5 pts

# Catz

1. Initialize 2 Rectangles for the cat and the mouse
2. Initialize an int variable timer that increases by 1 on every Update
3. Initialize KeyBoardStates for old and new keyboard status
4. Load Texture2Ds for the cat and the mouse
5. If arrow keys are pressed by using newKB && !oldKB, move the mouseRectangle in appropriate direction
6. If WASD keys are pressed by using newKB && !oldKB, move the catRectangle in appropriate direction
7. oldKB = newKB
8. Draw the time elapsed
9. DrawString containing “Time: ” + time / 60

# Follow Me

1. Initialize a Rectangle for Jimbo
2. Initialize old and new Mouse variables
3. Load Texture2D for him standing and him walking
4. Check if the Mouse’s leftButton was pressed by using newMouse && !oldMouse
   1. Record the click in a Boolean
   2. Change the texture to him walking
   3. Record the Mouse’s coordinates
   4. Keep moving him till his coordinates equal the mouse’s coordinates
      1. Left = -X and Right = X
      2. Up = -Y and Down = Y
   5. Change texture to him standing
5. oldMouse = new Mouse
6. Draw Jimbo on each Update iteration

# How Much

1. Initialize an int count = 0
2. Initialize old and new KeyboardStates
3. Initialize SpriteFont
4. If up arrow key is pressed in Update by using newKB && !oldKB
   1. As long as count is less than 10, increase count
5. If down arrow key is pressed in Update by using newKB && !oldKB
   1. As long as count is greater than -10, decrease count
6. oldKB = newKB
7. DrawString of the count on screen

# Beam Me Up

1. Initialize Rectangle for Trek image
2. Initialize SpriteFont
3. Initialize old and new KeyBoardState
4. Initialize array for SoundEffects
5. Initialize Boolean for played
6. Load the Texture2D for Trek image
7. Load each SoundEffect to be used
8. Check if any NumPad keys have been pressed by using by using newKB && !oldKB
   1. If true, play SoundEffect [<num>]
   2. Set played to true
   3. Wait until SoundEffect.Duration() == 0
   4. Then make played false
9. oldKB = newKB
10. Draw Trek image on screen
11. Use DrawString with <indexNo. (1-5)> + “: <Sound Name>”

# Avatar

1. Initialize Rectangle array for avatars
2. Initialize Texture2D array for corresponding textures
3. Initialize Rectangle for character selection highlight box
4. Initialize old and new GamePad for Xbox controller
5. Initialize index variable for currently selected character
6. Initialize Boolean for clicked
7. Load all Texture2D for avatar pictures
8. Load blank Texture2D for character selection highlight
9. Check if left or right on the DPad has been pressed by using newPad && !oldPad
   1. Shift index left or right accordingly
   2. If index == 0 && DPad.left, index == 4
   3. If index == 4 && DPad.Right, index == 0
   4. Character selection box Rectangle

= new(avatarArray [<index>].X - 10, avatarArray [<index>].Y - 10, avatarArray

[<index>].Width + 20, avatarArray [<index>].Height + 20)

1. Check if Start has been pressed by using newPad && !oldPad
   1. Set clicked to true
   2. Set new rectangle for the avatarTexture
   3. Disable character selection box
2. Check if Back has been pressed by using newPad && !oldPad
   1. Set clicked to false
3. oldPad = newPad
4. Draw all avatars in array if not clicked by using clicked
5. Draw character selection box if not clicked by using clicked
6. Draw enlarged avatar if clicked by using clicked

10 pts

# The Borg

1. Initialize 5 Rectangle variables for the turret and 4 pipes, which should all be connected and well placed
2. Initialize Boolean fire to check if torpedo has been fired
3. Initialize old and new KeyboardStates
4. Initialize tubeIndex
5. Initialize power, size, and timer
6. Initialize randomTube, randomTime, and randomDistance for Borg ships
7. Load Texture2Ds for the turret and 4 pipes
8. Load Texture2D for torpedo
9. Add to timer on every Update
10. If timer % 60 == 0, add 3 to power until it reaches 99
11. If randomTime == 0, make new values for ship variables with Math.Random()
    1. Set Rectangle for ship using randomTube and randomDistance
    2. Subtract from duration each time
12. Check if any number pad key has been pressed by using newKB && !oldKB
    1. If energy >= <num>, subtract <num> based on the key
13. Check if any arrow key has been pressed by using newKB && !oldKB
    1. Update tubeIndex accordingly (0 - up, 1 - right, 2 - down, 3 - left)
    2. Initialize torpedo Rectangle according to tubeIndex
14. Check if space has been pressed by using newKB && !oldKB
    1. Make fire true
    2. Change torpedo Rectangle parameters based on size
15. As long as fire = true, torpedo must move in the right direction (X+, X-, Y+, Y-)
16. Once torpedo goes off screen (X < 0 || X > 800 || Y < 0 || Y > 480), fire = false
17. oldKB = newKB
18. Draw all tubes and turrets
19. Draw selected tube with green color if !fired and red color if fired
20. Draw torpedo if fire = true
21. DrawString by using power available in real time

# More Borg

(Check earlier Borg for existing progress)

1. Initialize shipTorpedo Rectangle
2. Initialize oldMouse and new Mouse
3. Initialize torpedoPower
4. Load Texture2D for shipTorpedo
5. Load Texture2D for power bars
6. Use numbers input to determine size by using newKB && !oldKB
7. Shoot shipTorpedo towards turret
8. If shipTorpedo reaches turret or their randomTime is over
   1. Respawn ship and shipTorpedo
9. Set tubeIndex based on the mouse X and Y coordinates
   1. Find distance from the mouse to each tube
   2. Find the minimum amongst those tubes and set the tubeIndex accordingly
10. If mouse is clicked using newMouse && !oldMouse
    1. Set fire to true
    2. Change torpedo Rectangle parameters based on size
11. oldMouse = newMouse
12. Draw shipTorpedo
13. Draw Rectangles for charge remaining, torpedo power, and torpedo size using available variables
    1. Width of Rectangles depends on variable value

# Rocket Man

1. Initialize Rectangle for spaceship
2. Initialize GamePad
3. Initialize speed
4. Load Texture2D with spaceship image
5. Check if left thumbstick was used by using newPad && !oldPad
   1. If 1 < speed < 20,
   2. If thumbstick was up, increase speed
   3. If thumbstick was down, decrease speed
6. Check if right thumbstick was used
   1. Update spaceship Rectangle’s X and Y coordinates using thumbstick’s X and Y values
   2. If spaceship.X < 0, spaceship.X = 800
   3. If spaceship.X > 800, spaceship.X = 0
   4. If spaceship.Y < 0, spaceship.Y = 480
   5. If spaceship.Y > 480, spaceship.Y = 0
7. Initialize heading = tan inverse(thumbstick.Y / thumbstick.X)
8. oldPad = newPad
9. Draw spaceship
10. DrawString of speed
11. DrawString of heading

15 pts

# Mini Map

1. Initialize bigger-than-screen background and mini-map Rectangle
2. Initialize Rectangles for objects and character
3. Initialize Rectangles for all mini-map objects and character indicators
4. Initialize GamePad
5. Load Texture2Ds for background and mini-map
6. Load Texture2Ds for objects and character
7. Update character’s X and Y coordinate using GamePad’s X and Y values
   1. If character.X > 800, shift everything to the left by 800
   2. If character.X < 0, shift everything to the right by 800
8. Update mini-map elements by scaling down each actual element’s parameters
9. oldPad = newPad
10. Draw background and mini-map
11. Draw actual objects and character
12. Draw scaled down objects and character with simple Texture2Ds (background) having a single Color

30 pts

# Max Borg

(Check earlier Borg for existing progress)

1. Initialize old and new GamePad
2. Initialize phaserFire
3. Initialize phaserColor
4. Initialize phaserRectangle
5. Load Texture2D for explosion image
6. Load SoundEffect for shooting torpedos, shipTorpedos, and phasors
7. Play SoundEffects right after torpedo / shipTorpedo / phasor are shot
8. Detect press on DPad using newPad && !oldPad
   1. Change tubeIndex accordingly
9. Detect left thumbstick using newPad && !oldPad
   1. Change size and power accordingly
10. Detect left trigger using newPad && !oldPad
    1. fire = true
11. Detect A/B/X/Y using newPad && !oldPad
    1. phaserFire = true
    2. Set phaserColor based on button pressed
    3. Set phaserRectangle based on the button pressed and the tubeIndex
12. Detect right trigger using newPad && !oldPad
    1. phaserFire = true
13. Make torpedoPower a ratio of distance to the edge of the screen remaining
    1. (<num> \* distance to screen / 9.0)
14. Once torpedo reaches end or duration == 0
    1. Initialize explosiveTimer
    2. Decrease explosiveTimer on each Update
15. When explosiveTimer = 0
    1. Reset torpedo
16. If explosiveTimer > 0
    1. Change Texture2D for torpedo to explosive
17. Update power values using duration left for flight
18. If phaserFire = true
    1. Draw phasor