Validation & Verification

Trouble Seekers

Version 1.0

Prepared by

Group Name: Buttered Waffle

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

Instructor: Dr. Carette

Course: SFWR ENG 4GP6

Date: Feb. 11, 2013

Table of Contents

- 1. Menu Requirements
  - 1.1 Start New Game
  - 1.2 Save Game Menu
  - 1.3 Save Game
  - 1.4 Load Game Menu
  - 1.5 Load Game
  - 1.6 Main Menu
  - 1.7 Options Menu
  - 1.8 Game Menu
  - 1.9 Inventory
  - 1.10 Quests
- 2. In-Game Requirements
  - 2.1 Equipping Stat Changing Tablets
  - 2.2 Equipping Ability Changing Tablets
  - 2.3 Talking to NPCs
  - 2.4 Attacking with a Basic Attack
  - 2.5 Using a Targetable Ability
  - 2.6 Using a Non-Targetable Ability
  - 2.7 Map Menu
  - 2.8 Player Death
  - 2.9 Death Menu
  - 2.10 Picking Up Tablets
  - 2.11 Receiving a Quest
  - 2.12 Completing a Quest
  - 2.13 Enemy Death
  - 2.14 Triggering a Trap
- 3. Additional Requirements
  - 3.1 Updating NPC Info Bar
  - 3.2 Heads-up Display
  - 3.3 Tutorial Level

1. Menu Requirements

1.1 Start New Game

Verification:
At this current stage, the game does not have a “Start New Game” menu and the player simply starts the game from the beginning of the tutorial level. This fails the game requirement since the player must be able to select the option from a menu upon starting the game. We will be implementing a menu with animated background in the following version of the game as it did not seem like a priority for the first version.
Validation:
The user requires a menu in order to start the game and be able to select different options such as starting a new game, saving a game, loading a game as well as changing the options. Our design currently fails to meet user requirements as the user is not able to select a new game willingly and is instead forced to begin one. Once the saving a loading function will be implemented we will allow the user to willingly start a new game.

1.2 Save Game Menu

Verification:
The player is currently unable to access a menu to save the game, see *Requirement 1.3 Save Game* for more information.
Validation:
The player needs to be able to access a menu to save the game, see *Requirement 1.3 Save Game* for more information.

1.3 Save Game

Verification:
The player is currently unable to save the game, it is currently one of our highest priorities since the ability to save the game is crucial to any game in this generation. Currently this requirement is not met as the player cannot select the save option from the main menu of the game.
Validation:
The player needs to be able to save their progress at any point in time in order to pick up where they left off at any time; our game is currently lacking this feature and therefore fails this user requirement which shall be implemented in the next version of the game.

1.4 Load Game Menu

Verification:
The load game menu is not currently working in game, we have completed the Flash file which will be used to provide functionality to the menu but the buttons are not yet implemented in game. The requirement is therefore unfinished as we need to make sure the player can select the available load game options.

Validation:
The user needs to be able to select their load files from the menu in order to resume the game from that point, the menu needs to have buttons available to click for functionality. The user requirements are unfinished since the menu is created simply not functional yet.

1.5 Load Game

Verification:
The user is currently unable to load the game as the save function is not yet implemented, since these two functions go hand-in-hand once the save function is fully operational the loading function will also be working. We do not meet this requirement since loading does not currently work, it will be implemented for the next version.
Validation:
The player must be able to load their game from a saved state in order to resume progress in their game so they do not need to complete the game in one sitting. Our game is lacking this functionality and therefore fails to meet this requirement; it will be implemented for the next version of the game.

1.6 Main Menu

Verification:
The game currently lacks a main menu since the load, save and new game function are not yet implemented, the main menu would simply be empty and not contain any useful buttons. A menu will be implemented for the next version of the game as the lack of menu means we have not completed this requirement.
Validation:
The user needs to be able to access a main menu in order to select options such as saving, loading and starting a new game. The menu can be easily implemented in the game by using Flash to make an image with selectable buttons; the menu is required in order to correctly implement the user requirement.

1.7 Options Menu

Verification:
The options menu has not yet been implemented. The following is a description of the planned Options menu functionality.
When the user pressed the “Escape” key then they are showed a menu which contains several options such as save, load, exit etc. Options is one of these selections, and clicking it brings up another menu with several additional choices. These choices are input configuration, audio and graphics. Input configuration should allow the user override default key-bindings for actions and map them to custom keys. Audio should allow the user to change the volume of music and sound effects. Graphics should allow the user to change certain render settings of the UDK engine.
Validation:
The Options menu has not yet been implemented and as such validation cannot be completed at this moment for this feature.

1.8 Game Menu

Verification:
When the player presses the “Tab” key then they are presented with a menu which allows them to move items they have collected to different slots, however the player is not able to exit the game from this menu or save and load the game. This does not agree with the requirements we had previously set up, but will be fixed soon.
Validation:
The current menu allows the player to move tablets and items they have picked up to different slots, providing them with different bonuses. However the ability to save, load and exit the game would greatly increase the playability of the game, the fails to meet certain user requirements and will be fixed in the following updates to the game.

1.9 Inventory

Verification:
Currently the inventory screen meets the requirement of holding tablets in the inventory and allowing the user to equip and un-equip tablets. It fails to meet the requirement of holding weapons, miscellaneous items and quest related items, this is because those features have not been implemented yet, however extension of the inventory from tablets only to other items is not difficult and only requires that the code for the remaining items be extended on the TS_Items UC script. The player can successfully transfer tablets in the inventory to other inventory slots, or passive slots/active slots without duplication of the items or loss of the items 100% of the time tested. When the inventory screen is removed, the organizational structure of the items laid out in the inventory retain their order when the screen is brought back, 100% of the time tested. The inventory is successfully brought up using the hotkey “esc” but should be designated to its own hotkey, instead of functioning on the TogglePauseMenu() function, so in that respect it fails to meet that criteria.
Validation:
The current design implementation is sufficient for the user, allowing the user to successfully manage their tablets in terms of spatial organization as well as clear indications to the player of what tablets are being used for passives, which ones are being used for actives, and which ones are simply being held in the inventory.

1.10 Quests

Verification:
The quest screen has not yet been implemented, the goal is to display to the user a quest screen when they press the “q” key in-game. The quest screen displays all active quests, as well as all completed quests. There is one possible selected active quest, indicated by a special marker next to the quest’s name. This can be changed by double-clicking another quest, or by selecting one and pressing the “Change to current quest” button. Each quest, either completed or active can be expanded (if it contains multiple steps) to show the current progress through all the levels of the quest. Each completed portion is indicated with a special marker next to it, and the currently active portion is highlighted, with a description written in the description panel. The user can select other steps of the quest to view their descriptions as well.

Validation:
The quest system has not yet been implemented and as such validation cannot be completed at this moment for this feature.

2. In-Game Requirements

2.1 Equipping Stat Changing Tablets

Verification:
Currently the “equipping stat changing tablets” event does functionally work, allowing the user to move the tablet in the inventory to a passive slot by dragging the tablet to the slot. It fails to meet the requirement of forcing the user to be in a safe zone to equip the tablet. The necessity of such a function is under re-evaluation. Additionally, the gameplay effects of each tablet on the statistics of the player’s character can be seen to update in real-time (~1 tick) and correctly effect the statistic that the tablet is scripted to change. Switching of tablets reacts in real-time and has been tested for duplication of items, or any other unintentional functionality. One-hundred percent of the time, equipping tablets does not duplicate the tablet. An additional requirement that the statistics should not be duplicated or kept without the appropriate tablet has been tested and 100% of the time, the statistics are updated as expected with no duplication in statistics.
Validation:
The current design implementation of the “equipping stat changing tablets” event is sufficient for the needs of the user, allowing the user to successfully move the tablet from an inventory slot to a passive slot. The UI elements of this feature and backend work to allow the user to customize their character and their play style, additionally the tablets successfully change the appropriate statistic of the user and are reflected clearly in the “statistics” tab of the inventory screen providing the user with current information about their pawn’s gameplay related stats.

2.2 Equipping Ability Changing Tablets

Verification:
Currently the “equipping ability changing tablets” event does functionally work, allowing the user to move the tablet in the inventory to an active slot by dragging the tablet to the slot. It fails to meet the requirement of forcing the user to be in a safe zone to equip the tablet. The necessity of such a function is under re-evaluation. Additionally, although the logic and UI for this function is implemented, as there are no spells yet scripted, there is no effect on the gameplay. This function reacts in real-time (~1 tick) and has been tested for duplication of items, or any other unintentional functionality. One-hundred percent of the time, equipping tablets does not duplicate the tablet.
Validation:
The current design implementation of the “equipping ability changing tablets” event is sufficient for the needs of the user, allowing the user to successfully move the tablet from an inventory slot to an active slot. While it does not currently affect gameplay, the UI elements of this feature and backend work to allow the user to customize their character and their play style.

2.3 Talking to NPCs

Verification:
We are currently using the player model as the friendly NPC model in the game, however the player does not have the ability to interact with the NPCs in order to talk to them, exchange items with them or accept quests from them yet. The player should be able to initiate interaction with friendly NPCs whenever they choose, but this feature has not been implemented yet, failing to meet the desired requirement.
Validation:
Currently the NPCs simply stand around and the player cannot interact with them, this will be changed the next version of the game in order to allow the player to receive quests as well as learn more information about the current state of the world. This fails to meet user requirements since the player will not be able to progress further in the game with respect to the story.

2.4 Attacking with a Basic Attack

Verification:
The user is able to find a weapon in game and equip it. Currently, all UDK default and custom weapons are able to be used, but their availability is limited to what is placed in each level. Weapons held by the player are lost when a new map is loaded, and while the player is able to hold a weapon in their hand they lack the third-person attack animations to indicate that they are performing an attack. The sword is still swung when the user presses the left mouse button, and any enemy that the sword touches while it is swinging will take damage.
Validation:
The sword weapon works as intended, but the user must take the additional step of temporary switching to first person via console commands to properly load the sword's model and animations into memory to allow them to execute properly. This falls outside the scope of the game's control scheme and needs to be fixed before the next revision.

2.5 Using a Targetable Ability

Verification:
The user currently does not have the ability to use a targetable ability in the game, this ability was not deemed of high enough importance to implement in the game for the first revision and will therefore be added into the game for the next version. The functional requirement is not met in this case.
Validation:
The player does not have the ability to use a targetable ability yet which would greatly increase the options available to the player in terms of dealing with incoming enemies, our implementation fails to meet user requirements and will be fixed in the following version of the game.

2.6 Using a Non-Targetable Ability

Verification:
The player does not have the ability to use non-targetable abilities yet since they were not considered an important enough asset to have completed for the first version of the game. The abilities are a combination of particle effects with scripts to determine how much damage should be dealt. The requirement is therefore not complete until we add non-targetable abilities.
Validation:
Currently the user cannot use non-targetable abilities which would greatly increase the options available to the player in terms of dealing with incoming enemies. Our implementation fails to meet user requirements and will be fixed in the following version of the game.

2.7 Map Menu

Verification:
The map menu has not yet been implemented. Its implementation is not deemed of very high importance (due to a proposed in game minimap), but will be done if there is time.
Validation:
The map menu has not yet been implemented and as such validation cannot be completed at this moment for this feature.

2.8 Player Death

Verification:
Currently, when the player dies, the screen simply locks over top of the player while the game continues. The planned player death sequence will be an overlayed screen that appears overtop of the locked screen with the options “Continue” or “Quit”. The “Continue” option will load the players last save (auto-save, or otherwise) and the “Quit” option will exit the player back to the Main Menu.
Validation:
A player death sequence is partially implemented, as the screen locks over top of the dead character and the game continues. The overlayed screen is the only thing left to implement for this feature.

2.9 Death Menu

Verification:
This feature is going to be implemented as it is deemed of critical status. The planned player death sequence will be an overlayed screen that appears overtop of the locked screen with the options “Continue” or “Quit”. The “Continue” option will load the players last save (auto-save, or otherwise) and the “Quit” option will exit the player back to the Main Menu.
Validation:
The Death Menu has not yet been implemented and as such validation cannot be completed at this moment for this feature.

2.10 Picking Up Tablets

Verification:
Currently the “picking up tablets” event successfully allows the user controlled pawn to pick-up the tablet when the pawn is within a certain radius of the tablet designated by the UC script attached to the extended Actor function Touch(). This provides the user with an inventory item based upon the item information embedded in the picked-up tablet. This requirement is successfully met 100% of the times an item is picked up by the user.
Validation:
The current design implementation for picking up tablets is sufficient and meets the needs of the user to be able to pick-up items from dropped enemies to potentially increase their character diversity and enhance character progression. The items can be visibly seen by the user within the inventory screen for inspection.

2.11 Receiving a Quest

Verification:
Receiving a quest is an important mechanic of the game, and will be implemented as soon as possible. The planned sequence for receiving a quest is a notification in the top right hand corner of the screen, and adding the quest to the Active Quests portion of the quest screen.
Validation:
The quest system has not yet been implemented and as such validation cannot be completed at this moment for this feature.

2.12 Completing a Quest

Verification:
Completing a quest is also an important mechanic in the game that works in tandem with receiving a quest, it will be implemented alongside receiving a quest. The planned sequence for completing a quest is a notification in the top right hand corner of the screen, and moving the quest from the Active Quests to the Completed Quests portion of the quest screen.
Validation:
The quest system has not yet been implemented and as such validation cannot be completed at this moment for this feature.

2.13 Enemy Death

Verification:
Currently the enemy death event triggers correctly and removes the actor from the set of living pawns on the level. This event does not meet the requirement that the appropriate death animation is played when the enemy dies. This event also does not meet the requirement that all attached graphical elements are removed when the enemy dies 100% of the time.

Validation:
The current design implementation of the enemy death event is built upon the structured functions provided by the default UC script extensions. The current actions associated with the death do not meet the needs of the user because it is hard to differentiate whether or not the dead pawn is actually alive or dead due to lacking death animations. Additionally, due to the attached graphical elements not being removed when the enemy dies, the player may assume the enemy to still be alive, which is not sufficient.

2.14 Triggering a Trap

Verification:
The player is now able to trigger traps which will damage him; certain traps will also instantly kill the player. These traps come in a variety of shapes and sizes, forcing the player to adapt to the situation in order to survive. The traps will currently cause the player the correct amount of damage however some of them, such as darts coming out of the wall, do not disappear after hitting the player. This issue will be fixed in the following version of the game.
Validation:
The player is expecting the paths in the levels to be challenging in some ways and one of the most common way to challenge players is to include traps. The player feels challenged since the traps increase in difficulty as he progresses in the level. The traps meet the needs required by the user since they deal damage upon contact with the player, however they fail to disappear which is an issue which will be fixed shortly.

3. Additional Requirements

3.1 Updating NPC Info Bar

Verification:
Currently the updating NPC info bar follows the requirement that it must be small enough to fit on screen, and large enough to be easily read by the player. The updating info bar does not meet the functional requirement of being removed upon death of the NPC; it fails to remove itself from the screen at least 10% of the time due to a bug. The updating bar does correctly update the stats reflected for the NPC attached to the visual element correctly within real-time (~1 tick).
Validation:
The current design implementation of the NPC Info bar is useful to the user, providing the user with the current health of the enemy in real-time, so that the user can see how much damage he/she is doing and how much health the enemy has left. The current info is limited to health of the NPC however, this can be expanded easily through modular code and scaleform changes to reflect the enemy name, and level if necessary, at the moment we do not find it a necessary requirement through playtesting.

3.2 Heads-up Display

Verification:
Currently the HUD does not meet all the requirements listed in the design document. The HUD does display the current health of the user, the current mana of the user and the current passives of the user in real time (~1 tick). The HUD does not allow the user to open the inventory, the quest list, or map from a set of buttons on screen.
Validation:
The current design implementation meets some of the needs of the user, allowing the user to quickly access the important information about the character: current health, current mana, and current passives. The current design fails to provide the user with a way to access the other game menus without having to remember hotkeys associated with each menu.

3.3 Tutorial Level

Verification:
The tutorial is included in the game to help the player understand the concepts we introduce to the game since some of them may be new to some players. Every action is singled out in specific sections to force the player to use them and familiarize themselves with them, and by the end of the tutorial the player must use combinations of each ability in order to successfully complete the tutorial level. So far the level is about 50% completed and does not fully comply with the requirements but is coming along well.
Validation:
The player must be able to feel at ease with the abilities and commands of the game as well as familiarize themselves with various concepts. The tutorial level is designed in order to do exactly that and so far it achieves this purpose rather well. The level will soon be completed.