University of Nevada, Reno

College of Engineering

Department of Computer Science

Dragonlord Chronicles

Acceptance Criteria and Testing Strategy and Plan

Team 18

Sean Stevens

Jonathan Meade

Ryan Lieu

Christine Vaughan

Instructors

Sergiu Dascalu

Devrin Lee

Advisor

Eelke Folmer

CSE Department Chair

March 8, 2019

Table of Contents

Abstract

Dragonlord Chronicles is designed to be an interactive Role-Playing Game (RPG) that can be enjoyed by players of all ages. The primary focus of the game is fighting, capturing, and training dragons in a medieval fantasy setting, taking inspiration from Nintendo's *Pokémon* series. The game offers players with a complex strategy experience where tactical thinking is required for many of the combat encounters. The game will be developed using Unity and programmed in C#. It will feature 2D pixel art aesthetics to make it feel reminiscent of classic SNES RPGs.

Project Updates and Changes

Team 18 has made noteworthy progress on the project since the previous report. Most components are still in development, but some features have been completed and are ready to be demonstrated. The completed features are as follows:

- The main starting level for the player has been completed. The level consists of a small town, interactable NPCs, and encounterable enemies.
- Sprite work and animation for the starting level has been completed. The environment and all entities within the level have working sprites, and the player has animated movement.

Since the previous report, some changes have been made to the game's specifications. The changes and their reasons are as follows:

- The quest system has been removed entirely. The reason for this change is due to the complexity required to implement even a basic quest system. The time required to implement a functioning quest system is beyond what Team 18 will be able to accomplish by May.
- Rather than being able to save at any point in the game world, saving can only be done a predefined locations. This change was made to simplify the internal process of the save system so that the game only needs to keep track of the scene in which the player saved rather than the scene and coordinates in the world. Whenever the player loads a saved game, the character will be placed at the predetermined location in the scene instead of some arbitrary coordinates.
- Originally the team planned on implementing a full tutorial level in which the player learns the controls of the game, but this has since been reduced to a reference manual the player may access in the game. This change was made

along with the removal of the quest system to save on time and focus on more core aspects of the game.

User Stories and Acceptance Criteria

Character Dialogue:

As a Dragonlord Chronicles player, I want to talk to characters in the game so I can get more information about the game's world.

Acceptance criteria:

AC1. A dialogue box appears with dialogue specific to the character that the player is talking to.

AC2. The dialogue contains information about the lore of the world.

As a Dragonlord Chronicles player, I want to talk to shopkeepers so I can purchase or sell items.

Acceptance criteria:

AC1. A dialogue box with the shopkeeper's dialogue appears.

AC2. Interactable buttons for buying and selling items appear.

AC3. A small window displaying the player's current funds appears.

AC4. If the player presses the buy button, they will be shown a list of items they can buy, along with their respective prices.

AC5. If the player presses the sell button, they will be shown a list of items in their inventory they can sell, along with their respective prices and quantities.

Combat Management:

As a Dragonlord Chronicles player, I want to use attacks to damage enemies so I can defeat the enemies.

Acceptance criteria:

AC1. A dialogue box appears indicating which character used which attack, and if it missed.

AC2. The target's health bar decreases.

AC3. If the target's health bar decreases completely, it is removed from the battle, and if it was the only remaining enemy, the battle ends.

As a Dragonlord Chronicles player, I want to use magic to raise my stats so I can make my characters stronger.

Acceptance criteria:

AC1. A dialogue box appears indicating which character used which spell, and what effect it had.

AC2. One or more of the target's stats are raised, depending on the specific effect of the spell.

AC3. If the target's stat(s) that would be increased have already reached a maximum value, the spell will have no effect.

Inventory System:

As a Dragonlord Chronicles player, I want to use healing items so I can restore my characters' health or mana.

Acceptance criteria:

AC1. The number indicating the amount of that item decreases by one.

AC2. The target's health or mana increases, depending on the item used.

AC3. If the target's health or mana will not increase beyond their maximum value, even if the item would usually restore more points.

AC4. If the target's stat to be healed is already full, the item will not be used.

As a Dragonlord Chronicles player, I want to equip armor and weapons to my characters so I can make them more powerful.

Acceptance criteria:

AC1. The character's information menu will indicate the name of the weapon and armor they have equipped.

AC2. If the character already had a weapon equipped, the new weapon will replace the old. The old weapon will be unequipped and be available in the inventory to equip or sell. The same criteria applies to armor.

AC3. The character’s stats will change to reflect the change in stat boosts the new armor or weapon gives them.

Scene Management:

As a Dragonlord Chronicles player, I want to run into enemies on the overworld so I can start a battle.

Acceptance criteria:

AC1. If the player character collides with an enemy sprite, flags indicating the player, dragon, and enemy to enter combat will be saved to the game manager.

AC2. The game manager will load the battle scene as the active scene and use the flags to populate the scene with the correct characters.

AC3. While in the battle scene, no keyboard inputs will affect the overworld scene.

AC4. If the player wins the battle, the game manager will load the overworld scene again The player will be where they were when they encountered the enemy and the enemy will be gone.

AC5. The player and dragons’ stats will not change between the end of the battle and loading the overworld scene.

AC6. If the player loses the battle, the game manager will load a game over screen.

As a Dragonlord Chronicles player, I want walk into buildings and caves so I can explore.

Acceptance criteria:

AC1. If the player walks into a designated entrance, the player character will disappear from the current scene, and game manager will stop the current scene.

AC2. The game manager will load the appropriate scene depending on the entrance the player went into, and load the player character into the new scene.

Animation System:

As a Dragonlord Chronicles player, I want my character to move with natural-looking animation so I can stay immersed in the game and not be frustrated by strange-feeling movement.

Acceptance criteria:

AC1. When a character moves, its sprite will be animated to appear to be walking.

AC2. When the character changes direction, its sprite will change in order to match the direction it should be facing.

As a Dragonlord Chronicles player, I want the battles to have animations so I can have a more exciting visual cue as to which character did what action.

Acceptance criteria:

AC1. The acting character will make some movement, depending on their action.

AC2. The action will have an animation, specific to that action, that will connect the acting character and the target.

AC3. The target character will play an injured animation if the action damages them.

Testing Workflow

Battle System Workflow

Happy:

1) Player touches enemy sprite in overworld
2) Scene transitions to battle scene
3) Player and enemy take turns using items and attacking
4) Once enemy health reaches zero, player gets reward and returns to overworld
5) If player health reaches zero, player gets a game over and returns to closest checkpoint

Unhappy:

1) Player does not receive a reward for defeating enemies. Defeating an enemy should reward the player with experience, items, or both. Does this show on the victory screen?
2) Game does not transition to the correct scene when engaging enemies or when exiting the battle. Does the background of the battle scene match the corresponding scene in the overworld? For example, if an enemy was engaged in a cave, the battle scene should look like a cave.
3) Enemies do not populate on the battle scene or incorrect enemies are shown. Do the enemies in the battle scene correspond with the silhouette in the overworld?

Dialogue System Workflow

Happy:

1) Player approaches an NPC who isn’t an enemy
2) Player presses the action button
3) UI switches to the dialogue canvas
4) Text populates the UI
5) Player may respond to continue the conversation

Unhappy:

1) The incorrect dialogue tree is loaded. Is each NPC loading the dialogue tree that they are intended to use?

2) There are spelling errors in the dialogue tree or the sentences are worded poorly. Does every word in the dialogue text have correct spelling? Is it easy to understand what characters are saying?

Inventory System Workflow

Happy:

1) Player receives an item, so it is added to the inventory.
2) Player drops, sells, or uses an item, so it is removed from the inventory
3) Player presses a hotkey to bring up an inventory menu to use, drop, or equip an item
4) Items can be used during combat

Unhappy:

1) Player received an item, but it does not show up in the inventory. Does the inventory screen show the correct number of items after receiving new items?
2) Player dropped, sold, or used an item, but the count in the inventory menu did not change. Does the inventory screen show the correct number of items after an item is removed?
3) Player dropped, sold, or used an item that they didn't have. Does the inventory system restrict the player from removing items that they do not have?

Animation System Workflow

Happy:

1) Entity is assigned at least one array of sprites. Each array is given a name and a playback speed.
2) Depending on entity state (walking, standing, battling), one of the arrays of sprites will be selected
3) The sprite renderer will be assigned the next frame after a specified amount of time has passed

Unhappy:

1) Characters are not animated despite being assigned an animation. Is each character animated appropriately (idle animation for standing characters, talking characters should turn to face player)?
2) Animation does not correspond with their action (i.e. a spell animation being played when a melee attack was performed). Is each character showing the correct animation?

3) Entity calls an animation that does not exist. When the developer calls a different animation, did they make sure that the animation clip exists and is named correctly?
4) Animation is played at the incorrect speed. When a different animation clip is created, does it have the correct timing (typically will be 1/8th of a second)?

Testing Strategy

The team will be conducting unit testing, integration testing, system testing, user testing, and acceptance testing.

Each member of the team will be conducting unit testing on the system they have not been assigned to create. The five systems that are being created are the scene manager, combat manager, inventory system, dialogue system, and animation system. This will ensure that each system has been tested by different members of the group in order to find defects.

The integration testing will be done when two or more systems needs to be combined to function together. The individuals that designed the systems will work together to make sure their systems work with each other. After that the two group members that did not design the joint system will user test it.

System testing will be done by the entire group and we will also apply user testing in order to find if there was bugs during the gameplay. This part of testing will be the most effective to find defects in our game. When defects are found they will be reported to the entire team and the individual that created that system will attempt to fix it.

The project will be considered complete when all the acceptance criteria is met and passes the acceptance testing. This will mean that the game has all the basic systems necessary to be functional.

Table 1: Test Plan Chart

Test No.	Test Type	Target	Test Name	Purpose	Test Data or Situation	Expected Result	Actual Result	Outcome and Actions Required
1	Unit Testing	Dialogue System	Loading dialogue tree	Test that correct dialogue tree loads	1. NPC spoken to.	That the right dialogue tree will be loaded for the NPC being spoken to.	1. As Expected	Dialogue was presented as expected. No actions required.
2	Integration	Battle	Loading	Test if the	1. Player	The	1. As	Scene was

	Testing	system/ Scene manager	Battle Scene	Scene manager can switch from the overworld to the battle scene	interacts with enemy sprite. 2. Scene transitions to Battle scene.	scene will change to the battle scene after running into a monster.	Expected 2. As expected	able to transition successfully. No actions required.
3	Unit Testing	Battle System	Fighting Enemy	Test to see if a battle can be completed	1. Player can select attack. 2. Player can deal damage to enemy. 3. Battle can end.	The player will be able to fight the enemy and the battle can end.	1. As Expected 2. As Expected 3. As Expected	Battle was able to complete. No actions required.
4	Unit Testing	Inventory System	Add and Remove	Test to see if items can be added and removed	1. Item is stored in inventory. 2. Item is removed from inventory.	Item will be added to the inventory and then that item will be removed.	1. As Expected 2. As Expected	Item was able to add and remove. No actions required.
6.	Unit Testing	Animation System	Moving Sprites	Test to see if the sprites display the correct animation when walking or standing.	1. Stand still with sprite. 2. Walk with sprite.	The sprite will display the correct standing and walking animation	1. As Expected 2. As Expected	The sprites displayed the correct animations. No action required.
6	Integration Testing	Battle System/ Inventory System	Using and item during battle.	Test to see if an item can be used during a	Date set: 3/12/19 1. Player can access	Player can access inventory and use		

				battle.	inventory. 2. Player can use an item. 3. Item is removed. 4. Character status is updated.	an item to change the status of the character.		
7	Unit Testing	Animation System	Display battle animation	Test to see if animation in battle displays correctly.	Date set: 3/21/19 1. Attack enemy 2. Enemy attacks	Both sprites should display battle animation		

Contributions

Sean spent about two hours working on the Testing Workflow section.

Christine spent about an hour and a half working on the User Stories and Acceptance Criteria section.

Jonathan spent approximately an hour detailing the Progress Updates and Changes, as well as updating the abstract.

Ryan spent about two hours working on the Testing Strategy section.