Demon Destiny

Final Report

Comp 2501A

April 2016

Daniel Lau 100972729

Jeremy Denhartogh 100968279

Liam Costello 100977619

**Overview**

“Demon Destiny is a game where you tear through rooms of enemies, battle powerful bosses, and tame them to be your ridable companions.”

Demon Destiny is a top down action rpg. The game uses a consistent high fantasy aesthetic to bring the world and everything in it together. The player journeys across several varied areas with their creatures in order to defeat enemies and bosses. The interface is kept simple and is only presented to the player when the player requires the information it displays and disappearing when it’s no longer needed.

**Post-Mortem**

**Unimplemented Features**

For most of the unimplemented features that were discussed in the design document, the reason for them not being in the feature complete and final submission is that we did not have enough time to implement them. For all of the features that are not implemented, it was also due to us valuing other features in the game over them. The following features were left unimplemented for both these two reasons as well as more specific reasons listed below.

***Class System***

In the original plans for the game we wanted to have a class system and a skill tree that the player could go through, which would represent another aspect of customization and progression respectively. We planned for an archer, mage and warrior class with several abilities each. This would allow the player to have greater control over their character and increase the game’s replay ability.

In the end, there is only one attack available to the player(not including creature attacks), and the user may only choose one class to play. This particular feature we knew was very ambitious when we included in our design document, and we were only expecting to include it very late into development if time allowed it.

***Action Points***

The game was originally supposed to contain an action point system that would limit the player’s actions but this was replaced with a simple internal cooldown later in the design process for simplicity’s sake. This system would have allowed for more control over various different attacks. The idea was to have some resource(mana) that would need to be replenished if the player wished to continue using some action.

In the end, we put a cooldown on the basic attack of the player. One of the reasons for this is that, since our game is primarily based on fighting, if the player had to carefully use their attacks in fear of not being able to use actions anymore, they may be left walking around doing nothing. Also, since there was currently only one attack available to the player, we thought that it was not worth restricting it as the player wouldn’t be able to do anything without it.

***Multiple Level Designs/Objectives***

In the original design process we wanted to have numerous different level types (such as a wave based levels and defense levels). This was changed and while the game does feature different levels (hub zone, boss rooms) it was scaled back from what we originally intended. The main reason for this is time, as we currently have all the framework necessary to easily make these different objective based levels.

***Items***

Several of the item types were also not implemented as we had wished. We wanted to add potions, armor, creature eggs and currency. Items are really easy to add to the game, so we decided to add them in last after other more important features were implemented. Creature eggs and Armour were added to the game post feature complete, and are present in the final submission. The items currently present in the game are weapons, armour and creature eggs which all function as planned.

***AI***

We had plans for various enemy AIs. Some examples of AIs we planned were pathfinding, shortest path and multiple states. In the end, the enemies in our game use only a few AIs, and they are not as interesting as we had hoped for. Enemy AI was present in our game early in development, but we planned to save more complex behaviours for after they were discussed in lectures. Unfortunately, we were unable to include what we had learned in lectures into our game.

**Implemented Features**

Outside of the features that we didn’t end up putting in the game, there are several that are present, and function as planned. These include: differently behaving creatures, a boss, few enemies, different tile effects(damaging tile), inventory and items.

**Discussion**

Looking back we are overall happy with the final product but wish we had more content in the game in the form of bosses, levels, creatures or some form of final boss to fight to give the game a proper ending. We personally think that the only think setting us back from implementing everything we had planned is time. Much effort was spent on building the framework of our game, such that more content could be added with relative ease. We wish that we would have better used the time that we allotted to work together. Though we did spend a lot of time working on this project, had we put more of our efforts into sitting down and coding instead of theorizing the best ways to do things, our game would be closer to completion. The thing we wish that we did differently the most is assigning tasks to individual members and ensuring that they were completed by group meetups. This would allow each member to focus on individual portions of the game, and we would have been able to progress at a faster pace in several aspects of our game.

Added functionality/content:

-Ability to delete items from inventory

-Bosses drop Creature Egg, which allows them to switch creatures

-Added Ant creature

-Added “Tank” creature, provides a charged attack when riding

-Added Amor item, reduces damage the player takes

**Bugfixes and Efficiency Upgrades**

Between the feature complete version and the final submission several different bugs were fixed.

**Bugs**

***Inventory Bugs***

-Player is no longer able to ride/stop riding creature while in inventory screen

-Player can no longer pick up items while inventory is full

-Player can no longer pick up items while in the inventory screen

***Creature Bugs***

-Spider no longer shoots while in riding state

-Spider can no longer walk off the map

***Game Bugs***

-You can now close the game window while riding a creature

-Win condition fixed

-Game now properly stops after win or lose

**Optimization**

***Object Checks***

We changed the game to check only the enemies in the player’s current area instead of all the enemies in the game. A similar change was made with items where when the player tried to pick them up it would check the position of all the items in the world instead of just those in the player’s current area. So, now we are only checking items and enemies that are relevant to the player.

***Collision Check***

Previously, we only had one collision check for objects, and that was the AABB check. Now, we have added a circle collision check(without square roots), and if that returns true, then the program will execute the final collision check(AABB) to see if two objects collided.

Previously, we had all our collisions done based on a AABB with the dimensions of 64x64 pixels. Since we have several different sprites with different sizes, this caused there to be many false positives. We have now changed the size of the collision check(with circles and AABB) to be based on the sprite’s size rather than the arbitrary 64x64 pixel box.

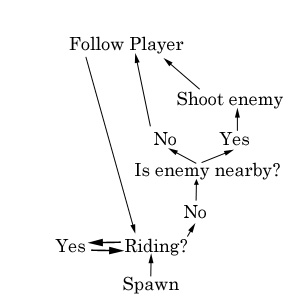
Artificial Intelligence

The enemies become active when the player enters their area. They set the player as their target and move towards them in order to attack. The creatures change between being AI controlled and player controlled depending on whether the player is riding them or not. The boss in the game strafes left and right and sets his target to the player’s current position. He then shoots a projectile that changes the terrain underneath the player.

We wished to include more sophisticated AIs but did not have the time to implement it. Some planned improvements would be to include the planned functionality in our design document of differently behaving enemies, but we also wished to have path finding too. An interesting improvement that we would like to add is having some sort of autonomous character steering for the creatures. We had planned for the creature’s follow state to behaviour differently based on the creature type, but never got around to doing it.

We plan on making this follow state more interesting by letting the creature move on its own based on pre-set conditions. We want to do something similar to Boids(discussed in lecture slides), where the creature would follow certain rules and act based on them. So, an example of something we were thinking of is, a creature’s movement and target locations would be based on the size of groups of enemies. A group would be defined as enemies within a certain proximity to each other. So, the creature would move toward and attack smaller groups of enemies, and try to avoid bigger groups. When there are no enemies present, the creature would return to the player and circle him until more enemies appeared.

Creature Finite State Machine:



Milestone Calendar

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  | Monday | Tuesday | Wednesday | Thursday | Friday | Saturday | Sunday |
| Week 3 |  | 19  January | 20  Register Group | 21 | 22 | 23 | 24 |
| Week 4 | 25 | 26 | 27 | 28 | 29 | 30 | 31 |
| Week 5  Framework Complete | 1  February | 2 | 3  Design Document | 4 | 5 | 6 | 7 |
| Week 6  Character movement, level changing | 8 | 9 | 10 | 11 | 12 | 13 | 14 |
| Break  Projectiles | 15 | 16 | 17 | 18 | 19 | 20 | 21 |
| Week 8  Complete Progress Report | 22 | 23 | 24  Progress Report | 25  Group Meeting | 26  J: scrolling  D:creature  L: textures | 27  Group Meeting | 28 |
| Week 9  Beast Riding, first phase of Inventory | 29  J: pathfinding  D: enemy type  L: weapon | 1  March | 2 | 3  Group Meeting | 4  J:spread sheet  D:boss  L: potions, levels | 5  Group Meeting | 6  J:Minimap  D:skills  L:armour |
| Week 10  Creature and enemies | 7 | 8  Each member finishes up what they need to. | 9  Vertical Slice | 10 | 11 | 12  Group Meeting | 13 |
| Week 11  Bosses, more levels | 14  J: Items  D:Inventory  L:Creatures | 15 | 16 | 17  Group Meeting | 18 | 19  Group Meeting | 20 |
| Week 12  Complete inventory, more content | 21  Group Meeting | 22  Finishing tasks | 23  Feature Complete | 24 | 25 | 26 | 27  Prepare for demo |
| Week 13  Bug fixes, content | 28  Rehearse demo | 29  Class Demo | 30 | 31  Class Demo | 1  April | 2  Group Meeting | 3 |
| Week 14  Final Report | 4  Bug Fixing | 5  Group Meeting | 6  Write final report |  | Final Report |  |  |