Requirements Specification

Trouble Seekers

Rev 0

Prepared by

Group Name: Buttered Waffle

Andrew Raudys	**0973268**
Nathan MainVille	**0960351**
Matt Leduc	**0958787**
Sean McLellan	**0963077**

Instructor: Dr. Carette
Course: SFWR ENG 4GP6
Date: Oct 29, 2012

Table of Contents

Project Drivers

1. Purpose of the project

1a. The User Business or Background of the Project Effort

The developers are building this game for a capstone project in university. The plan is to use the Unreal Development Kit (UDK) to develop an isometric 3D RPG within the following six months. Completing this project will help us gain an understanding in 3D game development and the usage of the UDK. The game is being developed to explore the process of game development. This project offers an opportunity to create a project that may give us the experience needed to have ourselves considered for positions in the game development industry.

1b. Goals of the Project

The main goal of this project is to give the players of our game a unique and entertaining experience. It also results in the creation of a marketable and entertaining game which will give the developers knowledge on the design process of a game. The game will be judged and graded by a group of unbiased professional software developers from different aspects of the industry.

2. Stakeholders

2a. The Client

The client is the board of experts in the industry who will be giving their judgement on our game at the end of the project. Another main client is the professor for the Capstone course as well the Teaching Assistant who is aiding him in the course. They should expect a unique and fully functional game.

2b. The Customer

The customers will be players who enjoy Role Playing Games with a cartoony art style. The players should be 14+ in order to adhere to the Teen rating of our game. The players should be expecting a fully functional and playable game in the form of at least one fully developed chapter of the storyline with potentially up to 3 hours of gameplay. The game should not have fatal crashes and should also allow the player to save their progress.

2c. Other Stakeholders

Other potential stakeholders would be people chosen by the developers in order to test the functionality and content of the game. They will be assisting the developers in refining the game by identifying potential issues with gameplay and art design.

2d. The Hands-On Users of the Product

The hands-on users of the product would include the players who are above the age of 14 and enjoy Role Playing Games, in addition to the board of judges we will be presenting our game to.

2e. Personas

N/A

2f. Priorities Assigned to Users

Key Users: The judges, developers, the TA and professor.
Secondary Users: N/A
Unimportant Users: N/A

2g. User Participation

Every developer in the project's group must put in equal parts of work to complete the project.

2h. Maintenance Users and Service Technicians

N/A

Constraints

3. Mandated Constraints

3a. Solution Constraints

Description:
The game will be built using the Unreal Development Kit.
Rationale:

Using the UDK as a starting point, more time can be spent working on the mechanics of the game as opposed to the engine.
Fit criterion:
The game will be compatible with devices that are able to run Unreal Engine games.

3b. Implementation Environment of the Current System

The game will be designed for personal computers able to satisfy the minimum requirements of the Unreal Engine. The game shall also take up a certain amount of space on the hard drive, the specific size of the game is not yet defined, but the user must have enough room for the game to save properly and complete its installation process.

3c. Partner or Collaborative Applications

The game will be compatible with the Windows Operating System.

3d. Off-the-Shelf Software

Blender will be utilized in order to create character and object models, seeing as Blender is compatible with the Unreal Development Kit and exports using the 3ds Max .fbx file format, which the UDK accepts.

The software to create the sound effects is still undetermined but Adobe Soundbooth will be used to edit or change any sounds we will be using in the game. FL Studio will be used to create some or all of the in-game music. The original files used for sound will either be taken online with respect to proper copyright laws or created them manually.

3e. Anticipated Workplace Environment

The workplace can vary from user to user, seeing as all that is required to use the product is a computer or laptop which can run the Unreal Engine. The most common workplace will be a desk/table on which the user can put down his computer or laptop. The users will either be in a public place or at home when using the product, which makes it difficult to predict their environment accurately.

3f. Schedule Constraints

The professor and TA for the course have given a list of deadlines to meet in order to complete the project on time. The visuals and sound must be completed by November 23rd. Without proper art and sound there will not be a satisfying look and feel for the game. The requirements for each system

component must be finished by December 19th, and the design of the components by January 11th. By completing these two steps there should be a playable version of the game ready in which the developers can observe the sound, art, and mechanics more clearly. It is also a stepping stone in order to create further aspects of the game such as level design.

The implementation of Revision 0 (Rev 0), the prototype of the game, is due on January 25th. The validation and verification of our Rev 0 is due on the 8th of February, by doing this soon after the implementation of Rev 0 the developers can quickly find issues and make potential changes to the game.

Once Rev 0 is finished, work begins on Rev 1 of the project. The Concept and all Requirements is due on February 22nd followed by Art and Design due on March 8th. These deadlines are essentially to make sure that the work due during Rev 0 is refined to a higher quality. The implementation of Rev 1 is due on the 22nd of March and the Validation and Verification is due on the 5th of April. This is essentially the final deadline to meet for the project, as there must be a working and playable version of the game by the beginning of April. The presentation of the product in front of the judges is slated for April 21st. This is a tentative date seeing as the judges have not confirmed their availability for the presentation.

Failure to meet any deadlines will have a cumulative ripple effect on the progress towards all future deadlines if lost time is not made up.

3g. Budget Constraints

Seeing as this is a school project, the only resource used is personal time. There should be no monetary budget to constrain the project.

3h. Enterprise Constraints

N/A

4. Naming Conventions and Terminology

4a. Definitions of All terms, Including Acronyms, Used by Stakeholders Involved in the Project

4a.1 Definitions

Abilities: Special powers the player can use to help himself or use to defeat enemy units.

Artifact: An item the player must pick up in order to access the next important area of the game.

NPC: A non-player character that the player can interact with through either conversation, combat, or merely observation.

Boss: A strong enemy unit which usually has a tie with the story.

Enemy Unit: An NPC which will attempt to kill the player.

Inventory: The player can hold items he finds in his inventory for future use or to gain abilities, the player can hold an unlimited number of items.

Item: An object the player can pick up and keep in his inventory.

Mana: A resource the player can consume to use abilities.

Stat: Any attribute of the player-controlled character that can have an adjustable metric applied to it. Examples can include defense, damage, attack speed, health, mana and movement speed.

Tablet: An item which grants the player either stat bonuses or abilities.

4a.2 Acronyms

AoE: Area of Effect

HUD: Heads-up Display

NPC: Non-Player Character

RPG: Role Playing Game

UDK: Unreal Development Kit

UDN: Unreal Development Network

UI: User Interface

TA: Teaching Assistant

5. Relevant facts and Assumptions

5a. Relevant Facts

N/A

5b. Business Rules

N/A

5c. Assumptions

All computers that meet the minimum requirements should be able to run the game without any problems. The UDK will be available to use throughout the entire duration of the project at no cost.

Functional Requirements

6. Scope of work

6a. The Current Situation

UDK and a 3D modelling program are required to develop the game, and game art respectively. The game is required to be an RPG. The game art has been determined to follow a cartoony and simple style. There must be a storyline in the RPG, of which the first chapter, involving an introductory portion of the story, must be completed for demonstration.

6b. The Context of the Work
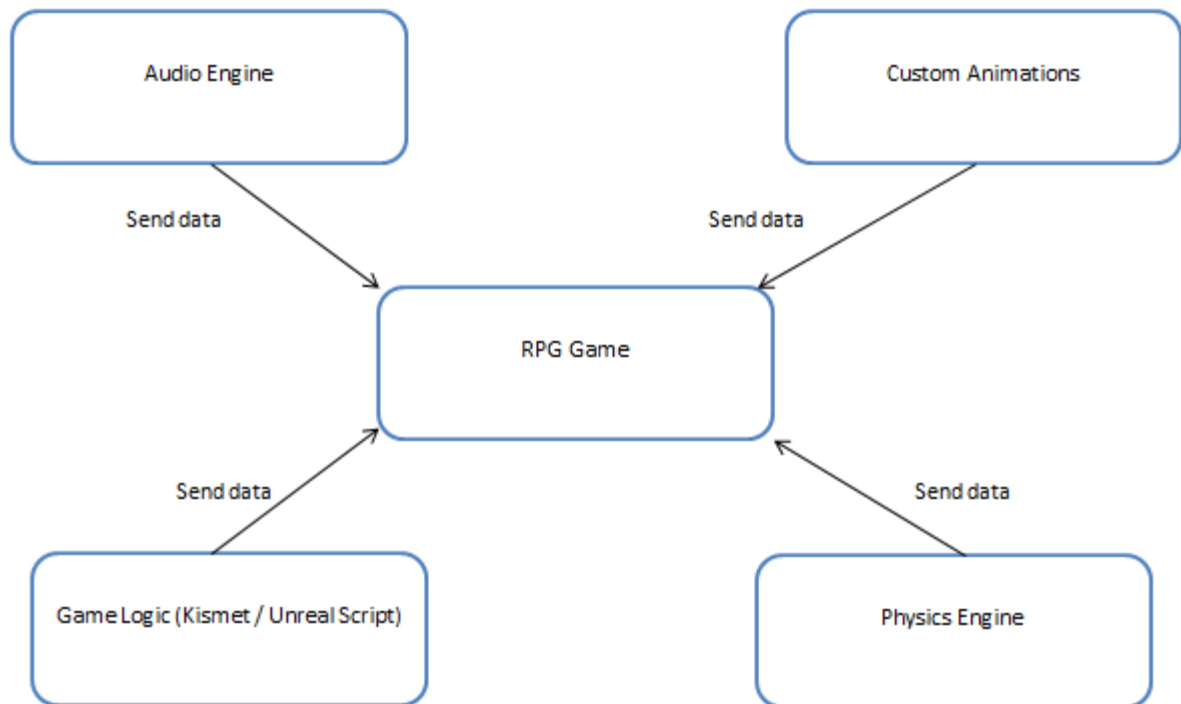
6c. Specific Use Cases and Data Requirements

6c.1 Menu Requirements

Event #: 1
Event Name: Start New Game
Input:
- User selects "New game" from the Main Menu using the mouse click or by navigating using the keyboard directionals and the spacebar to select.

Output:
- Game returns audio and visual feedback to player indicating "New game" was chosen.
- Game initializes and loads the logic, entities, and triggers for the first level of the game with the player's character at base stats.

Event #: 2
Event Name: Save Game Menu
Input

- User selects "Save game" from the game menu.

Output
- A save game screen appears with areas for typing the save name, as well as already existing save games in the form of items with a screenshot of the saved area, and the saved game's name.

Event #: 3
Event Name: Save Game
Input
- User types in a name for the save, or leaves it as the default, and clicks save game

Output
- If the game save name already exists, a prompt is displayed to the user asking if they would like to overwrite the save.
  - If the user clicks yes, the game save is put to the top of the list with an updated screenshot of the current game state.
  - If the user clicks no, the prompt disappears, the name remains unchanged.
- The game save is put to the top of the list with a screenshot of the current game state, as well as the name that the user has specified.

Event #: 4
Event Name: Load Game Menu
Input:
- User selects "Load game" from the Main Menu using the mouse click or by navigating using the keyboard directionals and the spacebar to select.

Output:
- Game returns audio and visual feedback to player indicating "Load game" was chosen.
- Game displays a Load game menu which contains all previous save games in the form of visual representations containing a screenshot of the game state and the save's name.

Event #: 5
Event Name: Load Game
Input:
- User selects one of the saved games, and clicks Load.

Output
- The game displays a loading progress bar while loading the game state, then initializes and loads the logic, player data, entities, and triggers for the loaded state of the game.

Event #: 6

Event Name: Main Menu
Input
- ○ User launches game, or returns to the main menu via exiting the currently active game by clicking "Quit" in the Game Menu.

Output
- ○ Displays a menu layout on the screen with the headings of New Game, Load Game, Options, and Exit.

Event #: 7
Event Name: Options Menu
Input
- ○ User selects the Options selection in the Main Menu, or in the In-Game Menu.

Output
- ○ Displays a menu (which takes up the entire viewing area) with options for customizing graphics, sound, and keyboard controls.

Event #: 8
Event Name: Game Menu
Input
- ○ The player presses escape, or clicks on the game menu icon located in the top right corner (subject to change).

Output
- ○ A menu about the quarter of the size of the screen appears in the center of the screen, while the last viewable in-game screen is grayed out as a background to the menu.
- ○ The menu displays options for saving, loading, customizing options, and exiting the game.

Event #: 9
Event Name: Inventory
Input:
- ○ The player presses the hotkey related to opening the inventory, or clicks on the inventory button.

Output:
- ○ The game state is paused, and an inventory screen is displayed on the screen, containing the following:
  - ■ Several equip slots for stat items
  - ■ Several equip slots for ability items
  - ■ Currently stored stat items
  - ■ Currently stored ability items
  - ■ Currently stored regular/miscellaneous/quest related items

Event #: 10
Event Name: Quest screen

Input:
- ○ The player presses the hotkey assigned to opening the quests screen, clicks on a quest notification (such as completing or receiving a quest or objective), or clicks on the quest icon in the HUD.
- ○ The player can click on individual quests to expand them to see the objectives that make up that particular quest.
- ○ The player can click on an expanded quest if they wish to hide the objectives of that particular quest.

Output:
- ○ A quest screen is displayed which contains the following:
  - ■ Two tabs, one for available quests, and one for completed quests.
  - ■ The available quests tab contains a list of currently active quests. Main quests are shown at the top of the list, and are in a regular font. Side quests are shown below the main quests in a lighter font.
  - ■ Each quest in the available quests tab has a box next to it, which can either be coloured or not coloured. Clicking on the box toggles this box, and it signifies whether the selected quest is the active quest. Only one box can be active at a given time.
    - ● An active quest will guide the user to the next objective of the quest via map notifications/directions.
  - ■ Each quest can be expanded to show the objectives of the quest by clicking on it. By default, the objectives are shown for the active quest.
  - ■ Completed objectives in a quest are noted with a green checkmark next to them.
  - ■ The completed quests tab contains a list of completed quests. Main quests are shown at the top of the list, and are in a regular font. Side quests are shown below the main quests in a lighter font.

6c.2 In-Game Requirements

Event #: 11
Event Name: Equipping Stat Changing Tablets
Input
- ○ The player opens the inventory menu using the assigned hot-key
- ○ On the inventory menu, the player drags an item from the inventory into the equipment slot.

Output
- ○ The game identifies whether the user is in a safe zone or not:
  - ■ If not in a safe zone the user cannot equip the stat item
  - ■ If the player is in a safe zone the stat item is equipped
- ○ The game reads the attributes of the item in the equipment slot.
- ○ The game adjusts the player's stats to reflect the change caused by the equipped item.

Event #: 12
Event Name: Equipping Ability Changing Tablets
Input
- ○ The player opens the inventory menu using the assigned hot-key
- ○ On the inventory menu, the player drags an item from the inventory into an ability slot.

Output
- ○ The game identifies whether the user is in a safe zone or not:
    - ■ If not in a safe zone the user cannot equip the ability item
    - ■ If the player is in a safe zone the ability item is equipped
- ○ The game reads the data regarding the type of the ability that is equipped.
- ○ The game adjusts the player's available abilities to reflect the newly equipped ability.

Event #: 13
Event Name: Talking to NPCs
Input:
- ○ User selects an NPC by left-clicking on the NPC with the mouse pointer.

Output:
- ○ Game identifies the selected NPC and presents the player with a pop-up dialog box with the related NPC conversation text after identifying the state of the game to ensure that the correct conversation options are available.
- ○ Game waits for the player to select an option if dialog has player responses.

Event #: 14
Event Name: Attacking with a basic attack
Input:
- ○ User swings their basic weapon by left-clicking on the screen with the mouse pointer in the direction the player wants to attack.

Output:
- ○ Game correlates the direction of the mouse with the rotational direction of the player's character and displays the character in the correct direction.
- ○ Game takes the input of the left-click on the screen and triggers the character to display an attack animation.
- ○ If a hittable target is within the range calculated by the Game, on hit effects are applied to the target entity and any animations of the target and feedback for the player related to the target being hit are shown.

Event #: 15
Event Name: Using a targetable ability
Input:

- User casts their ability by selecting an ability using the related hot-key to select it and left-clicking on the screen with the mouse pointer at the position where the ability should cast.

Output:
- Game identifies the position of the mouse pointer and identifies if the ability cast destination is within the ability's range:
  - If the ability is within range, the ability is cast at that location.
  - If the ability is not within range, the ability is cast in the rotational direction of the pointer and the extent of the range.
- Game takes the input of the left-click on the screen and triggers the character to display the related ability animation.
- If a hittable target is within the range of the ability's area of effect calculated by the Game, on hit effects are applied to the target entity and any animations of the target and feedback for the player related to the target being hit are shown.

Event #: 16
Event Name: Using a non-targetable ability
Input:
- User casts their ability by selecting an ability using the related hot-key to select it and cast it.

Output:
- Game identifies the ability being cast and applies its effects to the target entities.
- If a hittable target is within the range of the ability's area of effect calculated by the Game, on hit effects are applied to the target entity and any animations of the target and feedback for the player related to the target being hit are shown.

Event #: 17
Event Name: Map Menu
Input:
- User selects the hotkey assigned to open the map.

Output:
- The game identifies that the map hotkey has been pressed.
- The game checks to see that the player is NOT in combat:
  - If the player is in combat, audio and visual (text) feedback is relayed to the player identifying that the action to open the map cannot be taken.
  - If the player is NOT in combat, the map is opened with the player's current position being read to show the player's symbol in a relative location on the map.

Event #: 18
Event Name: Player Death
Input:

- ○ User's character has zero health points remaining.

Output:
- ○ The game identifies that the player has zero health points.
- ○ The game checks to see if the player has any items/passives that enable revival:
  - ■ If the condition is true, then the player revives with the revival conditions of the ability or item.
  - ■ If the condition is false the user is sent to the death menu.

Event #: 19
Event Name: Death Menu
Input:
- ○ User has died and is moved to the death menu.

Output:
- ○ The game presents the user with a series of options, the choice to Load Game or to return to the Main Menu.
- ○ The game waits for the user to select an option.

Event #: 20
Event Name: Picking Up Tablets
Input:
- ○ User moves the character over top of the Tablet.

Output:
- ○ The game identifies that user has come in contact with the Tablet entity.
- ○ The game identifies what the randomly generated Tablet's abilities/stats are and moves the instance of the Tablet into the player's inventory.
- ○ The game removes the Tablet entity from the game world.

Event #: 21
Event Name: Receiving a Quest
Input:
- ○ User initiates an event (talk to NPC or enter a trigger area) which gives the player a quest to complete.

Output:
- ○ The game identifies whether or not the requirements to receive the quest have been fulfilled based on the current game state and player state:
  - ■ If fulfilled, the game identifies that user has triggered a specific quest, and removes that quest from the list of quests that **can be taken** and moves it to the list of quests that **can be completed**.
  - ■ If not fulfilled, the game does not remove the quest from the list of quests that can be taken and does not allow the player to try and complete the quest.
- ○ The game provides the user with updated HUD feedback showing what quests are currently active.

Event #: 22
Event Name: Completing a Quest
Input:
- User completes objectives related to a quest in the list of quests that can be completed.

Output:
- The game identifies whether or not the requirements to complete the quest have been fulfilled based on the current game state and player state:
  - If fulfilled, the game identifies that user has completed a specific quest, and removes that quest from the list of quests that **can be completed** and moves it to the list of quests that **have been completed**.
    - The game also identifies what rewards are available to the player for completing the specific quest, and gives them to the player.
  - If not fulfilled, the game does not remove the quest from the list of quests that can be completed.
- The game provides the user with updated HUD feedback showing what quests are currently active, removing the completed quest.

7. Business Data Model and Data Dictionary

7a. Business Data Model
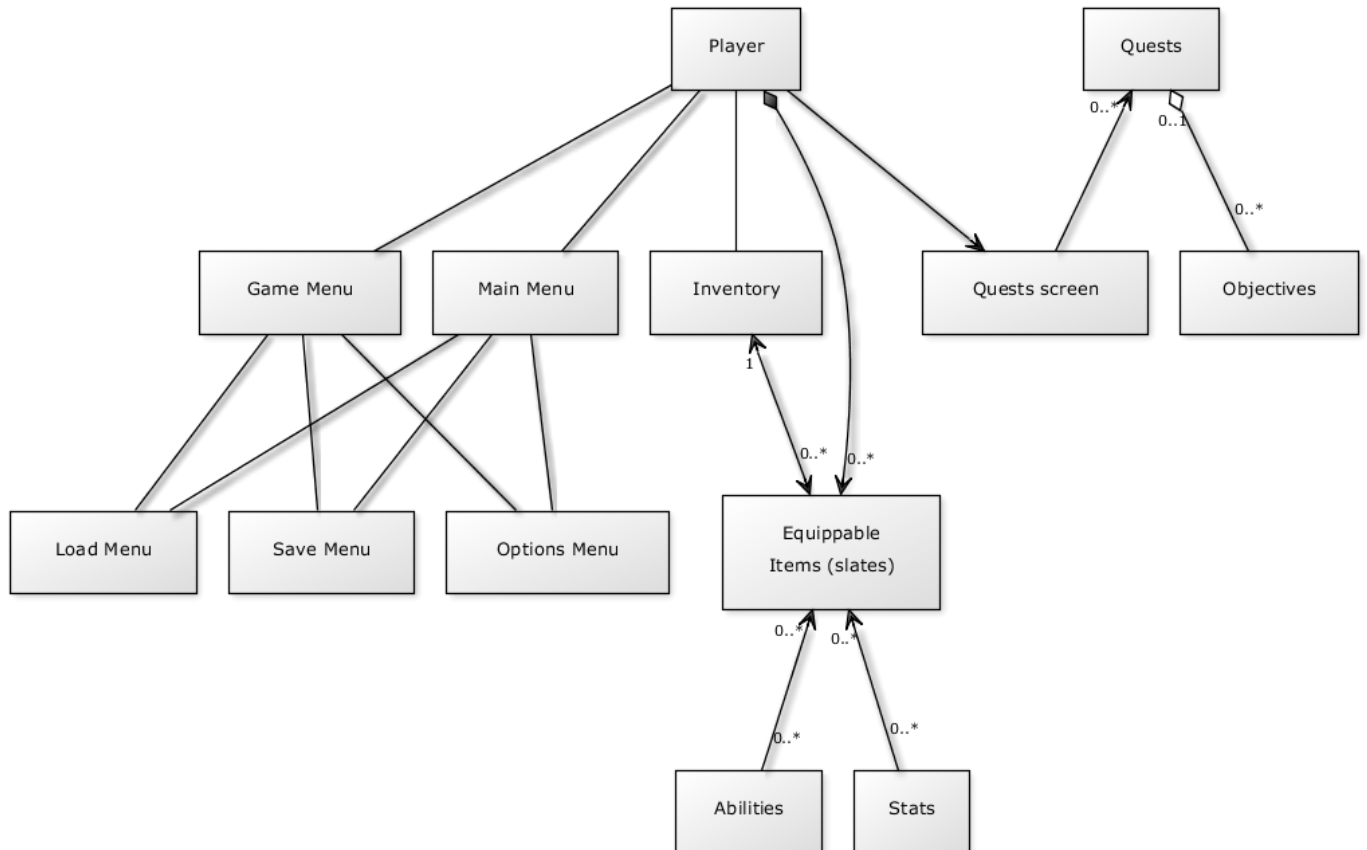
7b. Data Dictionary

Name	Content	Type
Player	User + In-game character	Class
Game Menu	Game Menu + Background Image	Class
Main Menu	Main Menu + Background Image	Class

Load Menu	Load Menu + Background Image	Class
Save Menu	Save Menu + Background Image	Class
Options Menu	Options Menu + Background Image	Class
Inventory	Inventory + Unlocked Slots	Class
Equippable Items (slates)	Slates	Class
Abilities	Magic + Offensive + Defensive Abilities	Class
Stats	Defensive + Offensive Stats	Class
Quests screen	Quests screen + Background Image	Class
Quests	Main Quests + Side Quests	Class
Objectives	Required + Optional Objectives	Class

8. Scope of Product

8a. Product Boundary

As the game is standalone and not part of any larger system, all actions of those playing the game are limited to the game itself.

8b. Product Use Case Table

N/A

8c. Individual Product Use Cases

The only intended use case for the game is for a user to play it for entertainment value.

9. Functional Requirements

9a. Functional Requirements

Requirement #: 1   Requirement Type: 9   Event/Use Case #: 1

Description: The Game shall start a new state initializing all characters, logic, assets, sounds, create an initial auto-save file, and begin the storyline.

Rationale: To start a new game from within the Main Menu.

Originator: Game Developer

Fit Criterion: The New Game button within the Main Menu is linked to this action.

Customer Satisfaction: 5   Customer Dissatisfaction: 5

Priority: High   Conflicts: None

Supporting Material: None

History: Created October 29, 2012

Requirement #: 2   Requirement Type: 9   Event/Use Case #: 2

Description: The Game shall open the Save Game menu displaying all existing saves, and providing the user the ability to name their save game file, and then proceed with saving.

Rationale: To save the game state, in order to be able to resume it at a later time.

Originator: Game Developer

Fit Criterion: The Save Game button is linked to opening the Save Game menu.

Customer Satisfaction: 3   Customer Dissatisfaction: 2

Priority: Medium   Conflicts: None

Supporting Material: None

History: Created October 29, 2012

Requirement #: 3   Requirement Type: 9   Event/Use Case #: 3

Description: Saving a game shall save the current game state and exactly where the player is located along with all other relevant game state information.

Rationale: To save the game state, in order to be able to resume it at a later time.

Originator: Game Developer

Fit Criterion: The Save function is accessed from within the Save Game menu.

Customer Satisfaction: 3   Customer Dissatisfaction: 2

Priority: Medium   Conflicts: None

Supporting Material: None

History: Created October 29, 2012

Requirement #: 4   Requirement Type: 9   Event/Use Case #: 4

Description: Opening the load game menu allows the player to see all previous game save states, along with an image of the state and the time it was saved. It gives the player the ability to choose which state to return to.

Rationale: To load the game into a previous state.

Originator: Game Developer

Fit Criterion: The Load Game menu is opened up from either the main menu or the game menu.

Customer Satisfaction: 3   Customer Dissatisfaction: 5

Priority: High   Conflicts: None

Supporting Material: None

Requirement #: 5　　　　　　Requirement Type: 9　　　　　　Event/Use Case #: 5

Description: Loading a game shall load the selected game state along with exactly where the player was located and all other relevant game state information.

Rationale: To load the game into a previous state..

Originator: Game Developer

Fit Criterion: The load function is accessed from within the load game menu.

Customer Satisfaction: 3　　　　　　　　　　　　　　　　Customer Dissatisfaction: 5

Priority: High　　　　　　　　　　　　　　　　　　　　Conflicts: None

Supporting Material: None

History: Created October 29, 2012

Requirement #: 6　　　　　　Requirement Type: 9　　　　　　Event/Use Case #: 6

Description: The main menu provides a launching pad for the user to either start a new game, load a previously saved game, enter the options menu, or quit the game.

Rationale: To provide the user with an interface to launch into the game itself, to customize options, or to quit.

Originator: Game Developer

Fit Criterion: The main menu is shown upon startup, in addition it is displayed after quitting from the playable game.

Customer Satisfaction: 3　　　　　　　　　　　　　　　　Customer Dissatisfaction: 5

Priority: High　　　　　　　　　　　　　　　　　　　　Conflicts: None

Supporting Material: None

History: Created October 29, 2012

Requirement #: 7   Requirement Type: 9   Event/Use Case #: 7

Description: The options menu provides an interface for the user to tweak graphics, sound and controls.

Rationale: To allow the player to tweak the game to their liking.

Originator: Game Developer

Fit Criterion: The load function is accessed from either the main menu or the game menu.

Customer Satisfaction: 4   Customer Dissatisfaction: 3

Priority: Medium   Conflicts: None

Supporting Material: None

History: Created October 29, 2012

Requirement #: 8   Requirement Type: 9   Event/Use Case #: 8

Description: The game menu is an in-game menu that appears as an overlay on the game when the player presses escape or clicks on the game menu icon. It gives the player the option to save/load or quit the game.

Rationale: To allow the player to save, load or quit the game.

Originator: Game Developer

Fit Criterion: The game menu is accessed in game through pressing escape or clicking on the game menu icon.

Customer Satisfaction: 3   Customer Dissatisfaction: 5

Priority: High   Conflicts: None

Supporting Material: None

History: Created October 29, 2012

Requirement #: 9   Requirement Type: 9   Event/Use Case #: 9

Description: The game allows the player to access an inventory to equip/unequip tablets and weapons, consume miscellaneous items, and hold quest related items.

Rationale: To allow the player to manage their tablets, miscellaneous and quest related items.

Originator: Game Developer

Fit Criterion: The inventory is brought up by pressing its hotkey, or pressing the inventory button.

Customer Satisfaction: 4   Customer Dissatisfaction: 4

Priority: High   Conflicts: None

Supporting Material: None

History: Created October 29, 2012

Requirement #: 10   Requirement Type: 9   Event/Use Case #: 10

Description: The quests screen provides an area for the player to keep track of their progress in the game, and to manage which quest to complete next.

Rationale: To allow the player to keep track of their current task and to view the game's progression.

Originator: Game Developer

Fit Criterion: The quest screen is brought up by either pressing the hotkey assigned to the quest screen, or by pressing the quest screen icon.

Customer Satisfaction: 4   Customer Dissatisfaction: 3

Priority: High   Conflicts: None

Supporting Material: None

History: Created October 29, 2012

Requirement #: 11   Requirement Type: 9   Event/Use Case #: 11

Description: Equipping stat changing tablets allows a player to augment their game character with unique stats pertaining to the given tablet.

Rationale: To allow the player to augment their in game character with unique stats.

Originator: Game Developer

Fit Criterion: The stat changing tablets are equipped in the inventory menu by dragging a tablet into an equipment slot. The user must be in a safe zone to equip the tablet.

Customer Satisfaction: 4   Customer Dissatisfaction: 3

Priority: High   Conflicts: None

Supporting Material: None

History: Created October 29, 2012

Requirement #: 12   Requirement Type: 9   Event/Use Case #: 12

Description: Equipping ability changing tablets allows a player to augment their game character with unique abilities pertaining to the given tablet.

Rationale: To allow the player to augment their in game character with unique abilities.

Originator: Game Developer

Fit Criterion: The ability changing tablets are equipped in the inventory menu by dragging a tablet into an ability slot. The user must be in a safe zone to equip the tablet.

Customer Satisfaction: 4   Customer Dissatisfaction: 3

Priority: Medium   Conflicts: None

Supporting Material: None

History: Created October 29, 2012

Requirement #: 13   Requirement Type: 9   Event/Use Case #: 13

Description: The game provides the player with the ability to talk to NPCs, reading dialogue that the NPC gives to the character, and allowing the player to respond in certain situations.

Rationale: To give the player a sense of interacting with the game world, as well as providing details for the story, and giving the player some options in interacting with NPCs that can influence the outcome of certain situations.

Originator: Game Developer

Fit Criterion: NPCs are targeted and selected with the mouse and clicked on when within a certain talking radius to the player.

Customer Satisfaction: 4   Customer Dissatisfaction: 3

Priority: Medium   Conflicts: None

Supporting Material: None

History: Created October 29, 2012

Requirement #: 14   Requirement Type: 9   Event/Use Case #: 14

Description: The player can attack with a basic melee attack which deals damage to enemies, and destroys certain static items such as bushes.

Rationale: To allow the player to destroy static items, defeat foes and resultantly collect dropped tablets and progress with the game.

Originator: Game Developer

Fit Criterion: The basic attack animates with a left click of the mouse button for the currently equipped weapon. It deals no damage to friendly NPCs, deals damage to enemy NPCs, and destroys certain static items if they are within the attack range of the player's weapon.

Customer Satisfaction: 4   Customer Dissatisfaction: 3

Priority: Medium   Conflicts: None

Supporting Material: None

History: Created October 29, 2012

Requirement #: 15   Requirement Type: 9   Event/Use Case #: 15

Description: The game will allow the player to use targeted abilities with various functions ranging from defensive, to healing, to offensive.

Rationale: To give the player gameplay depth in addition to basic attacks.

Originator: Game Developer

Fit Criterion: With the ability selected, the player points the mouse at where they would like to cast the ability, and when clicked, if it is in range and there is enough mana; it is cast.

Customer Satisfaction: 4   Customer Dissatisfaction: 1

Priority: Medium   Conflicts: None

Supporting Material: None

History: Created October 29, 2012

Requirement #: 16   Requirement Type: 9   Event/Use Case #: 16

Description: The game will allow the player to use non-targeted abilities with various functions ranging from defensive, to healing, to offensive.

Rationale: To give the player gameplay depth in addition to basic attacks.

Originator: Game Developer

Fit Criterion: With the ability selected, the player clicks the mouse button dedicated to the cast, it will apply the effects of the non-targeted ability provided there is enough mana.

Customer Satisfaction: 4   Customer Dissatisfaction: 1

Priority: Medium   Conflicts: None

Supporting Material: None

History: Created October 29, 2012

Requirement #: 17   Requirement Type: 9   Event/Use Case #: 17

Description: A map of the current area as well as total world can be opened and displayed to the player in game.

Rationale: To give the player a sense of location and distance.

Originator: Game Developer

Fit Criterion: Pressing the map hotkey, or clicking on the map icon brings up the map if the player is not in a combat situation. Otherwise, the user will be notified with a visual and audio feedback that they can not open it at the given time.

Customer Satisfaction: 4   Customer Dissatisfaction: 3

Priority: Low   Conflicts: None

Supporting Material: None

History: Created October 29, 2012

Requirement #: 18   Requirement Type: 9   Event/Use Case #: 18

Description: The player's character dies if their health reaches zero, and if the character possesses any revival items, the character is revived.

Rationale: To provide a sense of challenge and penalty if the player takes too much damage.

Originator: Game Developer

Fit Criterion: Every time the player takes damage, through attacks from enemies, or environmental hazards, their hitpoints decrease. Once their health reaches zero, the character dies.

Customer Satisfaction: 3   Customer Dissatisfaction: 3

Priority: Medium   Conflicts: None

Supporting Material: None

History: Created October 29, 2012

Requirement #: 19   Requirement Type: 9   Event/Use Case #: 19

Description: The game will display a death menu upon the characters death, it presents the user with the options to load a previous game, or return to the main menu.

Rationale: To demonstrate to the user that their character has died, and to provide options for what to do after their player character has died.

Originator: Game Developer

Fit Criterion: The death menu appears as soon as the player's death animation finishes, after the player's health has reached zero. It is a live menu which fades in while the colour of the still active game fades to grayscale.

Customer Satisfaction: 3   Customer Dissatisfaction: 3

Priority: Medium   Conflicts: None

Supporting Material: None

Requirement #: 20   Requirement Type: 9   Event/Use Case #: 20

Description: The game will allow the player to pick up tablets that are lying around in the game world, and dropped from certain enemies or hiding within static items.

Rationale: Picking up tablets gives the player a sense of accomplishment and searching for them gives the player a sense of purpose in addition to the main quests.

Originator: Game Developer

Fit Criterion: The player moves the character onto the tablet, which then is placed in the character's inventory.

Customer Satisfaction: 4   Customer Dissatisfaction: 2

Priority: High   Conflicts: None

Supporting Material: None

History: Created October 29, 2012

Requirement #: 21   Requirement Type: 9   Event/Use Case #: 21

Description: The game will allow the player to receive quests which follow a certain branch of the storyline.

Rationale: To give the player a direction in the game, and to advance the storyline.

Originator: Game Developer

Fit Criterion: The player initiates an event by talking to an NPC or entering a trigger area which adds a new quest to the player's current quests.

Customer Satisfaction: 3   Customer Dissatisfaction: 3

Priority: Medium   Conflicts: None

Supporting Material: None

History: Created October 29, 2012

Requirement #: 22   Requirement Type: 9   Event/Use Case #: 22

Description: The game will identify when the player has met all of a quest's objectives and marks it as complete in the quest menu, and notifies the player with audio and visual feedback.

Rationale: To give the player a sense of accomplishment and direction in the game's storyline.

Originator: Game Developer

Fit Criterion: When the player completes an objective related to a quest, they are notified of this with a visual and audio representation on the screen. When they have completed all the required objectives for a quest, they are further notified that they have completed a quest with an audio and visual notification. In addition, the quest is marked as complete in the

Customer Satisfaction: 4   Customer Dissatisfaction: 1

Priority: Medium   Conflicts: None

Supporting Material: None

History: Created October 29, 2012

Non-Functional Requirements

10. Look and Feel

10a. Appearance Requirements

Using a sample size of various users at or above the age of 14, 60% of users should believe that the visual representation of the game is adequate after being introduced to the setting and gameplay of the tutorial zone.

The game shall be visually cartoon-like using complementary color schemes for environment design with no more than 2 primary or secondary colors being used at a time with no limit on the number of analogous colors, tints, shades, or mixtures of the 2 colors being used.

The assets of the game will consist of low-poly triangular meshes that avoid long triangles and generally follow a poly count gradient where feet have the lowest polygon count and the upper torso/head has the highest polygon count.

The game shall display all key information (health, mana, ability cooldowns, map, and active missions) pertaining to the users character in a clear and concise manner through the use of a heads-up display utilizing a maximum of 30% of the screen.

10b. Style Requirements

After a sample group of players complete the tutorial level of the game, 60% of users should feel that the game is light-hearted, humorous but their purpose in the game world should be clearly identified.

After a sample group of players complete the tutorial level of the game, 80% of users should feel that the combat is fluid and controls for combat are highly responsive as opposed to a turn-based or point-and-click combat scheme.

After a sample group of players complete the tutorial level of the game, 80% of users who are familiar with other RPG games should find the layout of game information to be easily and intuitively found.

11. Usability and Humanity

11a. Ease of Use Requirements

The game should be easy to use for players that are at least 14-years-old. At least ninety percent of a sample group of users (minimum age of 14) shall be able to complete the initial tutorial stage which goes over the basic movement controls, basic action controls (fighting), and equipping character items and abilities within 1 hour.

The game shall be used by players who may have never played an RPG hack-and-slash before and possibly have no understanding of English. In a sample group consisting of people who are not familiar to the genre, at least 60% of the players should be able to identify certain elements of the game from a purely visual aspect

after completing the tutorial level.

The game should include toggleable tooltips to offer helpful advice for different parts of the game. In a sample group of users, there should be a higher percentage of the group that find that the tooltips are helpful to understand the game and its mechanics rather than useless. It is common that some gamers prefer to figure things out themselves so it is expected that some will prefer no tooltips.

The game shall be easy enough to control using a keyboard and mouse without a requirement for a high level of dexterity. In a sample group of players with an average level of computer usage, at least 60% of the group should find that the pace of the combat and the complexity of the controls are easy enough to manage after the completion of the tutorial level.

11b. Personalization and Internationalization Requirements

The game should contain mappable keyboard and mouse controls to allow users to adjust the control scheme to fit their needs.

The game should support the ability to localize language options but does not necessarily need to implement its usage.

11c. Learning Requirements

The game shall be used by members of the public who will receive no training before using it with the possible exception of players who are familiar with the RPG genre. At least ninety percent of players of a sample group with no prior experience with RPGs (but are familiar with video games in general) should be able to successfully complete the tutorial level as described in section (11a) within one hour.

11d. Understandability and Politeness Requirements

The game shall use easily identifiable and relatable symbols and words that are naturally understood by players and are mostly universal in the RPG genre (red for health, blue for mana, currency is represented by a coin, etc), anything that is introduced through the game that is not easily understood will be given more time to be explained in detail to the player using tooltips.

11e. Accessibility Requirements

The game shall utilize elements of design to assist partially sighted users play if they wanted to by using recognizable silhouette designs for characters in game, and an adjustable size for the heads-up display to allow for larger icons and text.

12. Performance Requirement

12a. Speed and Latency Requirements

The game shall respond to user input during real-time events such as combat and movement with a response time maximum of 8 ms, but should have an optimal response time of 2 ms for at least 80% of user actions to prevent interruption of user focus as well as combat flow.

The game shall load user requested menus within a maximum of 5 seconds from the time requested to the time accessible for 90% of in-game and title screen menus based on the average gamer's computer as specified by Valve's public data analysis on Steam users.

The game shall load levels/stages within 60 seconds of being tasked to transition to a new game level for 90% of level changes based on the average gamer's computer as specified by Valve's public data analysis on Steam users.

The game shall present a user on UDK recommended system settings with an average framerate of at least 30 frames per second during the time elapsed while playing the game.

12b. Safety-Critical Requirements

N/A

12c. Precision or Accuracy Requirements

An attempt will be made to maintain all numerical stats as integers. However, those elements of the game requiring statistical evaluation involving division/ multiplication such as critical chance, speed, chance to cast and other player or enemy characteristics will be adjusted to evaluate with 2 decimal places to offer a balance on statistical complexity, but player displayed information regarding base characteristics (i.e. health) will round up to the nearest integer to prevent user detail complexity.

12d. Reliability and Availability Requirements

The game shall be available for a player to use 24 hours a day, 365 days a year, unless a patch is deployed to fix important game bugs, then down time should last as long as the patch takes to download which is dependant on user internet speeds, as well as their client update settings. They may choose to opt out of downloading an update if they wish.

The game shall run correctly for an elapsed period of play time a player chooses to spend in the game world with a maximum 5% possibility of immediate game termination (crash to desktop).

12e. Robustness or Fault-Tolerance Requirements

The game shall include an auto-save functionality to periodically save the game so that data/progress loss is minimized if the game suddenly crashes.

12f. Capacity Requirements

The game shall be designed to handle one player during the entire playthrough with a minimum of 10% of total time spent in game resulting in irregular behaviour.

The game should be able to support at least 20 non-player characters (neutral or hostile) on screen at any given time, with additional factors being particle effects, animations, and environment assets, providing the user with a minimum framerate of 30 frames per second, and a response time of at most 8 ms.

12g. Scalability or Extensibility Requirements

The game/engine should be capable of supporting one player while remaining functionally correct, but should be able to raise support up to a minimum of 4 players if some form of multiplayer or cooperative mode is to be added after the single-player mode is completely built.

12h. Longevity Requirements

The game shall be expected to operate functionally upon release until the user is finished with the product. Support for the engine is dependant upon the developers of the current UDK version.

13. Operational and Environmental

13a. Expected Physical Environment

The game shall be played by a user sitting at a computer set-up (using a mouse not a touchpad) with specifications that meet the demands outlined by the minimum system requirements for the UDK.

13b. Requirements for Interfacing with Adjacent Systems

N/A

13c. Productization Requirements

The game shall be installed using a setup installation wizard that does not require a trained user to operate.

The game shall be distributed using a compressed file format passed by CD or downloaded using a direct-download link with potential to expand to distribution services like Steam or Origin.

13d. Release Requirements

The game will be released in episodic format in terms of context and storyline, but each new episode will not require the installation of the previous episode to play the game.

14. Maintainability and Support

14a. Maintenance Requirements

The game must be able to be updated/patched at anytime for all users either using an automatic distribution service requiring the user to do nothing but turn the game on to receive the patch (auto-updater) or issue patch-fixes using a direct-download link from the internet which the user can apply themselves.

14b. Supportability Requirements

The game is supported using a wiki, where detailed information about all aspects of the game including patch notes are available. The game itself also has an in-game tutorial section for players looking for game related help rather than technical help.

14c. Adaptability Requirements

The game is expected to run under Windows 7 and Windows 8 OS environments with design of controls built for keyboard and mouse however the ability to transition to consoles such as the XBOX 360 is a possible environment which would require remapping and re-evaluation of control schemes for the XBOX 360 controller.

15. Security

15a. Access Requirements

Only players with valid CD-Keys will be authorized to access the game upon installation.
Only developers will be able to see source code of game files.
Only developers will be able to see the metrics behind the game functionality.

15b. Integrity Requirements

N/A

15c. Privacy Requirements

The game will only collect information regarding the statistics of game related events and actions but may also collect system information (hardware) requiring the users to be aware of the information collection practices.

15d. Audit Requirements

N/A

15e. Immunity Requirements

N/A

16. Cultural

16a. Cultural Requirements

The game should not be offensive to religious groups and different ethnicities while still being able to use various deities and cultures as inspiration for the lore and myths behind the various abilities, items and entities in the game.

17. Legal

17a. Compliance Requirements

N/A

17b. Standards Requirements

The game should follow the *Games for Windows* certification requirements as a guideline for operational and environmental regulations.

Project Issues

18. Open Issues

There may be some compatibility issues with some of the tools used to develop assets for the UDK. These include (but are not limited to) 3ds Max and Blender.

19. Off-the-shelf Solutions

19a. Ready-Made Products

The game is being developed with the assumption that it is unique. While it may provide gameplay experiences similar to other games in its genre, the developers are currently unaware of any games that possess any similar mechanics.

19b. Reusable Components

The UDN and other places around the internet offer modules for the UDK that the development team will be able to implement in order to save time. Reusing and modifying code will save time as new code will not have to be created to suit specific gameplay mechanics.

19c. Products That Can Be Copied

Unreal Tournament 3 Black is the definitive Unreal Engine game, and as such there are a lot of modules from it that have been made available to the public for commercial and independent game development.

20. New Problems

20a. Effects on the Current Environment

N/A

20b. Effects on the Installed Systems

N/A

20c. Potential User Problems

It is possible that current players of RPG games will have a certain mind-set when initially playing the game and certain elements that are traditionally done a certain way may not be the same for this game in particular. In order to combat confusion when users switch from other traditional RPGs to this one, a tutorial level is set up to reintroduce certain concepts to players and a wiki exists online where players can gather more detailed information if necessary.

20d. Limitations in the Anticipated Implementation Environment That May Inhibit the New Product

It is possible that UDK may not have support for certain functionalities that the game may require during design and programming.

20e. Follow-Up Problems

N/A

21. Tasks

21a. Project Planning

The game's development will utilize a wiki with a ticket system in order to address deadlines and milestones. These will be arbitrarily assigned based on the game's progress throughout its life cycle.

21b. Planning of the Development Phases

The game's development phases are not distinct, but instead blend into each other, with each developer working within a given phase to meet certain criteria. Those phase consist of gameplay coding, asset creation, and testing/debugging.

22. Migration to the new Product

22a. Requirements for Migration to the New Product

N/A

22b. Data That Has to Be Modified or Translated for the New System

N/A

23. Risks

The biggest risk for the project is an incorrect estimation of the time needed for the developers to learn how to use the tools provided to create game assets and code.

24. Costs

There is no monetary cost associated with the game. As mentioned in section 3g, the only resource considered is the developer's time. The amount of time is static, and as such the only factor affecting it is how available time is managed.

25. User documentation and Training

25a. User Documentation Requirements

Game user manual describing all the controls for the game, and how to successfully navigate the various menus and use them efficiently and effectively, in essence, how to play the game.

25b. Training Requirements

The training for the game is left up to the game developers. The level design of the tutorial level will encompass all the basic functions of the game (movement, action, equipment and customization), teaching the player as they continue through that particular level.

26. Waiting room

Potential ideas we wanted to include in the initial design but decided against due to time constraints is the implementation of a "pet" system in which the player can obtain a pet which will fight alongside him. Another possible idea we wanted to include was a multitude of levels the player could have access to, again due to time constraints we had to limit the amount of levels the player had access to in the final release of the game.

27. Ideas for Solutions

As the designers are also the developers for the game, they will update the wiki with potential solutions to issues as development progresses.