File: index.html

<!--The html webpage of this game, the launch interface when the player click the webpage-->
<!--Initialize the game canvas and link to Main.js file to start the game-->

function init() {//initialize the game

File: Boss.js

```
/*
This function is the constructor of the object
Enemy1 is the object of the boss enemy tank, it is created and called by Scroller object
Once it is created, the will be added into an array list in the Scroller object. Sometimes, it
will also be called by Player objects and Bullet objects to make collision detections.
The purpose of this object is to create an enemy tank on the screen, it will shot the player
or be killed by the player. Parameter are rows and columns in the tile system
*/
function Boss(tileRow, tileCol, level)

/*
Update all movements of the enemy
Called by Scroller object, the parameter is the array of bricks from the Scroller
Determine the enemy's status and make movements including loading a bullet, moving
and rotating the turret.
It will call the object of CollisionDetector to check if the tank is able to move
The tank will randomly move in a certain direction for a random period of time.
Once the enemy finds the player, it will move toward to player
*/
Boss.prototype.update = function (bricks)

/*
Updating the body rotation of the tank
Called by update()
depends on the moving direction of the tank to change the radian of the image of the tank
*/
Boss.prototype.updateBodyRadian = function ()

/*
update the turret rotation of the tank
Called by update()
depends on the location of the player, change the rotation of the turret
*/
Boss.prototype.updateTurrentRotation = function ()
```

```
/*
fire a new bullet
Called by Scroller object
create a new Bullet object
*/
Boss.prototype.fireBullet = function ()

/*
Take hit by a bullet
Called by Scroller object, parameter bullet is a Bullet object
Calculate the damage from the bullet and deduct the health points
*/
Boss.prototype.takeBullet = function (bullet)

/*
Update the health bar
Called by takeBullet()
Calculate the new length of health bar and red bar and update them
*/
Boss.prototype.updateHealthBar = function ()
```

File: CollisionDetector.js

```
/*
This function is the constructor of the object
Collision Detector object
Created and called by the tanks' object
The object is able to make collision detection specifically for neared tiles of the tank
*/
function CollisionDetector(tileRow, tileCol, width, height, left, right, top, bottom)

/*
Check the left or right neared tiles
Called by update() of the tank
After it is called, it may return null means no bricks or the hitted brick
*/
CollisionDetector.prototype.checkX = function (speed, moveX)

/*
Check the top or bottom neared tiles
Called by update() of the tank
After it is called, it may return null means no bricks or the hitted brick
*/
CollisionDetector.prototype.checkY = function (speed, moveY)

/*
Use a nest for loop to check a short row or a column of tiles
Called by the CheckX() and CheckY()
The first collided brick will be returned
Null will be returned if no bricks are collided with the object
*/
CollisionDetector.prototype.checkTiles = function (tileRow, tileCol, row, col)
```

File: Concrete.js

```
/*
This function is the constructor of the object
Concrete object is object of one of bricks, this is the constructor function
Created by Scroller object depends on the number of row and column in the tiles matrix
It is an obstacle object that blocking tanks and bullets,it can not be destroyed by bullets
*/
function Concrete (x,y,tileRow,tileCol)

/*
Update the concrete brick
Called by Scroller object
update the position of the brick
*/
Concrete.prototype.update = function()
```

File: Drone.js

```
/*
This function is the constructor of the object
Drone is the object of the drone, it is created and called by Scroller object
Once it is created, the will be added into a array list in the Scroller object. Sometimes, it
will also be called by Player object and Bullet objects to make collision detections.
The purpose of this object is to create a drone on the screen, it will shot the enmies.
*/
function Drone()

/*
Update all movements of the drone
Called by Scroller object, the parameter is the array of bricks from Scroller
Determine the drone's status and make movements including loading a bullet, moving
and rotating the body.
The drone will either move toward an ememy or the player for a period of time.
Once the enemy find an enemy, it will move toward to it, once it fired all ammos,
it will fly back to player for reloading
*/
Drone.prototype.update = function (bricks)

/*
update the turret rotation of the tank
Called by update()
depends on the location of the player, change the rotation of the turret
*/
Drone.prototype.updateBodyRotation = function (target)

/*
fire a new bullet
Called by Scroller object
create a new Bullet object
*/
Drone.prototype.fireBullet = function ()

/*
Update the reloading time of machine gun
Called by update()
Update the the counter of reloading
*/
Drone.prototype.reloadMachineGun = function ()
```

```
/*
Search en enemy
Called by update()
find a target enemy from the array list of enemies
*/
Drone.prototype.searchEnemy = function (bricks)
```

File: DroneBullets.js

```
/*
This function is the constructor of the object
Bullet is the object for the drone's bullet, it is created and called by Drone objects
Once it is created, the will be added into a array list in the Scroller object. Sometimes, it
will also be called by Scroller to make collision detections.
The purpose of this object is to create an bullets on the screen, it may deduct the tanks's health
or even kill the tanks.
*/
function DroneBullet(drone, damage, pen)

/*
Updating all movement and status of the bullet
Called by Scroller object
If the bullet has exploded then increment the time counter and make rotations
*/
DroneBullet.prototype.update = function ()

/*
Switch to the next phase
Called by update()
Replace images for each phase
*/
DroneBullet.prototype.nextPhase = function ()

/*
Making collision detections between the bullet and tanks
Called by scroller object
The tank object will be returned it is hitted
Null will be returned if no tank is hitted
*/
DroneBullet.prototype.checkTanks = function ()

/*
Using bullet's CollisionDetector to check collision detections of neared tiles
Called by Scroller object
The brick object will be returned it is hitted
Null will be returned if no tank is hitted
*/
DroneBullet.prototype.checkTiles = function ()
```

File: enemy1.js

```
/*
This function is the constructor of the object
Enemy1 is the object of the first type enemy tank, it is created and called by Scroller object
Once it is created, the will be added into a array list in the Scroller object. Sometimes, it
will also be called by Player object and Bullet objects to make collision detections.
The purpose of this object is to create an enemy tank on the screen, it will shot the player
or be killed by the player.
*/
function Enemy1(tileRow, tileCol, level)

/*
Update all movements of the enemy
Called by Scroller object, the parameter is the array of bricks from Scroller
Determine the enemy's status and make movements including loading a bullet, moving
and rotating the turret.
It will call object of CollisionDetector to check if the tank is able to move
The tank will randomly moving at a certain direction for a random period of time.
Once the enemy find the player, it will move toward to player
*/
Enemy1.prototype.update = function (bricks)

/*
Updating the body rotation of the tank
Called by update()
depends on the moving direction of the tank to change the radian of image of the tank
*/
Enemy1.prototype.updateBodyRadian = function ()

/*
update the turret rotation of the tank
Called by update()
depends on the location of the player, change the rotation of the turret
*/
Enemy1.prototype.updateTurrentRotation = function ()

/*
fire a new bullet
Called by Scroller object
create a new Bullet object
*/
Enemy1.prototype.fireBullet = function ()
```

```
/*
Take hit by a bullet
Called by Scroller object, parameter bullet is a Bullet object
Calculate the damage from the bullet and deduct the health points
*/
Enemy1.prototype.takeBullet = function (bullet)

/*
Update the health bar
Called by takeBullet()
Calculate the new length of health bar and red bar and update them
*/
Enemy1.prototype.updateHealthBar = function ()
```

File: Enemy2.js

```
/*
This function is the constructor of the object
Enemy2 is the object of the second type enemy tank, it is created and called by Scroller object
Once it is created, the will be added into a array list in the Scroller object. Sometimes, it
will also be called by Player object and Bullet objects to make collision detections.
The purpose of this object is to create an enemy tank on the screen, it will shot the player
or be killed by the player.
*/
function Enemy2(tileRow, tileCol, level)

/*
Update all movements of the enemy
Called by Scroller object, the parameter is the array of bricks from Scroller
Determine the enemy's status and make movements including loading a bullet, moving
and rotating the turret.
It will call object of CollisionDetector to check if the tank is able to move
The tank will randomly moving at a certain direction for a random period of time.
Once the enemy find the player, it will move toward to player
*/
Enemy2.prototype.update = function (bricks)

/*
Updating the body rotation of the tank
Called by update()
depends on the moving direction of the tank to change the radian of image of the tank
*/
Enemy2.prototype.updateBodyRadian = function ()

/*
update the turret rotation of the tank
Called by update()
depends on the location of the player, change the rotation of the turret
*/
Enemy2.prototype.updateTurrentRotation = function ()

/*
fire a new bullet
Called by Scroller object
create a new Bullet object
*/
Enemy2.prototype.fireBullet = function ()
```

```
/*
Take hit by a bullet
Called by Scroller object, parameter bullet is a Bullet object
Calculate the damage from the bullet and deduct the health points
*/
Enemy2.prototype.takeBullet = function (bullet)

/*
Update the health bar
Called by takeBullet()
Calculate the new length of health bar and red bar and update them
*/
Enemy2.prototype.updateHealthBar = function ()
```

File: enemy3.js

```
/*
This function is the constructor of the object
Enemy3 is the object of the second type enemy tank, it is created and called by Scroller object
Once it is created, the will be added into a array list in the Scroller object. Sometimes, it
will also be called by Player object and Bullet objects to make collision detections.
The purpose of this object is to create an enemy tank on the screen, it will shot the player
or be killed by the player.
*/
function Enemy3(tileRow, tileCol, level)

/*
Update all movements of the enemy
Called by Scroller object, the parameter is the array of bricks from Scroller
Determine the enemy's status and make movements including loading a bullet, moving
and rotating the turret.
It will call object of CollisionDetector to check if the tank is able to move
The tank will randomly moving at a certain direction for a random period of time.
Once the enemy find the player, it will move toward to player
*/
Enemy3.prototype.update = function (bricks)

/*
Updating the body rotation of the tank
Called by update()
depends on the moving direction of the tank to change the radian of image of the tank
*/
Enemy3.prototype.updateBodyRadian = function ()

/*
update the turret rotation of the tank
Called by update()
depends on the location of the player, change the rotation of the turret
*/
Enemy3.prototype.updateTurrentRotation = function ()

/*
fire a new bullet
Called by Scroller object
create a new Bullet object
*/
Enemy3.prototype.fireBullet = function ()
```

```
/*
Take hit by a bullet
Called by Scroller object, parameter bullet is a Bullet object
Calculate the damage from the bullet and deduct the health points
*/
Enemy3.prototype.takeBullet = function (bullet)

/*
Update the health bar
Called by takeBullet()
Calculate the new length of health bar and red bar and update them
*/
Enemy3.prototype.updateHealthBar = function ()
```

File: explodeBullet.js

```
/*
This function is the constructor of the object
Bullet is the object for the all tanks bullet, it is created and called by Player, enemy and Boss object
Once it is created, the will be added into a array list in the Scroller object. Sometimes, it
will also be called by Scroller to make collision detections.
The purpose of this object is to create an bullets on the screen, it may deduct the tanks's health or even kill the tanks.
*/
function ExplodeBullet(tank, damage, pen)

/*
Updating all movement and status of the bullet
Called by Scroller object
If the bullet has exploded then increment the time counter and make rotations
*/
ExplodeBullet.prototype.update = function ()

/*
Switch to the next phase
Called by update()
Replace images for each phase
*/
ExplodeBullet.prototype.nextPhase = function ()

/*
Making collision detections between the bullet and tanks
Called by scroller object
The tank object will be returned it is hitted
Null will be returned if no tank is hitted
*/
ExplodeBullet.prototype.checkTanks = function ()

/*
Using bullet's CollisionDetector to check collision detections of neared tiles
Called by Scroller object
The brick object will be returned it is hitted
Null will be returned if no tank is hitted
*/
ExplodeBullet.prototype.checkTiles = function ()
```

File: ImageLoader.js

```
/*
This function is the constructor of the object
Load images
Called by construcor of Scroller object
After it is called, all nessary images will be loaded
*/
ImageLoader = function ()

/*
Swap the image of button of upgrading capacity of drone's machine gun
Called by updateButtons() in Scroller object
After it is called, the image of the button will be replaced
*/
ImageLoader.prototype.swapDroneMagImage = function()

/*
A listener of the mouse actions
Called by the image of buttons when the player clicked them
After it is called, the location of the image will be changed
*/
function mouseClick()

/*
A listener of the mouse actions
Called by the image of buttons when the player released mouse on them
After it is called, the location of the image will be changed
*/
function mouseRelease()

/*
A listener of the mouse actions
Called by the image of buttons when the player's mouse move over them
After it is called, the size of the image will be changed
*/
function mouseOver()

/*
A listener of the mouse actions
Called by the image of buttons when the player's mouse move out from them
After it is called, the size of the image will be changed
*/
function mouseOut()
```

File: KeyboardController.js

```
/*
This function is the constructor of the object
KeyboardController object
Create by Main, it is automatically detect players keyboard implementations
After keyboard keys are pressed, related variables will be changed
*/
function KeyboardController()
```

File: LootBox.js

```
/*
This function is the constructor of the object
LootBox is an object that contains awards for player when an enemy tank is killed
Created and called by Scroller object
After it is called, based on the killed tank, a loot box will be created and displayed on the screen
*/
function LootBox(tank)

/*
Update the status of the loot box
Called by Scroller
After it is called, the loot box's fire will be updated and make collision detection
bewteen the player's tank and the loot box
*/
LootBox.prototype.update = function ()

/*
Replace a new image of fire
Called by update()
After it is called, a new image of fire will be replaced
*/
LootBox.prototype.nextPhase = function ()
```

File: MachineGunBullets.js

```
/*
This function is the constructor of the object
Bullet is the object for the all tanks bullet, it is created and called by Player, enemy and Boss objects
Once it is created, the will be added into a array list in the Scroller object. Sometimes, it
will also be called by Scroller to make collision detections.
The purpose of this object is to create an bullets on the screen, it may deduct the tanks's health or even kill the tanks.
*/
function MachineGunBullet(tank, damage, pen)

/*
Updating all movement and status of the bullet
Called by Scroller object
If the bullet has exploded then increment the time counter and make rotations
*/
MachineGunBullet.prototype.update = function ()

/*
Switch to the next phase
Called by update()
Replace images for each phase
*/
MachineGunBullet.prototype.nextPhase = function ()

/*
Making collision detections between the bullet and tanks
Called by scroller object
The tank object will be returned it is hitted
Null will be returned if no tank is hitted
*/
MachineGunBullet.prototype.checkTanks = function ()

/*
Using bullet's CollisionDetector to check collision detections of neared tiles
Called by Scroller object
The brick object will be returned it is hitted
Null will be returned if no tank is hitted
*/
MachineGunBullet.prototype.checkTiles = function ()
```

File: Main.js

```
/*
The main function of this game
Called by index.html
After it is called, a Scroller object will be created
update all status and movements of objects in the game
*/
function Main()
```

File: PiercerBullet.js

```
/*
Bullet is the object for the all tanks bullet, it is created and called by Player, enemy and Boss object
Once it is created, the will be added into a array list in the Scroller object. Sometimes, it will also be called by Scroller to make collision detections.
The purpose of this object is to create an bullets on the screen, it may deduct the tanks's health or even kill the tanks.
*/
function PiercerBullet(tank, damage, pen, crit)

/*
Updating all movement and status of the bullet
Called by Scroller object
If the bullet has exploded then increment the time counter and make rotations
*/
PiercerBullet.prototype.update = function ()

/*
Switch to the next phase
Called by update()
Replace images for each phase
*/
PiercerBullet.prototype.nextPhase = function ()

/*
Making collision detections between the bullet and tanks
Called by scroller object
The tank object will be returned it is hitted
Null will be returned if no tank is hitted
*/
PiercerBullet.prototype.checkTanks = function ()

/*
Using bullet's CollisionDetector to check collision detections of neared tiles
Called by Scroller object
The brick object will be returned it is hitted
Null will be returned if no tank is hitted
*/
PiercerBullet.prototype.checkTiles = function ()
```

File: Player.js

```
/*
This function is the constructor of the object
Player is the object of the first enemy tank, it is created and called by Scroller object
Once it is created, Player can control the tank to move and fire. Sometimes, it
will also be called by Enemy object, Bullet objects and LootBox to make collision detections.
The purpose of this object is to create an player tank on the screen, it will shot the player
or be killed by the enemy tanks.
*/
function Player(tileRow, tileCol)

/*
Update all movements of the player's tank
Called by Scroller object
It will call object of CollisionDetector to check if the tank is able to move
Determine the player's status and make movements including loading a bullet, moving
and rotating the turret.
*/
Player.prototype.update = function ()

/*
Updating the body rotation of the tank
Called by update()
depends on the moving direction of the tank to change the radian of image of the tank
*/
Player.prototype.updateBodyRadian = function ()

/*
update the turret rotation of the tank
Called by update()
depends on the location of the player, change the rotation of the turret
*/
Player.prototype.updateTurrentRotation = function ()

/*
fire a new bullet
Called by Scroller object
create a new Bullet object
*/
Player.prototype.fireBullet = function ()
```

```
/*
Update the reloading time of main gun
Called by Scroller object
Update the the counter of reloading
*/
Player.prototype.reloadMainGun = function ()

/*
Update the reloading time of machine gun
Called by update()
Update the the counter of reloading
*/
Player.prototype.reloadMachineGun = function ()

/*
Take hit by a bullet
Called by Scroller object, parameter bullet is a Bullet object
Calculate the damage from the bullet and deduct the health points
*/
Player.prototype.takeBullet = function (bullet)

/*
Update the health bar
Called by takeBullet()
Calculate the new length of health bar and red bar and update them
*/
Player.prototype.updateHealthBar = function ()

/*
Pick up a loot box
Called by Scroller object
After it is called, the awards of the loot box will be applied on the player's tank
*/
Player.prototype.pickBox = function (box)

/*
Upgrade the piercer bullet
Called by updateButtons() in Scroller object
After it is called, the damage of the piercer bullet
will be increated by 2
*/
Player.prototype.upgradePerDmg = function ()
```

```
/*
Upgrade the explosive bullet
Called by updateButtons() in Scroller object
After it is called, the damage of the explosive bullet
will be increated by 2
*/
Player.prototype.upgradeExpDmg = function ()

/*
Upgrade the number of machine gun bullets
Called by updateButtons() in Scroller object
After it is called, the number of machine gun
bullets will be increated by 2
*/
Player.prototype.upgradeMcnMag = function ()

/*
Upgrade the possitility of critical hit of piercer bullet
Called by updateButtons() in Scroller object
After it is called, the possibility and critical damager of
piercer bullet will be increaed by 10% respectively
*/
Player.prototype.upgradePerCrt = function ()

/*
Upgrade the burning time of explosive bullets
Called by updateButtons() in Scroller object
After it is called, burning time of explosive
bullet will be increased by 20%
*/
Player.prototype.upgradeExpBurn = function ()

/*
Upgrade the damage of machine gun bullets
Called by updateButtons() in Scroller object
After it is called, damage of the machine gun will
be increased by 1
*/
Player.prototype.upgradeMcnDmg = function ()
```

```
/*
Upgrade the number of drone's machine gun bullets
Called by updateButtons() in Scroller object
After it is called, the number of drone's machine gun
bullets will be increated by 1
*/
Player.prototype.upgradeDroneMag = function ()

/*
Increase the number player's maximum health points
Called by updateButtons() in Scroller object
After it is called, the number of player's extra life will be increated by 1
*/
Player.prototype.upgradeHealth = function ()

/*
Increase the number player's extra life
Called by updateButtons() in Scroller object
After it is called, the number of player's extra life will be increated by 1
*/
Player.prototype.upgradeLife = function ()
```

File: Scroller.js

```
/*
This function is the constructor of the object
Scroller object is used to control all status and movements of objects on the screen
It is created by the Main object, also called by it to make updates
The main usage of this object is adding and deleting images of each objects such as tanks and
bullets
*/
function Scroller(stage)

/*
Initialize all global and local variables of the game
Called by update()
Initialize all necessary variables and
create the bricks, enemy and player's tanks and
display proper HUD information under the game canvas
*/
Scroller.prototype.initializeGame = function ()

/*
Update all object in the game
Called by Main object
Call and update Player object
Call and update all enemies objects
Call and update all bullets objects
Call and update all lootboxex objects
Add and remove images from the stage
*/
Scroller.prototype.update = function ()

/*
Tracking the mouse position
Be called when the Scroller object is created
it will automatically track the mouse position,
and update positions of mouse crusor images
*/
Scroller.prototype.getMousePosition = function (e)

/*
Creating one row of tiles
Called by initialize()
After it is called, a new row of tile will be created
*/
```

```
Scroller.prototype.createTilekRow = function ()

/*
Update all bricks
Called by update()
All bricks will be updated after called
*/
Scroller.prototype.updateBricks = function ()

/*
Update the player
Called by update()
After it is called, all players movements will be updated,
player also able to fire a bullet
*/
Scroller.prototype.updatePlayer = function ()

/*
Update all enemies
Called by update()
After it is called, all enemies' movements will be updated
Enemies also able to fire a bullet
*/
Scroller.prototype.updateEnemies = function ()

/*
Update all drones
Called by update()
After it is called, all drones' movements will be updated
Drones also able to fire bullets to enemies
*/
Scroller.prototype.updateDrones = function()

/*
Update all bullets
Called by update()
After it is called, all bullet's movements and staus will be updated
If an enemy is killed, a new loot box will be created
*/
Scroller.prototype.updateBullets = function ()
```

```
/*
Update all loot boxes
Called by Scroller object
After it is called, all status of loot boxex will be updated
*/
Scroller.prototype.updateLootBoxes = function ()

/*
Update all buttons
Called by update()
After it is called, it will check all status of buttons if they are pressed
Then the related functions will be called in the player's object
*/
Scroller.prototype.updateButtons = function ()

/*
Game is over
Called by update()
Stop the game and create a summary image for the player
Making the record form visibie in order to allow the player to make records
*/
Scroller.prototype.gameFinished = function ()

/*
Clear all images on the canvas
Called by update()
After it is called, all images on the canvas will be removed
Get ready for a new game
*/
Scroller.prototype.clearImages = function ()

/*
Eliminate all images of the tank
Called by updateBullets(), updateEnemies() and clearImages()
After it is called, all images of the tank will be eliminated from the stage
*/
Scroller.prototype.eliminateTank = function (tank)
```

```
/*
Switch imagee when the player clicked different tabs
Called by update()
After it is called, the all element in the keys or upgrade interface
will be removed and the other one will be added on the stage
*/
Scroller.prototype.switchPauseImages = function ()

/*
Add a drone for the player the kill enemies
Called by updateButtons()
After it is called, a drone will be created and added on the stage
to help the player to kill enemies
*/
Scroller.prototype.addDrone = function ()
```

File: Tile.js

```
/*
This function is the constructor of the object
Tile object is used contain bricks objects
Called by createTileRow() in Scroller object
It may be empty, tanks and bullets are able to go though
If it is not empty, then it can be considered as obstacles
*/
function Tile(row,col,totalCol)

/*
Set the tile to empty
Called by Scroller object
Once the tile is set to empty
All tanks and bullets are able to pass the tile
*/
Tile.prototype.setEmpty = function()
```

File: Wood.js

```
/*
This function is the constructor of the object
Concrete object is object of one of bricks
Created by Scroller object depends on the number of row and column in the tiles matrix
It is an obstacle object that blocking tanks and bullets, it can be destroyed by bullets
*/
function Wood (x,y,tileRow,tileCol)

/*
Update the concrete brick
Called by Scroller object
update the position of the brick
*/
Wood.prototype.update = function()
```