**Vashista Gande**

**Aladdin The Great 10% Program Design**

**Summary:**

* The game uses mathematical formulas and physics calculations to make coins fall of the screen, mathematical formulas such as sine, cos and tan functions.
* A function is created such as, in order to form user the mat and coins to fall, and it is also used to fall of various coins to fall off the screen the line, the location values are randomly assigned, so they fall at different time and different location.
* All the components in uses GDI

**Class Structures:**

**Intro.cs Partial Class Structure:**

* public partial class Form1: Form
* InitializeComponent();
* Public Static Class DataContainer
  + Data is inside a public static class and not in the partial class so other forms can access it and information can be passed around
* private void btnExit\_Clickt : Exits Entire Application
* private void btnStart\_Click: Continues to Game.cs
* private void btnHighscore\_Click: Continues to TheCircle.cs
* private void btnInstructions\_Click: Messagebox shows controls, objectives and tutorial videos
* End of Class

**Highscore.cs Partial Class Structure (WIP):**

* WORK IN PROGRESS

**Game.cs Partial Class Structure:**

* Form Initialized
* InitializeComponent();
* private void btnKeyDown\_Keys: uses Keys to play the game

**Game.cs Partial Class Structure:**

* Form initialized
* InitializeComponent();
* private void btnStart\_Click: The game starts with the mat at the centre of the screen and bottom of the screen
  + Mass declaration of local variables
  + public static void CheckWin(): Checks if player has lost every tick
  + private void Game\_Load: Loads all the components
  + private void Location\_Tick: Controls the location of the line
  + private void Circle\_Tick: Controls and executes the location and movement of all the coins
* private void Game\_KeyDown: Controls player movements by checking if arrow keys have been pressed to change the location of the mat
* private void Game\_Tick: Controls the Stars and other additional boosts in the game
* End of Class

**Pseudocode:**

**Form1:**

* If btnExit is pressed, exit the application -> END
* If btnStart is pressed, continue to ame.cs
* If btnHighScore is pressed, continue to HighScore.cs
* If btnInstruction is pressed, show tutorial messages through the message box
* End of Form

**Highscore:**

* Checks and add score

**LevelSelect:**

* + If btnSart is pressed, it continues to a different form when it is pressed, update the preview images and text
  + Get player’s selected color of the circle and name
  + When btnContinue is pressed, update the username and player image and continue to Game.cs
  + End of Form

**Game.cs:**

* + Declare local variables and enumerations
  + Load picture box components
  + Start all game timer
  + Checks if the user is hitting space
  + Keeps track of the mat movements and also the values of the function
  + End of Form

**Important Methods:**

**public static void Dop():** Uses Sine functions to drop the coins

**public static void Coins():** Controls the movement of the various coins

**public static void Score():** Keeps track of the score and boosts. It also functions to keep track of the

**Libraries:**

I will be requiring the default Windows form libraries plus one additional library, the system IO library so I will be able to write onto a text file to save and read high scores.

**List of Libraries:**

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Windows.Forms;

using System.IO

using System.Drawing;

using System.Linq;

using System.Text;