Project Plan

* Project Scope:
  + Deliverables
    - Demo of game
    - Documentation – Game play, technical specs and general code documentation
    - Happiness
  + Specific Project Goals
    - To demonstrate the physics engine of Unity 3D using C# Scripts
    - To immerse the user in a Virtual Reality environment using the Oculus Rift
      * Develop 3-5 Carnival Games
        + Ring Toss
        + Ball Toss Milk Jugs
        + Skee Ball
        + Dunk Tank (w/ Taunting Comments)
        + Squirt Gun Game
        + Free Throw Game
        + Saloon Shooting Game
        + Hammer Strength Game
        + Ferris Wheel?
  + Tasks
    - Build Room Template for standardization – (Shaheer)
    - Find and download models for different objects - (Shaheer)
    - Learn 3D Physics in Unity (Charlie)
  + Costs
    - $15 on Udemy to learn Unity 3d physics
  + Deadlines
    - First Game by March 30th -- Slack time to April 1st
* Project schedule
  + April 29th Deadline
* Project team organization
  + Charles Davis – Game Maker/Code Monkey
  + Shaheer Mahdi Jilanee – Code Monkey/Game Maker/VR Specialist/3D Object Juggler/Game Polisher
  + Patrick McDermott – Code Monkey/Game Maker/Tester/Game Breaker
  + Brady Rainey – Developer/Technical Writer
  + Philip Dukleth -- Game Maker/Code Monkey
* Technical description of system
  + Intel i5-4590 equivalent or greater.
  + NVIDIA GTX 970 or AMD 290 equivalent or greater.
  + 8GB of RAM.
  + HDMI 1.3 (or higher) video output.
  + Two USB 3.0 ports.
* Project standards and procedures
  + All games must have a defined goal that is obtainable by the player.
  + All games must be intuitive for a new player and have explained instructions.
* Quality assurance plan
  + First Line of Testing – Patrick McDermott
  + Second Line of Testing – Shaheer Mahdi Jilanee
  + And general testing by everyone else
* Configuration management plan
  + Don’t really know yet. We're working on it.
* Documentation plan
  + 3 Documents.
    - User Guide – This is a general manual found in most games. If you own any games, look at their manual to see how to properly structure.
    - Technical Manual – This acts with the internal programming and documents It.
    - Code Documentation - General Code Documentation In Code
* Data management plan
  + Data File Formats?
  + Naming Conventions
* Resource management plan
  + 3D Models – Open source or public domain. [Blender Repository](http://www.blender-models.com/)
  + [Unity Asset Store](https://www.assetstore.unity3d.com/en/)
* Test plan
  + Patrick – Tester.
* Training plan
  + Not necessary. Should be self explanatory as part of the game
* Security plan
  + None required, because offline game.
* Risk management plan
  + Risk: Not finishing. Management: Make sure we finish it.
* Maintenance plan
  + Patches in the future
* List of hardware and software
  + Oculus Rift DK2
  + Laptop/Desktop Computers
  + Unity 3D Engine and related software
* Standards and methods, such as
  + Algorithms
    - Basic C# Scripts
  + Tools
    - Unity 3D Game Engine
    - Oculus Rift
  + Review or Inspection Techniques
    - Patrick – Inspector
  + Design Language or Representations
    - UML
  + Coding Languages
    - C# and Unity
  + Testing Techniques
    - Playing the game. A lot
    - Check for clear documentation and object oriented programs in the code.
    - No spaghetti code.
    - All variables must be self-exclamatory. Any unusually named variables will fail QA.
    - Please separate variables from functions and loops.
    - Little to no self-entering code. If possible, use APIs