5 pts

# File Reader Demo

1. Follow instructions
2. Test program

# What Did You Say

1. Initialize List<String> to house all lines from the text file
2. LoadContent previously implemented ReadFileAsText method to parse each line as String and Add to List<String>
3. Loop through half of List<String>.Count and DrawString all pairs of String → [i] + [i+1]

# Slide Show

1. Initialize int current Rectangle and timer and List<Rectangle> for all Rectangles
2. Load previously implemented ReadFile method to parse each line as String and overall sprite sheet
3. Split each line of String and ParseInt into Rectangle constructor and Add Rectangle to List<String>
4. Draw destination Rectangle from sprite sheet in List<Rectangle> to the center of the screen
5. Add to current if timer % 120 == 0

# Start Me Up

1. Create enum GameState with Start, Play, Quit
2. Initialize gameState to Start and old KeyboardState and Texture2D[]
3. Load all Texture2D into Texture2D[]
4. If enter is pressed in Update, change gameState as needed and reinstantiate KeyboardState
5. Based on gameState, Draw appropriate Texture2D from Texture2D[] on screen

# Checkerboard

1. Initialize Texture2D[ , ]
2. Load Texture2D’s for red and black
3. Draw using Texture2D[ , ] and use loop variables i and j for location of Rectangle and make them all equal size

10 pts

# Space Invaders #2

1. Initialize Alien class
   1. Initialize Texture2D, Rectangle, and int speed in constructor
   2. Make Move(int, bool) method that moves Alien either on the side or down
   3. Make changeSpeed to reverse Alien movement direction
   4. Make Update(bool) method for one step of the Alien movement that will subsequently call Move and changeSpeed as needed
2. Initialize Alien[ , ] and Texture2D[ , ] which are both parallel and SoundEffect
3. Load all Texture2D into Texture2D[ , ] for Alien images and later all Alien objects into Alien[ , ] and SoundEffect which must be Play
4. In Update, for every second
   1. Check if group is at the edge of the screen and call Update as needed
5. Draw all Alien in Alien[ , ]
6. When Alien[ , ] is outside screen
   1. this.Initialize
   2. UnloadContent()
      1. SoundEffect.Stop

30 pts

# Breakout

1. Create enum GameState with MainMenu, Pause, Level1, Level2, GameOver
2. Change Window dimensions
3. Initialize Brick class
   1. Initialize Rectangle, Texture2D, int hits in constructor
   2. Make HasHit() to check if Ball.Rectangle hit Brick.Rectangle
   3. Make bool isDestroyed() to check if Brick.hits = 0 and let main class know when to remove Brick from memory
4. Initialize Bat class
   1. Initialize Rectangle and Texture2D in constructor
   2. Make Move(bool) that moves the Bat based on Keyboard input in main class
   3. Make Draw(SpriteBatch) to show Bat on screen
5. Initialize Ball class
   1. Initialize Rectangle, Texture2D, and int speedX and speedY in constructor
   2. Make Update() that moves ball for each Update in main class
   3. Make SwitchX() and SwitchY() to reverse speedX and speedY when needed
   4. Make Draw(SpriteBatch) to show Ball on screen
6. Set GameState to MainMenu
7. Initialize List<Texture2D> backgrounds, List<Brick>, Bat, Ball, int lives, KeyboardState
8. Load backgrounds and bricks Texture2D’s
9. Make ReadFile(String) to parse text file and set up List<Brick>
   1. For every character in each line,
      1. Create Rectangle based on x and y values
      2. Load correct texture based on char type
10. Update game with Keyboard input
    1. Get new KeyboardState
    2. If esc is pressed
       1. During game, set GameState to Pause
       2. Else, exit Game1
    3. If enter is pressed
       1. During Pause, set GameState back to Level1 or Level2
       2. During MainMenu, set GameState to Level1
       3. During GameOver, set GameState to MainMenu
    4. During Level1, if List<Brick>.Count = 0, set GameState to Level2
    5. During Level1 or Level2,
       1. If left or right is pressed, Bat.Move(true) or Bat.Move(false)
       2. Ball.Update()
       3. If Ball.Rectangle.Intersects(Bat.Rectangle), Switch(Y)
       4. If Ball.Rectangle.Intersects(Window.Left) or Ball.Rectangle.Intersects(Window.Right), Switch(X)
       5. If Ball.Rectangle.Intersects(Window.Top), Switch(Y)
       6. If Ball.Rectangle.Intersects(Window.Bottom), int lives--
       7. Foreach Brick in List<Brick>,
          1. If Ball.Rectangle.Intersects(Brick.Left) or Ball.Rectangle.Intersects(Brick.Right),
             1. Brick.hits--
             2. Ball.SwitchX()
          2. If Ball.Rectangle.Intersects(Brick.Top) or Ball.Rectangle.Intersects(Brick.Bottom),
             1. Brick.hits--
             2. Ball.SwitchY()
          3. If Brick.isDestroyed(), List<Brick>.Remove(Brick)
11. Draw based on GameState
    1. If GameState = MainMenu
       1. Draw using List<Texture2D> backgrounds
    2. If GameState = Pause
       1. Draw using List<Texture2D> backgrounds
    3. If GameState = GameOver
       1. Draw using List<Texture2D> backgrounds
    4. If GameState = Level1 or GameState = Level2
       1. Draw using List<Texture2D> backgrounds
       2. Ball.Draw()
       3. Bat.Draw()
       4. List<Bricks>.Draw()