University of Nevada, Reno
Computer Science and Engineering

Virtual Reality Physics Lab

Project Part 3: Acceptance Criteria and Testing Strategy and Plan

Team 10:
Nicholas Bolling
Christopher Lewis
Andrew Munoz
Willem Zandbergen

Instructors
Sergiu Dascalu & Devrin Lee

External Advisors
Eelke Folmer & Benjamin Brown

March 6, 2018

Table of Contents

1 Abstract

The goal of Team 10's Virtual Reality Physics Lab is to utilize a new, revolutionary product for educational purposes. Virtual reality hardware, such as that in the HTC Vive, allows for personalized, immersive experiences which can address all forms of VARK (Visual, Aural, Read/write, Kinesthetic) learning. Very few environments outside of private tutoring have the same potential for an entertaining, educational experience. The VR Lab project will be filling this gap currently present in the virtual reality repertoire. The final product of the VR Lab will provide an immersive and educational experience into kinematic physics while promoting STEM (Science, Technology, Engineering, Mathematics) principles.

2 Project Updates and Changes

The project is a Virtual Reality game in which the user will be placed into a simulation where they will be able to learn about kinematic force concepts while also being able to have a fun and engaging experience. Game elements will be using these learned concepts to complete an objective (main course of action is a basic game of target practice). Since completing project 2 of CS 426, there have been some notable strides. At the end of CS 425 we had many issues. Physics wasn't quite working, teleportation wasn't great, and we couldn't consistently grab an object. Now we have reached a point where we have dash teleportation, functional grabbing of objects, and physics seems to be very accurate. We are in the process of getting our cannon working and in game information to be displayed to the user that provides very consistent feedback.

Unfortunately, we have make a change to our project specification. We will be removing saving and loading from our game for now until the game is fully functional. This direction will allow focus on making the game rather than getting a consistent save/load. When the game is functional and complete we will add the save/loading later as it's not a strong functional requirement but still should exist in a complete build.

3 User Stories and Acceptance Criteria

Main Subsystems (as outlined in Project Assignment #1):

User Interaction - menus, interfaces, and player input - Will

Environment Design - models, textures, and object interaction - Andrew

Gameplay - Difficulty, balancing, and level design - Nick

Game Engine - physics, user score, and save data - Chris

User Stories / Acceptance Criteria:

1. User Interaction
    - As a player in the Virtual Reality Physics Lab, I want to be able to input my name to begin a game session.

    - AC1. In the menu screen, there is an option to input name.
    - AC2. Username is saved after it is submitted.
  - As a player in the Virtual Reality Physics Lab, I want to pick the level/simulation that I want to play.
    - AC1. Menu options displayed with buttons representing different levels.
    - AC2. When a button is pressed, then the player is placed into the level from the menu screen.
2. Environment Design
  - As a player in the Virtual Reality Physics Lab, I want to interact with the world around me and feel fully immersed.
    - AC1. World is displayed to the player using realistic textures and models.
    - AC2. Random objects set around the world space.
    - AC3. Player can interact with the world and collide with objects.
  - As a player in the Virtual Reality Physics Lab, I want to pick up different objects in the game space.
    - AC1. Different objects spawn within the world space when level starts.
    - AC2. Player can go around the world space and interact with different objects.
3. Gameplay
  - As a player in the Virtual Reality Physics Lab, I want to play multiple levels.
    - AC1. There are different levels/simulations available once the game starts.
    - AC2. Levels available are each different.
  - As a player in the Virtual Reality Physics Lab, I want to challenge myself through different difficulties.
    - AC1. The levels will vary in difficulty.
4. Game Engine
  - As a player in the Virtual Reality Physics Lab, I want to be immersed in the gameplay and see realistic physics.
    - AC1. Player motion is tracked by sensors and headset.
    - AC2. World space has physics that relate to real world scenarios.
  - As a player in the Virtual Reality Physics Lab, I want to go through and view my scores and statistics.
    - AC1. In the world space, player statistics and physics displayed on an in-game whiteboard.
    - AC2. Button in the main menu will display score screen after being pressed.
    - AC3. Scores will be saved under the player's profile.
  - As a player in the Virtual Reality Physics Lab, I want to be able to save my game.

- AC1. In the world space, there will be a pause menu where player can save their game.
- AC2. Player inputs their name in the menu screen to create an in game profile.

4 Testing Workflow

Happy Path Workflows

1. Load Simulation/Level
   - 1.1. User will put on Equipment and necessary VR playspace has been configured
   - 1.2. Unity will launch the application and display on the VR Headset Accordingly
   - 1.3. User is able to load into the Main Menu level and interact with objects
   - 1.4. User loads the Kinematics simulation and the level switches and loads
2. Interact with Environment smoothly (reduced VR sickness)
   - 2.1. User is able to move around in the playspace and necessary boundaries show
   - 2.2. User is able to teleport around the world-space (remaining inside the playspace) with little to no VR sickness
   - 2.3. Tracking and display of controllers and objects are accurate and uninhibiting for the user to use or interact with
   - 2.4. Item and object Interaction must react and behave as the user would expect in a real-world physics environment
3. "Win" simulation
   - 3.1. User is able to load into the Kinematics Level
   - 3.2. Firing cannon at desired angle and power (both adjustable) hits target
   - 3.3. Target will display hit when hit, and score will relay back to the user
   - 3.4. Last Target is hit, and score is finalized/saved - Simulation "won"
4. Save/Load simulation
   - 4.1. User is able to create a save file
   - 4.2. Save file will contain player information
   - 4.3. User is able to load said save file from the game menu
   - 4.4. Loading should complete and User is loaded into simulation

Happy Path Validation

1. Load Simulation/Level
    a. Testing will be lighter for this as we simply have only two levels, the main menu and kinematics simulation. To ensure proper functionality, we will always load into the main menu and then start the Kinematics simulation from there, and should only be accessible while the User is in the main menu.
    b. Validation for this Happy Path will be simply being able to load into the simulation after requesting to at the main menu and ensuring that only the Kinematics simulation loads when selected at the menu (and that the user is loaded into the Kinematics simulation subsequently).
    c. Timing - we believe this to be one of our first delivered and tested and finally on the road to validation as it requires us to create our primary two levels and then link them with a button in VR, which will be helpful in testing and validation of other happy paths.
2. Interact with Environment smoothly (reduced VR sickness)
    a. This is extremely important but also extremely hard to judge how to validate as each individual will be different in their tolerances to VR locomotion. However, one of our team members does experience VR sickness to some degree, so Chris will be a quick member to test with.
    b. We have also implemented other smaller details into the game like a cursor to show your teleport location, and objects follow you when you teleport giving a sense of continuity. UI elements at roughly eye to chest level and easy to point at using the controller and cursor tracer will help make things smooth for the user to interact with what they want.
    c. Validation - This validation will be harder than the others as it will require multiple users to comment on objective elements of the simulation, and for proper validation there should be reduced sickness for typically VR sick individuals. UI elements should be implemented with proper cursor and tracer, along with teleportation cursor.
    d. Timing - This is a tricky one to time as it will be fluctuating throughout development as we implement our features, but we should be incorporating elements of this concept/happy path while doing our other paths.
3. “Win” simulation
    a. This will be tested/played much more frequently than many of our other components as this is the main simulation, ensuring proper target collision with projectiles and counting score will be important functionalities to test.

b. Validation for this is extremely easy - only when all the targets are hit will the proper win sequence start. Ensuring only allowed win conditions trigger the win sequence.
c. Timing - This will be our next deliverable after loading the simulations/levels, as this is the objective of our game and essentially the focus of the assignment as many elements will be required to fit together to make this a coherent experience and one that will function together correctly.

4. Save/Load simulation
    a. Saving and Loading a simulation is a way to personalize the simulation and give the User a sense of pride and accomplishment when achieving their goal and winning the simulation and subsequently saving.
    b. Validation for this happy path will be simply being able to store and recall player variables (name, previous score, ect.) at will through the save/load feature.
    c. Timing - This is our last deliverable as most of the game will be setup with our gameplay and then saving attributes will be the final step in the journey to ensure proper completion and returning to level should also function with the load properly, but these need to be implemented after the main functionality is finished.

Unhappy Path Workflows

1. Components fail to load (VR Headset)
    1.1. Unity does not recognize the VR Headset and only plays on the computer
    1.2. VR Equipment (Headset, Lighthouses, and controllers) are damaged or unplugged
    1.3. Proper Drivers or Prerequisite software not installed or corrupt
2. Requirements not met for running application
    2.1. User must have Prerequisite software installed:
        2.1.1. Steam
        2.1.2. Vive Drivers
        2.1.3. Video Drivers
    2.2. Hardware requirements will be similar to running the HTC Vive at normal capacity:
        2.2.1. GPU: Nvidia GeForce GTX 970, AMD Radeon R9 290 equivalent or better
        2.2.2. CPU: Intel i5-4590, AMD FX 8350 equivalent or better
        2.2.3. RAM: 4 GB or more
        2.2.4. Video Output: HDMI 1.4, DisplayPort 1.2 or newer
        2.2.5. USB Port: 1x USB 2.0 or better port

                2.2.5.1. Operating System: Windows 7 SP1, Windows 8.1 or later, Windows 10

3. Unable to "win" simulation
    3.1. Simulation does not "finish" after last target is hit
    3.2. Simulation does not accumulate or store score correctly
    3.3. User is unable to interact or move cannon to hit enemies

Unhappy Path Validation

1. Components fail to load (VR Headset)
    a. This will be tested with unplugging and trying to run the game without components connected, with some connected (Mix of all the different combinations), and all components connected.
    b. This should only launch and work properly when all components are successfully loaded and working - Validation will be load when all connected, and warning/failure to launch without VR hardware.
    c. Timing on unhappy paths are important to keep in mind as these are potentially VR sickness inducing, part of our happy paths mentioned above. Proper hardware drivers are not developed by us, only HTC and Nvidia (or AMD) and only interfaced with so many of these should be on the manufacturer to continue supporting our HTC Vive and respective GPUs - many of these are taken care of by the interfacing done through these and should be maintained continuously leaving it another continuously working path.
2. Requirements not met for running application
    a. For this unhappy path validation, we will be testing the game on a handful of computers ranging from low to high end hardware, both on the desktop and on the laptop - only those who pass minimum specifications should run the application.
    b. Requirements should be fairly strict, so attempting to run these applications without minimum hardware specifications will not launch the application - Validation will be refusal to run on low-end hardware.
    c. Timing - This should be similar to components failing to load, however this will be even less a worry as many users will have appropriate hardware. This is something that needs to be taken care of first however, as we need the higher end to be able to develop and test on effectively to meet other paths validations.
3. Unable to "win" simulation
    a. This will have much more rigorous testing, as it is the end goal of the simulation and the goal of the User in the simulation.
    b. Testing will be done on a number of computers, both at home and in the VR lab provided on campus, to ensure "winning" is always a possibility

c. Errors in collisions, target placements, and projectiles not launching will all need to be properly fixed to ensure winnable gameplay.
d. Validation for this will be being able to load in, and from start to finish play the simulation and ensure proper "win" scenarios.
e. Timing - This should be completed similar time as the "Win" simulation happy path as we will be using the best coding practices to ensure that invalid wins do not happen while we achieve a proper and validated win.

5 Testing Strategy

Tests to be conducted

- Acceptance tests
    - Ensure that the project adheres to the team's promises.
- Unit tests
    - Make sure that the game world is properly built and that players have a game to play.
- User tests
    - These tests will be necessary since the project his heavily focused on player experience.
- Manual exploratory tests
    - Input from players will not always be available and game breaking bugs might not be found by players.

Testing Responsibilities

The team agreed to be responsible for testing the subsystems they were assigned as described in '3 User Stories and Acceptance Criteria'. Testing will occur intermittently during development to confirm that systems are working properly. Continuous testing will start after feedback is given from stakeholders, which is estimated to occur early April.

Handling Defects

Defects found by a tester will be reported to the team through two programs. The first would be the team's primary mode of contact, Discord. The second would be through GitHub, by creating an issue related to the affected branch. The tester who found the issue and the team member responsible for the subsystem the issue was found in would coordinate to replicate and then fix the issue.

Project Completion

The project will be marked complete after the Test Plan shown below is filled out, and when no additional actions need to be taken. Ultimately, if the actual results meet the expected results, the project has reached completion.

Test Plan

Test #	Test Type	Target File or Screen	Test Name	Purpose of Test	Test Data or Situation	Expected Result	Actual Result	Outcome and Actions Required
1	User Interaction	Menu.unity	Player input	The player can begin the game.		A player profile can be made. The player can select the level they want to play.		
2	Environment Design	Prototype.unity	Immersion	The world looks and acts believable.		The game environment is a realistic scene. The player can interact with the game world.		
3	Gameplay	Prototype.unity	Fun	The player will be able to play an enjoyable game.		The levels load with different challenges. Each level is more challenging than the previous level.		
4	Game Engine	Prototype.unity	Mechanics	The game runs well and looks appealing.		Player interaction with the world is noticeable and accurate. Player progress is recorded and tallied.		
5	Load Simulation/ Level	Menu.unity	Progression	Levels load correctly and the simulation runs.		The player can run the game. The player can load into a working level.		
6	Smooth Interaction	Prototype.unity	Reception	The player is physically immersed in the		The player experiences no “VR sickness”. The player is not		

				game world.		disconnected from the experience.		
7	Win Condition	Prototype.unity	Completion	The challenges can be completed in a satisfying manner.		The player knows that the level has been completed. The player can progress to the next level.		
8	Save/Load Simulation	Menu.unity	Persistence	Player progress is kept and games can be resumed.		The player profile is saved. Multiple profiles can be saved and loaded.		

6 Contribution of Team Members

Working hours:

Andrew: 2.5 hours + (7 hours for development)

Modified teleportation physics, game logic, created models, particle effects, UI interaction with controller implementation and testing, User Stories and Acceptance Criteria

Chris: 2.5 + (8 hours for development)

Modified teleportation physics, added models to scene, changed version control, worked on lever action logic, Cover page, Table of Contents, Abstract, and Contributions of Team Members

Nick: 3.0 hours + (6 hours for development)

Modified Controller Scripts, changed UI using VRTK, Created temp target entities, particle effects, UI interaction with controller implementation and testing, Testing Workflow and Validation, Layed out deliverables for demo, Menu level testing and partial configuration.

Will: 3.0 hours + (4.5 hours for development)

Manual exploratory tests involving UI, menus, and user interaction. Planning for improved level, UI, and menu design. Abstract and Testing Strategy.