PROJECT REPORT

Sydney Melvin -
Implemented the boss battle, title screen, bug fixes

Keenan Coleman -
Implemented player, and enemies, and collision

Mickey Stephenson -
Transitions between scenes and removing objects between scenes, game over, bug fixes.

Source Code:
Main.lua:

```
---------------------------------------------------------------------------------------
--
-- main.lua
--
---------------------------------------------------------------------------------------
display.setStatusBar( display.HiddenStatusBar )


local composer = require( "composer" )
composer.gotoScene( "sceneTitle" )
```

sceneGame.lua:

```
---------------------------------------------------------------------------------------
--
-- sceneGame.lua
--
---------------------------------------------------------------------------------------
-- Set up physics
local physics = require("physics");
physics.start();
physics.setGravity(0,0);

-- Load sound
local soundTable=require("soundTable");

-- Set up composer scene
local composer = require("composer")
local scene = composer.newScene()
```

```
local starfield1
local starfield2
local runtime = 0
local scrollSpeed = 1.4
local hud = nil
local delay = 4000


local enemies = {}
local sceneGroup = display.newGroup()
-- "scene:create()"
function scene:create(event)



end

-- "scene:show()"
function scene:show(event)
   local phase = event.phase

   if (phase == "will") then

         score = 0
         health = 5
         -- Score display
         textNum = display.newText(score, 400, 70, native.systemFont, 36)
         textScore = display.newText("Score: ", 300, 70, native.systemFont, 36)
         sceneGroup:insert(textNum)
         sceneGroup:insert(textScore)

         -- Health display
         textNum2 = display.newText(health, 380, 120, native.systemFont, 36)
         textHealth = display.newText("HP: ", 300, 120, native.systemFont, 36)
         sceneGroup:insert(textNum2)
         sceneGroup:insert(textHealth)

               -- Create the control Bar for the player character
         local controlBar = display.newRect(0, 320, 140, display.contentHeight);
         controlBar:setFillColor(1,1,1,0.5);
         sceneGroup:insert(controlBar)

         -- Create the player character
```

```
	local player = display.newCircle(display.contentCenterX-450, display.contentHeight/2,
15);
	physics.addBody(player, "dynamic");
	sceneGroup:insert(player)

	-- Function to move the player character using the control bar
	local function move ( event )
		if event.phase == "began" then
			player.markY = player.y
		elseif event.phase == "moved" then
			local y = (event.y - event.yStart) + player.markY

			if (y <= 20 + player.height/2) then
				player.y = 20+player.height/2;
			elseif (y >= display.contentHeight-20-player.height/2) then
				player.y = display.contentHeight-20-player.height/2;
			else
				player.y = y;
			end
		end
	end

	-- Add event listener to the control bar to move the player character
	controlBar:addEventListener("touch", move);

	-- Function to spawn projectiles from the player character when the screen is tapped
	local function fire (event)
		local projectile = display.newCircle (player.x + 25, player.y, 5);
		projectile.anchorY = 1;
		projectile:setFillColor(0,1,0);
		physics.addBody (projectile, "dynamic", {radius=5} );
		projectile:applyForce(2,0, projectile.x, projectile.y);

		audio.play( soundTable["shootSound"] );
		projectile.parent = player

		local function removeProjectile (event)
			if (event.phase=="began") then
				event.target:removeSelf();
				event.target=nil;
				if (event.other.tag == "enemy") then
					event.other.pp:hit();
					if(event.other.pp:getHealth() <= 0) then
						score = score + 100;
```

```
                                        textNum.text = score
                                end
                        elseif (event.other.tag == "boss") then
                                event.other.pp:hit();
                                if(event.other.pp:getHealth() <= 0) then
                                        score = score + 10000;
                                        textNum.text = score
                                        gameOver(true)
                                end
                        end
                end
        end
        projectile:addEventListener("collision", removeProjectile);
end

-- Add event listener to the screen to spawn projectiles
Runtime:addEventListener("tap", fire)

--Enemy
local Enemy1 = require ("Enemy1");
local Enemy2 = require ("Enemy2");
local Boss = require ("Boss");

local bossLevelDuration = 120 -- 2 minutes in seconds
local elapsedTime = 0
local bossSpawned = false

function spawner()
        elapsedTime = elapsedTime + 4

        -- Check if the elapsed time has reached the boss level duration
        if elapsedTime > bossLevelDuration and not bossSpawned then
                bossSpawned = true

                local boss = Boss:new({xPos=1200, yPos=math.random(1, 640)});
                --sceneGroup:insert(boss)
                boss:spawn();
                boss:move();
                boss:shoot();

                -- loop over all items in the enemies array
                for i = #enemies, 1, -1 do
                        local enemy = enemies[i]
                        enemy:offScreen()
```

```
                end
            elseif not bossSpawned then
                --Square
                en1 = Enemy1:new({xPos=1200, yPos= math.random(1, 640)});
                --sceneGroup:insert(en1)
                en1:spawn();
                en1:move();
                sceneGroup:insert(en1.shape)
                table.insert(enemies, en1)

                --Triangle
                en2 = Enemy2:new({xPos=1200, yPos=math.random(1, 640)});
                --sceneGroup:insert(en2)
                en2:spawn();
                en2:move();
                sceneGroup:insert(en2.shape)
                table.insert(enemies, en2)
            end
        end

        local spawnEnemies = timer.performWithDelay(delay, spawner, 100)

local function addScrollableBg()
    local starfield = { type="image", filename="starfield.png" }
    -- Add First bg image
    starfield1 = display.newRect(0, 0, display.contentWidth, display.actualContentHeight)
    starfield1.fill = starfield
    starfield1.x = display.contentCenterX
    starfield1.y = display.contentCenterY

    -- Add Second bg image
    starfield2 = display.newRect(0, 0, display.contentWidth, display.actualContentHeight)
    starfield2.fill = starfield
    starfield2.x = display.contentCenterX  - display.actualContentWidth
    starfield2.y = display.contentCenterY
end

local function moveBg(dt)
    starfield1.x = starfield1.x + scrollSpeed * dt
    starfield2.x = starfield2.x + scrollSpeed * dt

    if (starfield1.x - display.contentWidth/2) > display.actualContentWidth then
        starfield1:translate(-starfield1.contentWidth * 2, 0)
    end
```

```
    if (starfield2.x - display.contentWidth/2) > display.actualContentWidth then
        starfield2:translate(-starfield2.contentWidth * 2, 0)
    end
end

local function getDeltaTime()
local temp = system.getTimer()
local dt = (temp-runtime) / (1000/60)
runtime = temp
return dt
end

local function enterFrame()
    local dt = getDeltaTime()
    moveBg(dt)
end

function init()
    addScrollableBg()
    Runtime:addEventListener("enterFrame", enterFrame)
end

init()

        local function restartGame()
                Runtime:removeEventListener("tap", restartGame)
                timer.cancel(spawnEnemies)
                composer.gotoScene(
        "sceneTitle",
        {
            effect = "slideRight",
                                time = 4000
        }
    )
        end

        --True if player won, False if player lost
        local function gameOver(playerWon)
                local gameOverString = playerWon and "You Won!" or "You lost! Game Over."
                gameOverText = display.newText(gameOverString, display.contentCenterX,
display.contentCenterY, native.systemFont, 100)
                sceneGroup:insert(gameOverText)
                Runtime:removeEventListener("tap", fire)
                Runtime:addEventListener("tap", restartGame)
```

```
        end

        -- Function to handle collisions between the player character and other objects
        local function playerHit(event)
                if event.phase == "began" then
                        health = health - 1;
                        textNum2.text = health
                        if(health <= 0) then
                                --Player Lost
                                gameOver(false)
                        end
                        if (event.other.tag == "enemy" or event.other.tag == "boss") then
                                event.other.pp:offScreen();
                        end
                end
        end

        player:addEventListener("collision", playerHit);
    -- Called when the scene is still off screen (but is about to come on screen).
  elseif (phase == "did") then


  -- Called when the scene is now on screen.
  -- Insert code here to make the scene come alive.
  -- Example: start timers, begin animation, play audio, etc.
  end
end

-- "scene:hide()"
function scene:hide(event)
  local phase = event.phase

  if (phase == "will") then
    -- Called when the scene is on screen (but is about to go off screen).
    -- Insert code here to "pause" the scene.
    -- Example: stop timers, stop animation, stop audio, etc.

                starfield1.isVisible = false
                starfield2.isVisible = false
                for i = sceneGroup.numChildren, 1, -1 do
      local child = sceneGroup[i]
      display.remove(child)
    end
                for i = 1, #enemies do
```

```
                        local enemy = enemies[i]
                        enemy:offScreen()
                end
    elseif (phase == "did") then
    -- Called immediately after scene goes off screen.
    end
end

-- "scene:destroy()"
function scene:destroy(event)

    -- Called prior to the removal of scene's view ("sceneGroup").
    -- Insert code here to clean up the scene.
    -- Example: remove display objects, save state, etc.
end
```

```
-- Listener setup
scene:addEventListener("create", scene)
scene:addEventListener("show", scene)
scene:addEventListener("hide", scene)
scene:addEventListener("destroy", scene)
```

```
return scene
```

Enemy.lua:

```
local physics = require("physics");
local soundTable=require("soundTable");
```

```
local Enemy = {tag="enemy", HP=1, xPos=0, yPos=0, fR=0, sR=0, bR=0, fT=1000, sT=500, bT
=500, red=1, green=1, blue=0};

function Enemy:new (o)    --constructor
  o = o or {};
  setmetatable(o, self);
  self.__index = self;
  return o;
end

function Enemy:spawn()
 self.shape=display.newCircle(self.xPos, self.yPos,15);
 self.shape.pp = self;  -- parent object
 self.shape.tag = self.tag; -- “enemy”
 self.shape:setFillColor (self.red,self.green,self.blue);
 physics.addBody(self.shape, "kinematic");
end


function Enemy:back ()
  transition.to(self.shape, {x=self.shape.x, y=150,
  time=self.fB, rotation=self.bR,
  onComplete=function (obj) self:forward() end} );
end

function Enemy:side ()
   transition.to(self.shape, {x=self.shape.x,
   time=self.fS, rotation=self.sR,
   onComplete=function (obj) self:back() end } );
end

function Enemy:forward ()
   transition.to(self.shape, {x=self.shape.x, y=800,
   time=self.fT, rotation=self.fR } );
end

function Enemy:move ()
          self:forward();
end

function Enemy:hit ()
          self.HP = self.HP - 1;
          if (self.HP > 0) then
```

```
                audio.play( soundTable["hitSound"] );
                self.shape:setFillColor(0.5,0.5,0.5);

        else
                audio.play( soundTable["explodeSound"] );

    transition.cancel( self.shape );

                if (self.timerRef ~= nil) then
                        timer.cancel ( self.timerRef );
                end

                -- die
                self.shape:removeSelf();
                self.shape=nil;
        end
end

function Enemy:getHealth ()
        return self.HP;
end

function Enemy:offScreen()
  if self.shape then
    self.shape:removeSelf();
          self.shape=nil;
  end
end

function Enemy:shoot (interval)
  interval = interval or 1500;

  local function createShot(obj)
    local p = display.newRect(obj.shape.x - 120, obj.shape.y + 10, 10, 10);
    p:setFillColor(1, 0, 0);
    p.anchorY = 0;
    physics.addBody(p, "dynamic");
    p:applyForce(-2, 0, p.x, p.y);

    local function shotHandler (event)
      if (event.phase == "began" and event.other.tag ~= 'enemy') then
            event.target:removeSelf();
          event.target = nil;
      end
```

```
    end
    p:addEventListener("collision", shotHandler);
  end

  self.timerRef = timer.performWithDelay(
    interval,
    function (event) createShot(self) end,
    -1
  );
end

return Enemy
```

Enemy1.lua:

```
local Enemy = require("Enemy");

local Enemy1 = Enemy:new( {HP=2, fR=720, fT=700,
                                    bT=700} );

function Enemy1:spawn()
  self.shape = display.newRect (self.xPos,
                                        self.yPos, 30, 30);
  self.shape.pp = self;
  self.shape.tag = "enemy";
  self.shape:setFillColor ( 0, 1, 1);
  physics.addBody(self.shape, "kinematic");
end

function Enemy1:back ()
   transition.to(self.shape, {x=self.shape.x-1163, time=self.bT, rotation=self.sR});
end

function Enemy1:forward ()
          transition.to(self.shape, {x=-20,time=10000, rotation=self.fR,
          onComplete= function (obj) self:offScreen() end});
end

return Enemy1;
```

Enemy2.lua:

```
local Enemy = require("Enemy");
```

```
local Enemy2 = Enemy:new( {HP=3, bR=360, fT=500, bT=300});

function Enemy2:spawn()
 self.shape = display.newPolygon(self.xPos, self.yPos,{-15,-15,15,-15,0,15});
 self.shape.pp = self;
 self.shape.tag = "enemy";
 self.shape:setFillColor ( 1, 0, 1);
 physics.addBody(self.shape, "kinematic",{shape={-15,-15,15,-15,0,15}});
end


function Enemy2:back ()
  transition.to(self.shape, {x=self.shape.x-600,
    y=self.shape.y-self.dist, time=self.bT, rotation=self.bR,
    onComplete= function (obj) self:forward() end } );
end

function Enemy2:side ()
   transition.to(self.shape, {x=self.shape.x + 400,
      time=self.sT, rotation=self.sR,
      onComplete= function (obj) self:back () end });
end

function Enemy2:forward ()
  self.dist = math.random (40,70) * 10;
  transition.to(self.shape, {x=-20,
    y= 320, time= 10000, rotation=self.fR,
          onComplete= function (obj) self:offScreen() end} );
end


return Enemy2;
```

Boss.lua:

```
local Enemy = require("Enemy");
local soundTable=require("soundTable");

function createFish()
          --group where all body parts will be placed
```

```
local fish = display.newGroup()

--the frames or each fish body part
local opt =
{
	frames = {
		{ x = 22, y = 8, width = 167, height = 50}, -- 1. Body
		{ x = 207, y = 27, width = 16, height = 9}, -- 2. Snout 1
		{ x = 228, y = 27, width = 16, height = 9}, -- 3. Snout 2
		{ x = 249, y = 27, width = 16, height = 9}, -- 4. Snout 3
		{ x = 281, y = 20, width = 56, height = 26}, -- 5. Mouth 1
		{ x = 344, y = 20, width = 56, height = 26}, -- 6. Mouth 2
		{ x = 407, y = 20, width = 56, height = 26}, -- 7. Mouth 3
		{ x = 22, y = 93, width = 52, height = 37}, -- 8. Pectoral Fin 1
		{ x = 80, y = 99, width = 53, height = 31}, -- 9. Pectoral Fin 2
		{ x = 140, y = 102, width = 54, height = 28}, -- 10. Pectoral Fin 3
		{ x = 210, y = 70, width = 48, height = 92}, -- 11. Caudal Fin 1
		{ x = 267, y = 82, width = 55, height = 70}, -- 12. Caudal Fin 2
		{ x = 331, y = 89, width = 60, height = 55}, -- 13. Caudal Fin 3
		{ x = 405, y = 93, width = 60, height = 46} -- 14. Dorsal Fin
	}
}


--loads in sprite sheet
local sheet = graphics.newImageSheet( "KingBayonet.png", opt);

--sets the sequence for the fish animations
local sequences_fish = {

	{
		name = "Body",
		frames = { 1 },
		time = 800,
		loopCount = 0
	},

	{
		name = "Mouth",
		frames = { 5, 6, 7 },
		time = 800,
		loopCount = 0
	},
```

```
        {
                name = "Caudal fin",
                frames = { 11, 12,13 },
                time = 400,
                loopCount = 0
        },

        {
                name = "Pectoral fin",
                frames = {8, 9, 10 },
                time = 400,
                loopCount = 0
        },

        {
                name = "Snout",
                frames = { 2, 3, 4 },
                time = 400,
                loopCount = 0
        },


        {
                name = "Dorsal fin",
                frames = {14},
                time = 400,
                loopCount = 0
        },

}

--following code piecies the fishes body parts together
local body = display.newSprite(sheet, sequences_fish)

body.x = 0
body.y = 0

body:setSequence("Body")

local mouth = display.newSprite(sheet, sequences_fish)

mouth.x = (body.x) -39
mouth.y = (body.y) +5
```

```
	mouth:setSequence("Mouth")

	local snout = display.newSprite(sheet, sequences_fish)

	snout.x = (body.x) -90
	snout.y = (body.y) +3

	snout:setSequence("Snout")

	local caudal = display.newSprite(sheet, sequences_fish)

	caudal.x = (body.x) +100
	caudal.y = (body.y) -4

	caudal:setSequence("Caudal fin")

	local dorsal = display.newSprite(sheet, sequences_fish)

	dorsal.x = (body.x) +15
	dorsal.y = (body.y) -38

	dorsal:setSequence("Dorsal fin")

	local pectoral = display.newSprite(sheet, sequences_fish)

	pectoral.x = (body.x) +25
	pectoral.y = (body.y) +28

	pectoral:setSequence("Pectoral fin")

	--insets body parts into group
	fish:insert(body)
	fish:insert(mouth)
	fish:insert(snout)
	fish:insert(caudal)
	fish:insert(dorsal)
	fish:insert(pectoral)

	return fish
end

local Boss = Enemy:new( { HP=30 } );

function Boss:spawn()
```

```
  self.shape = createFish();
  self.shape.x = self.xPos;
  self.shape.y = self.yPos;
  self.shape.pp = self;
  self.shape.tag = "boss";
  physics.addBody(self.shape, "kinematic");
end

function Boss:move()
  transition.to (
    self.shape, {
       x = math.random(display.contentCenterX-450, display.contentWidth),
       y = math.random(0, display.contentHeight),
       time = 1000,
       onComplete = function() self:move() end
    }
  )
end

function Boss:hit ()
          self.HP = self.HP - 1;
          if (self.HP > 0) then
                    audio.play( soundTable["hitSound"] );
          else
                    audio.play( soundTable["explodeSound"] );

    transition.cancel( self.shape );

                    if (self.timerRef ~= nil) then
                              timer.cancel ( self.timerRef );
                    end

                    -- die
                    self.shape:removeSelf();
                    self.shape=nil;
          end
end

return Boss;
```

sceneTitle.lua:

```
-----------------------------------------------------------------------------------------
--
-- sceneTitle.lua
--
-----------------------------------------------------------------------------------------
local widget = require( "widget" )
local composer = require( "composer" )
local scene = composer.newScene()

-- "scene:create()"
function scene:create( event )

    local sceneGroup = self.view

    -- Display names of group members
    local names = display.newText(
        "Sydney Melvin, Keenan Coleman, Mickey Stephenson",
        0,
        0,
        native.systemFont,
        40
    )
    names.x = 150 ; names.y = display.contentCenterY - 200
    names:setFillColor( 1, 1, 1 )
    names.anchorX = 0

    -- Create start button to initiate game
    startButton = widget.newButton(
        {
            x = display.contentCenterX,
            y = display.contentCenterY,
            id = "startButton",
            label = "START",
            labelColor = {
                default={ 0, 0, 1 },
                over={ 0, 1, 1, }
            },
            shape = "roundedRect",
            width = 120,
            height = 80
        }
    )
```

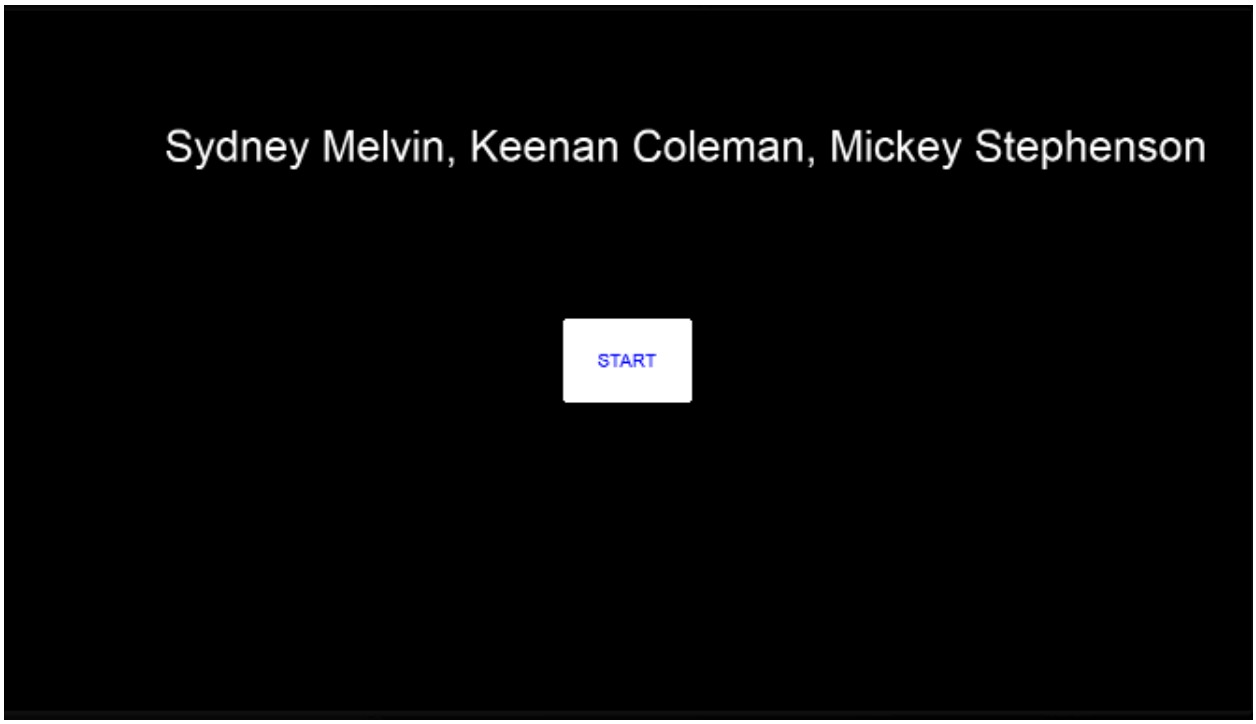
```
        selectedCharacter = ""

    --Use tap to transition to game scene
    function startButton:tap(event)
        composer.gotoScene(
            "sceneGame",
            {
                effect = "slideLeft",
                params = {
                    character = selectedCharacter
                }
            }
        )
        return true
    end

    startButton:addEventListener("tap", tap)

    -- Add names and start button to scene group
    sceneGroup:insert( names )
    sceneGroup:insert( startButton )
end

-- "scene:show()"
function scene:show( event )

  local sceneGroup = self.view
  local phase = event.phase

  if ( phase == "will" ) then
    -- Called when the scene is still off screen (but is about to come on screen).
  elseif ( phase == "did" ) then
    -- Called when the scene is now on screen.
    -- Insert code here to make the scene come alive.
    -- Example: start timers, begin animation, play audio, etc.
  end
end

-- "scene:hide()"
function scene:hide( event )

  local sceneGroup = self.view
  local phase = event.phase
```

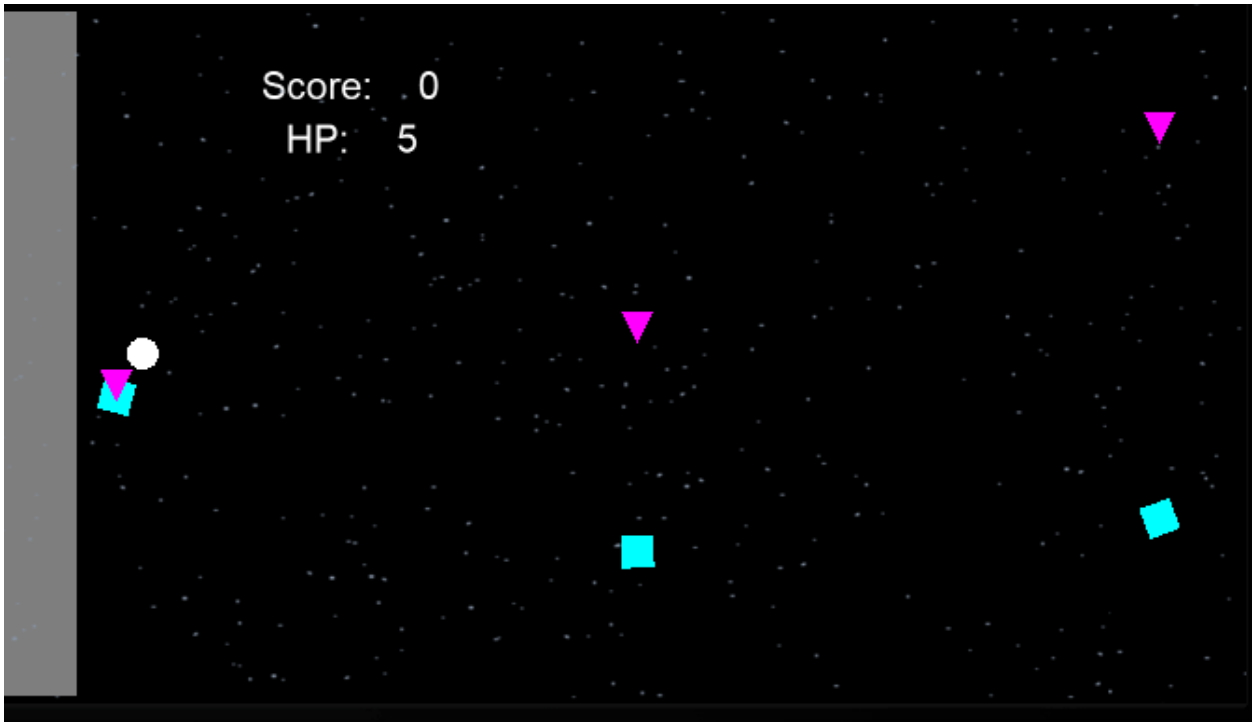
```
  if ( phase == "will" ) then
    -- Called when the scene is on screen (but is about to go off screen).
    -- Insert code here to "pause" the scene.
    -- Example: stop timers, stop animation, stop audio, etc.
  elseif ( phase == "did" ) then
    -- Called immediately after scene goes off screen.
  end
end

-- "scene:destroy()"
function scene:destroy( event )

  local sceneGroup = self.view

  -- Called prior to the removal of scene's view ("sceneGroup").
  -- Insert code here to clean up the scene.
  -- Example: remove display objects, save state, etc.
end

---------------------------------------------------------------------------------

-- Listener setup
scene:addEventListener( "create", scene )
scene:addEventListener( "show", scene )
scene:addEventListener( "hide", scene )
scene:addEventListener( "destroy", scene )

---------------------------------------------------------------------------------

return scene
```

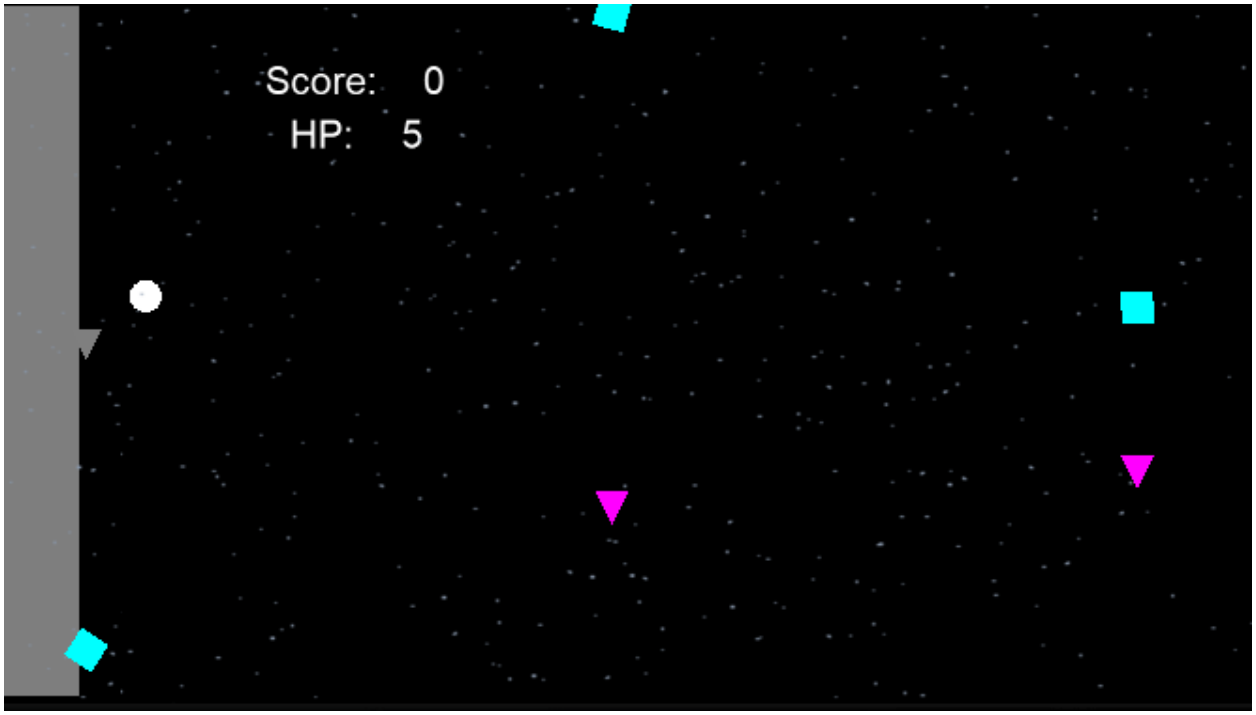
soundTable.lua:

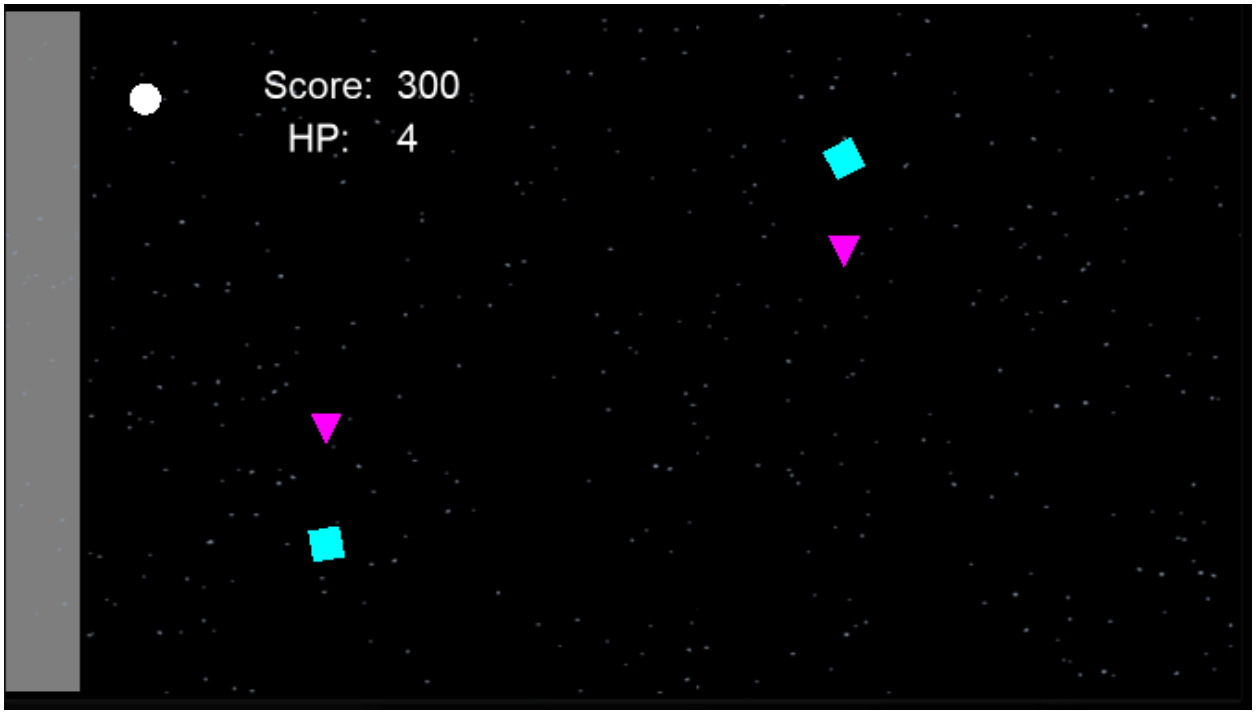
```
local soundTable = {
    shootSound = audio.loadSound( "shoot.wav" ),
    hitSound = audio.loadSound( "hit.wav" ),
    explodeSound = audio.loadSound( "explode.wav" ),
}

return soundTable;
```

Screen Shot:




Score: 0
HP: 5


Score: 0
HP: 5


Score: 300
HP: 4


CS371TeamProject - Solar2D Simulator
File Hardware View Help
Score: 2900
HP: 4