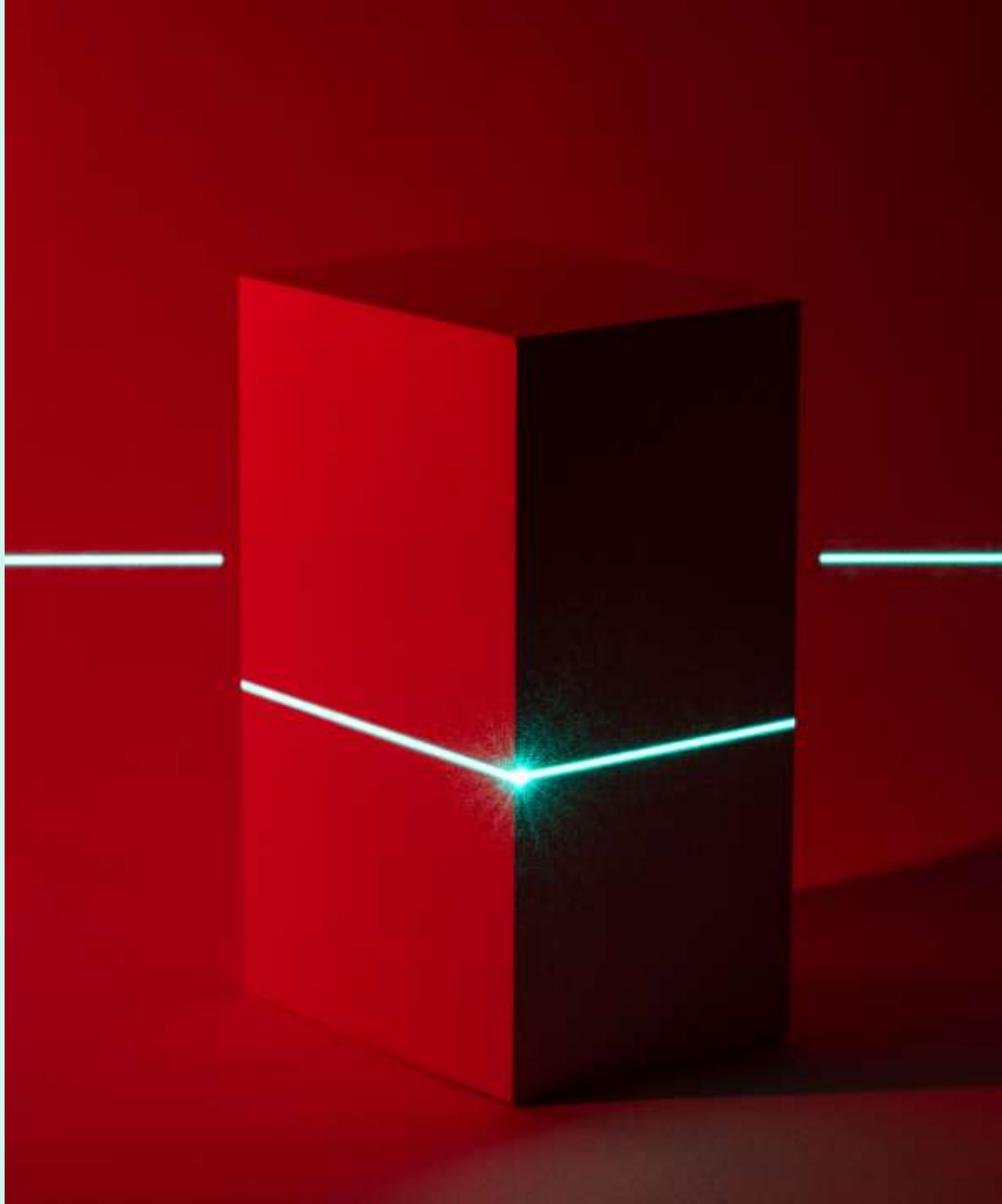
Pixel Dash

BY ANDY LE, JARED LICHT, DAYMIAN NGUYEN



Quick Summary of Game

- ❑ A “Geometry Dash”-inspired game where repeatedly generated levels with four obstacles that the player must avoid colliding.
- ❑ The player cube can dash, jump, and fast fall to get over the obstacles.
- ❑ Each level is determined when the green obstacle successfully moves from the right of the screen to the left edge. The speed of the obstacles will increase after reaching specific rounds.
- ❑ The game's objective is to survive and pass as many levels as possible without colliding into obstacles.

Steps in Creating the Game

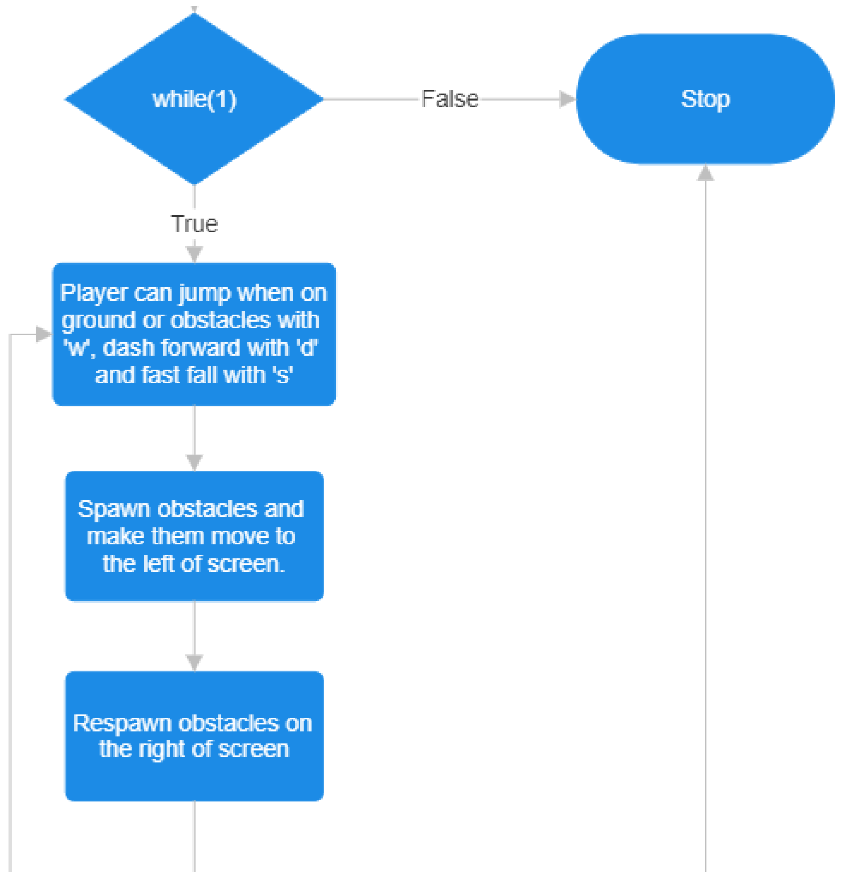
- ❑ Establish the player cube and its movement
- ❑ Create game environment with ground level and obstacles
- ❑ Create core game mechanics such as gravity, the ability to stand on top of ground and obstacles
- ❑ Develop a collision system for the box; when it hits an obstacle, the game ends
- ❑ Create level difficulty by increasing the speed of obstacles after reaching a specific round
- ❑ Display “lvl” and the level number on the seven segment displays on DE10-LITE

Responsibilities Among Team Members

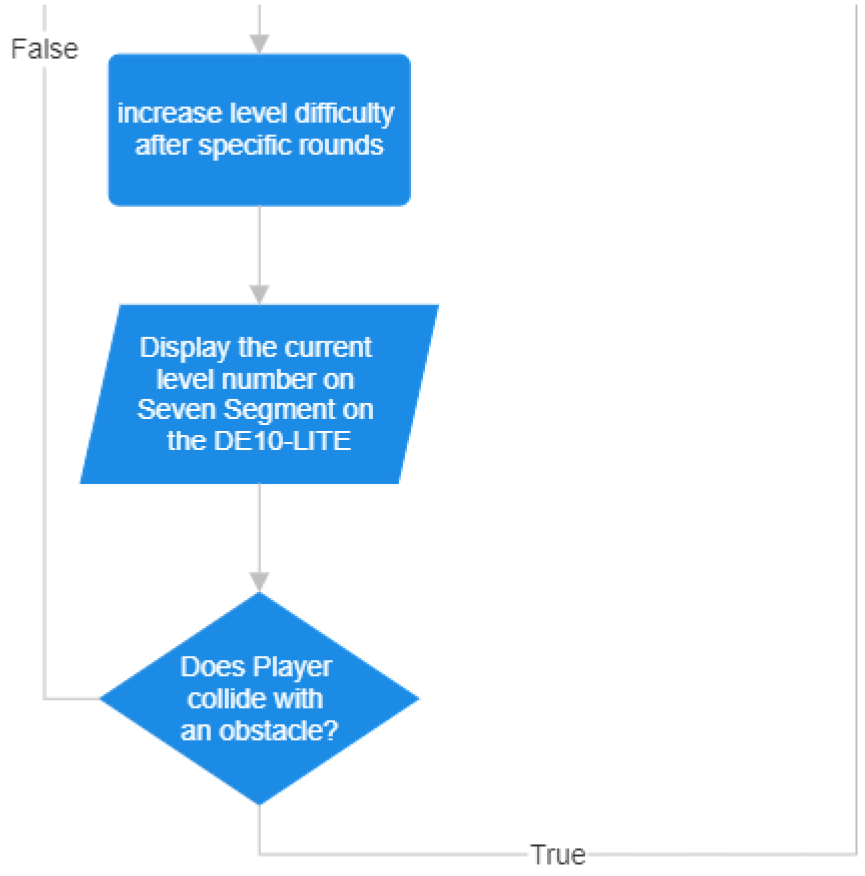
- Jared: Project Report, Notes/observations on coding process, Presentation of Game
- Andy: Developed struct variables(Playerbox, obstacles, ground). Created game mechanics. Displayed “LvL” and level number on DE10-LITE
- Daymian: Assisted in displaying the levels on seven-segment display of DE10-LITE. Partially configured controls for player box.

Software Flowchart

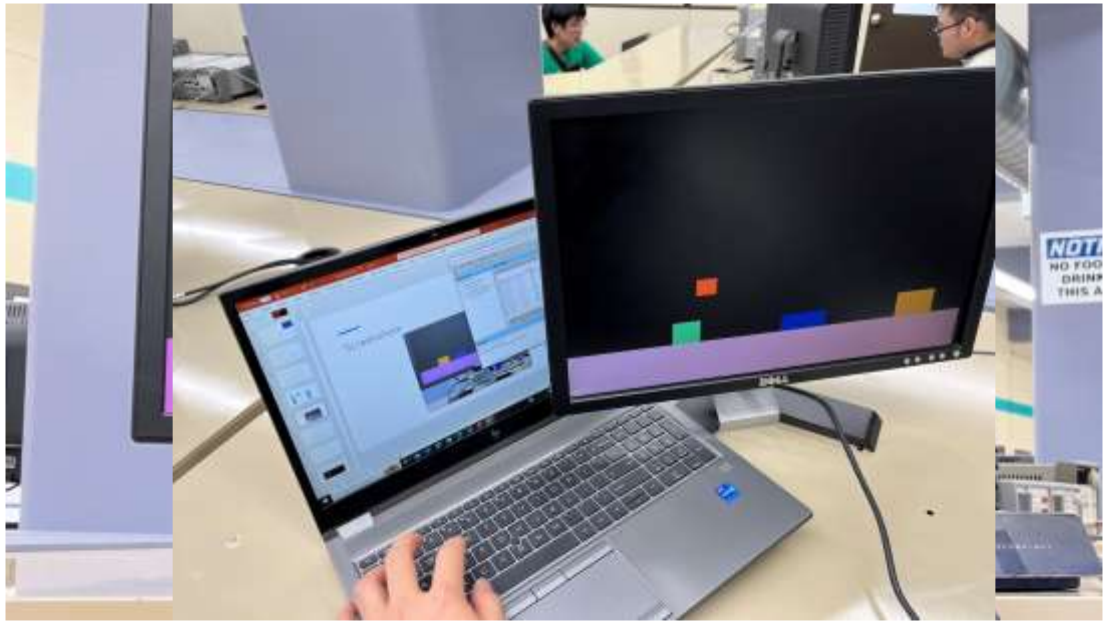
1.


while(1)
False
Stop
True
Player can jump when on ground or obstacles with 'w', dash forward with 'd' and fast fall with 's'
Spawn obstacles and make them move to the left of screen.
Respawn obstacles on the right of screen

2.


False
increase level difficulty after specific rounds
Display the current level number on Seven Segment on the DE10-LITE
Does Player collide with an obstacle?
True

Screenshots



Encountered Problems and their Solutions

❑ How to make player box stand on top of the obstacles.

Solution: Created struct variables and if-statements when the player is at a specific x and y range, the player coords are updated to be one pixel above the original location.

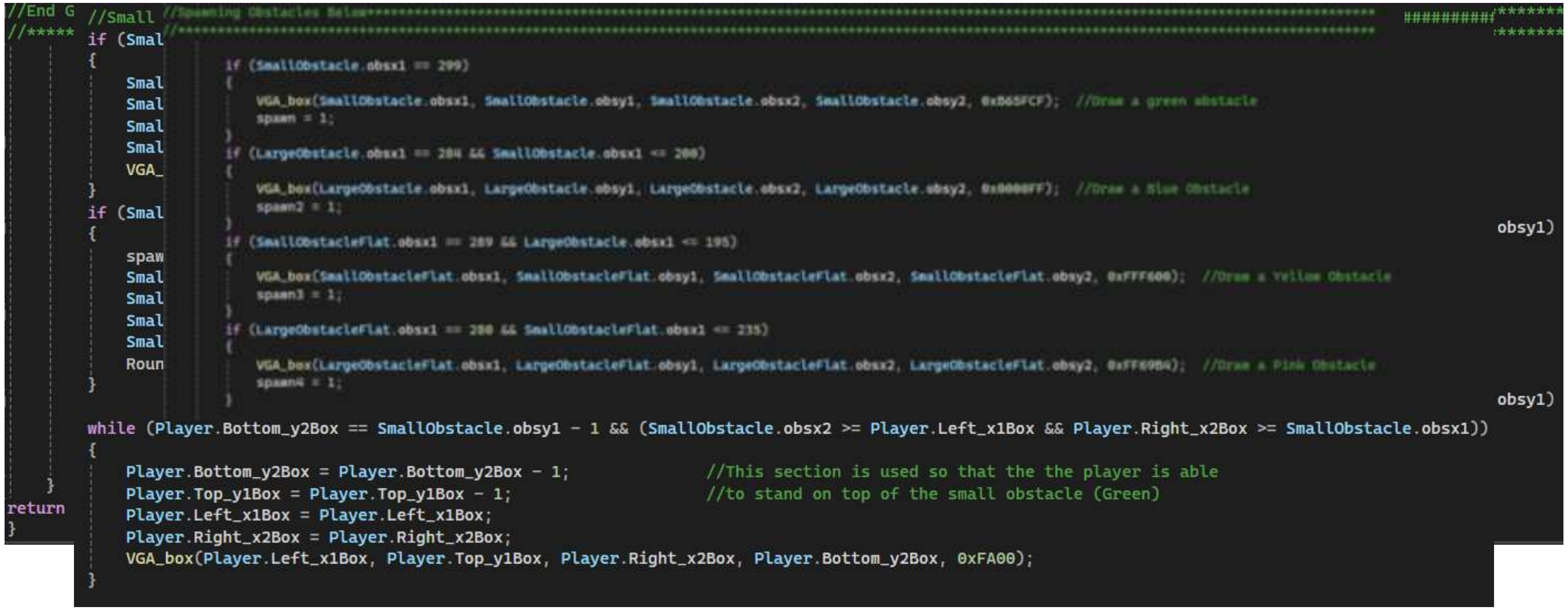
❑ How to make the player box collide with obstacles and end the game.

Solution: Created if-statements when the player is between the x-ranges of the obstacle AND if the player is below the top of the obstacle, then break/end the game.

New Acquired skills/knowledge

- ❑ Jared: I learned the overall structure of coding with respect to the DE-10 and how the player box interacts with the obstacles using coordinates of the VGA monitor while utilizing the address map of the DE-10.
- ❑ Andy: I learned that struct variables are very useful in the organization of the code and special characteristics/variables can be created for each type of struct.
- ❑ Daymian: I became more proficient when it came to programming on the DE-10 especially with the 7 segment hexes because I struggled a lot when it came to programming them.

Code with Comments



```
//End G
//*****
```

```
//Small
if (Smal
{
    Smal
    Smal
    Smal
    Smal
    VGA_
}
if (Smal
{
    spaw
    Smal
    Smal
    Smal
    Smal
    Roun
}
```

```
//Spawning Obstacles Below*****************************************************************************************
//*******************************************************************************************************************

        if (SmallObstacle.obsx1 == 299)
        {
            VGA_box(SmallObstacle.obsx1, SmallObstacle.obsy1, SmallObstacle.obsx2, SmallObstacle.obsy2, 0xB65FCF);  //Draw a green obstacle
            spawn = 1;
        }
        if (LargeObstacle.obsx1 == 284 && SmallObstacle.obsx1 <= 200)
        {
            VGA_box(LargeObstacle.obsx1, LargeObstacle.obsy1, LargeObstacle.obsx2, LargeObstacle.obsy2, 0x0000FF);  //Draw a Blue Obstacle
            spawn2 = 1;
        }
        if (SmallObstacleFlat.obsx1 == 289 && LargeObstacle.obsx1 <= 195)
        {
            VGA_box(SmallObstacleFlat.obsx1, SmallObstacleFlat.obsy1, SmallObstacleFlat.obsx2, SmallObstacleFlat.obsy2, 0xFFF600);  //Draw a Yellow Obstacle
            spawn3 = 1;
        }
        if (LargeObstacleFlat.obsx1 == 280 && SmallObstacleFlat.obsx1 <= 235)
        {
            VGA_box(LargeObstacleFlat.obsx1, LargeObstacleFlat.obsy1, LargeObstacleFlat.obsx2, LargeObstacleFlat.obsy2, 0xFF69B4);  //Draw a Pink Obstacle
            spawn4 = 1;
        }
```

```
##########/*******
          *******
obsy1)
obsy1)
    }
return
}
```

```
while (Player.Bottom_y2Box == SmallObstacle.obsy1 - 1 && (SmallObstacle.obsx2 >= Player.Left_x1Box && Player.Right_x2Box >= SmallObstacle.obsx1))
{
    Player.Bottom_y2Box = Player.Bottom_y2Box - 1;              //This section is used so that the the player is able
    Player.Top_y1Box = Player.Top_y1Box - 1;                    //to stand on top of the small obstacle (Green)
    Player.Left_x1Box = Player.Left_x1Box;
    Player.Right_x2Box = Player.Right_x2Box;
    VGA_box(Player.Left_x1Box, Player.Top_y1Box, Player.Right_x2Box, Player.Bottom_y2Box, 0xFA00);
}
```

Video of Game