

## **Amazeballs: Demonstration Instructions**

*Project:* Doodle aMazing

*Team:* Amazeballs

*Members:* Matt Beachly, Glenn Kiefer, Frank Smith

### Instructions: Playing the Game on Android

#### Installation and Startup

1. Connect an Android device to your PC using a USB cable
2. Open windows explorer and find the connected device under My Computer
3. Browse the file structure on the Android device to a folder such as the Download folder
4. Copy the Amazeballs.apk file from your computer to the Android device
5. On the Android device, open its file browser (usually called My Files, or Files, or File Browser)
6. Browse to the Amazeballs.apk file (might be found under Installation files) and click on it until an Install popup appears
7. Install the Amazeballs.apk file
8. Go to Apps on the device and find the installed AmazeBalls application
9. Launch the AmazeBalls application
10. The Amazeballs intro screen loads after the Unity screen, you can click anywhere on the screen to create more balls, then click on Start to proceed

#### Viewing Instructions

1. On the main menu screen click Instructions
2. On the Instructions screen use the scroll bar to read text, then click Back to return to the main menu

## Creating a Maze (Taking a Picture)

1. On the main menu screen click New Maze, say No if you are asked to load last saved game
2. On the camera screen click Start Camera
3. A popup will appear asking if you wish to allow Amazeballs to take pictures, click allow
4. On the camera screen again click Start Camera again
5. The camera view will appear, aim it at a maze drawing (see next section), then click Capture Image (same button as Start Camera) to take that picture
6. The camera turns off and the maze game will load with the captured image

## Drawing a Maze

1. Draw the maze on a sheet of printer paper or other some white background with a dark black marker
  - Don't draw lines too close together or you will need to shrink the ball size in Options menu
2. Draw a light blue spot where you want the ball to start
3. Draw a light red spot where you want the ball to end
  - Dark spots might get interpreted as walls and not be reachable

## Playing a Maze

1. The start and end point will be automatically detected from the biggest blue spot and red spot, respectively
  - You can disable Detect Start/End in Options, in which case the game will ask you to tap where you want to start and end
2. Hold the device so that the surface is flat/horizontal
3. Tilt the device to roll the ball down towards that direction
4. Navigate the ball through the maze to the red end point
5. Touch the Quit button if there you are unable to reach the end point
  - You can adjust ball size, ball speed, and edge threshold in Options to make the maze more navigable
6. If you make it to the You Won screen, click Replay if you wish to play the maze again, or Main Menu to return to the Main Menu

## Selecting a Maze

1. On the main menu screen click Select Maze
2. If there is a saved maze you will be asked if you want to load it, click YES to replay the saved maze or click NO if you want to select a different maze

3. Click the Select Your Maze button to access the phone's photos, you can choose either Photos or Gallery
4. Select the photo you wish to load as a maze
5. The selected photo should appear in the background, click Play Game
6. Begin playing the maze if start point, end point, edges, and ball size look correct
7. If settings need to be adjusted click Quit and go to Options, the maze will be saved so that you can load it again quickly from Select Your Maze

#### Options: Changing Ball Settings

1. On the main menu screen go to Options
2. Drag the slider under Ball Size left to make the ball smaller, or right to make the ball bigger
3. Drag the slider under Ball Speed left to make the ball slower, or right to make the ball faster
4. Click on a ball theme to choose space, smiley face, or marble
5. Click Back to return to main menu, the options will take effect the next time you play a maze

#### Options: Changing Edge Detection Threshold

1. Go to Options then Advanced
2. Drag the slider under Edge Threshold left to only interpret darker areas as walls (making fewer wall areas)
3. Drag the slider under Edge Threshold right to make interpret less dark areas as walls (making more wall areas)
4. Click Back to return to main menu, the options will take effect the next time you play a maze

#### Options: Disable or Enable Automatic Start and End Point Detection

1. Go to Options then Advanced
2. Uncheck the Detect Start/End check box to make the game always ask you where you would like to place the start and end points
3. Check the Detect Start/End check box to let the game try to automatically detect the start and end points based on widest blue and red areas, respectively
  - The game will still ask you to place start and end points if it does not find suitable blue and red areas
4. Click Back to return to main menu, the options will take effect the next time you play a maze

### Options: Changing Start Point Detection Threshold

1. Go to Options then Advanced
2. Drag the slider with blue shades left to only detect very-blue areas for the start point
3. Drag the slider with blue shades right to detect less-blue areas for the start point
4. Click Back to return to main menu, the options will take effect the next time you play a maze

### Options: Changing End Point Detection Threshold

1. Go to Options then Advanced
2. Drag the slider with red shades left to only detect very-red areas for the end point
3. Drag the slider with red shades right to detect less-red areas for the end point
4. Click Back to return to main menu, the options will take effect the next time you play a maze

### Deleting Saved Maze

1. On the main menu screen click Delete Maze Data, this will delete the maze file saved on your phone that is normally retained between loading the app
2. The next time you go to New Maze or Select Maze you will not be asked if you want to load the last saved game

### Instructions: Viewing the Unity Project and Scripts on PC

#### Installing Unity

1. Go to <https://unity3d.com/get-unity/download> and click Download Unity Hub
2. Run the UnityHubSetup.exe file that downloads
3. Click I Agree, Install and then Finish
4. Run Unity Hub
5. Click the person icon in the corner, click Sign in
6. Create a Unity ID if you don't have one, otherwise enter your email and password and click Accept
7. Click Activate New License, select Unity Personal, I don't use Unity in a professional capacity, and then DONE
8. Click the corner arrow to go back until you reach Projects, then click ADD

9. Browse to the folder containing the Amazeballs project and select it (it can be downloaded from <https://github.com/mbeachly/Amazeballs>)
10. The Amazeballs project will now be listed under Projects, but now requires that the proper version of Unity Editor be installed
11. Click the project and click INSTALL when the popup appears showing that the required Editor version is missing
12. The Add Unity Version popup will appear
  - If you don't have Visual Studio installed select Microsoft Visual Studio Community 2019
  - If you need to build the game and deploy it on Android select Android Build Support, Android SDK & NDK Tools, and OpenJDK
13. Click Install and allow Unity Hub to make changes to your device, the install may take 20 minutes
14. Once installs are finished, go back to Projects and double-click the Amazeballs project to open it in Unity Editor (skip new version if it asks)

## Viewing Scenes

1. Inside Unity Editor in the Amazeballs project go to the Project tab, then Assets>Scenes to view all the scenes the game is composed of:
  - Intro
  - MainMenu
  - Instructions
  - Options
  - OptionsAdv
  - CaptureImage
  - SelectMaze
  - LoadSavedMaze
  - WinGame
2. The first scene encountered in the game is the Intro Scene, select and open it for this example
3. The Intro contains the Amazeballs title (created with TextMeshPro), Start button, and 2D physics ball sprites that collide with 2D wall sprites
4. The Scene tab shows that the scene is actually composed from two different areas:
  - The Canvas box holds the text and button objects (and is 100 times larger than the sprites for some reason)
  - The Camera box contains the 2D sprites and background, it is located at the bottom left corner of the Canvas so you must zoom in close with the scroll button to be able to see it

5. The Hierarchy tab shows all the objects by name and order of what objects are children of other objects
  - Expand objects with the triangle button to see their children
  - For example, Main Camera has ButtonCanvas as a child, which has StartButton as a child, which has StartText as a child
  - If objects on the same Hierarchy level overlap on screen, lower objects in the hierarchy will be appear in front of objects above them
6. The Inspector tab shows properties of objects such as position, scale, color, text, physics, and scripts
  - Select Start Button and look at the Inspector, in the Button (Script) section it shows that the `OnClick()` event calls the `SceneLoader.LoadMainMenu` function of the Scene Loader object to change to the next scene
  - Select the Scene Loader object to see that it contains the `SceneLoader.cs` script
  - Not-so-obvious objects can have scripts, for example, Main Camera calls the `instantiateBall.cs` script to make 2D physics balls appear wherever the screen is touched

# AmazeBalls Flow Chart

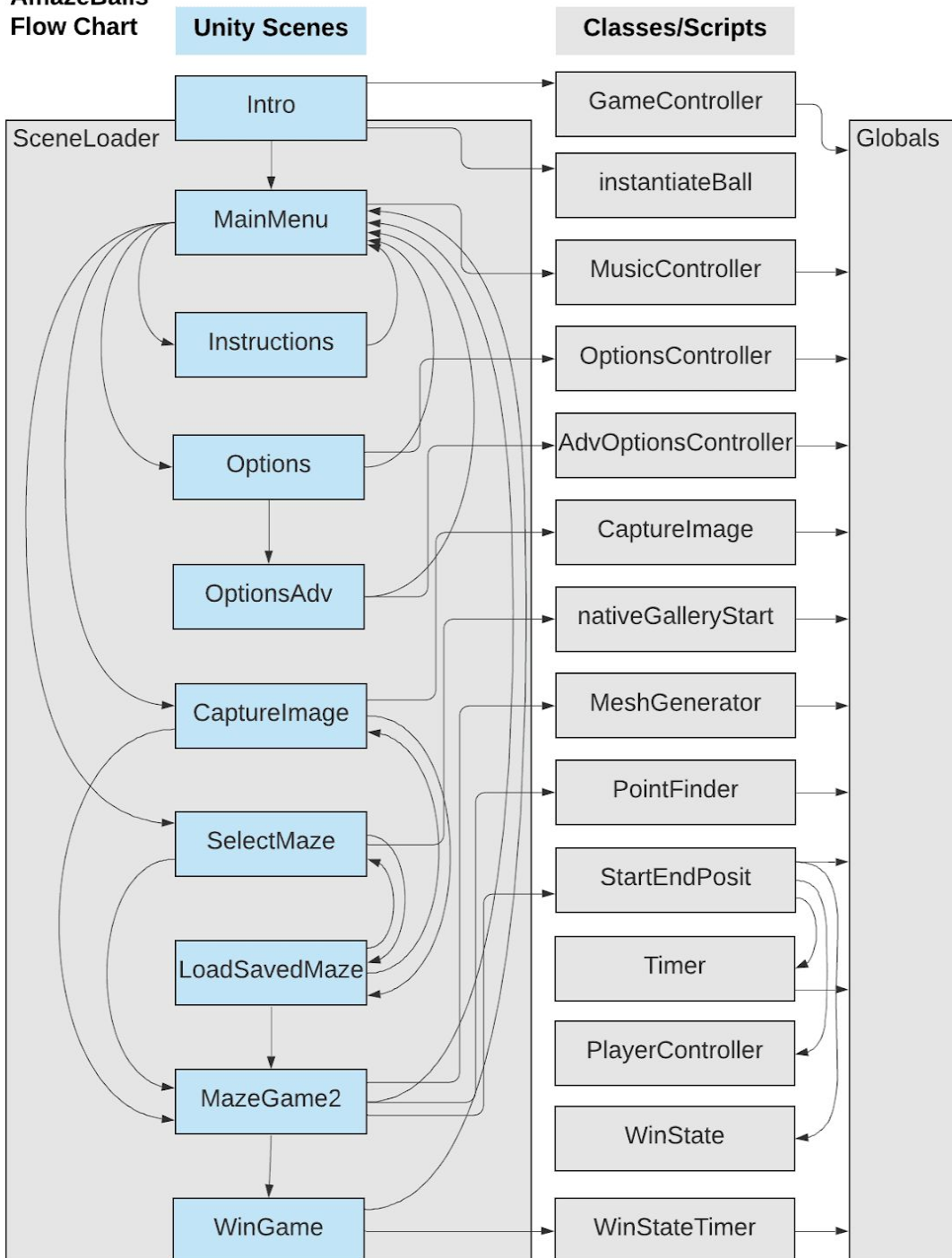


Figure: This flowchart shows how scenes and scripts/classes interact in Amazeballs. The SceneLoader class is called by every scene in order to change scenes. Each scene also has at least one class used only by that scene. Most classes interact with the Globals class to get or set variables that need to be used across multiple scenes.

### Viewing Scripts

1. Each script contains a single class, which contains methods and variables
2. We located all our scripts in one project folder to be able to find them easily; in the Project tab go to Assets>Scripts to view all the scripts:
  - SceneLoader.cs
  - Globals.cs
  - GameController.cs
  - instantiateBall.cs
  - MusicController.cs
  - OptionsController.cs
  - AdvOptionsController
  - CaptureImage.cs
  - nativeGalleryStart.cs
  - MeshGenerator.cs
  - PointFinder.cs
  - StartEndPosit.cs
  - Timer.cs
  - PlayerController.cs
  - WinState.cs
  - WinStateTimer.cs
3. Double-click a script such as SceneLoader.cs to open it in Visual Studio
4. Each class contains variables and methods used by a particular scene, or used by many scenes in the case of Globals and SceneLoader
5. To debug a script, place break points on the script in Visual Studio, then click Attach to Unity (green triangle), then in Unity Editor click Play (black triangle)
  - Note: play will cause Unity to hang in most scenes if you do not start play from the Intro scene
6. Continue debugging in Visual studio when a breakpoint is hit
7. To stop debugging both stop debugging in Visual Studio and stop play in Unity Editor

### Viewing Prefabs



1. While the MazeGame2 scene is the most important scene in Amazeballs, it shows almost nothing in the editor because the mesh, ball, endpoint, and text boxes are created by scripts at runtime
2. The ball, endpoint, and text prompts for manually picking start and end points are saved as prefab objects
3. To view all Prefab objects go to Assets>Prefabs:
  - Player
  - End
  - StartObj
  - EndObj
  - Others were not used in final game
4. To view a single Prefab object select one such as StartObj
5. Now right-click in the scene area to look around and see that this prefab contains a text box that says "Select Your Start Position!"
6. Select the Player or End prefabs, then look in Inspector to see what physics settings and scripts they contain

#### Viewing Sprites

1. Sprites are basically 2D prefabs, find them in Assets>Sprites
2. Select a sprite such as ballObj to view its settings in inspector

#### Viewing Other Resources

1. Go to Assets>Resources
2. Here we store textures that can be applied to the player ball, default maze images, and the music track for the game