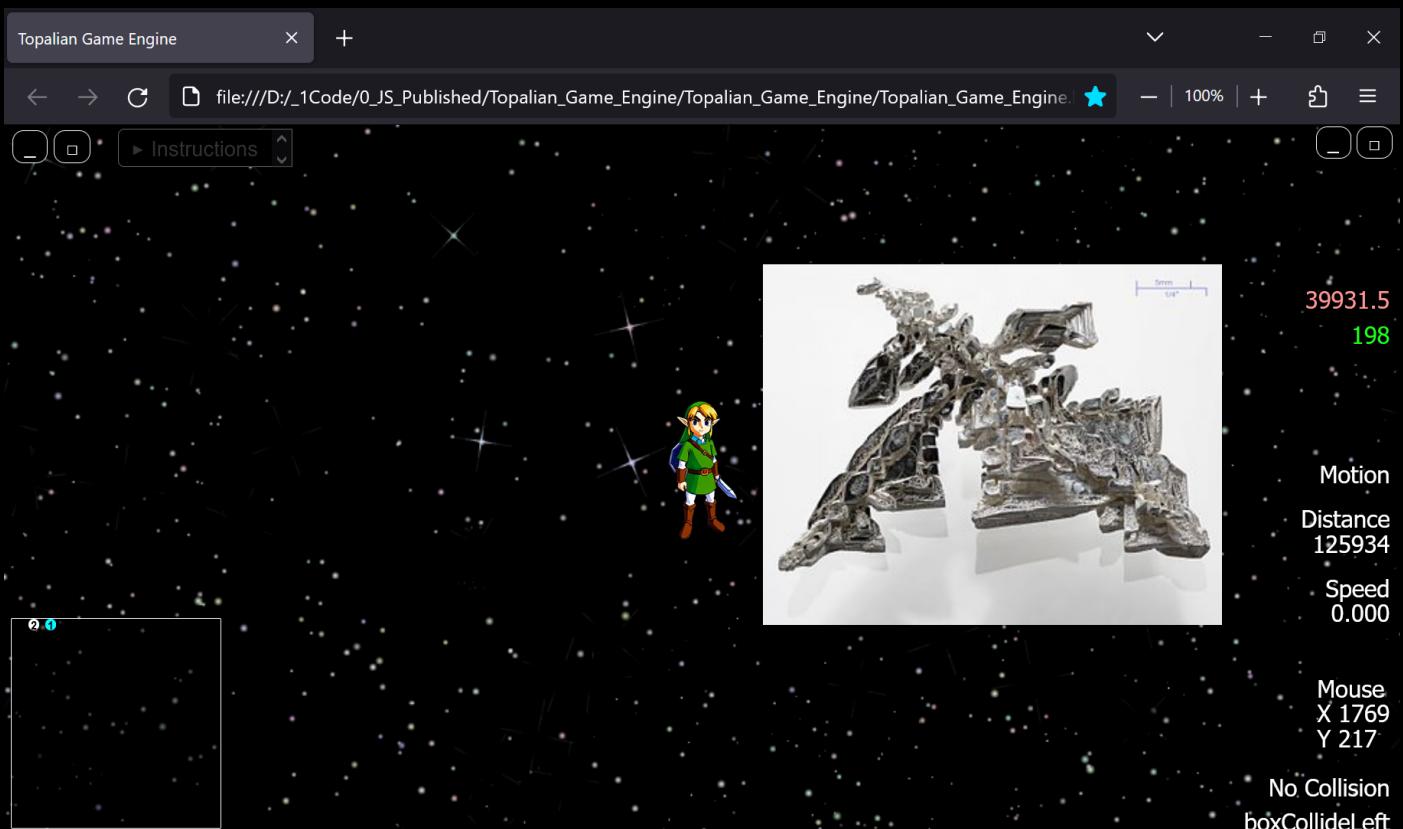
No Collision floor2

<https://en.wikipedia.org/wiki/Hydrogen>

Hydrogen 1
Helium 2
Lithium 3
Beryllium 4
Boron 5
Carbon 6
Nitrogen 7
Oxygen 8
Fluorine 9
Neon 10
Sodium 11
Magnesium 12
Aluminum 13
Silicon 14







Topalian Game Engine

file:///D:/_1Code/0_JS_Published/Topalian_Game_Engine/Topalian_Game_Engine/Topalian_Game_Engine.html

Instructions

19924.5
198

Motion

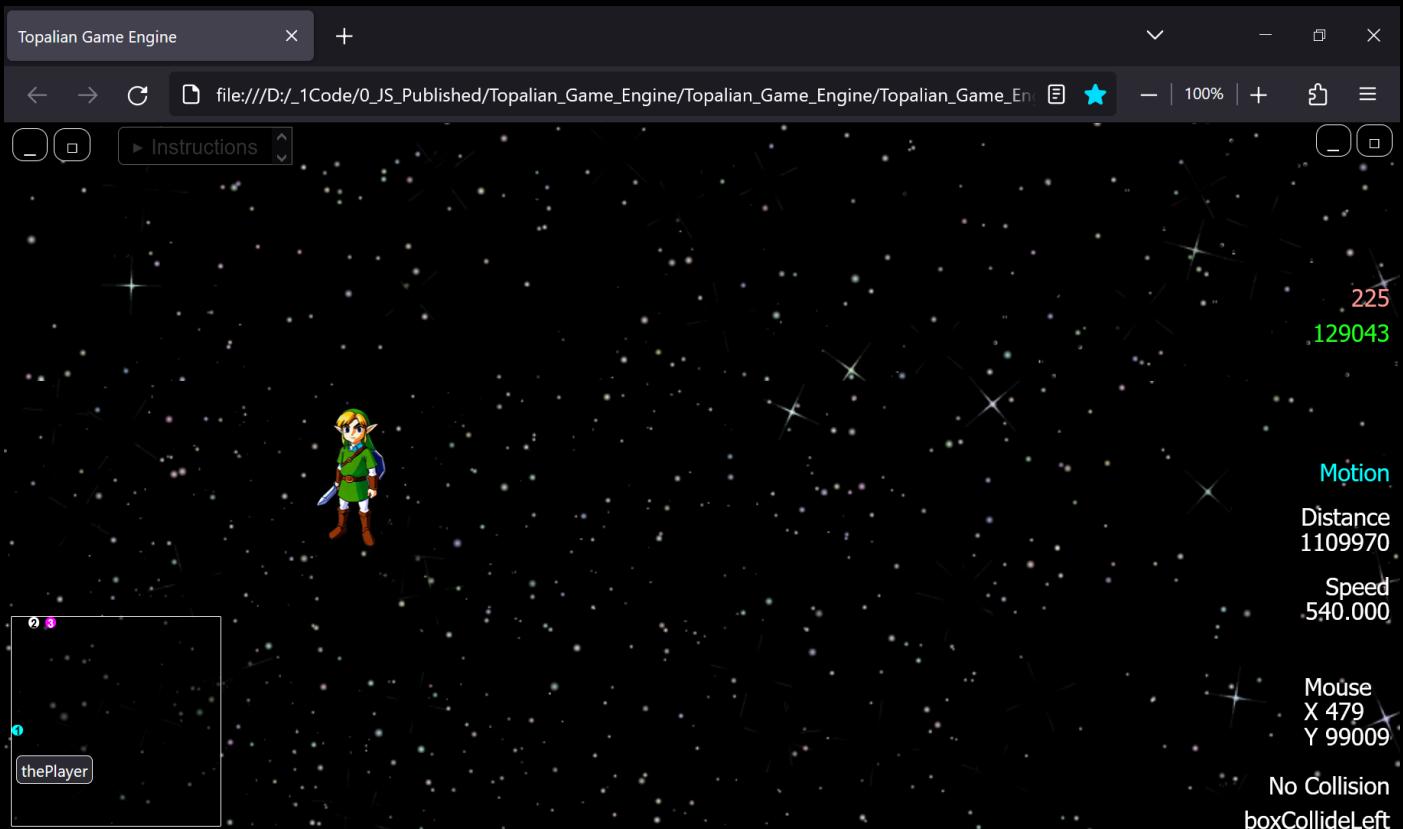
Distance 146481

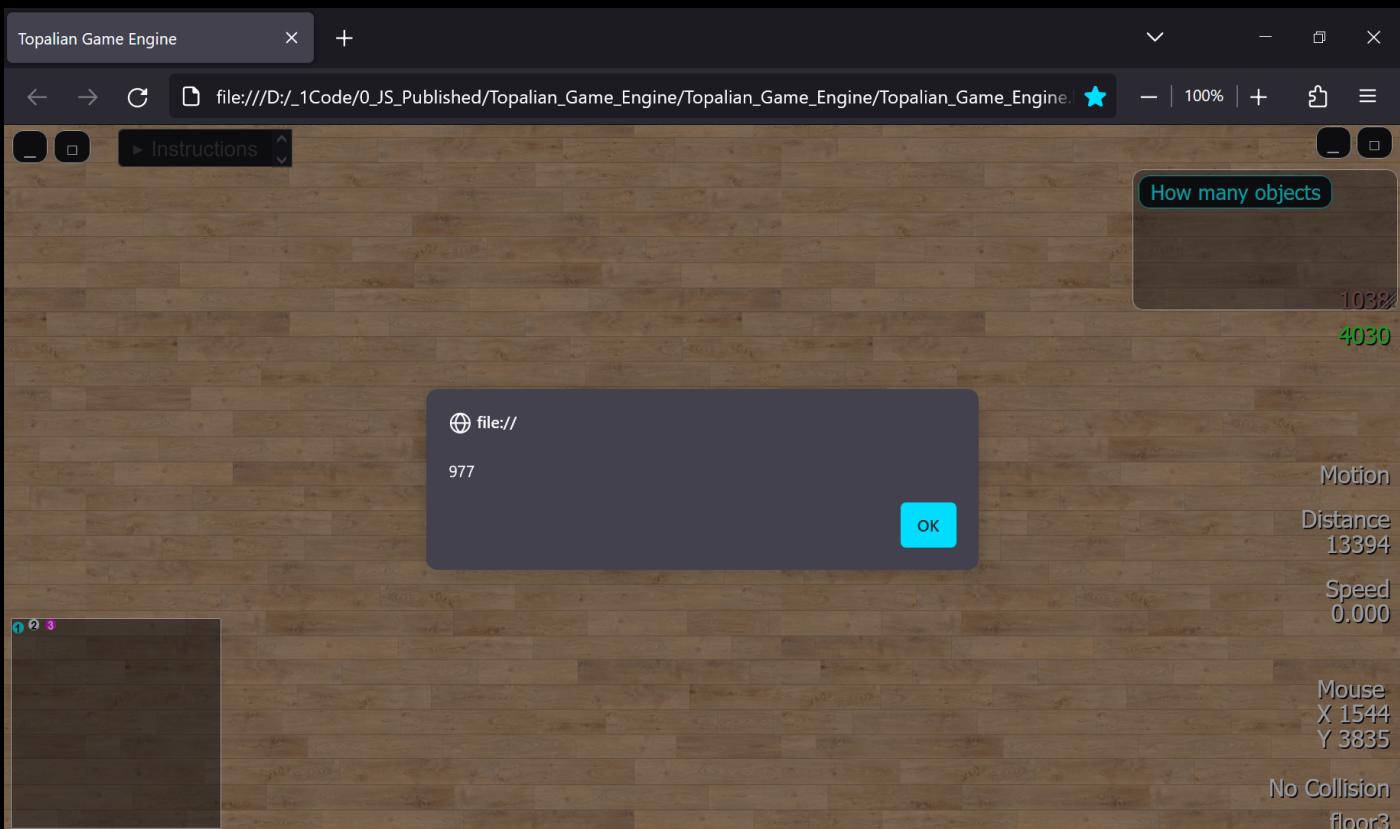
Speed 0.000

Mouse X 39926 Y 338

No Collision boxCollideLeft

The screenshot shows a game engine interface with a dark background featuring a starry sky. In the center, there is a 3D model of a Link-like character standing next to a large, reddish-brown copper nugget. A small blue rectangular box labeled "treasureCopper" is positioned near the nugget. On the left side, there is a UI panel with a grid containing the text "rium" and "2". To the right of this are three more panels: one for "Europium" (atomic number 63), one for "Gadolinium" (atomic number 64), and one for "Terbium" (atomic number 65). Each panel contains a "Link" button. On the far right, there is a vertical column of status information: "19924.5", "198", "Motion", "Distance 146481", "Speed 0.000", "Mouse X 39926 Y 338", and "No Collision boxCollideLeft".





Topalian Game Engine

Instructions
by Paul Sherman
for wpclipart.com
Graphic is Public Domain

Dedicated to God the Father
TOPALIAN GAME ENGINE

<https://github.com/ChristopherTopalian>

1824 67

Motion

Distance 42509

Speed 0.000

Mouse X 1837 Y 272

No Collision boxCollideLeft

Element	Symbol	Name	Atomic Number	Atomic Mass
He	He	HELIUM	2	4.002602
F	F	FLUORINE	9	18.9984032
Ne	Ne	NEON	10	20.1797
Cl	Cl	CHLORINE	17	35.453
Ar	Ar	ARGON	18	39.948
Br	Br	BROMINE	35	79.904
Kr	Kr	KRYPTON	36	83.798
I	I	IODINE	53	126.904
Xe	Xe	XENON	54	131.30

boxCarb
onTrig
ger

Carbon
6
C
[Link](#)

boxNitr
ogenTri
gger

Nitrogen
7
N
[Link](#)



_instructions



css



js

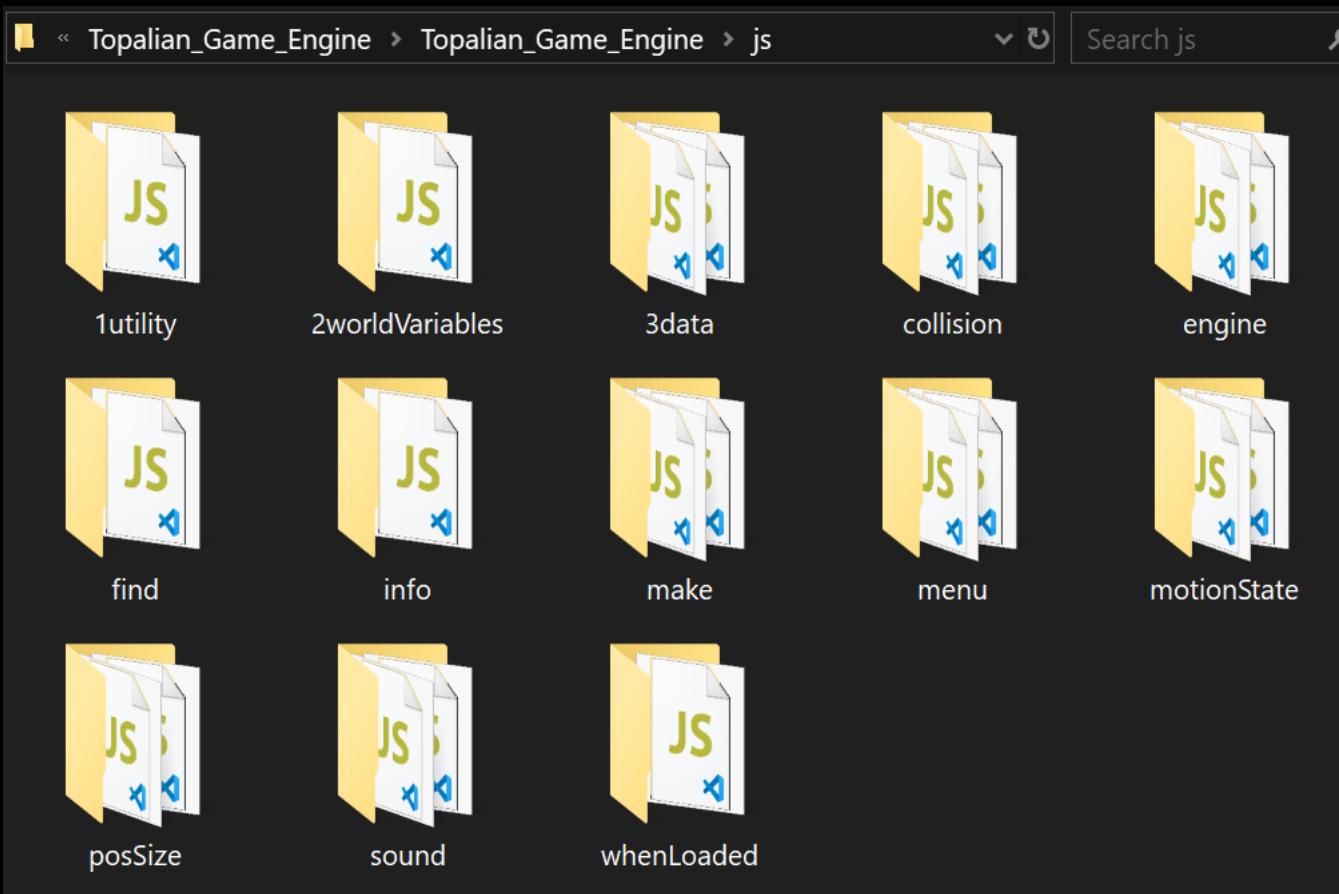


media



Topalian_Game_Engine.htm

|



// Dedicated to God the Father

// All Rights Reserved Christopher Andrew
Topalian Copyright 2000-2024

// <https://github.com/ChristopherTopalian>

//
<https://github.com/ChristopherAndrewTopalia>
n

// HowToCombineJSFiles.js

TUTORIAL:
**How to Combine all .js files in all folders that
are in our js folder.**

Getting things ready:

We should add two new lines at the end every script. This way they will combine nicely with a line break in between each script.

Step One: Open our js folder

Step Two: Type in the address bar of the js folder, cmd, press Enter

This opens our js folder in the command prompt

Step Three: Type the command shown below in the command prompt and then press Enter

for /r "%CD%" %i in (*.js) do type "%i" >> main.js

Now we have a newly created .js file named main.js that has all of our js files included into one file.

This makes it easy to upload our application and easy to find out how many lines of code our project is.

To use our main.js file, we include it in our html file code:

```
<script src = 'js/main.js'></script>
```

Happy Scripting :-)

<!-- Dedicated to God the Father -->

<!-- All Rights Reserved Christopher Andrew
Topalian Copyright 2000-2024 -->

<!-- <https://github.com/ChristopherTopalian> -->

<!--
[https://github.com/ChristopherAndrewTopalia
n](https://github.com/ChristopherAndrewTopalia
n) -->

<!--
[https://github.com/ChristopherTopalian/Topal
ian-Game-Engine](https://github.com/ChristopherTopalian/Topal
ian-Game-Engine) -->

<!--

https://github.com/ChristopherTopalian/Topalian_Game_Engine-->

<!-- Topalian_Game_Engine.html -->

<!-- Version 019 - (2024-02-15) -->

<html>

<head>

<title> Topalian Game Engine </title>

<link rel = 'stylesheet' href =
'css/style001.css'>

<!-- UTILITY -->

<script src = 'js/1utility/utility.js'></script>

<!-- WORLD VARIABLES -->

<script src = 'js/2worldVariables
/worldVariables.js'></script>

<!-- DATA -->

<script src = 'js/3data/dataFloors.js'></script>

<script src =
'js/3data/dataObjectsPeriodicTable.js'></scrip
t>

<script src =
'js/3data/dataVideosChemistry.js'></script>

```
<script src =  
'js/3data/dataPeriodicElements.js'></script>
```

```
<script src =  
'js/3data/dataObjectsCollisionSides.js'></scri  
pt>
```

```
<script src =  
'js/3data/dataObjectsPeriodicElements2.js'></  
script>
```

```
<script src =  
'js/3data/dataObjectsPeriodicElements3.js'></  
script>
```

```
<!-- MAKE -->
```

```
<script src =  
'js/make/makeBackground.js'></script>
```

```
<script src =  
'js/make/makePlayer.js'></script>
```

```
<script src =  
'js/make/makePeriodicTable.js'></script>
```

```
<script src =  
'js/make/makeCollisionDiv.js'></script>
```

```
<!-- MINIMAP -->
```

```
<script src =  
'js/make/minimap/makeMinimap.js'></script>
```

```
<script src =  
'js/make/minimap/makeMinimapPlayer.js'></s  
cript>
```

```
<script src =  
'js/make/minimap/updateMinimapPlayer.js'></  
script>
```

```
<script src =  
'js/make/minimap/makeMinimapTreasure.js'>  

```

```
<script src =  
'js/make/minimap/updateMinimapTreasure.js'  
></script>
```

<!-- MAKE -->

```
<script src =  
'js/make/makeTreasure.js'></script>
```

```
<script src =  
'js/make/makeMousePosDiv.js'></script>
```

```
<script src =  
'js/make/makeXPosDiv.js'></script>
```

```
<script src =  
'js/make/makeYPosDiv.js'></script>
```

```
<script src =  
'js/make/makeDedication.js'></script>
```

```
<!-- INSTRUCTIONS -->
```

```
<script src =  
'js/make/instructions/makeInstructions.js'></  
script>
```

```
<script src =  
'js/make/instructions/showInstructions.js'></  
script>
```

```
<script src =  
'js/make/instructions/hideInstructions.js'></s  
cript>
```

```
<!-- MAKE -->
```

```
<script src =  
'js/make/makeInfoDiv.js'></script>
```

```
<script src =  
'js/make/makeIsMovingDiv.js'></script>
```

```
<script src =  
'js/make/makeDistanceTraveledDiv.js'></script  
t>
```

```
<script src =  
'js/make/makeElementInfoBox.js'></script>
```

```
<script src =  
'js/make/speedometer/makeSpeedometer.js'  
</script>
```

```
<script src =  
'js/make/speedometer/updateSpeedometer.js'  
></script>
```

```
<script src =  
'js/make/makeTilesOneRow.js'></script>
```

```
<script src = 'js/make/makeTiles.js'></script>
```

```
<script src =  
'js/make/makePeriodicElement.js'></script>
```

```
<script src =  
'js/make/makeOneElement.js'></script>
```

```
<script src =  
'js/make/makePeriodicElements.js'></script>
```

```
<script src =  
'js/make/makePeriodicElementsNameAsLink.j  
s'></script>
```

```
<script src =  
'js/make/makeRowOfElements.js'></script>
```

```
<script src =  
'js/make/makeObjectsToScreen.js'></script>
```

```
<script src =  
'js/make/makeRowOfWebsitesPdfsOrVideos.j  
s'></script>
```

<!-- MENU -->

```
<script src = 'js/menu/menuLeft.js'></script>
```

```
<script src =  
'js/menu/menuRight.js'></script>
```

<!-- POS-SIZE -->

<script src = 'js/posSize/getPos.js'></script>

<script src = 'js/posSize/getSize.js'></script>

<script src =
'js/posSize/mousePos.js'></script>

<!-- FIND -->

<script src =
'js/find/findIndexByName.js'></script>

<!-- INFO -->

<script src =
'js/info/countAllElements.js'></script>

<!-- COLLISION -->

<script src =
'js/collision/collision.js'></script>

<script src =
'js/collision/collidedWithClass.js'></script>

<script src =
'js/collision/collisionTopSide.js'></script>

<script src =
'js/collision/collisionBottomSide.js'></script>

<script src =
'js/collision/collisionLeftSide.js'></script>

```
<script src =  
'js/collision/collisionRightSide.js'></script>
```

```
<script src =  
'js/collision/wasObjectLineCrossed.js'></scri  
pt>
```

```
<script src =  
'js/collision/wasLineCrossed.js'></script>
```

```
<!-- ENGINE -->
```

```
<script src =  
'js/engine/keepPlayerInWorld.js'></script>
```

```
<script src =  
'js/engine/cameraFollowsPlayer.js'></script>
```

```
<script src =  
'js/engine/movePlayer.js'></script>
```

```
<script src =  
'js/engine/controlsKeyboard.js'></script>
```

```
<script src =  
'js/engine/playerMotion.js'></script>
```

```
<script src = 'js/engine/gravity.js'></script>
```

```
<script src =  
'js/engine/gameLoop.js'></script>
```

<!-- MOTION STATE -->

```
<script src =  
'js/motionState/checkPlayerMovement.js'></s  
cript>
```

```
<script src =  
'js/motionState/isPlayerMoving.js'></script>
```

```
<script src =  
'js/motionState/calculateDistanceTraveled.js'  
></script>
```

```
<!-- SOUND -->
```

```
<script src =  
'js/sound/loadSounds.js'></script>
```

```
<script src =  
'js/sound/loadSongs.js'></script>
```

```
<script src = 'js/sound/audioPlay.js'></script>
```

```
<script src =
'js/sound/audioPause.js'></script>
```

```
<!-- WHEN LOADED -->
```

```
<script src =
'js/whenLoaded/whenLoaded.js'></script>
```

```
</head>
```

```
<body onload = 'whenLoaded();'>
```

```
</body>
```

```
</html>
```


/* Dedicated to God the Father */

**/* All Rights Reserved Christopher Andrew
Topalian Copyright 2000-2024 */**

/* https://github.com/ChristopherTopalian */

**/*
https://github.com/ChristopherAndrewTopalia
n */**

/* style001.css */

body
{
height: 100%;
overflow: hidden;
color: rgb(255, 255, 255);

}

```
a:link  
{  
    color: rgb(0, 200, 255);  
}
```

```
a:visited  
{  
    color: rgb(200, 200, 200);  
}
```

```
audio  
{  
    display: none;  
}
```

video

```
{  
    width: 500px;  
    border: solid;  
    border-width: 1px;  
    border-color: rgb(255, 255, 255);  
}
```

.instructions

```
{  
    margin: 2px;  
    padding-left: 10px;  
    padding-right: 10px;  
    padding-top: 4px;  
    padding-bottom: 4px;  
    border-style: solid;  
    border-width: 0.5px;  
    border-color: rgb(100, 100, 100);  
    border-radius: 4px;
```

```
background: rgb(0, 0, 0);
z-index: 998;
font-size: 15px;
color: rgb(255, 255, 255);
word-wrap: break-word;
overflow-y: scroll;
}
```

```
.textStyle001
{
    padding-left: 4px;
    padding-right: 8px;
    padding-top: 0px;
    padding-bottom: 0px;
    font-family: tahoma;
    font-size: 17px;
    color: rgb(255, 255, 255);
    line-height: 18px;
}
```

```
word-wrap: break-word;  
}
```

```
.textStyle002  
{  
padding-left: 4px;  
padding-right: 8px;  
padding-top: 0px;  
padding-bottom: 8px;  
font-family: arial;  
font-size: 18px;  
color: rgb(255, 255, 255);  
line-height: 18px;  
word-wrap: break-word;  
}
```

```
.tile  
{
```

```
font-family: tahoma;  
font-size: 15px;  
color: rgb(0, 0, 0);  
word-wrap: break-word;  
}
```

.theDedication

```
{  
position: absolute;  
left: 1580px;  
top: 40px;  
width: 500px;  
height: 400px;  
z-index: 5;  
font-family: arial;  
font-size: 28px;  
font-weight: bold;  
color: rgb(255, 255, 255);
```

```
text-align: center;  
}
```

```
#periodicTable  
{  
    position: absolute;  
    left: 0px;  
    top: 0px;  
    width: 1535;  
    height: 912px;  
    z-index: 3;  
    background-image:  
        url("media/textures/periodicTable.png");  
    background-size: cover;  
}
```

```
.elementDiv  
{
```

```
position: absolute;  
left: 190px;  
top: 170px;  
width: 400px;  
height: 150px;  
padding-left: 10px;  
padding-right: 10px;  
padding-top: 10px;  
padding-bottom: 10px;  
margin-bottom: 0px;  
z-index: 4;  
background: rgb(0, 0, 0);  
font-family: tahoma;  
font-size: 30px;  
font-weight: bold;  
color: rgb(255, 255, 255);  
text-align: center;  
}
```

```
.thelimage
{
    height: 80%;
    z-index: 10;
    vertical-align: middle;
}
```

```
.boxStyle001
{
    padding-left: 4px;
    padding-right: 4px;
    padding-top: 8px;
    padding-bottom: 8px;
    z-index: 4;
    border-style: solid;
    border-width: 3px;
    border-color: rgb(0, 0, 0);
```

```
background: rgb(171, 171, 171);  
font-family: tahoma;  
font-size: 15px;  
font-weight: bold;  
color: rgb(255, 255, 255);  
text-align: center;  
word-wrap: break-word;  
}
```

```
.boxStyle002  
{  
padding-left: 4px;  
padding-right: 4px;  
padding-top: 8px;  
padding-bottom: 8px;  
z-index: 4;  
border-style: solid;  
border-width: 3px;
```

```
border-color: rgb(170, 170, 170);  
background: rgb(0, 0, 0);  
font-family: tahoma;  
font-size: 15px;  
font-weight: bold;  
color: rgb(255, 255, 255);  
text-align: center;  
word-wrap: break-word;  
}
```

```
.buttonStyle001  
{  
margin: 4px;  
padding-left: 8px;  
padding-right: 8px;  
padding-top: 2px;  
padding-bottom: 2px;  
border-style: solid;
```

```
border-width: 1px;  
border-radius: 8px;  
border-color: rgb(255, 255, 255);  
background-color: rgb(0, 0, 0);  
font-size: 15px;  
color: rgb(255, 255, 255);  
}  
  
.buttonStyle001:hover  
{  
    border-color: rgb(0, 255, 255);  
}  
  
.buttonStyle001:active  
{  
    color: rgb(0, 255, 255);  
}
```

// Dedicated to God the Father

// All Rights Reserved Christopher Andrew
Topalian Copyright 2000-2024

// <https://github.com/ChristopherTopalian>

//
<https://github.com/ChristopherAndrewTopalian>

// utility.js

```
function ge(whichId)
{
    let result =
        document.getElementById(whichId);
```

```
return result;  
}  
  
function ce(whichType)  
{  
    let result =  
document.createElement(whichType);  
  
    return result;  
}  
  
function ba(whichElement)  
{  
    let result =  
document.body.append(whichElement);  
  
    return result;  
}
```

// Dedicated to God the Father

// All Rights Reserved Christopher Andrew
Topalian Copyright 2000-2024

// <https://github.com/ChristopherTopalian>

//
<https://github.com/ChristopherAndrewTopalia>
n

// worldVariables.js

```
let ourPlayer =  
{  
  x: 100,  
  y: 200,  
  width: 50,
```

```
height: 100,  
speedMultiplier: 2,  
id: 'thePlayer',  
playerColor: 'rgb(73, 100, 150)',  
texture:  
'url("media/textures/player001.png")',  
texturePos: '0% 0%',  
velocityY: 0,  
};
```

//-//

```
let backgroundSizeX = 250000;
```

```
let backgroundSizeY = 250000;
```

//-//

```
let collisionColor001 = "rgb(0, 255, 255)";
```

```
let collisionColor002 = "rgb(170, 170, 170)";
```

```
// initial X position
```

```
let playerPrevX = 0;
```

```
// initial Y position
```

```
let playerPrevY = 0;
```

```
// distance traveled by player
```

```
let distanceTraveled = 0;
```

```
// initial player speed
```

```
let playerSpeed = 0;
```

// Dedicated to God the Father

// All Rights Reserved Christopher Andrew
Topalian Copyright 2000-2024

// <https://github.com/ChristopherTopalian>

//
<https://github.com/ChristopherAndrewTopalia>
n

// dataFloors.js

let dataFloors =

[

{

name: "floor1",
type: "div",

```
positionType: "absolute",
x: 100,
y: 100,
width: 500,
height: 50,
text: "floor1",
id: "floor1",
styleName: "boxStyle001"
},
{
name: "floor2",
type: "div",
positionType: "absolute",
x: 190,
y: 310,
width: 400,
height: 20,
```

```
text: "floor2",
id: "floor2",
styleName: "boxStyle001"
},
{
  name: "floor3",
  type: "div",
  positionType: "absolute",
  x: 1015,
  y: 135,
  width: 400,
  height: 20,
  text: "floor3",
  id: "floor3",
  bstyleName: "boxStyle001"
}
];
```

// Dedicated to God the Father

// All Rights Reserved Christopher Andrew
Topalian Copyright 2000-2024

// <https://github.com/ChristopherTopalian>

//
<https://github.com/ChristopherAndrewTopalia>
n

// dataObjectsCollisionSides.js

```
let objectsPart3 =  
[  
  {  
    name: "boxCollideTop",  
    type: "div",
```

```
positionType: "absolute",
x: 100,
y: 1100,
width: 150,
height: 150,
text: "Collide Top boxCollideTop",
id: "boxCollideTop",
styleName: "boxStyle002"
},
{
name: "boxCollideBottom",
type: "div",
positionType: "absolute",
x: 400,
y: 1100,
width: 150,
height: 150,
```

```
    text: "Collide Bottom  
boxCollideBottom",  
    id: "boxCollideBottom",  
    styleName: "boxStyle002"  
},  
  
{  
    name: "boxCollideLeft",  
    type: "div",  
    positionType: "absolute",  
    x: 700,  
    y: 1100,  
    width: 150,  
    height: 150,  
    text: "Collide Left boxCollideLeft",  
    id: "boxCollideLeft",  
    styleName: "boxStyle002"  
},
```

```
{  
    name: "boxCollideRight",  
    type: "div",  
    positionType: "absolute",  
    x: 1000,  
    y: 1100,  
    width: 150,  
    height: 150,  
    text: "Collide Right boxCollideRight",  
    id: "boxCollideRight",  
    styleName: "boxStyle002"  
}  
];
```

// Dedicated to God the Father

// All Rights Reserved Christopher Andrew
Topalian Copyright 2000-2024

// <https://github.com/ChristopherTopalian>

//
<https://github.com/ChristopherAndrewTopalia>
n

// dataObjectsPeriodicElements2.js

```
let objectsPart2 =  
[  
 {  
   name: "boxCarbonTrigger",  
   type: "div",
```

```
positionType: "absolute",
x: 1650,
y: 400,
width: 65,
height: 60,
text: "boxCarbonTrigger",
id: "boxCarbonTrigger",
styleName: "boxStyle002"
},
```

```
{
  name: "boxNitrogenTrigger",
  type: "div",
  positionType: "absolute",
  x: 1965,
  y: 400,
  width: 65,
  height: 60,
```

```
text: "boxNitrogenTrigger",
id: "boxNitrogenTrigger",
styleName: "boxStyle002"
}
];
```

// Dedicated to God the Father

// All Rights Reserved Christopher Andrew
Topalian Copyright 2000-2024

// <https://github.com/ChristopherTopalian>

//
<https://github.com/ChristopherAndrewTopalia>
n

// dataObjectsPeriodicElements3.js

```
let objectsTriggers =  
[  
  {  
    name: "boxSilverTrigger",  
    type: "div",
```

```
positionType: "absolute",
x: 186,
y: 2280,
width: 65,
height: 60,
text: "boxSilverTrigger",
id: "boxSilverTrigger",
styleName: "boxStyle002"
},
{
name: "boxGoldTrigger",
type: "div",
positionType: "absolute",
x: 186,
y: 2500,
width: 65,
height: 60,
```

```
text: "boxGoldTrigger",
id: "boxGoldTrigger",
styleName: "boxStyle002"
}
];
```

// Dedicated to God the Father

// All Rights Reserved Christopher Andrew
Topalian Copyright 2000-2024

// <https://github.com/ChristopherTopalian>

//
<https://github.com/ChristopherAndrewTopalia>
n

// dataObjectsPeriodicTable.js

```
let dataObjectsPeriodicTable =  
[  
 {  
   name: "boxIron",  
   type: "div",
```

```
positionType: "absolute",
x: 603,
y: 353,
width: 65,
height: 60,

text: "IRON <br> 26 <br> Fe",
id: "boxIron",
styleName: "boxStyle001"

},

{
name: "boxCopper",
type: "div",
positionType: "absolute",
x: 855,
y: 354,
width: 65,
```

```
height: 60,  
text: "COPPER <br> 29 <br> Cu",  
id: "boxCopper",  
styleName: "boxStyle001"  
},  
  
{  
name: "boxSilver",  
type: "div",  
positionType: "absolute",  
x: 855,  
y: 437,  
width: 65,  
height: 60,  
text: "SILVER <br> 47 <br> Ag",  
id: "boxSilver",  
styleName: "boxStyle001"  
},
```

```
{  
    name: "boxGold",  
    type: "div",  
    positionType: "absolute",  
    x: 855,  
    y: 522,  
    width: 65,  
    height: 60,  
    text: "GOLD <br> 79 <br> Au",  
    id: "boxGold",  
    styleName: "boxStyle001"  
}  
];
```

// Dedicated to God the Father

// All Rights Reserved Christopher Andrew
Topalian Copyright 2000-2024

// <https://github.com/ChristopherTopalian>

//
<https://github.com/ChristopherAndrewTopalia>
n

// dataPeriodicElements.js

```
let dataPeriodicElements = [  
{  
  name: "Hydrogen",  
  number: "1",  
  abbreviation: "H",
```

```
url: "https://en.wikipedia.org/wiki/Hydrogen",
tag: "chemicalElement"
},
{
name: "Helium",
number: "2",
abbreviation: "He",
url: "https://en.wikipedia.org/wiki/Helium",
tag: "chemicalElement"
},
{
name: "Lithium",
number: "3",
abbreviation: "Li",
url: "https://en.wikipedia.org/wiki/Lithium",
tag: "chemicalElement"
},
{
```

```
name: "Beryllium",
number: "4",
abbreviation: "Be",
url: "https://en.wikipedia.org/wiki/Beryllium",
tag: "chemicalElement"
},
{
name: "Boron",
number: "5",
abbreviation: "B",
url: "https://en.wikipedia.org/wiki/Boron",
tag: "chemicalElement"
},
{
name: "Carbon",
number: "6",
abbreviation: "C",
url: "https://en.wikipedia.org/wiki/Carbon",
```

```
tag: "chemicalElement"
},
{
  name: "Nitrogen",
  number: "7",
  abbreviation: "N",
  url: "https://en.wikipedia.org/wiki/Nitrogen",
  tag: "chemicalElement"
},
{
  name: "Oxygen",
  number: "8",
  abbreviation: "O",
  url: "https://en.wikipedia.org/wiki/Oxygen",
  tag: "chemicalElement"
},
{
  name: "Fluorine",
```

```
number: "9",
abbreviation: "F",
url: "https://en.wikipedia.org/wiki/Fluorine",
tag: "chemicalElement"
},
{
name: "Neon",
number: "10",
abbreviation: "Ne",
url: "https://en.wikipedia.org/wiki/Neon",
tag: "chemicalElement"
},
{
name: "Sodium",
number: "11",
abbreviation: "Na",
url: "https://en.wikipedia.org/wiki/Sodium",
tag: "chemicalElement"
```

```
},
{
  name: "Magnesium",
  number: "12",
  abbreviation: "Mg",
  url:
  "https://en.wikipedia.org/wiki/Magnesium",
  tag: "chemicalElement"
},
{
  name: "Aluminum",
  number: "13",
  abbreviation: "Al",
  url: "https://en.wikipedia.org/wiki/Aluminum",
  tag: "chemicalElement"
},
{
  name: "Silicon",
```

```
number: "14",
abbreviation: "Si",
url: "https://en.wikipedia.org/wiki/Silicon",
tag: "chemicalElement"
},
{
name: "Phosphorus",
number: "15",
abbreviation: "P",
url:
"https://en.wikipedia.org/wiki/Phosphorus",
tag: "chemicalElement"
},
{
name: "Sulfur",
number: "16",
abbreviation: "S",
url: "https://en.wikipedia.org/wiki/Sulfur",
```

```
tag: "chemicalElement"
},
{
  name: "Chlorine",
  number: "17",
  abbreviation: "Cl",
  url: "https://en.wikipedia.org/wiki/Chlorine",
  tag: "chemicalElement"
},
{
  name: "Argon",
  number: "18",
  abbreviation: "Ar",
  url: "https://en.wikipedia.org/wiki/Argon",
  tag: "chemicalElement"
},
{
  name: "Potassium",
```

```
number: "19",
abbreviation: "K",
url: "https://en.wikipedia.org/wiki/Potassium",
tag: "chemicalElement"
},
{
name: "Calcium",
number: "20",
abbreviation: "Ca",
url: "https://en.wikipedia.org/wiki/Calcium",
tag: "chemicalElement"
},
{
name: "Scandium",
number: "21",
abbreviation: "Sc",
url: "https://en.wikipedia.org/wiki/Scandium",
tag: "chemicalElement"
```

```
},
{
  name: "Titanium",
  number: "22",
  abbreviation: "Ti",
  url: "https://en.wikipedia.org/wiki/Titanium",
  tag: "chemicalElement"
},
{
  name: "Vanadium",
  number: "23",
  abbreviation: "V",
  url: "https://en.wikipedia.org/wiki/Vanadium",
  tag: "chemicalElement"
},
{
  name: "Chromium",
  number: "24",
```

```
abbreviation: "Cr",
url: "https://en.wikipedia.org/wiki/Chromium",
tag: "chemicalElement"
},
{
name: "Manganese",
number: "25",
abbreviation: "Mn",
url:
"https://en.wikipedia.org/wiki/Manganese",
tag: "chemicalElement"
},
{
name: "Iron",
number: "26",
abbreviation: "Fe",
url: "https://en.wikipedia.org/wiki/Iron",
tag: "chemicalElement"
```

```
},
{
  name: "Cobalt",
  number: "27",
  abbreviation: "Co",
  url: "https://en.wikipedia.org/wiki/Cobalt",
  tag: "chemicalElement"
},
{
  name: "Nickel",
  number: "28",
  abbreviation: "Ni",
  url: "https://en.wikipedia.org/wiki/Nickel",
  tag: "chemicalElement"
},
{
  name: "Copper",
  number: "29",
```

```
abbreviation: "Cu",
url: "https://en.wikipedia.org/wiki/Copper",
tag: "chemicalElement"
},
{
name: "Zinc",
number: "30",
abbreviation: "Zn",
url: "https://en.wikipedia.org/wiki/Zinc",
tag: "chemicalElement"
},
{
name: "Gallium",
number: "31",
abbreviation: "Ga",
url: "https://en.wikipedia.org/wiki/Gallium",
tag: "chemicalElement"
},
```

```
{  
  name: "Germanium",  
  number: "32",  
  abbreviation: "Ge",  
  url:  
    "https://en.wikipedia.org/wiki/Germanium",  
  tag: "chemicalElement"  
},  
{  
  name: "Arsenic",  
  number: "33",  
  abbreviation: "As",  
  url: "https://en.wikipedia.org/wiki/Arsenic",  
  tag: "chemicalElement"  
},  
{  
  name: "Selenium",  
  number: "34",  
}
```

```
abbreviation: "Se",
url: "https://en.wikipedia.org/wiki/Selenium",
tag: "chemicalElement"
},
{
name: "Bromine",
number: "35",
abbreviation: "Br",
url: "https://en.wikipedia.org/wiki/Bromine",
tag: "chemicalElement"
},
{
name: "Krypton",
number: "36",
abbreviation: "Kr",
url: "https://en.wikipedia.org/wiki/Krypton",
tag: "chemicalElement"
},
```

```
{  
  name: "Rubidium",  
  number: "37",  
  abbreviation: "Rb",  
  url: "https://en.wikipedia.org/wiki/Rubidium",  
  tag: "chemicalElement"  
},  
{  
  name: "Strontium",  
  number: "38",  
  abbreviation: "Sr",  
  url: "https://en.wikipedia.org/wiki/Strontium",  
  tag: "chemicalElement"  
},  
{  
  name: "Yttrium",  
  number: "39",  
  abbreviation: "Y",  
}
```

```
url: "https://en.wikipedia.org/wiki/Yttrium",
tag: "chemicalElement"
},
{
name: "Zirconium",
number: "40",
abbreviation: "Zr",
url: "https://en.wikipedia.org/wiki/Zirconium",
tag: "chemicalElement"
},
{
name: "Niobium",
number: "41",
abbreviation: "Nb",
url: "https://en.wikipedia.org/wiki/Niobium",
tag: "chemicalElement"
},
{
```

```
name: "Molybdenum",
number: "42",
abbreviation: "Mo",
url:
"https://en.wikipedia.org/wiki/Molybdenum",
tag: "chemicalElement"
},
{
name: "Technetium",
number: "43",
abbreviation: "Tc",
url:
"https://en.wikipedia.org/wiki/Technetium",
tag: "chemicalElement"
},
{
name: "Ruthenium",
number: "44",
```

```
abbreviation: "Ru",
url:
"https://en.wikipedia.org/wiki/Ruthenium",
tag: "chemicalElement"
},
{
name: "Rhodium",
number: "45",
abbreviation: "Rh",
url: "https://en.wikipedia.org/wiki/Rhodium",
tag: "chemicalElement"
},
{
name: "Palladium",
number: "46",
abbreviation: "Pd",
url: "https://en.wikipedia.org/wiki/Palladium",
tag: "chemicalElement"
```

```
},
{
  name: "Silver",
  number: "47",
  abbreviation: "Ag",
  url: "https://en.wikipedia.org/wiki/Silver",
  tag: "chemicalElement"
},
{
  name: "Cadmium",
  number: "48",
  abbreviation: "Cd",
  url: "https://en.wikipedia.org/wiki/Cadmium",
  tag: "chemicalElement"
},
{
  name: "Indium",
  number: "49",
```

```
abbreviation: "In",
url: "https://en.wikipedia.org/wiki/Indium",
tag: "chemicalElement"
},
{
name: "Tin",
number: "50",
abbreviation: "Sn",
url: "https://en.wikipedia.org/wiki/Tin",
tag: "chemicalElement"
},
{
name: "Antimony",
number: "51",
abbreviation: "Sb",
url: "https://en.wikipedia.org/wiki/Antimony",
tag: "chemicalElement"
},
```

```
{  
  name: "Tellurium",  
  number: "52",  
  abbreviation: "Te",  
  url: "https://en.wikipedia.org/wiki/Tellurium",  
  tag: "chemicalElement"  
},  
{  
  name: "Iodine",  
  number: "53",  
  abbreviation: "I",  
  url: "https://en.wikipedia.org/wiki/Iodine",  
  tag: "chemicalElement"  
},  
{  
  name: "Xenon",  
  number: "54",  
  abbreviation: "Xe",  
}
```

```
url: "https://en.wikipedia.org/wiki/Xenon",
tag: "chemicalElement"
},
{
name: "Cesium", // also spelled Caesium
number: "55",
abbreviation: "Cs",
url: "https://en.wikipedia.org/wiki/Caesium",
tag: "chemicalElement"
},
{
name: "Barium",
number: "56",
abbreviation: "Ba",
url: "https://en.wikipedia.org/wiki/Barium",
tag: "chemicalElement"
},
{
```

```
name: "Lanthanum",
number: "57",
abbreviation: "La",
url:
"https://en.wikipedia.org/wiki/Lanthanum",
tag: "chemicalElement"
},
{
name: "Cerium",
number: "58",
abbreviation: "Ce",
url: "https://en.wikipedia.org/wiki/Cerium",
tag: "chemicalElement"
},
{
name: "Praseodymium",
number: "59",
abbreviation: "Pr",
```

```
url: "https://en.wikipedia.org/wiki/Praseodymium",
},
{
tag: "chemicalElement"
},
{
name: "Neodymium",
number: "60",
abbreviation: "Nd",
url: "https://en.wikipedia.org/wiki/Neodymium",
tag: "chemicalElement"
},
{
name: "Promethium",
number: "61",
abbreviation: "Pm",
```

```
url: "https://en.wikipedia.org/wiki/Promethium",
tag: "chemicalElement"
},
{
name: "Samarium",
number: "62",
abbreviation: "Sm",
url: "https://en.wikipedia.org/wiki/Samarium",
tag: "chemicalElement"
},
{
name: "Europium",
number: "63",
abbreviation: "Eu",
url: "https://en.wikipedia.org/wiki/Europium",
tag: "chemicalElement"
},
```

```
{  
  name: "Gadolinium",  
  number: "64",  
  abbreviation: "Gd",  
  url:  
    "https://en.wikipedia.org/wiki/Gadolinium",  
  tag: "chemicalElement"  
},  
{  
  name: "Terbium",  
  number: "65",  
  abbreviation: "Tb",  
  url: "https://en.wikipedia.org/wiki/Terbium",  
  tag: "chemicalElement"  
},  
{  
  name: "Dysprosium",  
  number: "66",  
}
```

```
abbreviation: "Dy",
url:
"https://en.wikipedia.org/wiki/Dysprosium",
tag: "chemicalElement"
},
{
name: "Holmium",
number: "67",
abbreviation: "Ho",
url: "https://en.wikipedia.org/wiki/Holmium",
tag: "chemicalElement"
},
{
name: "Erbium",
number: "68",
abbreviation: "Er",
url: "https://en.wikipedia.org/wiki/Erbium",
tag: "chemicalElement"
```

```
},
{
  name: "Thulium",
  number: "69",
  abbreviation: "Tm",
  url: "https://en.wikipedia.org/wiki/Thulium",
  tag: "chemicalElement"
},
{
  name: "Ytterbium",
  number: "70",
  abbreviation: "Yb",
  url: "https://en.wikipedia.org/wiki/Ytterbium",
  tag: "chemicalElement"
},
{
  name: "Lutetium",
  number: "71",
```

```
abbreviation: "Lu",
url: "https://en.wikipedia.org/wiki/Lutetium",
tag: "chemicalElement"
},
{
name: "Hafnium",
number: "72",
abbreviation: "Hf",
url: "https://en.wikipedia.org/wiki/Hafnium",
tag: "chemicalElement"
},
{
name: "Tantalum",
number: "73",
abbreviation: "Ta",
url: "https://en.wikipedia.org/wiki/Tantalum",
tag: "chemicalElement"
},
```

```
{  
  name: "Tungsten",  
  number: "74",  
  abbreviation: "W",  
  url: "https://en.wikipedia.org/wiki/Tungsten",  
  tag: "chemicalElement"  
},  
{  
  name: "Rhenium",  
  number: "75",  
  abbreviation: "Re",  
  url: "https://en.wikipedia.org/wiki/Rhenium",  
  tag: "chemicalElement"  
},  
{  
  name: "Osmium",  
  number: "76",  
  abbreviation: "Os",  
}
```

```
url: "https://en.wikipedia.org/wiki/Osmium",
tag: "chemicalElement"
},
{
name: "Iridium",
number: "77",
abbreviation: "Ir",
url: "https://en.wikipedia.org/wiki/Iridium",
tag: "chemicalElement"
},
{
name: "Platinum",
number: "78",
abbreviation: "Pt",
url: "https://en.wikipedia.org/wiki/Platinum",
tag: "chemicalElement"
},
{
```

```
name: "Gold",
number: "79",
abbreviation: "Au",
url: "https://en.wikipedia.org/wiki/Gold",
tag: "chemicalElement"
},
{
name: "Mercury",
number: "80",
abbreviation: "Hg",
url:
"https://en.wikipedia.org/wiki/Mercury_(element)",
tag: "chemicalElement"
},
{
name: "Thallium",
number: "81",
```

```
abbreviation: "Tl",
url: "https://en.wikipedia.org/wiki/Thallium",
tag: "chemicalElement"
},
{
name: "Lead",
number: "82",
abbreviation: "Pb",
url: "https://en.wikipedia.org/wiki/Lead",
tag: "chemicalElement"
},
{
name: "Bismuth",
number: "83",
abbreviation: "Bi",
url: "https://en.wikipedia.org/wiki/Bismuth",
tag: "chemicalElement"
},
```

```
{  
  name: "Polonium",  
  number: "84",  
  abbreviation: "Po",  
  url: "https://en.wikipedia.org/wiki/Polonium",  
  tag: "chemicalElement"  
},  
{  
  name: "Astatine",  
  number: "85",  
  abbreviation: "At",  
  url: "https://en.wikipedia.org/wiki/Astatine",  
  tag: "chemicalElement"  
},  
{  
  name: "Radon",  
  number: "86",  
  abbreviation: "Rn",  
}
```

```
url: "https://en.wikipedia.org/wiki/Radon",
tag: "chemicalElement"
},
{
name: "Francium",
number: "87",
abbreviation: "Fr",
url: "https://en.wikipedia.org/wiki/Francium",
tag: "chemicalElement"
},
{
name: "Radium",
number: "88",
abbreviation: "Ra",
url: "https://en.wikipedia.org/wiki/Radium",
tag: "chemicalElement"
},
{
```

```
name: "Actinium",
number: "89",
abbreviation: "Ac",
url: "https://en.wikipedia.org/wiki/Actinium",
tag: "chemicalElement"
},
{
name: "Thorium",
number: "90",
abbreviation: "Th",
url: "https://en.wikipedia.org/wiki/Thorium",
tag: "chemicalElement"
},
{
name: "Protactinium",
number: "91",
abbreviation: "Pa",
```

```
url: "https://en.wikipedia.org/wiki/Protactinium",
tag: "chemicalElement"
},
{
name: "Uranium",
number: "92",
abbreviation: "U",
url: "https://en.wikipedia.org/wiki/Uranium",
tag: "chemicalElement"
},
{
name: "Neptunium",
number: "93",
abbreviation: "Np",
url:
"https://en.wikipedia.org/wiki/Neptunium",
tag: "chemicalElement"
```

```
},
{
  name: "Plutonium",
  number: "94",
  abbreviation: "Pu",
  url: "https://en.wikipedia.org/wiki/Plutonium",
  tag: "chemicalElement"
},
{
  name: "Americium",
  number: "95",
  abbreviation: "Am",
  url:
  "https://en.wikipedia.org/wiki/Americium",
  tag: "chemicalElement"
},
{
  name: "Curium",
```

```
number: "96",
abbreviation: "Cm",
url: "https://en.wikipedia.org/wiki/Curium",
tag: "chemicalElement"
},
{
name: "Berkelium",
number: "97",
abbreviation: "Bk",
url: "https://en.wikipedia.org/wiki/Berkelium",
tag: "chemicalElement"
},
{
name: "Californium",
number: "98",
abbreviation: "Cf",
url:
"https://en.wikipedia.org/wiki/Californium",
```

```
tag: "chemicalElement"
},
{
  name: "Einsteinium",
  number: "99",
  abbreviation: "Es",
  url:
  "https://en.wikipedia.org/wiki/Einsteinium",
  tag: "chemicalElement"
},
{
  name: "Fermium",
  number: "100",
  abbreviation: "Fm",
  url: "https://en.wikipedia.org/wiki/Fermium",
  tag: "chemicalElement"
},
{
```

```
name: "Mendelevium",
number: "101",
abbreviation: "Md",
url:
"https://en.wikipedia.org/wiki/Mendelevium",
tag: "chemicalElement"
},
{
name: "Nobelium",
number: "102",
abbreviation: "No",
url: "https://en.wikipedia.org/wiki/Nobelium",
tag: "chemicalElement"
},
{
name: "Lawrencium",
number: "103",
abbreviation: "Lr",
```

```
url: "https://en.wikipedia.org/wiki/Lawrencium",
tag: "chemicalElement"
},
{
name: "Rutherfordium",
number: "104",
abbreviation: "Rf",
url: "https://en.wikipedia.org/wiki/Rutherfordium",
tag: "chemicalElement"
},
{
name: "Dubnium",
number: "105",
abbreviation: "Db",
url: "https://en.wikipedia.org/wiki/Dubnium",
tag: "chemicalElement"
```

```
},  
{  
  name: "Seaborgium",  
  number: "106",  
  abbreviation: "Sg",  
  url:  
    "https://en.wikipedia.org/wiki/Seaborgium",  
  tag: "chemicalElement"  
},  
{  
  name: "Bohrium",  
  number: "107",  
  abbreviation: "Bh",  
  url: "https://en.wikipedia.org/wiki/Bohrium",  
  tag: "chemicalElement"  
},  
{  
  name: "Hassium",  
  number: "108",  
  abbreviation: "Ts",  
  url: "https://en.wikipedia.org/wiki/Hassium",  
  tag: "chemicalElement"
```

```
number: "108",
abbreviation: "Hs",
url: "https://en.wikipedia.org/wiki/Hassium",
tag: "chemicalElement"
},
{
name: "Meitnerium",
number: "109",
abbreviation: "Mt",
url:
"https://en.wikipedia.org/wiki/Meitnerium",
tag: "chemicalElement"
},
{
name: "Darmstadtium",
number: "110",
abbreviation: "Ds",
```

```
url: "https://en.wikipedia.org/wiki/Darmstadtium",
tag: "chemicalElement"
},
{
name: "Roentgenium",
number: "111",
abbreviation: "Rg",
url:
"https://en.wikipedia.org/wiki/Roentgenium",
tag: "chemicalElement"
},
{
name: "Copernicium",
number: "112",
abbreviation: "Cn",
url:
"https://en.wikipedia.org/wiki/Copernicium",
```

```
tag: "chemicalElement"
},
{
  name: "Nihonium",
  number: "113",
  abbreviation: "Nh",
  url: "https://en.wikipedia.org/wiki/Nihonium",
  tag: "chemicalElement"
},
{
  name: "Flerovium",
  number: "114",
  abbreviation: "Fl",
  url: "https://en.wikipedia.org/wiki/Flerovium",
  tag: "chemicalElement"
},
{
  name: "Moscovium",
```

```
number: "115",
abbreviation: "Mc",
url:
"https://en.wikipedia.org/wiki/Moscovium",
tag: "chemicalElement"
},
{
name: "Livermorium",
number: "116",
abbreviation: "Lv",
url:
"https://en.wikipedia.org/wiki/Livermorium",
tag: "chemicalElement"
},
{
name: "Tennessine",
number: "117",
abbreviation: "Ts",
```

```
url: "https://en.wikipedia.org/wiki/Tennessine",
tag: "chemicalElement"
},
{
name: "Oganesson",
number: "118",
abbreviation: "Og",
url:
"https://en.wikipedia.org/wiki/Oganesson",
tag: "chemicalElement"
}
];
```

// Dedicated to God the Father

// All Rights Reserved Christopher Andrew
Topalian Copyright 2000-2024

// <https://github.com/ChristopherTopalian>

//
<https://github.com/ChristopherAndrewTopalia>
n

// dataVideosChemistry.js

let dataVideosChemistry =
[

{
 title: "The Periodic Table Explained",
 date: "July 27",

```
year: "2017",
url:
"https://www.youtube.com/embed/wXRHz5ZE
IK0?start=13",
tag: "youtubeVideo"
},
{
  title: " Atomic Number Atomic Mass And
Mass Number ",
  date: "March 25",
  year: "2015",
  url:
"https://www.youtube.com/embed/iTpadg_fa
Dk?start=53",
tag: "youtubeVideo"
},
{
```

```
    title: " Chemical Reactions And  
Equations ",  
    date: "March 25",  
    year: "2015",  
    url:  
    "https://www.youtube.com/embed/WlISgj-  
BriE",  
    tag: "youtubeVideo"  
},  
{  
    title: "Faraday's First Law of  
Electrolysis",  
    date: "April 1",  
    year: "2013",  
    url:  
    "https://www.youtube.com/embed/sZ8Z54E4  
WXI",  
    tag: "youtubeVideo"
```

```
 }  
];
```

// Dedicated to God the Father

// All Rights Reserved Christopher Andrew
Topalian Copyright 2000-2024

// <https://github.com/ChristopherTopalian>

//
<https://github.com/ChristopherAndrewTopalia>
n

// collidedWithClass.js

function collidedWithClass(obj1, className)
{

let collidedAny = false;

// if obj1 exists

```
if (ge(obj1))
{
    // get collision box of obj1
    let theRect1 =
        ge(obj1).getBoundingClientRect();

    // get all objects with specified class
    let objectsWithClass =
        document.querySelectorAll('.' + className);

    for (let i = 0; i < objectsWithClass.length;
i++)
    {
        let obj = objectsWithClass[i];

        let theRect2 =
            obj.getBoundingClientRect();
```

```
if (theRect1.right >= theRect2.left &&
    theRect1.left <= theRect2.right &&
    theRect1.bottom >= theRect2.top &&
    theRect1.top <= theRect2.bottom)
{
    collidedAny = true;
}
}
return collidedAny;
}
```

// Dedicated to God the Father

// All Rights Reserved Christopher Andrew
Topalian Copyright 2000-2024

// <https://github.com/ChristopherTopalian>

//
<https://github.com/ChristopherAndrewTopalia>
n

// collision.js

let collidedElementId;

let isColliding = false;

function collision(obj1, obj2)

```
{  
    // if both objects exist  
    if (ge(obj1) && ge(obj2))  
    {  
        // get collision box of obj1  
        let theRect1 =  
            ge(obj1).getBoundingClientRect();  
  
        // get collision box of obj2  
        let theRect2 =  
            ge(obj2).getBoundingClientRect();  
  
        let collidedAny = (  
            theRect1.right >= theRect2.left &&  
            theRect1.left <= theRect2.right &&  
            theRect1.bottom >= theRect2.top &&  
            theRect1.top <= theRect2.bottom  
        );  
    }  
}
```

// collided left side

```
let collidedLeft = theRect1.left >  
theRect2.left || theRect1.right >  
theRect2.right;
```

// collided right side

```
let collidedRight = theRect1.left <  
theRect2.left || theRect1.right <  
theRect2.right;
```

// collided top side

```
let collidedTop = theRect1.top >  
theRect2.top || theRect1.bottom >  
theRect2.bottom;
```

// collided bottom side

```
let collidedBottom = theRect1.top <  
theRect2.top || theRect1.bottom <  
theRect2.bottom;
```

```
if (collidedAny)  
{  
    isColliding = true;
```

```
collidedElementId = obj2;
```

```
audioPlay('sfx_blip_001', 1.0);
```

```
if (collidedLeft)  
{  
    movePlayer(1,0);  
}  
if (collidedRight)  
{
```

```
    movePlayer(-1,0);
}
if (collidedTop)
{
    movePlayer(0,1);
}
if (collidedBottom)
{
    movePlayer(0,-1);
}
return true;
}
```

```
}
```

```
isColliding = false;
```

```
// if no collision
return null;
```

}

// Dedicated to God the Father

// All Rights Reserved Christopher Andrew
Topalian Copyright 2000-2024

// <https://github.com/ChristopherTopalian>

//
<https://github.com/ChristopherAndrewTopalia>
n

// collisionBottomSide.js

// detects collisions from bottom side only
function collisionBottomSide(obj1, obj2)
{
 if (ge(obj1) && ge(obj2))
 {

```
let theRect1 =  
ge(obj1).getBoundingClientRect();
```

```
let theRect2 =  
ge(obj2).getBoundingClientRect();
```

```
let collided = !(theRect1.right <  
theRect2.left ||  
theRect1.left > theRect2.right ||  
theRect1.bottom < theRect2.top ||  
theRect1.top > theRect2.bottom);
```

```
let collidedY1 = theRect1.top <  
theRect2.top || theRect1.bottom <  
theRect2.bottom;
```

```
if (collided == true && collidedY1 != true)  
{
```

```
audioPlay("sfx_blip_001", 1.0);

ge('minimap').style.borderColor =
collisionColor001;

ge(obj2).style.backgroundColor =
collisionColor002;

collidedElementId = obj2;

setTimeout(function()
{
    ge(obj2).style.backgroundColor =
"rgb(0, 0, 0)";

    ge('minimap').style.borderColor =
'rgb(255, 255, 255)';
}, 500);
```

```
    }  
}  
}
```

// Dedicated to God the Father

// All Rights Reserved Christopher Andrew
Topalian Copyright 2000-2024

// <https://github.com/ChristopherTopalian>

//
<https://github.com/ChristopherAndrewTopalia>
n

// collisionLeftSide.js

// detects collisions from left side only
function collisionLeftSide(obj1, obj2)
{
 if (ge(obj1) && ge(obj2))
 {

```
let theRect1 =  
ge(obj1).getBoundingClientRect();
```

```
let theRect2 =  
ge(obj2).getBoundingClientRect();
```

```
let collided = !(theRect1.right <  
theRect2.left ||  
theRect1.left > theRect2.right ||  
theRect1.bottom < theRect2.top ||  
theRect1.top > theRect2.bottom);
```

```
let collidedY1 = theRect1.left >  
theRect2.left || theRect1.right >  
theRect2.right;
```

```
if (collided == true && collidedY1 != true)  
{
```

```
audioPlay("sfx_blip_001", 1.0);

ge('minimap').style.borderColor =
collisionColor001;

ge(obj2).style.backgroundColor =
collisionColor002;

collidedElementId = obj2;

setTimeout(function()
{
    ge(obj2).style.backgroundColor =
"rgb(0, 0, 0)";

    ge('minimap').style.borderColor =
'rgb(255, 255, 255)';
}, 500);
```

```
    }  
}  
}
```

// Dedicated to God the Father

// All Rights Reserved Christopher Andrew
Topalian Copyright 2000-2024

// <https://github.com/ChristopherTopalian>

//
<https://github.com/ChristopherAndrewTopalia>
n

// collisionRightSide.js

// detects collisions from right side only
function collisionRightSide(obj1, obj2)
{
 if (ge(obj1) && ge(obj2))
 {

```
let theRect1 =  
ge(obj1).getBoundingClientRect();
```

```
let theRect2 =  
ge(obj2).getBoundingClientRect();
```

```
let collided = !(theRect1.right <  
theRect2.left ||  
theRect1.left > theRect2.right ||  
theRect1.bottom < theRect2.top ||  
theRect1.top > theRect2.bottom);
```

```
let collidedY1 = theRect1.left <  
theRect2.left || theRect1.right <  
theRect2.right;
```

```
if (collided == true && collidedY1 != true)  
{
```

```
audioPlay("sfx_blip_001", 1.0);

ge('minimap').style.borderColor =
collisionColor001;

ge(obj2).style.backgroundColor =
collisionColor002;

collidedElementId = obj2;

setTimeout(function()
{
    ge(obj2).style.backgroundColor =
"rgb(0, 0, 0)";

    ge('minimap').style.borderColor =
'rgb(255, 255, 255)';
}, 500);
```

```
    }  
}  
}
```

// Dedicated to God the Father

// All Rights Reserved Christopher Andrew
Topalian Copyright 2000-2024

// <https://github.com/ChristopherTopalian>

//
<https://github.com/ChristopherAndrewTopalia>
n

// collisionTopSide.js

// detects collisions from top side only
function collisionTopSide(obj1, obj2)
{
 if (ge(obj1) && ge(obj2))
 {

```
let theRect1 =  
ge(obj1).getBoundingClientRect();
```

```
let theRect2 =  
ge(obj2).getBoundingClientRect();
```

```
let collided = !(theRect1.right <  
theRect2.left ||  
theRect1.left > theRect2.right ||  
theRect1.bottom < theRect2.top ||  
theRect1.top > theRect2.bottom);
```

```
let collidedY1 = theRect1.top >  
theRect2.top || theRect1.bottom >  
theRect2.bottom;
```

```
if (collided == true && collidedY1 != true)  
{
```

```
audioPlay("sfx_blip_001", 1.0);

ge('minimap').style.borderColor =
collisionColor001;

ge(obj2).style.backgroundColor =
collisionColor002;

collidedElementId = obj2;

setTimeout(function()
{
    ge('minimap').style.borderColor =
'rgb(255, 255, 255)';

    ge(obj2).style.backgroundColor =
"rgb(0, 0, 0)";
}, 500);
```

```
    }  
}  
}
```

// Dedicated to God the Father

// All Rights Reserved Christopher Andrew
Topalian Copyright 2000-2024

// <https://github.com/ChristopherTopalian>

//
<https://github.com/ChristopherAndrewTopalia>
n

// detect if player crossed a specific x or y
coordinate

function wasLineCrossed()
{

// if player object exists
if (ge(ourPlayer.id))
{

```
// if player is located at a certain y
position
if (getPos(ourPlayer.id).y > 1700 - 20 &&
getPos(ourPlayer.id).y < 1700 + 20)
{
    audioPlay("sfx_crystal_bell_001", 1.0);

    ge('infoDiv').innerHTML = "1700 Y
Crossed<br>Hydrogen";

    ge('minimap').style.borderColor =
collisionColor001;

// creates only one
if (lineAt1700Flag == 0)
{
    audioPlay("sfx_zap_001", 1.0);
```

```
makePeriodicElement(dataPeriodicElements,
"Hydrogen", 50, 1700);
}

setTimeout(function()
{
    ge('minimap').style.borderColor =
'rgb(255, 255, 255)';
}, 500);

lineAt1700Flag = 1;
}

// if player is located at a certain y
position
if (getPos(ourPlayer.id).y > 1900 - 20 &&
getPos(ourPlayer.id).y < 1900 + 20)
{
```

```
audioPlay("sfx_crystal_bell_001", 1.0);

ge('minimap').style.borderColor =
collisionColor001;

ge('infoDiv').innerHTML = "1900 Y
Crossed<br>Helium";

// creates only one
if (lineAt1900Flag == 0)
{
    audioPlay("sfx_zap_001", 1.0);

makePeriodicElement(dataPeriodicElements,
"Helium", 50, 1900);
}

setTimeout(function()
```

```
{  
    ge('minimap').style.borderColor =  
'rgb(255, 255, 255);  
}, 500);  
  
lineAt1900Flag = 1;  
}  
  
// if player is located at a certain y  
position  
if (getPos(ourPlayer.id).y > 2830 - 20 &&  
getPos(ourPlayer.id).y < 2830 + 20)  
{  
    audioPlay("sfx_crystal_bell_001", 1.0);  
  
    ge('infoDiv').innerHTML = "2830 Y  
Crossed";
```

```
ge('minimap').style.borderColor =  
collisionColor001;  
  
// creates only one  
if (lineAt2830Flag == 0)  
{  
    audioPlay("sfx_crystal_bell_001",  
1.0);  
  
    audioPlay("sfx_zap_001", 1.0);  
  
makePeriodicElements(dataPeriodicElements  
, 250, 2832, 300);  
}  
  
setTimeout(function()  
{
```

```
    ge('minimap').style.borderColor =  
'rgb(255, 255, 255);  
}, 500);  
  
lineAt2830Flag = 1;  
}  
  
// if player is located at a certain y  
position  
if (getPos(ourPlayer.id).y > 5730 - 20 &&  
getPos(ourPlayer.id).y < 5730 + 20)  
{  
    audioPlay("sfx_crystal_bell_001", 1.0);  
  
    ge('infoDiv').innerHTML = "5730 Y  
Crossed";
```

```
ge('minimap').style.borderColor =  
collisionColor001;
```

```
// creates only one  
if (lineAt5730Flag == 0)  
{  
    audioPlay("sfx_crystal_bell_001",  
1.0);
```

```
    audioPlay("sfx_zap_001", 1.0);
```

```
makePeriodicElementsNameAsLink(dataPeri  
odicElements, 800, 5730);  
}
```

```
setTimeout(function()  
{
```

```
    ge('minimap').style.borderColor =  
    'rgb(255, 255, 255);  
}, 500);  
  
    lineAt5730Flag = 1;  
}  
}  
}  
}
```

// Dedicated to God the Father

// All Rights Reserved Christopher Andrew
Topalian Copyright 2000-2024

// <https://github.com/ChristopherTopalian>

//
<https://github.com/ChristopherAndrewTopalia>
n

// wasObjectCrossed.js

let carbonFlag = 0;
let nitrogenFlag = 0;

let lineYBoxThreeFlag = 0;
let lineYBoxFourFlag = 0;

```
let lineAt1700Flag = 0;  
let lineAt1900Flag = 0;  
let lineAt2830Flag = 0;  
let lineAt5730Flag = 0;
```

```
// detect if player crossed a specified object's  
x or y coordinate  
function wasObjectLineCrossed(whichPlayer)  
{  
    // if objects exist  
    if (ge(whichPlayer.id) &&  
        ge('boxCarbonTrigger'))  
    {  
        // if player is located at a certain x  
        position, compared with boxCarbonTrigger  
        if (getPos(whichPlayer.id).x >  
            getPos("boxCarbonTrigger").x - 20 &&
```

```
getPos(whichPlayer.id).x <
getPos("boxCarbonTrigger").x + 20)
{
    audioPlay("sfx_crystal_bell_001", 1.0);

ge("boxCarbonTrigger").style.backgroundColor
or = collisionColor002;

// creates only one
if (carbonFlag == 0)
{
    audioPlay("sfx_zap_001", 1.0);

makePeriodicElement(dataPeriodicElements,
"Carbon", getPos("boxCarbonTrigger").x,
getPos("boxCarbonTrigger").y);

carbonFlag = 1;
```

```
}
```

```
    ge('minimap').style.borderColor =  
collisionColor001;
```

```
    ge('infoDiv').innerHTML =  
"boxCarbonTrigger X Crossed";
```

```
setTimeout(function()  
{  
    ge('minimap').style.borderColor =  
'rgb(255, 255, 255)';  
  
    ge("boxCarbonTrigger").style.backgroundCol  
or = "rgb(0, 0, 0)";  
}, 500);  
}
```

```
// if objects exist
if (ge(whichPlayer.id) &&
    ge("boxNitrogenTrigger"))
{
    // if player is located at a certain x
    position, compared with boxNitrogenTrigger
    if (getPos(whichPlayer.id).x >
        getPos("boxNitrogenTrigger").x - 20 &&
        getPos(whichPlayer.id).x <
        getPos("boxNitrogenTrigger").x + 20)
    {
        audioPlay("sfx_crystal_bell_001",
1.0);

        ge('minimap').style.borderColor =
collisionColor001;
```

```
ge('infoDiv').innerHTML =  
"boxNitrogenTrigger X Crossed";
```

```
ge("boxNitrogenTrigger").style.backgroundColor = collisionColor002;
```

```
// creates only one  
if (nitrogenFlag == 0)  
{  
    audioPlay("sfx_zap_001", 1.0);
```

```
makePeriodicElement(dataPeriodicElements,  
"Nitrogen", getPos("boxNitrogenTrigger").x,  
getPos("boxNitrogenTrigger").y);
```

```
    nitrogenFlag = 1;  
}
```

```
setTimeout(function()
{
    ge("boxNitrogenTrigger").style.backgroundColor =
        "rgb(0, 0, 0)";

    ge('minimap').style.borderColor =
        'rgb(255, 255, 255)';
}, 500);
}

// if objects exist
if (ge(whichPlayer.id) &&
    ge("boxSilverTrigger"))
{
    // if player is located at a certain
    position, compared with boxSilverTrigger
```

```
if (getPos(whichPlayer.id).y >
getPos("boxSilverTrigger").y - 20 &&
getPos(whichPlayer.id).y <
getPos("boxSilverTrigger").y + 20)
{
    audioPlay("sfx_crystal_bell_001",
1.0);
```

```
        ge('minimap').style.borderColor =
collisionColor001;
```

```
        ge("boxSilverTrigger").style.backgroundColor =
collisionColor002;
```

```
        ge('infoDiv').innerHTML=
"boxSilverTrigger Y Crossed<br>SILVER";
```

// creates only one

```
if (lineYBoxThreeFlag == 0)
{
    audioPlay("sfx_zap_001", 1.0);

makePeriodicElement(dataPeriodicElements,
"Silver", getPos("boxSilverTrigger").x,
getPos("boxSilverTrigger").y);

    lineYBoxThreeFlag = 1;
}

setTimeout(function()
{
    ge("boxSilverTrigger").style.backgroundColor
    r = "rgb(0, 0, 0)";

        ge('minimap').style.borderColor =
    'rgb(255, 255, 255);
```

```
        }, 500);
    }
}

// if objects exist
if (ge(whichPlayer.id) &&
    ge("boxGoldTrigger"))
{
    // if player is located at a certain
    position, compared with boxGoldTrigger
    if (getPos(whichPlayer.id).y >
getPos("boxGoldTrigger").y - 20 &&
        getPos(whichPlayer.id).y <
getPos("boxGoldTrigger").y + 20)
    {
        audioPlay("sfx_crystal_bell_001",
1.0);
```

```
ge('minimap').style.borderColor =  
collisionColor001;
```

```
ge("boxGoldTrigger").style.backgroundColor  
= collisionColor002;
```

```
ge('infoDiv').innerHTML =  
"boxGoldTrigger Y Crossed<br>GOLD";
```

```
// creates only one  
if (lineYBoxFourFlag == 0)  
{  
    audioPlay("sfx_zap_001", 1.0);
```

```
makePeriodicElement(dataPeriodicElements,  
"Gold", getPos("boxGoldTrigger").x,  
getPos("boxGoldTrigger").y);
```

```
lineYBoxFourFlag = 1;  
}  
  
setTimeout(function()  
{  
ge("boxGoldTrigger").style.backgroundColor  
= "rgb(0, 0, 0);  
  
    ge('minimap').style.borderColor =  
'rgb(255, 255, 255)';  
    }, 500);  
}  
}  
}
```

// Dedicated to God the Father

// All Rights Reserved Christopher Andrew
Topalian Copyright 2000-2024

// <https://github.com/ChristopherTopalian>

//
<https://github.com/ChristopherAndrewTopalia>
n

// cameraFollowsPlayer.js

```
function cameraFollowsPlayer()
{
    if (ge(ourPlayer.id))
    {
        ge(ourPlayer.id).scrollIntoView()
```

```
{  
    block: "center",  
    inline: "center"  
});  
}  
}  
}
```

// Dedicated to God the Father

// All Rights Reserved Christopher Andrew
Topalian Copyright 2000-2024

// <https://github.com/ChristopherTopalian>

//
<https://github.com/ChristopherAndrewTopalia>
n

// controlsKeyboard.js

// keyboard letter codes being pressed
let keyboard = {};

// 87 is w, 38 is up arrow
keyboard.UP = 87;

```
// 83 is s, 40 is down arrow  
keyboard.DOWN = 83;
```

```
// 65 is a, 37 is left arrow  
keyboard.LEFT = 65;
```

```
// 68 is d, 39 is right arrow  
keyboard.RIGHT = 68;
```

```
// Shift Button  
keyboard.SHIFT = 16;
```

```
function theControls(e)  
{  
    // AUDIO  
    if (e.keyCode == 86) // Press Letter v  
    {
```

```
audioPlay("Action_Strike", 0.5);

    ge('infoDiv').innerHTML = "Song
Playing";
}

if (e.keyCode == 66) // Press Letter b
{
    audioPause("Action_Strike", 0.5);

    ge('infoDiv').innerHTML = "Song
Stopped";
}

if (e.keyCode == 27) // Press Esc
{
    audioPause("Action_Strike");
```

```
    ge('infoDiv').innerHTML = "Song  
Stopped";  
}  
  
// TRANSPARENCY  
if (e.keyCode == 77) // Letter m  
{  
    ge(ourPlayer.id).style.opacity = "1.0";  
}  
if (e.keyCode == 78) // Letter n  
{  
    ge(ourPlayer.id).style.opacity = "0.2";  
  
    ge(ourPlayer.id).innerHTML = "";  
}  
  
// SPEED  
if (e.keyCode == 192) // tilda ~
```

```
{  
    ourPlayer.speedMultiplier = 0.5;  
  
    ge(ourPlayer.id).innerHTML = "Speed  
0.5";  
  
    setTimeout(function()  
    {  
        ge("thePlayer").innerHTML = "";  
    }, 3000);  
}  
  
if (e.keyCode == 49 || e.keyCode == 97) //  
number 1, or numpad 1  
{  
    ourPlayer.speedMultiplier = 1;  
  
    ge(ourPlayer.id).innerHTML = "Speed 1";
```

```
setTimeout(function()
{
    ge("thePlayer").innerHTML = "";
}, 3000);

}

if (e.keyCode == 50 || e.keyCode == 98) //  
number 2
{
    ge(ourPlayer.id).speedMultiplier = 2;

    ge(ourPlayer.id).innerHTML = "Speed 2";

setTimeout(function()
{
    ge(ourPlayer.id).innerHTML = "";
}, 3000);
```

```
}

if (e.keyCode == 51 || e.keyCode == 99) //  
number 3  
{  
    ourPlayer.speedMultiplier = 3;  
  
    ge(ourPlayer.id).innerHTML = "Speed 3";  
  
    setTimeout(function()  
    {  
        ge("thePlayer").innerHTML = "";  
    }, 3000);  
}  
  
if (e.keyCode == 52 || e.keyCode == 100) //  
number 4  
{
```

```
ourPlayer.speedMultiplier = 4;  
  
ge(ourPlayer.id).innerHTML = "Speed 4";  
  
setTimeout(function()  
{  
    ge("thePlayer").innerHTML = "";  
}, 3000);  
  
}  
  
if (e.keyCode == 53 || e.keyCode == 101) //  
number 5  
{  
    ourPlayer.speedMultiplier = 5;  
  
    ge(ourPlayer.id).innerHTML = "Speed 5";  
  
    setTimeout(function()  
}
```

```
{  
    ge("thePlayer").innerHTML = "";  
, 3000);  
  
}  
  
if (e.keyCode == 54 || e.keyCode == 102) //  
number 6  
{  
    ourPlayer.speedMultiplier = 6;  
  
    ge(ourPlayer.id).innerHTML = "Speed 6";  
  
setTimeout(function()  
{  
    ge("thePlayer").innerHTML = "";  
, 3000);  
}
```

```
if (e.keyCode == 55 || e.keyCode == 103) //  
number 7  
{  
    ourPlayer.speedMultiplier = 7;  
  
    ge(ourPlayer.id).innerHTML = "Speed 7";  
  
setTimeout(function()  
{  
    ge("thePlayer").innerHTML = "";  
}, 3000);  
}  
  
if (e.keyCode == 56 || e.keyCode == 104) //  
number 8  
{  
    ourPlayer.speedMultiplier = 8;
```

```
ge(ourPlayer.id).innerHTML = "Speed 8";  
  
setTimeout(function()  
{  
    ge("thePlayer").innerHTML = "";  
, 3000);  
}  
  
if (e.keyCode == 57 || e.keyCode == 105) //  
number 9  
{  
    ourPlayer.speedMultiplier = 9;  
  
    ge(ourPlayer.id).innerHTML = "Speed 9";  
  
    setTimeout(function()  
{  
        ge("thePlayer").innerHTML = "";
```

```
    }, 3000);  
}  
  
if (e.keyCode == 48 || e.keyCode == 96) //  
number 0  
{  
    ourPlayer.speedMultiplier = 200;  
  
    ge(ourPlayer.id).innerHTML = "Speed  
10";  
  
setTimeout(function()  
{  
    ge("thePlayer").innerHTML = "";  
}, 3000);  
}  
  
let theKeyCode = e.keyCode || e.which;
```

```
if (e.type === 'keydown')
{
    keyboard[theKeyCode] = true;
}
else if (e.type === 'keyup')
{
    keyboard[theKeyCode] = false;
}
```

// Dedicated to God the Father

// All Rights Reserved Christopher Andrew
Topalian Copyright 2000-2024

// <https://github.com/ChristopherTopalian>

//
<https://github.com/ChristopherAndrewTopalia>
n

// gameLoop.js

function gameLoop()
{

playerMotion();

cameraFollowsPlayer();

```
// gravity(9.8);
```

```
//-//
```

```
// text update of player position
if (ge('xPosDiv') && ge('yPosDiv'))
{
    ge('xPosDiv').innerHTML = ourPlayer.x;
    ge('yPosDiv').innerHTML = ourPlayer.y;
}
```

```
//-//
```

```
if (ge('infoDiv'))
{
    if (collidedWithClass(ourPlayer.id,
'boxStyle001'))
```

```
{  
    ge('infoDiv').innerHTML =  
collidedElementId;  
}  
else  
{  
    ge('infoDiv').innerHTML = 'No  
Collision';  
}  
}  
//--//
```

```
// minimap player update  
if (ge('minimap') && ge('minimapPlayer'))  
{  
    updateMinimapPlayer();  
}
```

//-//

```
// minimap copper treasure update
if (ge('minimap') && ge('treasureCopper'))
{
    // id of icon, id of element we are tracking

updateMinimapTreasure('minimapCopperIcon'
, 'treasureCopper');
}
```

//-//

```
// minimap silver treasure update
if (ge('minimap') && ge('treasureSilver'))
{
    // id of icon, id of element we are tracking
```

```
updateMinimapTreasure('minimapSilverIcon',
'treasureSilver');
}
```

//-//

```
// update speed
if (ge('speedometerDiv'))
{
    ge('speedometerDiv').innerHTML =
'Speed' + '<br>' +
updateSpeedometer().toFixed(3);
}
```

//-//

```
isPlayerMoving();
```

//-//

```
if (ge('collisionInfoDiv'))  
{  
    ge('collisionInfoDiv').innerHTML =  
collidedElementId;  
}
```

//-//

```
collision(ourPlayer.id, "floor1");  
collision(ourPlayer.id, "floor2");  
collision(ourPlayer.id, "floor3");
```

```
collision(ourPlayer.id, "boxIron");  
collision(ourPlayer.id, "boxCopper");  
collision(ourPlayer.id, "boxSilver");
```

collision(ourPlayer.id, "boxGold");

//-//

**collisionTopSide(ourPlayer.id,
"boxCollideTop");**

**collisionBottomSide(ourPlayer.id,
"boxCollideBottom");**

**collisionLeftSide(ourPlayer.id,
"boxCollideLeft");**

**collisionRightSide(ourPlayer.id,
"boxCollideRight");**

//-//

```
wasObjectLineCrossed(ourPlayer);  
  
wasLineCrossed();  
  
//--  
  
keepPlayerInWorld();  
  
requestAnimationFrame(gameLoop);  
}
```

// Dedicated to God the Father

// All Rights Reserved Christopher Andrew
Topalian Copyright 2000-2024

// <https://github.com/ChristopherTopalian>

//
<https://github.com/ChristopherAndrewTopalia>
n

// gravity.js

```
function gravity(whichAmount)
{
    // gravity strength
    let gravity = whichAmount;
```

```
// apply gravity  
ourPlayer.y += gravity;  
  
movePlayer();  
}
```

// Dedicated to God the Father

// All Rights Reserved Christopher Andrew
Topalian Copyright 2000-2024

// <https://github.com/ChristopherTopalian>

//
<https://github.com/ChristopherAndrewTopalia>
n

// keepPlayerInWorld.js

```
function keepPlayerInWorld()
{
    if (ge(ourPlayer.id))
    {
        // if left
```

```
if (ourPlayer.x <= 0)
{
    ourPlayer.x = 0;
}

// if right
if (ourPlayer.x > backgroundSizeX)
{
    ourPlayer.x = backgroundSizeX;
}

// if top
if (ourPlayer.y <= 0)
{
    ourPlayer.y = 0;
}

// if bottom
```

```
if (ourPlayer.y >= backgroundSizeY)
{
    ourPlayer.y = backgroundSizeY;
}
}
```

// Dedicated to God the Father

// All Rights Reserved Christopher Andrew
Topalian Copyright 2000-2024

// <https://github.com/ChristopherTopalian>

//
<https://github.com/ChristopherAndrewTopalia>
n

// movePlayer.js

```
function movePlayer(xPos, yPos)
{
    if (ge(ourPlayer.id))
    {
```

```
ourPlayer.x += (xPos || 0) *  
ourPlayer.speedMultiplier;
```

```
ourPlayer.y += (yPos || 0) *  
ourPlayer.speedMultiplier;
```

```
ge(ourPlayer.id).style.left = ourPlayer.x  
+ "px";
```

```
ge(ourPlayer.id).style.top = ourPlayer.y  
+ "px";  
}  
}
```

// Dedicated to God the Father

// All Rights Reserved Christopher Andrew
Topalian Copyright 2000-2024

// <https://github.com/ChristopherTopalian>

//
<https://github.com/ChristopherAndrewTopalia>
n

// playerMotion.js

```
// player Movement Controls
function playerMotion()
{
    if (keyboard[keyboard.LEFT])
    {
```

```
movePlayer(-1, 0);

ge(ourPlayer.id).style.transform =
'scaleX(-1)';
}

if (keyboard[keyboard.RIGHT])
{
    movePlayer(1, 0);

    ge(ourPlayer.id).style.transform =
'scaleX(1)';
}

if (keyboard[keyboard.UP])
{
    movePlayer(0, -1);
}
```

```
if (keyboard[keyboard.DOWN])
{
    movePlayer(0, 1);
}

// boost speed
if (keyboard[keyboard.LEFT] &&
keyboard[keyboard.SHIFT])
{
    movePlayer(-2, 0);
}

if (keyboard[keyboard.RIGHT] &&
keyboard[keyboard.SHIFT])
{
    movePlayer(2, 0);
}
```

```
if (keyboard[keyboard.UP] &&
keyboard[keyboard.SHIFT])
{
    movePlayer(0, -2);
}

if (keyboard[keyboard.DOWN] &&
keyboard[keyboard.SHIFT])
{
    movePlayer(0, 2);
}
```

// Dedicated to God the Father

// All Rights Reserved Christopher Andrew
Topalian Copyright 2000-2024

// <https://github.com/ChristopherTopalian>

//
<https://github.com/ChristopherAndrewTopalia>
n

// findIndexByName.js

function findIndexByName(whichArray,
searchWord)

{

// index in array of found word

let index = whichArray.findIndex(

```
function(whichArray)
{
    return whichArray.name ===
searchWord;
}
);
return index;
}
```

// Dedicated to God the Father

// All Rights Reserved Christopher Andrew
Topalian Copyright 2000-2024

// <https://github.com/ChristopherTopalian>

//
<https://github.com/ChristopherAndrewTopalia>
n

// countAllElements.js

```
function countAllElements()
{
    let elements =
document.getElementsByTagName('*');
```

```
let amount = elements.length;  
  
return amount;  
}
```

// Dedicated to God the Father

// All Rights Reserved Christopher Andrew
Topalian Copyright 2000-2024

// <https://github.com/ChristopherTopalian>

//
<https://github.com/ChristopherAndrewTopalia>
n

// makeBackground.js

```
function makeBackground()
{
    let theBackground = ce('div');
    theBackground.id =
'theBackgroundDimensions';
```

```
theBackground.style.position = 'absolute';
theBackground.style.left = '0px';
theBackground.style.top = '0px';
theBackground.style.width =
backgroundSizeX + 'px';
theBackground.style.height =
backgroundSizeY + 'px';
// theBackground.style.background =
'rgb(0, 0, 0)';
theBackground.style.background =
'url("media/textures/stars-nice-
edited_orig.png")';
ba(theBackground);
}
```

// Dedicated to God the Father

// All Rights Reserved Christopher Andrew
Topalian Copyright 2000-2024

// <https://github.com/ChristopherTopalian>

//
<https://github.com/ChristopherAndrewTopalia>
n

// makeCollisionDiv.js

```
function makeCollisionDiv()
{
    let collisionInfoDiv = ce('div');
    collisionInfoDiv.textContent = 'Collision
Info';
```

```
collisionInfoDiv.id = 'collisionInfoDiv';
collisionInfoDiv.className =
'textStyle001';
collisionInfoDiv.title = 'makeCollisionDiv()';
collisionInfoDiv.style.position = 'fixed';
collisionInfoDiv.style.right = '0px';
collisionInfoDiv.style.bottom = '0px';
collisionInfoDiv.style.zIndex = 1000;
collisionInfoDiv.style.textShadow = '1px
1px #000000';
ba(collisionInfoDiv);
}
```

// Dedicated to God the Father

// All Rights Reserved Christopher Andrew
Topalian Copyright 2000-2024

// <https://github.com/ChristopherTopalian>

//
<https://github.com/ChristopherAndrewTopalia>
n

// makeDedication.js

```
function makeDedication()
{
    let theDedication = ce('div');
    theDedication.id = 'theDedication';
    theDedication.className = 'theDedication';
```

```
theDedication.style.zIndex = 1000;  
theDedication.innerHTML = `Dedicated to  
God the Father<br>  
<b>TOPALIAN GAME ENGINE</b><br>  
<a href =  
"https://github.com/ChristopherTopalian"  
target = "_blank" class =  
"textStyle002">https://github.com/Christophe  
rTopalian</a>  
`;  
ba(theDedication);  
}
```

// Dedicated to God the Father

// All Rights Reserved Christopher Andrew
Topalian Copyright 2000-2024

// <https://github.com/ChristopherTopalian>

//
<https://github.com/ChristopherAndrewTopalia>
n

// makeDistanceTraveledDiv.js

```
function makeDistanceTraveledDiv()
{
    let distanceTraveledDiv = ce('div');
    distanceTraveledDiv.id =
'distanceTraveledDiv';
```

```
distanceTraveledDiv.className =
'textStyle001';
distanceTraveledDiv.title =
'makeSpeedometer()';
distanceTraveledDiv.style.position = 'fixed';
distanceTraveledDiv.style.right = '0px';
distanceTraveledDiv.style.bottom = '200px';
distanceTraveledDiv.style.zIndex = 1000;
distanceTraveledDiv.style.textAlign =
'right';
distanceTraveledDiv.innerHTML =
'Distance <br>0';
distanceTraveledDiv.style.textShadow =
'1px 1px #000000';
ba(distanceTraveledDiv);
}
```

// Dedicated to God the Father

// All Rights Reserved Christopher Andrew
Topalian Copyright 2000-2024

// <https://github.com/ChristopherTopalian>

//
<https://github.com/ChristopherAndrewTopalia>
n

// makeElementInfoBox.js

```
function makeElementInfoBox()
{
    let elementDiv = ce('div');
    elementDiv.id = 'elementInfo';
    elementDiv.className = 'elementDiv';
```

```
ba(elementDiv);  
}
```

// Dedicated to God the Father

// All Rights Reserved Christopher Andrew
Topalian Copyright 2000-2024

// <https://github.com/ChristopherTopalian>

//
<https://github.com/ChristopherAndrewTopalia>
n

// makeInfoDiv.js

```
function makeInfoDiv()
{
    let infoDiv = ce('div');
    infoDiv.textContent = 'Info';
    infoDiv.id = 'infoDiv';
```

```
infoDiv.className = 'textStyle001';
infoDiv.title = 'makeInfoDiv()';
infoDiv.style.position = 'fixed';
infoDiv.style.right = '0px';
infoDiv.style.bottom = '25px';
infoDiv.style.zIndex = 1000;
infoDiv.style.textShadow = '1px 1px
#000000';
ba(infoDiv);
}
```

// Dedicated to God the Father

// All Rights Reserved Christopher Andrew
Topalian Copyright 2000-2024

// <https://github.com/ChristopherTopalian>

//
<https://github.com/ChristopherAndrewTopalia>
n

// makelsMovingDiv.js

```
function makelsMovingDiv()
{
    let isMovingDiv = ce('div');
    isMovingDiv.textContent = 'Motion';
    isMovingDiv.id = 'isMovingDiv';
```

```
isMovingDiv.className = 'textStyle001';
isMovingDiv.title = 'makelsMovingDiv()';
isMovingDiv.style.position = 'fixed';
isMovingDiv.style.right = '0px';
isMovingDiv.style.bottom = '250px';
isMovingDiv.style.zIndex = 1000;
isMovingDiv.style.textShadow = '1px 1px
#000000';
ba(isMovingDiv);
}
```

// Dedicated to God the Father

// All Rights Reserved Christopher Andrew
Topalian Copyright 2000-2024

// <https://github.com/ChristopherTopalian>

//
<https://github.com/ChristopherAndrewTopalia>
n

// makeMousePosDiv.js

```
function makeMousePosDiv()
{
    let infoMousePosition = ce('div');
    infoMousePosition.textContent = 'Mouse
    Pos';
```

```
info.mousePosition.id =  
'infoMousePositionId';  
    info.mousePosition.className =  
'textStyle001';  
    info.mousePosition.title =  
'makeMousePosDiv()';  
    info.mousePosition.style.position = 'fixed';  
    info.mousePosition.style.right = '0px';  
    info.mousePosition.style.bottom = '60px';  
    info.mousePosition.style.zIndex = 1000;  
    info.mousePosition.style.textShadow = '1px  
1px #000000';  
    ba(info.mousePosition);  
}
```

// Dedicated to God the Father

// All Rights Reserved Christopher Andrew
Topalian Copyright 2000-2024

// <https://github.com/ChristopherTopalian>

//
<https://github.com/ChristopherAndrewTopalia>
n

// makeObjectsToScreen.js

```
function makeObjectsToScreen(whichArray)
{
    for (let x = 0; x < whichArray.length; x++)
    {
        let theElement = ce(whichArray[x].type);
```

**theElement.style.position =
whichArray[x].positionType;**

**theElement.style.left = whichArray[x].x +
"px";**

**theElement.style.top = whichArray[x].y +
"px";**

**theElement.style.width =
whichArray[x].width + "px";**

**theElement.style.height =
whichArray[x].height + "px";**

theElement.id = whichArray[x].id;

```
theElement.innerHTML =  
whichArray[x].text;
```

```
theElement.title = whichArray[x].name;
```

```
theElement.className =  
whichArray[x].styleName;
```

```
ba(theElement);
```

```
}
```

// Dedicated to God the Father

// All Rights Reserved Christopher Andrew
Topalian Copyright 2000-2024

// <https://github.com/ChristopherTopalian>

//
<https://github.com/ChristopherAndrewTopalia>
n

// makeOneElement.js

function makeOneElement(theName,
theNumber, theLink, xPos, yPos)

{
let topicDiv = ce("div");
topicDiv.style.position = "absolute";

```
topicDiv.style.left = xPos + 50;  
topicDiv.style.top = yPos - 50;  
topicDiv.style.width = "163px";  
topicDiv.style.padding = "10px";  
topicDiv.style.zIndex = "2";  
topicDiv.style.borderStyle = "solid";  
topicDiv.style.borderColor = "rgb(255, 255,  
255)";  
topicDiv.style.background = "rgb(0, 0, 0)";  
topicDiv.style.fontFamily = "tahoma";  
topicDiv.style.fontSize = "20px";  
topicDiv.style.fontWeight = "bold";  
topicDiv.style.color = "rgb(255, 255, 255)";  
topicDiv.style.textAlign = "center";  
topicDiv.innerHTML = theName + "<br>";  
topicDiv.innerHTML += theNumber +  
"<br>";
```

```
topicDiv.innerHTML += '<a href = \''  
+theLink+'\' target = "_blank"> Link </a>';  
topicDiv.id = theName;  
ba(topicDiv);  
}
```

// Dedicated to God the Father

// All Rights Reserved Christopher Andrew
Topalian Copyright 2000-2024

// <https://github.com/ChristopherTopalian>

//
<https://github.com/ChristopherAndrewTopalia>
n

// makePeriodicElement.js

```
function makePeriodicElement(whichArray,  
theChoice, xPos, yPos)  
{
```

```
let theName =  
whichArray[findIndexByName(whichArray,  
theChoice)].name;
```

```
let theNumber =  
whichArray[findIndexByName(whichArray,  
theChoice)].number;
```

```
let theAbbreviation =  
whichArray[findIndexByName(whichArray,  
theChoice)].abbreviation;
```

```
let theLink =  
whichArray[findIndexByName(whichArray,  
theChoice)].url;
```

//-

```
let topicDiv = ce("div");
topicDiv.innerHTML = theName + "<br>";
topicDiv.innerHTML += theNumber +
"<br>";
topicDiv.innerHTML += theAbbreviation +
"<br>";
topicDiv.innerHTML += '<a href = \''
+theLink+'\' target = "_blank"> Link </a>';
topicDiv.id = theName;
topicDiv.title = theName + '\n' +
theAbbreviation;
topicDiv.style.position = "absolute";
topicDiv.style.left = xPos + 50;
topicDiv.style.top = yPos - 50;
topicDiv.style.width = "163px";
topicDiv.style.padding = "10px";
topicDiv.style.zIndex = "2";
topicDiv.style.borderStyle = "solid";
```

```
topicDiv.style.borderColor = "rgb(255, 255,  
255)";  
topicDiv.style.background = "rgb(0, 0, 0)";  
topicDiv.style.fontFamily = "tahoma";  
topicDiv.style.fontSize = "20px";  
topicDiv.style.fontWeight = "bold";  
topicDiv.style.color = "rgb(255, 255, 255)";  
topicDiv.style.textAlign = "center";  
ba(topicDiv);  
}
```

// Dedicated to God the Father

// All Rights Reserved Christopher Andrew
Topalian Copyright 2000-2024

// <https://github.com/ChristopherTopalian>

//
<https://github.com/ChristopherAndrewTopalia>
n

// makePeriodicElements.js

```
// shows the list of periodic elements in a div
function makePeriodicElements(whichArray,
xPos, yPos)
{
    let elementDiv = ce("div");
```

```
elementDiv.style.position = "absolute";
elementDiv.style.left = xPos + "px";
elementDiv.style.top = yPos + "px";
elementDiv.style.paddingLeft = 10 + "px";
elementDiv.style.paddingRight = 10 + "px";
elementDiv.style.paddingTop = 10 + "px";
elementDiv.style.paddingBottom = 14 +
"px";
elementDiv.style.zIndex = 2;
elementDiv.style.borderStyle = "solid";
elementDiv.style.borderColor = "rgb(255,
255, 255)";
elementDiv.style.borderRadius = "4px";
elementDiv.style.backgroundColor =
"rgb(0, 0, 0)";
elementDiv.style.fontFamily = "Tahoma";
elementDiv.style.fontWeight = "bold";
```

```
elementDiv.style.color = "rgb(255, 255,  
255)";  
elementDiv.style.textAlign = "right";  
elementDiv.id = "periodicTableElements";  
ba(elementDiv);  
  
let output = "";  
  
for (let x = 0; x < whichArray.length; x++)  
{  
    if (whichArray[x].tag ==  
    "chemicalElement")  
    {  
        output += whichArray[x].name + " ";  
        output += whichArray[x].number + " ";  
        output += '<a href = \'' +  
whichArray[x].url + '\'' target = "_blank"> Link  
</a>';
```

```
    output += '<br>';
}
}
elementDiv.innerHTML = output;
}
```

// Dedicated to God the Father

// All Rights Reserved Christopher Andrew
Topalian Copyright 2000-2024

// <https://github.com/ChristopherTopalian>

//
<https://github.com/ChristopherAndrewTopalia>
n

// makePeriodicElementsNameAsLink.js

// shows the list of periodic elements in a div
function
makePeriodicElementsNameAsLink(whichArr
ay, xPos, yPos)
{

```
let elementDiv = ce("div");
elementDiv.style.position = "absolute";
elementDiv.style.left = xPos + "px";
elementDiv.style.top = yPos + "px";
elementDiv.style.paddingLeft = 10 + "px";
elementDiv.style.paddingRight = 10 + "px";
elementDiv.style.paddingTop = 10 + "px";
elementDiv.style.paddingBottom = 14 +
"px";
elementDiv.style.zIndex = 2;
elementDiv.style.borderStyle = "solid";
elementDiv.style.borderColor = "rgb(255,
255, 255)";
elementDiv.style.borderRadius = "4px";
elementDiv.style.backgroundColor =
"rgb(0, 0, 0)";
elementDiv.style.fontFamily = "Tahoma";
elementDiv.style.fontSize = 20 + "px";
```

```
elementDiv.style.fontWeight = "bold";
elementDiv.style.color = "rgb(255, 255,
255)";
elementDiv.style.textAlign = "right";
elementDiv.id = "periodicTableElements";
ba(elementDiv);

let output = "";

for (let x = 0; x < whichArray.length; x++)
{
    if (whichArray[x].tag ==
"chemicalElement")
    {
        output += '<a href = \'' +
whichArray[x].url + '\' target = "_blank">
'+whichArray[x].name+'</a>';
    }
}
```

```
    output += ' ';  
  
    output += whichArray[x].number;  
  
    output += '<br>';  
}  
}  
elementDiv.innerHTML = output;  
}
```

// Dedicated to God the Father

// All Rights Reserved Christopher Andrew
Topalian Copyright 2000-2024

// <https://github.com/ChristopherTopalian>

//
<https://github.com/ChristopherAndrewTopalia>
n

// makePeriodicTable.js

```
function makePeriodicTable()  
{
```

```
    let periodicTable = ce('img');
```

```
    periodicTable.id = 'periodicTable';
```

```
periodicTable.src =  
'media/textures/periodicTableColorInverted.p  
ng';  
  
ba(periodicTable);  
}
```

// Dedicated to God the Father

// All Rights Reserved Christopher Andrew
Topalian Copyright 2000-2024

// https = //github.com/ChristopherTopalian

// https =
//github.com/ChristopherAndrewTopalian

// makePlayer.js

```
function makePlayer(whichPlayer)
{
    let player = ce('div');
    player.id = whichPlayer.id;
    player.style.position = 'absolute';
    player.style.left = whichPlayer.x;
```

```
player.style.top = whichPlayer.y;
player.style.width = whichPlayer.width;
player.style.height = whichPlayer.height;
player.style.borderRadius = '0px';
player.style.background =
whichPlayer.texture;
player.style.backgroundSize = 'cover';
player.style.zIndex = '10';
player.style.fontFamily = 'tahoma';
player.style.fontSize = '13px';
player.style.fontWeight = 'bold';
player.style.textAlign = 'center';
player.style.wordWrap = 'break-word';
ba(player);
}
```

// Dedicated to God the Father

// All Rights Reserved Christopher Andrew
Topalian Copyright 2000-2024

// <https://github.com/ChristopherTopalian>

//
<https://github.com/ChristopherAndrewTopalia>
n

// makeRowOfElements.js

```
// makes divs in a row
function makeRowOfElements(whichArray,
xPos, yPos, sizeX)
{
    // correct x position by subtracting sizeX
```

```
xPos -= sizeX;  
  
for (let x = 0; x < whichArray.length; x++)  
{  
    // add sizeX to xPos and add an offset  
    xPos = xPos + sizeX + 75;  
  
    //--//
```

```
let theDiv = ce("div");  
theDiv.style.position = "absolute";  
theDiv.style.left = xPos;  
theDiv.style.top = yPos;  
theDiv.style.width = sizeX;  
theDiv.style.zIndex = "2";  
theDiv.style.padding = "25px";  
theDiv.style.borderStyle = "solid";  
theDiv.style.borderWidth = "2px";
```

```
theDiv.style.borderColor = "rgb(255, 255,  
255)";  
theDiv.style.background = "rgb(0, 0, 0)";  
theDiv.style.fontFamily = "tahoma";  
theDiv.style.fontWeight = "bold";  
theDiv.style.fontSize = "25px";  
theDiv.style.color = "rgb(255, 255, 255)";  
theDiv.style.textAlign = "center";  
theDiv.style.lineHeight = 30 + "px";  
theDiv.innerHTML = whichArray[x].name  
+ "<br>";  
theDiv.innerHTML +=  
whichArray[x].number + "<br>";  
theDiv.innerHTML += '<a href = \'' +  
whichArray[x].url + '\' target = "_blank"> Link  
</a>';  
ba(theDiv);  
}
```

}

// Dedicated to God the Father

// All Rights Reserved Christopher Andrew
Topalian Copyright 2000-2024

// <https://github.com/ChristopherTopalian>

//
<https://github.com/ChristopherAndrewTopalia>
n

// makeRowOfWebsitesPdfsOrVideos.js

// makes iframes in a row to add content to
our level

function

**makeRowOfWebsitesPdfsOrVideos(whichArr
ay, xPos, yPos, sizeX, sizeY)**

```
{  
    // correct x position by subtracting the  
    sizeX  
    xPos -= sizeX;  
  
    for (let x = 0; x < whichArray.length; x++)  
    {  
        // add sizeX to xPos and add an offset  
        xPos = xPos + sizeX + 40;  
  
        let theFrame = ce("iframe");  
        theFrame.src = whichArray[x].url;  
        theFrame.style.position = "absolute";  
        theFrame.style.left = xPos;  
        theFrame.style.top = yPos;  
        theFrame.style.width = sizeX;  
        theFrame.style.height = sizeY;  
        theFrame.style.zIndex = "2";
```

```
theFrame.style.background = "rgb(0, 0,  
0)";  
theFrame.style.fontFamily = "tahoma";  
theFrame.style.fontWeight = "bold";  
theFrame.style.fontSize = "20px";  
theFrame.style.color = "rgb(255, 255,  
255)";  
theFrame.style.textAlign = "center";  
ba(theFrame);  
}  
}
```

// Dedicated to God the Father

// All Rights Reserved Christopher Andrew
Topalian Copyright 2000-2024

// <https://github.com/ChristopherTopalian>

//
<https://github.com/ChristopherAndrewTopalia>
n

// makeTiles.js

// make Tiles by specifying the amount of
rows and columns for a Background
function makeTiles(theTexture, rows,
columns, xPos, yPos, sizeX, sizeY)
{

```
let startingPosition = xPos;  
  
// rows  
for (let j = 0; j < rows; j++)  
{  
    // correct x position by subtracting the  
    sizeX  
    xPos -= sizeX;  
  
    // columns  
    for (let x = 0; x < columns; x++)  
    {  
        // add the sizeX to xPos  
        xPos = xPos + sizeX;  
  
        let makeTile = ce("img");  
        makeTile.src = theTexture;  
        makeTile.style.position = "absolute";
```

```
makeTile.style.left = xPos;  
makeTile.style.top = yPos;  
makeTile.style.width = sizeX;  
makeTile.style.height = sizeY;  
makeTile.style.color = "rgb(255, 255,  
255)";  
makeTile.style.zIndex = "1";  
makeTile.style.fontFamily = "tahoma";  
makeTile.style.fontWeight = "bold";  
makeTile.style.fontSize = "25px";  
makeTile.style.textAlign = "center";  
makeTile.style.background = "rgb(0, 0,  
0)";  
ba(makeTile);  
}  
xPos = startingPosition;  
  
yPos += sizeY;
```

```
}
```

```
}
```

// Dedicated to God the Father

// All Rights Reserved Christopher Andrew
Topalian Copyright 2000-2024

// <https://github.com/ChristopherTopalian>

//
<https://github.com/ChristopherAndrewTopalia>
n

// makeTilesOneRow.js

// make Tiles in one row for a Background
function makeTilesOneRow(theTexture,
howManyX, xPos, yPos, sizeX, sizeY)
{

```
// correct x position by subtracting the  
sizeX
```

```
xPos -= sizeX;
```

```
for (let x = 0; x < howManyX; x++)
```

```
{
```

```
// add the sizeX to xPos
```

```
xPos = xPos + sizeX;
```

```
let makeTile = ce("img");
```

```
makeTile.src = theTexture;
```

```
makeTile.style.position = "absolute";
```

```
makeTile.style.left = xPos;
```

```
makeTile.style.top = yPos;
```

```
makeTile.style.width = sizeX;
```

```
makeTile.style.height = sizeY;
```

```
makeTile.style.background = "rgb(0, 0,
```

```
0)";
```

```
makeTile.style.zIndex = "1";
makeTile.style.fontFamily = "tahoma";
makeTile.style.fontWeight = "bold";
makeTile.style.fontSize = "25px";
makeTile.style.color = "rgb(255, 255,
255)";
makeTile.style.textAlign = "center";
ba(makeTile);
}
}
```

// Dedicated to God the Father

// All Rights Reserved Christopher Andrew
Topalian Copyright 2000-2024

// <https://github.com/ChristopherTopalian>

//
<https://github.com/ChristopherAndrewTopalia>
n

// makeTreasure.js

```
function makeTreasure(whichId, whichSrc,  
whichX, whichY)  
{  
    let treasure = ce('img');  
    treasure.id = whichId;
```

```
treasure.title = whichId;  
treasure.src = whichSrc;  
treasure.style.position = 'absolute';  
treasure.style.left = whichX + 'px';  
treasure.style.top = whichY + 'px';  
treasure.style.zIndex = 3000;  
ba(treasure);  
}
```

// Dedicated to God the Father

// All Rights Reserved Christopher Andrew
Topalian Copyright 2000-2024

// <https://github.com/ChristopherTopalian>

//
<https://github.com/ChristopherAndrewTopalia>
n

// makeXPosDiv2.js

```
function makeXPosDiv()
{
    let xPosDiv = ce('div');
    xPosDiv.textContent = 'x';
    xPosDiv.id = 'xPosDiv';
```

```
xPosDiv.className = 'textStyle001';
xPosDiv.title = 'makeXPosDiv()';
xPosDiv.style.position = 'fixed';
xPosDiv.style.right = '0px';
xPosDiv.style.bottom = '375px';
xPosDiv.style.zIndex = 1000;
xPosDiv.style.color = 'rgb(255, 150, 150)';
xPosDiv.style.textShadow = '1px 1px
#000000';
ba(xPosDiv);
}
```

// Dedicated to God the Father

// All Rights Reserved Christopher Andrew
Topalian Copyright 2000-2024

// <https://github.com/ChristopherTopalian>

//
<https://github.com/ChristopherAndrewTopalia>
n

// makeYPosDiv.js

```
function makeYPosDiv()
{
    let yPosDiv = ce('div');
    yPosDiv.textContent = 'y';
    yPosDiv.id = 'yPosDiv';
```

```
yPosDiv.className = 'textStyle001';
yPosDiv.title = 'makeYPosDiv()';
yPosDiv.style.position = 'fixed';
yPosDiv.style.right = '0px';
yPosDiv.style.bottom = '350px';
yPosDiv.style.zIndex = 1000;
yPosDiv.style.color = 'rgb(30, 255, 30)';
yPosDiv.style.textShadow = '1px 1px
#000000';
ba(yPosDiv);
}
```

// Dedicated to God the Father

// All Rights Reserved Christopher Andrew
Topalian Copyright 2000-2024

// <https://github.com/ChristopherTopalian>

//
<https://github.com/ChristopherAndrewTopalia>
n

// hideInstructions.js

```
function hideInstructions()
{
    ge('instructionsMainDiv').style.display =
'none';
}
```

// Dedicated to God the Father

// All Rights Reserved Christopher Andrew
Topalian Copyright 2000-2024

// <https://github.com/ChristopherTopalian>

//
<https://github.com/ChristopherAndrewTopalia>
n

// makeInstructions.js

```
function makeInstructions()
{
    let instructionsMainDiv = ce('div');
    instructionsMainDiv.id =
    'instructionsMainDiv';
```

```
ba(instructionsMainDiv);
```

```
//-//
```

```
let detailsInstructions = ce('details');
detailsInstructions.className =
'Instructions';
detailsInstructions.style.position = 'fixed';
detailsInstructions.style.top = 1;
detailsInstructions.style.left = 80;
detailsInstructions.style.maxWidth =
'275px';
detailsInstructions.style.maxHeight =
'125px';

instructionsMainDiv.append(detailsInstructions);
```

//-//

```
let summaryInstructions = ce('summary');
summaryInstructions.textContent =
'Instructions';
summaryInstructions.style.color = 'rgb(50,
50, 50)';

detailsInstructions.append(summaryInstructions);
```

//-//

```
let textInstructions = ce('div');
textInstructions.textContent =
'Instructions';
textInstructions.id = 'textInstructions';
```

```
textInstructions.className =  
'textStyle001';  
textInstructions.innerHTML = `  
W is up, S is down, <br>  
A is left, D is Right <br>  
~ 1 2 3 4 5 6 7 8 9 0 for speed <br>  
Hold Shift to Boost Speed <br>  
Letter V to Start Song <br>  
Letter B to Stop Song <br>  
Letter N player is see-through <br>  
Letter M player is solid  
`;
```

```
detailsInstructions.append(textInstructions);  
}
```

// Dedicated to God the Father

// All Rights Reserved Christopher Andrew
Topalian Copyright 2000-2024

// <https://github.com/ChristopherTopalian>

//
<https://github.com/ChristopherAndrewTopalia>
n

// showInstructions.js

```
function showInstructions()
{
    ge('instructionsMainDiv').style.display =
'block';
}
```

// Dedicated to God the Father

// All Rights Reserved Christopher Andrew
Topalian Copyright 2000-2024

// <https://github.com/ChristopherTopalian>

//
<https://github.com/ChristopherAndrewTopalia>
n

// makeMinimap.js

```
function makeMinimap()
{
    let minimapDiv = ce('div');
    minimapDiv.id = 'minimap';
    minimapDiv.className = 'minimapDiv';
```

```
minimapDiv.style.position = 'fixed';
minimapDiv.style.left = '5px';
minimapDiv.style.bottom = '4px';
minimapDiv.style.width = '150px';
minimapDiv.style.height = '150px';
minimapDiv.style.backgroundColor =
'rgba(0, 0, 0, 0.5)';
minimapDiv.style.border = 'solid';
minimapDiv.style.borderWidth = '1px';
minimapDiv.style.borderRadius = '1px';
minimapDiv.style.zIndex = 2000;
minimapDiv.style.overflow = 'hidden';
minimapDiv.style.cursor = 'pointer';
ba(minimapDiv);
}
```

// Dedicated to God the Father

// All Rights Reserved Christopher Andrew
Topalian Copyright 2000-2024

// <https://github.com/ChristopherTopalian>

//
<https://github.com/ChristopherAndrewTopalia>
n

// makeMinimapPlayer.js

```
function makeMinimapPlayer()
{
    let minimapPlayer = ce('div');
    minimapPlayer.innerHTML = '1';
    minimapPlayer.id = 'minimapPlayer';
```

```
minimapPlayer.title = ourPlayer.id;  
minimapPlayer.style.position = 'absolute';  
minimapPlayer.style.width = '8px';  
minimapPlayer.style.height = '8px';  
minimapPlayer.style.backgroundColor =  
'rgb(0, 255, 255)';  
minimapPlayer.style.zIndex = 5;  
minimapPlayer.style.borderRadius = '50%';  
minimapPlayer.style.fontSize = '8px';  
minimapPlayer.style.color = 'rgb(0, 0, 0)';  
minimapPlayer.style.fontWeight = 'bold';  
minimapPlayer.style.textAlign = 'center';  
minimapPlayer.style.lineheight = '0px';  
  
if (ge('minimapPlayer'))  
{  
    ge('minimapPlayer').remove();  
}
```

```
// append minimap player to minimap  
ge('minimap').append(minimapPlayer);  
}
```

// Dedicated to God the Father

// All Rights Reserved Christopher Andrew
Topalian Copyright 2000-2024

// <https://github.com/ChristopherTopalian>

//
<https://github.com/ChristopherAndrewTopalia>
n

// makeMinimapTreasure.js

```
function makeMinimapTreasure(whichId,  
whichHTML, whichBgColor, whichColor)  
{  
    let minimapTreasure = ce('div');
```

```
minimapTreasure.innerHTML =  
whichHTML;  
minimapTreasure.id = whichId;  
minimapTreasure.title = whichHTML;  
minimapTreasure.style.position =  
'absolute';  
minimapTreasure.style.width = '8px';  
minimapTreasure.style.height = '8px';  
minimapTreasure.style.backgroundColor =  
whichBgColor;  
minimapTreasure.style.zIndex = 3;  
minimapTreasure.style.borderRadius =  
'50%';  
minimapTreasure.style.fontSize = 8 + 'px';  
minimapTreasure.style.fontWeight = 'bold';  
minimapTreasure.style.color = whichColor;  
minimapTreasure.style.textAlign = 'center';
```

```
if (ge(whichId))
{
    ge(whichId).remove();
}

ge('minimap').append(minimapTreasure);
}
```

// Dedicated to God the Father

// All Rights Reserved Christopher Andrew
Topalian Copyright 2000-2024

// <https://github.com/ChristopherTopalian>

//
<https://github.com/ChristopherAndrewTopalia>
n

// updateMinimapPlayer.js

function updateMinimapPlayer()
{

// calculate minimap pos for ourPlayer.id
// convert to percentage

```
let minimapX = (ourPlayer.x /  
backgroundSizeX) * 100;  
  
// convert to percentage  
let minimapY = (ourPlayer.y /  
backgroundSizeY) * 100;  
  
// set pos of minimap player  
ge('minimapPlayer').style.left = `'$  
{minimapX}%`;  
ge('minimapPlayer').style.top = `'$  
{minimapY}%`;  
}
```

// Dedicated to God the Father

// All Rights Reserved Christopher Andrew
Topalian Copyright 2000-2024

// <https://github.com/ChristopherTopalian>

//
<https://github.com/ChristopherAndrewTopalia>
n

// updateMinimapTreasure.js

function updateMinimapTreasure(whichId,
targetElementId)

{

```
let treasureLeft =  
parseFloat(getComputedStyle(ge(targetEleme  
ntId)).left);
```

```
let treasureTop =  
parseFloat(getComputedStyle(ge(targetEleme  
ntId)).top);
```

//-

```
// calculate minimap pos for treasure  
// convert to percentage  
let minimapX = (treasureLeft /  
backgroundSizeX) * 100;
```

```
let minimapY = (treasureTop /  
backgroundSizeY) * 100;
```

//-//

```
// set pos on minimap
ge(whichId).style.left = `${minimapX}%`;
ge(whichId).style.top = `${minimapY}%`;
}
```

// Dedicated to God the Father

// All Rights Reserved Christopher Andrew
Topalian Copyright 2000-2024

// <https://github.com/ChristopherTopalian>

//
<https://github.com/ChristopherAndrewTopalia>
n

// makeSpeedometer.js

```
function makeSpeedometer()
{
    let speedometerDiv = ce('div');
    speedometerDiv.textContent = 'Speed';
    speedometerDiv.id = 'speedometerDiv';
```

```
speedometerDiv.className =  
'textStyle001';  
    speedometerDiv.title =  
'makeSpeedometer()';  
    speedometerDiv.style.position = 'fixed';  
    speedometerDiv.style.right = '0px';  
    speedometerDiv.style.bottom = '150px';  
    speedometerDiv.style.zIndex = 1000;  
    speedometerDiv.style.textAlign = 'right';  
    speedometerDiv.style.textShadow = '1px  
1px #000000';  
    ba(speedometerDiv);  
}
```

// Dedicated to God the Father

// All Rights Reserved Christopher Andrew
Topalian Copyright 2000-2024

// <https://github.com/ChristopherTopalian>

//
<https://github.com/ChristopherAndrewTopalia>
n

// updateSpeedometer.js

```
function updateSpeedometer()
{
    // calculate player speed based on change
    // in pos
    let deltaX = ourPlayer.x - playerPrevX;
```

```
let deltaY = ourPlayer.y - playerPrevY;  
  
let distance = Math.sqrt(deltaX ** 2 + deltaY  
** 2);  
  
// if game is running at 60fps  
playerSpeed = distance * 60;  
  
return playerSpeed;  
}
```

// Dedicated to God the Father

// All Rights Reserved Christopher Andrew
Topalian Copyright 2000-2024

// <https://github.com/ChristopherTopalian>

//
<https://github.com/ChristopherAndrewTopalia>
n

// menuLeft.js

```
let menuFunctions =  
[  
  {  
    name: `showInstructions()`,  
    fullName: "Show Instructions",
```

```
    description: "Show Instructions"  
},  
  
{  
  name: `hideInstructions()  
fullName: "Hide Instructions",  
description: "Hide Instructions"  
},  
  
{  
  name: `audioPlay('Guerilla_Tactics',  
1.0)  
fullName: "Play Song",  
description: "Play Default Song"  
},  
  
{  
  name: `audioPause('Guerilla_Tactics')`  
},
```

```
fullName: "Pause Song",
description: "Pause Default Song"
}
];
```

```
function menuLeft()
{
  if (ge("menuLeftDiv"))
  {
    ge("menuLeftDiv").remove();
  }
}
```

```
let divHeight = 90;
```

```
let menuLeftDiv = ce("div");
menuLeftDiv.id = "menuLeftDiv";
menuLeftDiv.style.position = "fixed";
menuLeftDiv.style.left = 0 + "px";
```

```
menuLeftDiv.style.top = 20 + "px";
menuLeftDiv.style.margin = '2px';
menuLeftDiv.style.zIndex = 1000;
ba(menuLeftDiv);
```

//-//

```
let minimizeButton = ce("button");
minimizeButton.innerHTML = "_";
minimizeButton.className =
"buttonStyle001";
minimizeButton.style.position =
"absolute";
minimizeButton.style.left = 0 + "px";
minimizeButton.style.top = -22 + "px";
minimizeButton.onclick = function()
{
    audioPlay("sfx_warp_001", 1.0);
```

```
// minimize height to show only _ and □  
ge("menuLeftSubDiv").style.height = 0 +  
"px";  
  
ge("menuLeftSubDiv").style.display =  
'none';  
};  
  
menuLeftDiv.append(minimizeButton);  
  
//--//  
  
let maximizeButton = ce("button");  
maximizeButton.innerHTML = "□";  
maximizeButton.className =  
"buttonStyle001";
```

```
maximizeButton.style.position =
"absolute";
maximizeButton.style.left = 30 + "px";
maximizeButton.style.top = -22 + "px";
maximizeButton.onclick = function()
{
    audioPlay("sfx_warp_001", 1.0);

    ge("menuLeftSubDiv").style.height =
divHeight + "px";

    ge("menuLeftSubDiv").style.display =
'block';
};
menuLeftDiv.append(maximizeButton);

//--
```

```
let menuLeftSubDiv = ce("div");
menuLeftSubDiv.id = "menuLeftSubDiv";
menuLeftSubDiv.style.position = "fixed";
menuLeftSubDiv.style.left = 0 + "px";
menuLeftSubDiv.style.top = 35 + "px";
menuLeftSubDiv.style.margin = 2 + "px";
menuLeftSubDiv.style.width = '190px';
menuLeftSubDiv.style.height = divHeight +
"px";
menuLeftSubDiv.style.overflowY = "auto";
menuLeftSubDiv.style.resize = "both";
menuLeftSubDiv.style.whiteSpace =
'nowrap';
menuLeftSubDiv.style.zIndex = 1001;
menuLeftSubDiv.style.borderStyle = 'solid';
menuLeftSubDiv.style.borderWidth = '1px';
menuLeftSubDiv.style.borderColor =
'rbg(255, 255, 255)';
```

```
menuLeftSubDiv.style.borderRadius =  
'8px';  
menuLeftSubDiv.style.backgroundColor =  
'rgba(0, 0, 0, 0.4)';  
ba(menuLeftSubDiv);
```

//-//

```
for (let x = 0; x < menuFunctions.length; x+)  
{  
    let theButton = ce("button");  
    theButton.innerHTML =  
menuFunctions[x].description;  
    theButton.className =  
"buttonStyle001";  
    theButton.title =  
menuFunctions[x].fullName;
```

```
theButton.style.width = 'auto';
theButton.style.fontSize = '15px';
theButton.style.display = "flex";
theButton.style.flexDirection = "row";
theButton.onclick = function()
{
    audioPlay("sfx_blip_001", 1.0);

    eval(""+menuFunctions[x].name+"");
};

console.log(""+menuFunctions[x].name+"");
};
menuLeftSubDiv.append(theButton);

//--
```

```
    ge("menuLeftSubDiv").style.height = 0 +  
    "px";  
  
    ge("menuLeftSubDiv").style.display =  
    'none';  
}  
}
```

// Dedicated to God the Father

// All Rights Reserved Christopher Andrew
Topalian Copyright 2000-2024

// <https://github.com/ChristopherTopalian>

//
<https://github.com/ChristopherAndrewTopalia>
n

// menuRight.js

```
function menuRight()
{
    if (ge("menuRight"))
    {
        ge("menuRight").remove();
```

```
}
```

```
let divHeight = 100;
```

```
let menuRightDiv = ce("div");
menuRightDiv.id = 'menuRight';
menuRightDiv.style.position = "fixed";
menuRightDiv.style.right = 0 + "px";
menuRightDiv.style.top = 17 +
"px";menuRightDiv.style.margin = '2px';
menuRightDiv.style.backgroundColor =
"rgba(0, 0, 0, 0.5)";
menuRightDiv.style.zIndex = 1000;
ba(menuRightDiv);
```

```
//-//
```

```
let closeButton = ce("button");
```

```
closeButton.innerHTML = "_";
closeButton.className =
"buttonStyle001";
closeButton.style.position = "absolute";
closeButton.style.right = 30 + "px";
closeButton.style.top = -22 + "px";
closeButton.onclick = function()
{
    audioPlay("sfx_warp_001", 1.0);

    // minimize height to show only _ and □
    ge("menuRightSubDiv").style.height = 0
+ "px";

    ge("menuRightSubDiv").style.display =
'none';
};
menuRightDiv.append(closeButton);
```

//-//

```
let maximizeButton = ce("button");
maximizeButton.innerHTML = "□";
maximizeButton.className =
"buttonStyle001";
maximizeButton.style.position =
"absolute";
maximizeButton.style.right = 0 + "px";
maximizeButton.style.top = -22 + "px";
maximizeButton.onclick = function()
{
    audioPlay("sfx_warp_001", 1.0);

    ge("menuRightSubDiv").style.height =
divHeight + "px";
```

```
ge("menuRightSubDiv").style.display =  
'block';  
};  
menuRightDiv.append(maximizeButton);  
  
//--
```

```
let menuRightSubDiv = ce("div");  
menuRightSubDiv.id =  
"menuRightSubDiv";  
menuRightSubDiv.style.position = "fixed";  
menuRightSubDiv.style.right = 0 + "px";  
menuRightSubDiv.style.top = 30 + "px";  
menuRightSubDiv.style.margin = 2 + "px";  
menuRightSubDiv.style.width = '190px';  
menuRightSubDiv.style.height = divHeight  
+ "px";  
menuRightSubDiv.style.zIndex = 1001;
```

```
menuRightSubDiv.style.borderWidth =  
'solid';  
menuRightSubDiv.style.borderColor =  
'rgb(255, 255, 255)';  
menuRightSubDiv.style.borderRadius =  
'8px';  
menuRightSubDiv.style.backgroundColor =  
"rgba(0, 0, 0, 0.4)";  
menuRightSubDiv.style.overflowY =  
"auto";  
menuRightSubDiv.style.resize = "both";  
menuRightSubDiv.style.whiteSpace =  
'nowrap';  
ba(menuRightSubDiv);
```

//-//

```
// countButton
let countButton = ce('button');
countButton.textContent = 'How many
objects';
countButton.id = 'countButton';
countButton.className = 'buttonStyle001';
countButton.onclick = function()
{
    audioPlay("sfx_blip_001", 1.0);
    alert(countAllElements());
};
menuRightSubDiv.append(countButton);

//-
ge("menuRightSubDiv").style.height = 0 +
"px";
```

```
ge("menuRightSubDiv").style.display =  
'none';  
}
```

// Dedicated to God the Father

// All Rights Reserved Christopher Andrew
Topalian Copyright 2000-2024

// <https://github.com/ChristopherTopalian>

//
<https://github.com/ChristopherAndrewTopalia>
n

// calculateDistanceTraveled.js

**function calculateDistanceTraveled()
{**

**// calculate distance moved in current
frame**

```
let distanceX = Math.abs(ourPlayer.x -  
playerPrevX);
```

```
let distanceY = Math.abs(ourPlayer.y -  
playerPrevY);
```

```
// player previous pos  
playerPrevX = ourPlayer.x;  
playerPrevY = ourPlayer.y;  
  
// update total distance traveled  
distanceTraveled += Math.sqrt(distanceX **  
2 + distanceY ** 2);  
}
```

// Dedicated to God the Father

// All Rights Reserved Christopher Andrew
Topalian Copyright 2000-2024

// <https://github.com/ChristopherTopalian>

//
<https://github.com/ChristopherAndrewTopalia>
n

// checkPlayerMovement.js

```
function checkPlayerMovement()
{
    // store movement state
    let isPlayerMovingLocal = ourPlayer.x !==
playerPrevX || ourPlayer.y !== playerPrevY;
```

```
// return movement state  
return isPlayerMovingLocal;  
}
```

// Dedicated to God the Father

// All Rights Reserved Christopher Andrew
Topalian Copyright 2000-2024

// <https://github.com/ChristopherTopalian>

//
<https://github.com/ChristopherAndrewTopalia>
n

// isPlayerMoving.js

```
function isPlayerMoving()
{
    // is player moving
    let movingOrNot =
        checkPlayerMovement();
```

```
if (movingOrNot)
{
    if (ge('isMovingDiv'))
    {
        calculateDistanceTraveled();

        ge('isMovingDiv').innerHTML =
'Motion';

        ge('isMovingDiv').style.color = 'rgb(0,
255, 255)';

    if (ge('distanceTraveledDiv'))
    {

        ge('distanceTraveledDiv').innerHTML =
```

```
'Distance' + '<br>' +  
distanceTraveled.toFixed(0);  
    }  
}  
}  
else  
{  
    if (ge('isMovingDiv'))  
{  
    ge('isMovingDiv').innerHTML =  
'Motion';  
  
    ge('isMovingDiv').style.color =  
'rgb(255, 255, 255)';  
}  
}  
}
```

// Dedicated to God the Father

// All Rights Reserved Christopher Andrew
Topalian Copyright 2000-2024

// <https://github.com/ChristopherTopalian>

//
<https://github.com/ChristopherAndrewTopalia>
n

// getPos.js

```
function getPos(theId)
{
    theId = ge(theId);
```

```
let theRect =  
theId.getBoundingClientRect();
```

```
let theRectY =  
theId.getBoundingClientRect().top +  
window.scrollY;
```

```
let posXY =  
{  
    x: parseInt(theRect.x) +  
parseInt(theRect.width) / 2 + window.scrollX,  
    y: parseInt(theRectY) +  
parseInt(theRect.height) / 2  
};  
  
return posXY;  
}
```

// Dedicated to God the Father

// All Rights Reserved Christopher Andrew
Topalian Copyright 2000-2024

// <https://github.com/ChristopherTopalian>

//
<https://github.com/ChristopherAndrewTopalia>
n

// getSize.js

// get Size of the specified element by id
function getSize(theld)
{
 theld = ge(theld);

```
let theRect =  
theId.getBoundingClientRect();
```

```
let sizeXY =  
{  
    x: parseInt(theRect.width),  
    y: parseInt(theRect.height)  
};  
  
return sizeXY;  
}
```

// Dedicated to God the Father

// All Rights Reserved Christopher Andrew
Topalian Copyright 2000-2024

// <https://github.com/ChristopherTopalian>

//
<https://github.com/ChristopherAndrewTopalia>
n

// mousePos.js

let mouseClick = 0;

function mousePos()
{
 mouseClick += 1;

```
let mouseX = event.pageX;  
let mouseY = event.pageY;  
  
let mousePos = "Mouse" + "<br>" + "X " +  
mouseX + "<br>" + "Y " + mouseY;  
  
if (ge("infoMousePositionId"))  
{  
    ge("infoMousePositionId").innerHTML =  
mousePos;  
}  
  
return mousePos;  
}
```

// Dedicated to God the Father

// All Rights Reserved Christopher Andrew
Topalian Copyright 2000-2024

// <https://github.com/ChristopherTopalian>

//
<https://github.com/ChristopherAndrewTopalia>
n

// audioPause.js

```
function audioPause(whichAudioId)  
{
```

```
    let theAudio = ge(whichAudioId);
```

```
    theAudio.pause();
```

```
ge("infoSongStatusId").innerHTML = "Song  
Stopped";  
}
```

// Dedicated to God the Father

// All Rights Reserved Christopher Andrew
Topalian Copyright 2000-2024

// <https://github.com/ChristopherTopalian>

//
<https://github.com/ChristopherAndrewTopalia>
n

// audioPlay.js

```
function audioPlay(whichAudioId, theVolume)
{
    let theAudio = ge(whichAudioId);
    theAudio.play();
```

```
theAudio.volume = theVolume;  
  
if (ge("infoSongStatusDiv"))  
{  
    ge("infoSongStatusDiv").innerHTML =  
whichAudiod;  
}  
}
```

// Dedicated to God the Father

// All Rights Reserved Christopher Andrew
Topalian Copyright 2000-2024

// <https://github.com/ChristopherTopalian>

//
<https://github.com/ChristopherAndrewTopalia>
n

// loadSongs.js

```
let songsArray =  
[  
  {  
    name: "Lonely Mountain",  
    theld: "Lonely_Mountain",
```

```
    theLocation:  
    "./media/songs/Lonely_Mountain.mp3",  
    theURL:  
    "https://collegeofscripting.weebly.com/uploads/6/4/4/8/64482293/lonely_mountain.mp3"  
,  
{  
    name: "Guerilla_Tactics",  
    theID: "Guerilla_Tactics",  
    theLocation:  
    "./media/songs/Guerilla_Tactics.mp3",  
    theURL: ""  
,  
{  
    name: "Action_Strike",  
    theID: "Action_Strike",  
    theLocation:  
    "./media/songs/Action_Strike.mp3",
```

```
theURL:  
"https://collegeofscripting.weebly.com/uploads/6/4/4/8/64482293/action_strike.mp3"  
}  
];
```

```
function loadSongs()  
{  
let theX = 20;  
let theY = 20;  
  
for (let x = 0; x < songsArray.length; x++)  
{  
let ourSound =  
document.createElement("audio");  
ourSound.style.position = "absolute";  
ourSound.style.left = theX + 'px';  
ourSound.style.top = theY +'px';
```

```
ourSound.setAttribute('type',
'audio/mpeg');
ourSound.setAttribute('controls',
'controls');
ourSound.setAttribute('loop', 'loop');
ourSound.setAttribute('preload',
'preload');

if (online == true)
{
    ourSound.setAttribute('src',
songsArray[x].theURL);
}
else
{
    // if the location of the file is in our
    folder, then we use theLocation as the source
}
```

```
    ourSound.setAttribute('src',
songsArray[x].theLocation);
}

ourSound.setAttribute('id',
songsArray[x].theld);

// do not show the element on the page
ourSound.style.display = 'none';
ba(ourSound);
}

}
```

// Dedicated to God the Father

// All Rights Reserved Christopher Andrew
Topalian Copyright 2000-2024

// <https://github.com/ChristopherTopalian>

//
<https://github.com/ChristopherAndrewTopalia>
n

// soundLoad.js

let online = false;

let soundsArray =
[
 {

```
name: "Blip",
theID: "sfx_blip_001",
theLocation:
"./media/sounds/sfx_blip_001.mp4",
theURL: "https://64482293-
555277121853380144.preview.editmysite.com
/uploads/6/4/4/8/64482293/blip.mp4"
},
{
  name: "Crystal Bell",
  theID: "sfx_crystal_bell_001",
  theLocation:
"./media/sounds/sfx_crystal_bell_001.mp4",
  theURL: "https://64482293-
555277121853380144.preview.editmysite.com
/uploads/6/4/4/8/64482293/pop_sound.mp4"
},
```

```
name: "Warp",
theID: "sfx_warp_001",
theLocation:
"./media/sounds/sfx_warp_001.mp4",
theURL:
"https://collegeofscripting.weebly.com/uploads/6/4/4/8/64482293/blipphaser.mp3"
},
{
  name: "Zap",
  theID: "sfx_zap_001",
  theLocation:
"./media/sounds/sfx_zap_001.mp4",
  theURL:
"https://collegeofscripting.weebly.com/uploads/6/4/4/8/64482293/blipdeep.mp3"
}
];
```

```
function loadSounds()
{
    let theX = 20;
    let theY = 20;

    for (let x = 0; x < soundsArray.length; x++)
    {
        let ourSound =
            document.createElement("audio");
        ourSound.style.position = "absolute";
        ourSound.style.left = theX + 'px';
        ourSound.style.top = theY +'px';
        ourSound.setAttribute('type',
        'audio/mpeg');
        ourSound.setAttribute('controls',
        'controls');
    }
}
```

```
ourSound.setAttribute('preload',
'preload');

if (online == true)
{
    ourSound.setAttribute('src',
soundsArray[x].theURL);
}

else
{
    // if the location of the file is in our
    // folder, then we use theLocation as the source
    ourSound.setAttribute('src',
soundsArray[x].theLocation);
}

ourSound.setAttribute('id',
soundsArray[x].theld);
```

```
// do not show the element on the page
ourSound.style.display = 'none';
ba(ourSound);
}
}
```

// Dedicated to God the Father

// All Rights Reserved Christopher Andrew
Topalian Copyright 2000-2024

// <https://github.com/ChristopherTopalian>

//
<https://github.com/ChristopherAndrewTopalia>
n

// whenLoaded.js

```
function whenLoaded()  
{
```

```
    makeDedication();
```

```
    loadSongs();
```

loadSounds();

makePlayer(ourPlayer);

makeBackground();

makePeriodicTable();

makeMousePosDiv();

makeInfoDiv();

makeCollisionDiv();

makeIsMovingDiv();

makeDistanceTraveledDiv();

```
makeSpeedometer();  
  
makeMinimap();  
  
makeMinimapPlayer();  
  
// whichId, whichSrc, whichX, whichY  
makeTreasure('treasureCopper',  
'media/textures/copper.jpg', 20000, 80);  
  
// whichId, whichSrc, whichX, whichY  
makeTreasure('treasureSilver',  
'media/textures/silver.jpg', 40000, 100);  
  
// whichId, whichHTML, whichBgColor,  
whichColor  
makeMinimapTreasure(  
'minimapCopperIcon',
```

```
'2',
'rgb(255, 255, 255)',
'rgb(0, 0, 0'
);

// whichId, whichHTML, whichBgColor,
whichColor
makeMinimapTreasure(
'minimapSilverIcon',
'3',
'rgb(255, 0, 255)',
'rgb(255, 255, 255)'
);

menuLeft();
menuRight();

makeElementInfoBox();
```

```
makeXPosDiv();
makeYPosDiv();

makeInstructions();

makeObjectsToScreen(dataFloors);

makeObjectsToScreen(dataObjectsPeriodicTable);
    makeObjectsToScreen(objectsTriggers);
    makeObjectsToScreen(objectsPart2);
    makeObjectsToScreen(objectsPart3);

// whichArray, xPos, yPos, sizeX, sizeY

makeRowOfWebsitesPdfsOrVideos(dataVideosChemistry, 2465, 670, 550, 550);
```

```
// shows periodic table on one row  
// xPos, yPos, sizeX, sizeY  
  
makeRowOfElements(dataPeriodicElements,  
2465, 388, 200);  
  
// below is a way to make a tiled  
background, one row at a time  
  
// theTexture, howManyX, xPos, yPos,  
sizeX, sizeY  
  
makeTilesOneRow("media/textures/asphalt.jp  
g", 10, 0, 0, 500, 500);
```

```
makeTilesOneRow("media/textures/asphalt.jpg", 10, 0, 500, 500, 500);
```

```
makeTilesOneRow("media/textures/asphalt.jpg", 10, 0, 1000, 500, 500);
```

// here is a much easier way to make a tiled background with rows and columns specified

```
makeTiles(
```

```
    "media/textures/wood_003.jpg", // url
```

```
    12,      // rows
```

```
    4,       // columns
```

```
    120,     // xPos
```

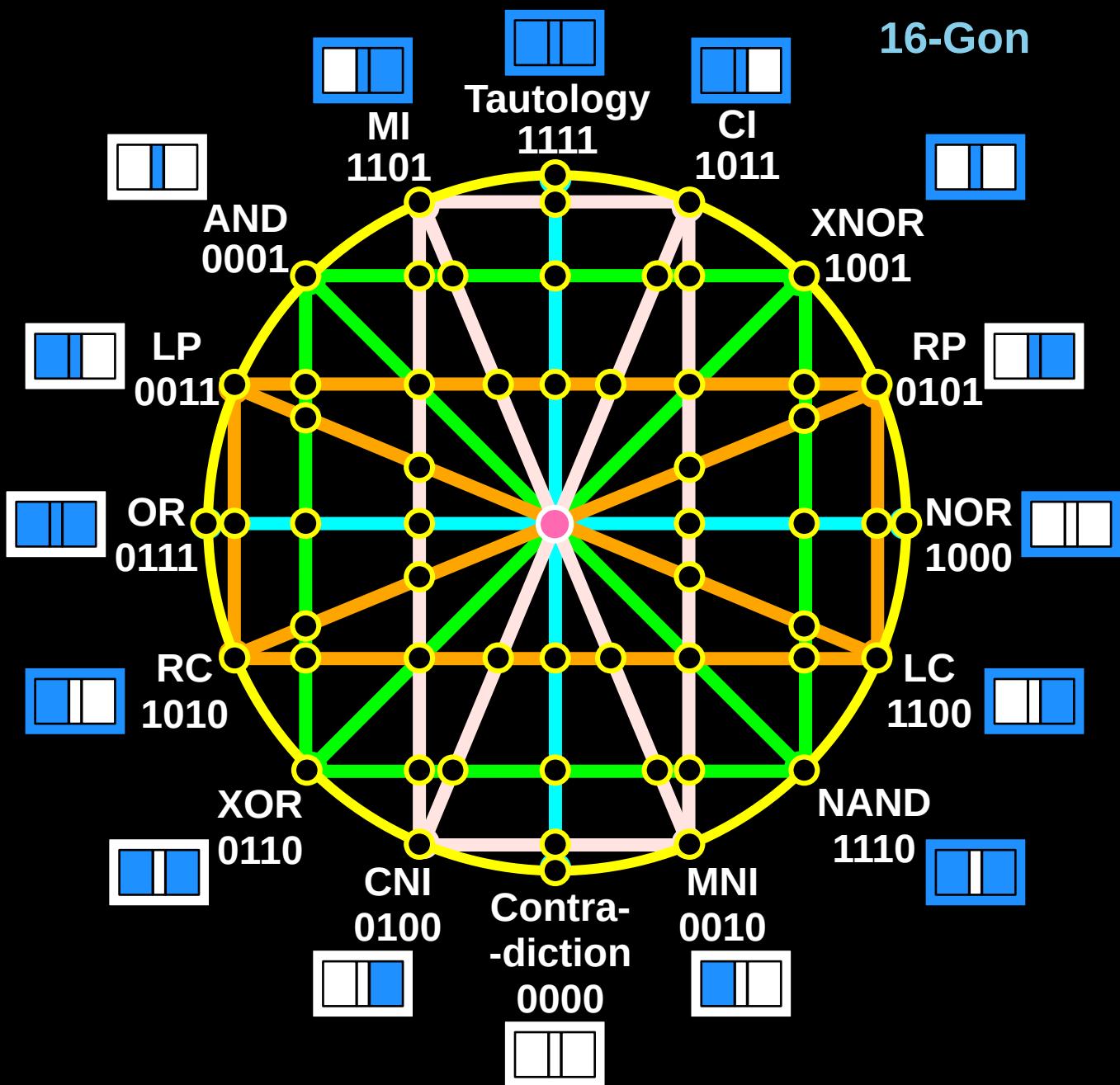
```
    2780,    // yPos
```

```
    500,     // sizeX
```

```
    500      // sizeY
```

```
});  
  
// listen for mouse clicks and get mouse  
pos  
document.addEventListener("click",  
mousePos, false);  
  
// listen for keydown events  
document.addEventListener("keydown",  
theControls, false);  
  
// listen for keyup events  
document.addEventListener("keyup",  
theControls, false);  
  
// start gameLoop  
gameLoop();  
}
```

True Artificial Intelligence System



For More Tutorials:

CollegeOfScripting.weebly.com

CollegeOfScripting.wordpress.com

Youtube.com/ScriptingCollege

Twitter.com/CollegeOfScript

GitHub.com/ChristopherTopalian

GitHub.com/ChristopherAndrewTopalian

Sites.google.com/view/CollegeOfScripting

Dedicated to God the Father

**This book is created by the
College of Scripting Music & Science.**

**Always remember, that each time you write a script
with a pencil and paper, it becomes imprinted so
deeply in memory that the material and methods are
learned extremely well.**

**When you Type the scripts, the same is true. The
more you type and write out the scripts by keyboard
or pencil and paper, the more you will learn
programming!**

**Write and Type every example that you find.
Keep all of your scripts organized.**

**Every script that you create increases your
programming abilities.**

**SEEING CODE, is one thing,
but WRITING CODE is another.**

Write it, Type it, Speak It, See It, Dream It.

CollegeOfScripting.weebly.com