# Game Adaptivity

## What is our GOAL for this MODULE?

We added AI to the game and made the game increasingly complex as the game progresses.

## What did we ACHIEVE in the class TODAY?

- Used OOP concept to create a Box Object
- Added sounds to the T rex game.
- Made the game increasingly complex as the game progressed.
- Added AI to the Trex.

## Which CONCEPTS/ CODING BLOCKS did we cover today?

- OOP concepts
- sound.play()
- Concept of AI

## How did we DO the activities?

1. Revise the concepts of OOP, create a class for making a rectangle object.
    - A class required constructor () to define properties.
    - Also a method to declare its functionality.
    - A class can include more than one method.
    - Multiple objects can be created from a single class.

```
class Box
  {
    constructor()
    {
      this.x =100;
      this.y = 200
      this.w = 50;
      this.h = 50;
    }

    show()
    {
      rect(this.x,this.y,this.w,this.h)
    }

    set_speed(v)
    {
      this.x = this.x + v;
    }

  }
```

```
var box1;
var box2;

function setup() {
  createCanvas(600, 400);
  box1 = new Box();
  box2 = new Box();
}

function draw() {
  background(220);
  box1.show();
  box1.set_speed(2);
  box2.show();
  box2.set_speed(1);

}
```

2. Add sounds in Trex game::
    - *jump.mp3* - Trex jump sound - Plays when the user presses “space”.
    - *die.mp3* - Trex dying sound - Plays when the Trex touches an obstacle.
    - *checkPoint.mp3* - Trex crossing 100th milestone in score.

```
jumpSound = loadSound("jump.mp3")
dieSound = loadSound("die.mp3")
checkPointSound = loadSound("checkPoint.mp3")
```

```
//jump when the space key is pressed
if(keyDown("space")&& trex.y >= 362) {
    trex.velocityY = -12;
    jumpSound.play();
}
```

```
if(obstaclesGroup.isTouching(trex)){
    gameState = END;
    dieSound.play()
}
```

```
ground.velocityX = -4;
//scoring
score = score + Math.round(getFrameRate()/60);

if(score>0 && score%100 === 0){
    checkPointSound.play()
}

if (ground.x < 0){
  ground.x = ground.width/2;
}

//jump when the space key is pressed
if(keyDown("space")&& trex.y >= 100) {
    trex.velocityY = -12;
    jumpSound.play();
}
```

3. Increase the speed in the game as the game progresses. Increase the ground as well as obstacle velocity after every 100th score.

```
if(gameState === PLAY){

  gameOver.visible = false;
  restart.visible = false;

  ground.velocityX = -(4 + 3* score/100)
  //scoring
  score = score + Math.round(getFrameRate()/60);

  if(score>0 && score%100 === 0){
     checkPointSound.play()
  }

  if (ground.x < 0){
    ground.x = ground.width/2;
  }

  //jump when the space key is pressed
  if(keyDown("space")&& trex.y >= 100) {
      trex.velocityY = -12;
      jumpSound.play();
  }
```

```
function spawnObstacles(){
 if (frameCount % 60 === 0){
   var obstacle = createSprite(400,365,10,40);
   obstacle.velocityX = -(6 + score/100);

    //generate random obstacles
    var rand = Math.round(random(1,6));
    switch(rand) {
      case 1: obstacle.addImage(obstacle1);
              break;
      case 2: obstacle.addImage(obstacle2);
              break;
      case 3: obstacle.addImage(obstacle3);
              break;
      case 4: obstacle.addImage(obstacle4);
              break;
      case 5: obstacle.addImage(obstacle5);
              break;
      case 6: obstacle.addImage(obstacle6);
              break;
      default: break;
    }
```

4. Make the Trex artificially intelligent so that it jumps on its own when it sees the obstacle.
    - Change the collider radius of Trex.
    - Give vertical velocity to Trex when it touched an obstacle and commented on the statement to change the gameState to END.

```
//create Obstacle and Cloud Groups
obstaclesGroup = createGroup();
cloudsGroup = createGroup();

console.log("Hello" + 5);

trex.setCollider("rectangle",0,0,trex.width,trex.height);


score = 0;

}
```

```
  //spawn obstacles on the ground
  spawnObstacles();

  if(obstaclesGroup.isTouching(trex)){
      trex.velocityY = -12;
      jumpSound.play();
      //gameState = END;
      //dieSound.play()

  }
}
 else if (gameState === END) {
    gameOver.visible = true;
    restart.visible = true;

    ground.velocityX = 0;
    trex.velocityY = 0

    //change the trex animation
    trex.changeAnimation("collided", trex_collided);

    //set lifetime of the game objects so that they are never destroyed
  obstaclesGroup.setLifetimeEach(-1);
  cloudsGroup.setLifetimeEach(-1);
```



## What's next?

We will learn the "scope"  of variables in programming and add functionality to restart the game.

## Extend Your Knowledge:

1. You can read more about the Game Adaptivity: https://medium.com/@brucerobbins/adaptive-games-48e6d2c21821