Deliverable 1:

Sub-Task 1:
Pong game

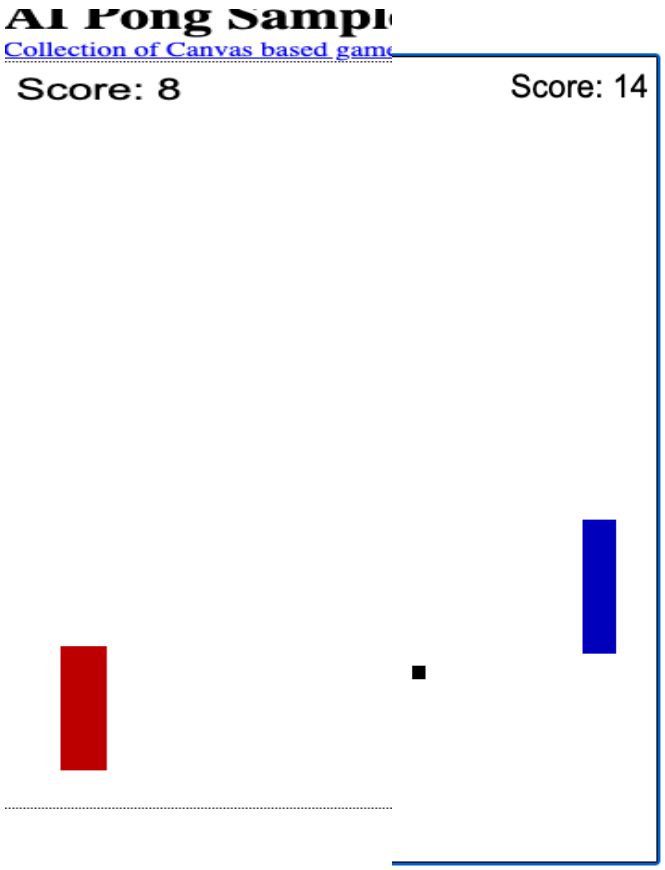
Sub-Task 2:
https://web.njit.edu/~jeb79/jeb79-dev/M6/html5.html

Sub-Task 3:
https://github.com/j0rdanbruce/IT202-007/pull/12

Deliverable 2:

Sub-Task 1:
For modification 1, I changed the AI difficulty according to the score of the rightPaddle and the leftPaddle. If the right paddle player had a significant lead in terms of the score, the AI difficulty would increase by adjusting how much of the screen the AI can see. If the AI has a significant score lead, the AI visibility of the screen would decrease instead of increase.

Sub-Task 2 and 3:



In this example, I tried to show that the difference in score of the AI and the human user is greater than 3. Because the current AI is beating the human user, the visibility percentage will increase by 0.1. If the human user is winning by a difference greater than 3, the AI visibility variable decrements by 0.1. Both the increase and decrease of visibilityPercent takes in effect for each time the ball is reset in-game.

The aiDifficulty() function uses an if statement to check if the difference in the left – right paddle score is greater than 3. If the difference in score is greater than 3, then the visibilityPercent global value is decremented by 0.1. The reverse occurs for the difference in Right side paddle – left side paddle. If the difference is greater than 3 in this case, the AI is able to see more of the game screen, which then allows it to react sooner to the ball moving towards it.

```
//UCID: 31569180      DATE: 11/20/22 <- #40-58 function init()
function resetBall() {
    roundCounter++;
    aiDifficulty();

    ball.x = canvas.width / 2 - ball.w / 2;
    ball.y = canvas.height / 2 - ball.w / 2;
    // Modify the ball object to have two speed properties, one for
    ball.sX = ballSpeed;
    ball.sY = ballSpeed / 2;

    // Randomize initial direction
    if (Math.random() > 0.5) {
        ball.sX *= -1;
    }
    // Randomize initial direction
    if (Math.random() > 0.5) {
        ball.sY *= -1;
    }
} <- #64-79 for (b in ballList) <- #59-81 function resetBall()
```

The second line of code shows that the aiDifficulty() function is called for everytime the resetBall() function is called. Because of this, AI visibility changes do not take effect until the next round when the ball is reset.

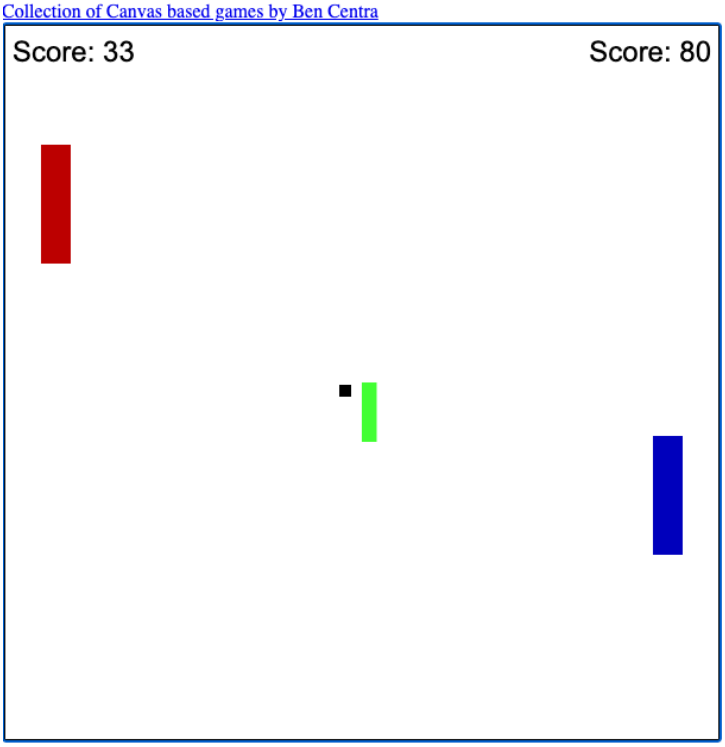
Deliverable 3:

Sub-Task 1:
For modification 2, I added a speed block accelerator. Whenever, the ball passes through the rectangular block, the speed of the ball is incremented by 1.

Sub-Task 2 and 3:

AI Pong Sample Back

Collection of Canvas based games by Ben Centra



This image depicts the ball passing through the middle speed block accelerator. When it passes through the block, it increments the ball speed by 1. The ball speed is then reset to normal speed whenever the AI or human-user scores a point. The speed of the ball still increases when the ball comes in contact with either of the paddles.

```
//UCID: 31569180      DATE: 11/20/22      You, now • Uncommitted changes
var speedBlock;
var speedBlockSpeed = 1;
```

The above code shows the added variables for the speedblock and speedblock movement speed value.

```
//UCID: 31569180      DATE: 11/20/22
function checkSpeedBlockBoundaries(){
  let ballcY = ball.y + ball.h/2;
    if (ballcY == speedBlock.y && ballcY == speedBlock.y + speedBlock.h) {
        ball.sX+= 1;
    }
}
```

The center of the ball is represented by the let ballcY… statement. The center of the ball comes in contact with the speedBlock, then the value of the balls movement speed is incremented by 1.

```
//UCID: 31569180      DATE: 11/20/22
function gameLoop() {
    erase();
    movePaddle();
    doAI();
    moveBall();

    checkPaddleCollision();
    checkScore();
    drawScores();

    checkSpeedBlockBoundaries();
    //draw stuff
    for (let i = 0; i < drawables.length; i++) {
        drawables[i].draw();
    }
}
```

This picture shows my code for the gameloop() function nbeing updated to include the checkSpeedBlockBoundaries() function.

Deliverable 4:

I had initial difficulty actually getting the game running using Ben Centra's original code for the pong game. However, I used the revised version that was created by Professor Matt. I also learned more about using canvas. A tricky part of the game assignment was the boundary assignment and figuring out the logic for when certain objects in the game came in to collision with each other. It was tough figuring out the upper, lower, left, and right side boundaries for the speedblock and when the ball came into contact with the block.