Ethan Dunham

Final Project design

3/9/17

Vampire Slayer:

I will create a dungeon where each square that points to 4 other squares, some of them will be walls. Each square will either be empty, filled with treasure, a trap, or a creature. I will use the creatures from other projects with some slight changes. The final boss will be a vampire. To beat the vampire, one must have either garlic or a wooden stake. The player will be given the option to pick their attack style, and roll their attributes, much like a D&D character. They will fight their way through the dungeon, getting better gear and leveling up. Eventually, they will find their way to the vampire, where they will fight it out to the death. The game will be turn based. Each move by the player will count as 1 turn. There will be a set limit of turns until the player automatically loses. There will be an option to cheat and have a map displayed for the user to help them through the dungeon.

There will be a main class called Square. Each square will have 4 pointers pointing to another square. The Space class will be the parent class for a Square, BarbarianSquare, WizardSquare ChestSquare, DoorSquare, KeySquare, TrapSquare, PlayerSquare, and VampireBossSquare. Each subclass will have something special happen when the player arrives at that square.

The game will start with the player picking their attributes with 3? Extra points they can allocate. The player will then be shown an empty array with a P for “player” marking their location. As they move around and explore, the map will be filled in. There will also be an option to cheat and show a fully explored map. The player will be given a backpack consisting of 3 available slots. The first slot is prefilled will 5? Health potions. If they drink a health potion, their health will return to max. If drunk in a fight, they will drink the potion and then be attacked by the monster.

The Space class will contain several virtual functions like Type, which returns the char type of the space, to be used by the Space class to determine where a player moves. When the player tries to move, they will send a pointer of the current space the player is on and the direction they want to move to the Space function that corresponds with moving. It will then check the equivalent pointer to see if it runs into a wall or can be moved. If it cannot move, the number of turns the player has to finish the game will not be decreased. If it can move, the number of turns will be decreased main.

As the player moves around the dungeon, they will find various items, a sword that adds a +5 modifier to their attack rolls, a key that opens the door to the vampire, traps, etc.

The traps will take in the player’s dexterity and return the amount of dmg done to the player. The dmg done is divided by the player’s dexterity in the trap function, so having good dexterity is a plus.

The chest and key will be auto looted. They key will be placed in the player’s inventory with a quantity of 1.

Each room will have an action function that displays the info about the room and returns an int to main notifying it what type of room it is. If it is a monster or vampire room, it will call the fight function and

the player will have to fight. They will be given the choice of attack or drink potion. Both the player and monster attack at the same time in the fight, so both will get at least 1 chance to attack. Defense is also done while attacking, which calculates the dmg received. After each round, the dmg dealt to the monster and the player will be displayed. The player's current health will also be displayed in each fight.

The main choices while moving around will be in a menu that loops until 1)Player dies 2)Vampire dies 3)number of turns reaches 35. The options for the menu will be moving in the 4 cardinal directions, displaying inventory, drinking potion, cheat display map. Cheat display map will increment the number of turns by 1, even though the player does not move. All other options will not increment the turn counter unless the player successfully moves. Running into a wall does not increment the counter. Running into a locked door does not increment the counter. Once the user finds and kills the vampire, the game is over.

Action return value indicates the room

1=empty hall

2=door

3=monsters and vampire

Calls fight function (Space* currentSquare, player* player)

Fight will call the corresponding fight and defend functions and the player will fight.

4=chest

Calls addModifier function () in player

5=key

Calls key function (Space* currentSquare, player* player)

6=trap

Calls trap function (Space* currentSquare, player* player)

				w			
			w	P	w		
	w	w	w		w	w	
w	C	M	T		w	k	w
	w	w	w	M	w	M	w
			w			T	w
			w	D	w	w	
			w		w		
			w		w		
			w	V	w		
				w			

Map

W=wall

P=player start

C=chest

M=monster

T=trap

K=key

D=door

V=vampire boss