Augmented Reality Game

Software Requirements Specification

COP 4331C, Fall 2015

Team Name: Project Pals

Team Members:

* Eric Peralli - Eric.peralli@gmail.com
* Connor Heckman – Connor.heckman@me.com
* Clayton Cuteri – Cuteri.clayton@knights.ucf.edu

|  |  |  |  |
| --- | --- | --- | --- |
| Version | Date | Who | Comment |
| V0.0 | 10/04/15 | Connor Heckman | Original Specs Draft |
| V1.0 | 10/04/15 | Clayton Cuteri | Editing and Supplemental Information |
| V2.0 | 10/06/15 | Eric Peralli | Added Event Table |
| V3.0 | 10/07/15 | All Members | Added Use Case diagram and Use Case descriptions |

**Contents of this Document**

SECTION 1 - Introduction

* Software to be Produced
* Reference Documents
* Applicable Standards
* Definition, Acronyms, and Abbreviations

SECTION 2 - Product Overview

* Assumptions
* Stakeholders
* Event Table
* Use Case Diagram
* Use Case Descriptions

SECTION 3 - Specific Requirements

* Functional Requirements
* Interface Requirements
* Physical Environment Requirements
* Users and Human Factors Requirements
* Documentation Requirements
* Data Requirements
* Resource Requirements
* Security Requirements
* Quality Assurance Requirements
* Supporting Materials

**SECTION 1 - Introduction**

**Software to be Produced:**

The software in production is an augmented reality application built for smartphone users. The software places the user in a first person perspective. The software provides a menu for customization of the user’s augmented reality experience. When the user begins a game, the software populates the user’s screen with enemies of varying difficulty as well as a heads up display of user’s score and health. Users defeat enemies by tapping on the enemy’s virtual sprite. The software utilizes existing smartphone camera API to display the generated enemies on the user’s immediate surroundings.

**Reference Documents:**

* [Concept of Operations](https://docs.google.com/document/d/1x4QyHfMRQdcK_UUwJiVtlEQeu1ca7ca3vmrsmUwGgYY/edit?usp=sharing)
* [Project Management Plan](https://docs.google.com/document/d/1_ODvCtTKPSJEnTv0dxVnQx6iGd40UmvQDcgeTsX3Kjo/edit?usp=sharing)
* [Test Plan](https://docs.google.com/document/d/1cSxtwkrlGLdXxo6UWzqB2D9D3ob08vY_kFgXF0AnbmI/edit?usp=sharing)

**Applicable Standards:**

* Application built for Android Lollipop 5.0
* [Android Core App Quality Standards](http://developer.android.com/distribute/essentials/quality/core.html)

**Definitions, Acronyms, Abbreviations:**

* API - Application Programming Interface
* GUI – Graphical User Interface
* Augmented Reality – a live view of the real-world with elements supplemented by computer-generated input such as sound or visuals
* Appropriate gaming environment – a setting where the user can play our augmented reality game without endangering themselves or others through negligence of environmental hazards
* OS – operating system

**SECTION 2 - Product Overview**

**Assumptions**

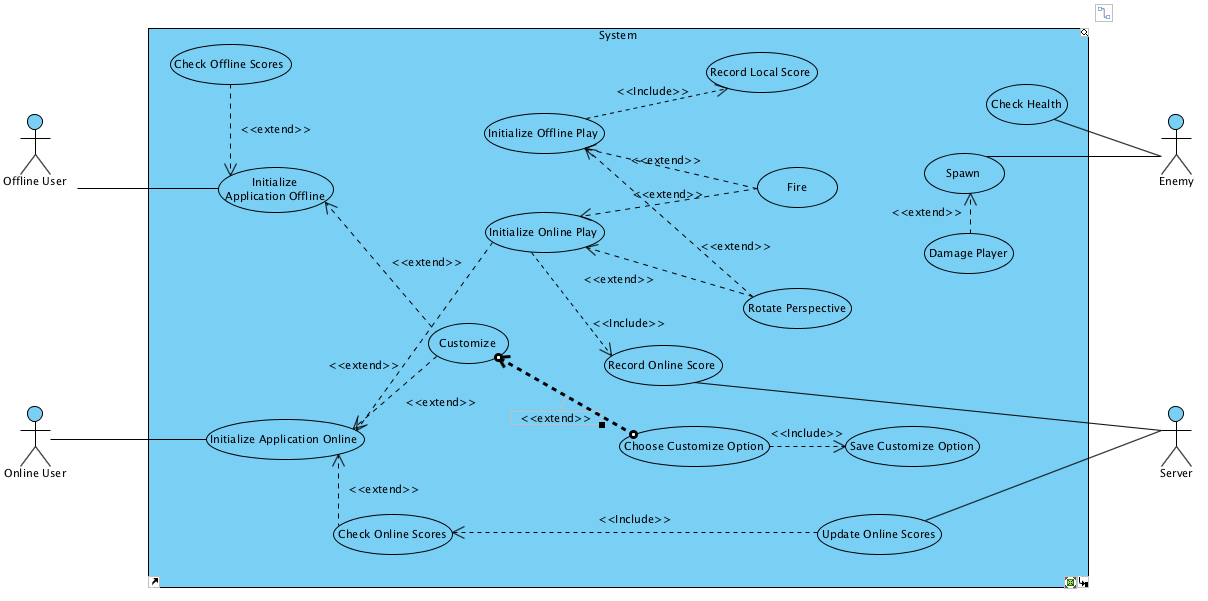
* The application will run on the most current Android OS
* The application will require permissions to utilize the user’s camera
* The user will need a smartphone with an Android OS to use the application
* The user will have the most recent version Android OS to use the application
* The user will need internet connectivity in order to use the online scoring system
* The user will use the application only in an appropriate gaming environment

**Stakeholders**

* Professor – This stakeholder is in charge of evaluating the execution of our application as well as accounting for proper documentation throughout the software development process.
* Team Members (Developers) – The developers will be responsible for the functionality of the application, the creation and organization of documentation deliverables, and the accuracy of the documentation in regard to the application.
* Competitive Users – these users will be interested in attainting a top score. They will be most concerned with the applications ability to networking features.
* Casual Users – These users are those playing the game purely as a recreational activity or as an entertaining distraction. They will be most interested in the customization options available to keep the application feeling fresh in each play through.

**Event Table**

|  |  |  |  |
| --- | --- | --- | --- |
| **Event Name** | **External Stimuli** | **External Response** | **Internal Data/State** |
| User fires at enemy | User taps on screen enemy | Enemy takes damage | Targeted enemy decreases its health instance variable |
| User eliminates enemy | User taps on screen enemy | Enemy disappears from screen | Targeted enemy decreases its health variable to 0. |
| Users turns left | User rotates smartphone camera to the left | Enemy position shifts right relative to user | GUI removes enemy sprite from camera if enemy is shifted off-screen. Enemy is added to list of unseen but active enemies |
| User turns right | User rotates smartphone camera to the right | Enemy position shifts left relative to user | GUI removes enemy sprite from camera if enemy is shifted off-screen. Enemy is added to list of unseen but active enemies |
| User takes damage | Enemy is onscreen and has not been targeted by user for appointed period of time | User’s health bar decreases on the Heads up display | User instance variable “health” is decreased. GUI is updated |
| Enemy Takes Damage | User successfully targets enemy sprite | Hit indicator displayed | Enemy health variable is updated. Check run to see if health = 0, if this is true, GUI is updated to remove enemy sprite. Score is updated. |
| “Play” Selected | User taps “Play” on the main menu | Heads up display appears and screen begins to populate with enemies | A new instance of the game is created |
| "Check Scores” Selected (Online) | User taps “Scores” on the main menu | A display with the top global scores appears | The application fetches the top scores from the server |
| “Check Scores” Selected (Offline) | User taps “Scores” on the main menu | A display with the top local scores appears | The application reads the local file that is storing the scores |
| “Customize” Selected | User taps “Customize” option on main menu | Main menu disappears. Customization menu appears | GUI updates to display Customization menu |
| User receives notification (text, call, etc.) while using application | User receives notification (text, call, etc.) while using application | Some of the user’s screen is hindered by the notification | Game functions normally |
| Application is minimized | The user’s mobile device incurred an event that forced the application to minimize | Application is no longer being displayed on the screen | User becomes immune to damage until the application is open again. The application will resume once opened again. |
| Application is closed | The user’s mobile device incurred an event that forced the application to minimize | Application is no longer being displayed on the screen | The application process is ended on the mobile device, and the user’s play session is terminated. An attempt to save scores will be made. |

**Use Case Diagram**

**Use Case Descriptions**

* Offline User – user playing the game without an Internet connection
* Online User – user playing the game with an Internet connection
* Server – stores and updates high scores.
* Enemy – AI that targets and is targeted by the user during gameplay.
* Initialize application online – User has an Internet connection when they open the application. The menu options “Play”, “Customize”, and “Check Scores” are displayed.
* Initialize application offline – User has no Internet connection when they open the application. The menu options “Play”, “Customize”, and “Check Scores” are displayed.
* Check Offline Scores – Offline user selects “Check Scores” option from
* Check Online Scores – Online user selects “Check Scores” option from the main menu. All time top scores recorded by the server are displayed to the user.
* Customize – Menu displayed when Customize is selected on the start menu. Displays experience customization options to user.
* Chose Customization Option – User selects the customized skin they would like to use when playing the game.
* Save Customization Option – Customization skin selected by used is saved as new default.
* Initialize play Online – User selects the “Play” option from the main menu with an Internet connection.
* Initialize play offline – User selects the “Play” option from the main menu without an Internet connection.
* Fire – User taps the screen while in game, attempting to target an enemy sprite.
* Rotate Perspective – User rotates the camera of their smartphone either right or left. GUI updates to show enemies existing in that direction.
* Damage user – enemy applies damage to user health if they have not been targeted in the given timeframe.
* Check Health – Enemy checks its health to ensure it should remain active.
* Record Online Score – Server records the score of the online player as they play, checking each time against existing top scores to see if it is a new top score.
* Update Online Score – Server updates the online scoreboard if an online player breaks a top score.
* Record Local Score – application updates offline player’s offline scoreboard.

**SECTION 3 – Specific Requirements**

**Functional Requirements**

1. GUI

* Our application will require a graphical user interface utilized to customize the gaming experience, insatiate the game, and play the game. This GUI will be created through the use of the Java Swing library.
* Source: The GUI is required for the user to interact with the enemies as well as to select options in the Start and Customization menus.
* Dependency: None.
* Conflicts: None.
* Supporting Materials: [About Java Swing](https://docs.oracle.com/javase/tutorial/uiswing/start/about.html)
* Evaluation Method: we will test the GUI by exploring all options and interactions available to the user and ensuring there are no bugs or dead ends in the GUI.
* Revision History: Created originally by Connor Heckman on October 4th 2015

1. Start, Customization, and Online Scoring Menus

* Our application will require a start menu that appears when the application is opened and gives the user the option to Play, Customize, or View online scores.
* Source: Clearly displayed, minimal options for the user to choose from. Highly responsive to user input.
* Dependency: #1 (Menus will be displayed by updating the GUI)
* Conflicts: None.
* Supporting Material: None
* Evaluation Method: All options available to the user will be reviewed in test play to ensure functionality of menus.
* Revision History: Created originally by Connor Heckman on October 4th 2015

1. User Environment / Game Environment

* Statement: The game will utilize the user’s immediate surroundings to create an augmented reality gaming experience.
* Source: augmented environment provided for the user to interact with while in game.
* Dependency: #1 (Enemy sprites and user Heads-up Display will be created with GUI, the GUI must also update as gameplay unfolds). #2 (Options selected in menus must be saved to cater to the user’s desired experience. An instance of the gameplay environment will be instantiated through user interaction with the menus).
* Conflicts: None.
* Supporting Materials: [Android Camera API](http://developer.android.com/reference/android/hardware/package-summary.html)
* Evaluation Method: Interaction between user and game environment will be tested in our test play phase. All interactions between user and augmented environment will be explored.
* Revision History: Created originally by Connor Heckman on October 4th 2015

1. Game Customization skins

* Statement: The application will provide customization options to the user’s augmented reality experience.
* Source: The user will chose the appearance of the enemy sprites as their own heads up display.
* Dependency: #1/#2 (The options are chosen through the Customize menu displayed by the GUI) #3 (The game environment must update to suit the user’s choice)
* Conflicts: None.
* Supporting Materials: None.
* Evaluation Method: Each customization option will be played to assure there is no miscommunication between the Customize Menu and the game environment GUI
* Revision History: Created originally by Connor Heckman on October 4th 2015

**Interface Requirements**

1. Responsive Play

* Statement: The game environment must update each time the user triggers a fire action by tapping on their screen.
* Source: User/Game Environment
* Dependency: Game Environment
* Conflicts: None.
* Supporting Materials: None.
* Evaluation Method: Test to ensure hits register in gameplay. Test to ensure misses do not affect enemy health and produce a miss indicator.
* Revision History: Created originally by Connor Heckman on October 4th 2015

1. Responsive Environment

* Statement: Enemies must remain in active play even when the user rotates their camera so that they appear off screen. On screen enemies must shift position relative to user when user rotates camera.
* Source: User/Game Environment
* Dependency: GUI
* Conflicts: None.
* Supporting Materials: None.
* Evaluation Method: Responsiveness of environment will be tested during test play phase.
* Revision History: Created originally by Connor Heckman on October 4th 2015

**Physical Environment Requirements**

1. Hardware

* Statement: User must have an Android smartphone in order to play.
* Source: Concept of Operations.
* Dependency: None.
* Conflicts: Unable to reach users who do not own smartphones.
* Supporting Materials: None.
* Evaluation Method: None.
* Revision History: Created originally by Connor Heckman on October 4th 2015

1. Appropriate Gaming Environment

* Statement: User must be in an appropriate gaming environment in order to use application.
* Source: Assumptions
* Dependency: User’s physical location.
* Conflicts: User will have limited available settings to use application.
* Supporting Materials: None.
* Evaluation Method: None.
* Revision History: Created originally by Connor Heckman on October 4th 2015

**Users and Human Factor Requirements**

1. Network Effect

* Statement: the user’s enjoyment of the application will increase and decrease along with the number of total users.
* Source: Concept of Operations.
* Dependency: reception of the application by gaming community
* Conflicts: Competitive app users may be discouraged from using the application due to a lack of other users to challenge on the online scoreboard.
* Supporting Materials: None.
* Evaluation Method: None.
* Revision History: Created originally by Connor Heckman on October 4th 2015

**Documentation Requirements**

1. Online Documentation

* Statement: All documentation for the project will be kept and updated on the team’s website to expedite updates and revisions to documents.
* Source: Project Management Plan.
* Dependency: Team member’s organization.
* Conflicts: None.
* Supporting Materials:
* Evaluation method: None.
* Revision History: Created originally by Connor Heckman on October 4th 2015

**Data Requirements**

1. Consistent Gaming Environment

* Statement: The gaming environment must remain consistent as the user rotates the camera. Enemy sprites must continue to approach and cause damage to user health even when off-screen.
* Source: Functional Requirements
* Dependency: GUI and Gameplay Environment in functional operations.
* Conflicts: None.
* Supporting Materials: None.
* Evaluation Method: Enemy consistency will be examined during test play phase.
* Revision History: Created originally by Connor Heckman on October 4th 2015

1. Online Scoring

* Statement: User score must be updated each time user defeats an enemy. Online scores must be updated if the user has broken onto the leaderboard.
* Source: Concept of Operations
* Dependency: Networking abilities of developers.
* Conflicts: What will occur if user ties with a leaderboard score?
* Supporting Materials: None.
* Evaluation Method: Test the score counter and leaderboard accuracy during the test play phase.
* Revision History: Created originally by Connor Heckman on October 4th 2015

**Resource Requirements**

1. Developer Time

* Statement: Developers must treat programming and test play time before project deadline as a commodity and ration it appropriately.
* Source: general knowledge of programming deadlines.
* Dependency: Developers’ skill and class/work schedules.
* Conflicts: Some preferences may have to be compromised in order to produce a functioning and enjoyable application before deadline.
* Supporting Materials: None.
* Evaluation Method: The test of time.
* Revision History: Created originally by Connor Heckman on October 4th 2015

**Security Requirements**

1. Camera Permission

* Statement: Application will require permission from user to utilize camera for application.
* Source: Concept of Operations.
* Dependency: User willingness to grant camera permission.
* Conflicts: May lose users due to their unwillingness to grant camera permission to the application.
* Supporting Materials: None.
* Evaluation Method: None.
* Revision History: Created originally by Connor Heckman on October 4th 2015

**Quality Assurance Requirements**

1. Difficulty Scaling

* Statement: Difficulty of the game must scale will time elapsed. Game must get more difficult the higher the player score.
* Source: Experience with smartphone applications.
* Dependency: Difficulty will depend on time elapsed since instance of game was created.
* Conflicts: Some users may feel game is too hard or too easy. Developers will have to find a compromise during test play.
* Supporting Materials: [Subjective Difficulty Article](http://www.gamasutra.com/view/feature/134950/examining_subjective_difficulty_.php)
* Evaluation Method: During test play, developers will determine a compromise between competitive and casual gameplay.
* Revision History: Created originally by Connor Heckman on October 4th 2015.

1. Continuity of Platform Priority

* Statement: User’s incoming text messages and phone calls supersede the game’s functionality.
* Source: Application Platform
* Dependency: User Environment/Game Environment
* Conflicts: None.
* Supporting Materials: None.
* Evaluation Method: Ensure that working prototype of application does not interfere with smartphones ability to receive notifications and communications.
* Revision History: Created originally by Connor Heckman on October 4th 2015.

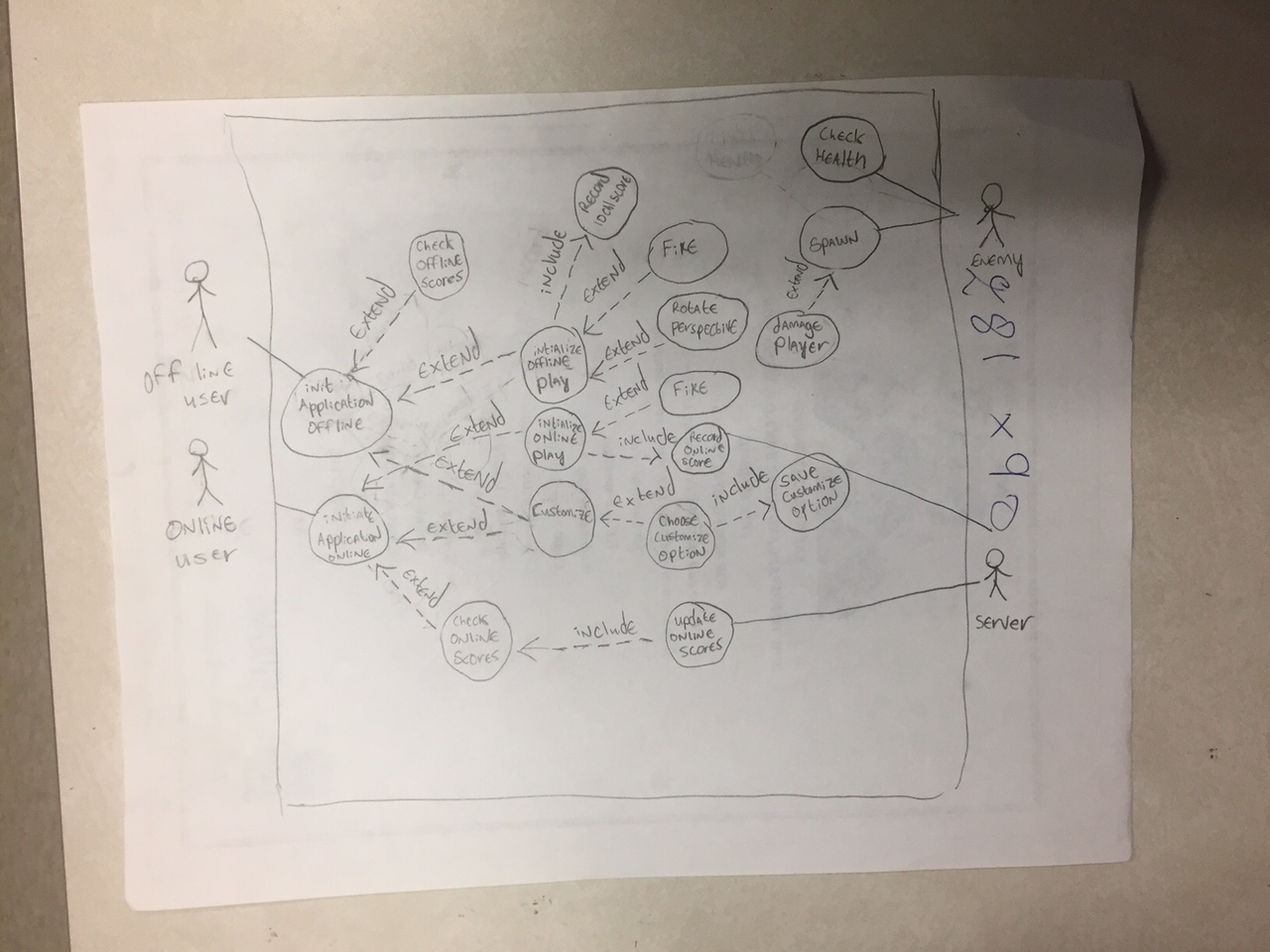
1. Accurate Ray Casting

* Statement: The taps from the user on any part of the enemy sprites must register as a hit.
* Source: Game Environment/User Environment
* Dependency: GUI and Responsive Play requirements
* Conflicts: Developers must ensure that enemy size is accurately displayed by the GUI.
* Supporting Materials: None.
* Evaluation Method: Hit registering will be thoroughly test during the test play phase of the project.
* Revision History: Created originally by Connor Heckman on October 4th 2015.

**Supporting Materials**

Links to Meeting Minutes

* [October 4th, 2015](https://docs.google.com/document/d/1fvVnba10j99guWE6N2lCItwzECNavkzwzdBA0CLk_Ag/edit?usp=sharing)
* [October 7th, 2015](https://docs.google.com/document/d/1g1kG1wMUSNRSA8lrjyOQD1CcZpJIDVywdRBaeqFiybs/edit?usp=sharing)

****First Draft sketch of Use Case Diagram