**References**

* <http://unity3d.com/5>
  + Homepage of the Unity engine.
* <http://docs.unity3d.com/Manual/index.html>
  + Unity documentation.

**Software Documentation (scripts)**

* AutoAddUIToCanvas.cs
  + Automatically adds the in-game UI to the canvas to be displayed during runtime.
    - Score Text.
    - Hearts.
* AutoRouteThroughMainMenu.cs
  + Displays log error to inform testers how to test correctly when attempting to start the game outside the Main Menu.
  + "When testing you must hit play from the main\_menu level. Then press the number on the keyboard for the index of the level you want to load.\n Make sure the level you want is added in File->Build Options".
* BaseEnemy.cs
  + Movement patterns and death actions.
  + Registers collision boxes and behavior.
  + Incoming damage and health reduction.
* CameraFollow.cs
  + Finds the player and moves the camera along with the player according to the follow speed.
* DebugLevelLoader.cs
  + Allows testers to load to their desired level from the starting menu based on number key pressed.
* DontDestroyMusic.cs
  + Allows music to be persistent throughout game
* EndOfLevelPoint.cs
  + Loads the next level when the player enters the end level collision box.
* KillPlayerVolume.cs
  + “Kill zone” placed below the levels that removes the player’s remaining health when the player enters the collision box.
* LevelManager.cs
  + Adds to the score
  + Times level loading and displays loading screen if necessary
  + Loads levels and destroys main menu buttons on level load
* PlayerController.cs
  + Handles interaction between PlayerView and PlayerModel
* PlayerModel.cs
  + Physics for moving and jumping
  + Handles damage taken and death
  + Collision and grounding functionality
  + Respawns player and reloads current level
* PlayerView.cs
  + Handles user input
  + Updates health display
  + Changes player sprite when player is hurt and back when hurting stops
  + Switches to death sprite on death
  + Displays respawn button on death
* SpikeTrap.cs
  + Deals damage to player if player enters collision box
* TimerManager.cs
  + Contains list of structs that consist of pointers to functions and times for the functions to be called and calls any function whose time has been passed.
  + Increments time based on time since last frame.
* UIGroup.cs
  + Displays and hides UI components