Comp 441: Project 2 Report

Group Members: Andrew Miller, Spencer Kershaw, Garrett Young  
Game Name: American Hobo: The Resistance

Summary:

Our game is a top down sword combat game that uses a wave mechanic to dictate its levels and scoring. Armed with a sword, you fight waves of enemy Hobos, also wielding swords, and try to kill the waves of enemies dictated by the level. Each level has progressively harder waves. There are two enemy types, the second having more health and moving faster. When you are injured by the enemies, you lose health, which you can replenish between levels using your score points. In this way, the less you are injured, the higher your score will be. This is the main way we linked the levels together, so your actions in the prior levels have an impact at the middle and end of the game.

We initially planned on having multiple items as part of the in-game store, but after exploring initial implementations of these, we decided on focusing on a single weapon. Our reasoning for this was mostly in implementation complexity. The benefit gained from having different weapons did not add a ton to the game, and the extra time it would take to make final animations and code concessions for each item would have left us much less time to focus on the core mechanic. We decided to drop these, but implement the rest of the game so that we could add them at a later date.

We really liked how the old Legend of Zelda combat mechanics worked, but we also wanted to have free movement, not bound by the cardinal directions. We decided to have this style movement, but only have attacking in the four directions. After experimenting with it, we actually really liked the mechanic, since you would have to maneuver more to attack.

After sitting back and examining what each of us did, we determined it was not really feasible to assign total ownership over any one part of the game to any of us. We were all modifying all parts of the code, and while we definitely took the lead on different parts, we did not have any exclusive ownership. For the purposes of the spec however, we would determine the following as our “leads:”

Garrett: Menu Class, Game Logic  
Spencer: Enemy Classes, Game State Logic  
Andrew: Hero Class, Sword Class, Animation Lead

However, we can’t emphasize enough that we all worked on ALL portions of the code.

PROTIP: If you feel like cheating, you can use the F1, F2, and F3 keys to switch between the levels.

Total Points: 100 /100

|  |  |  |
| --- | --- | --- |
|  | Points Percentage | Comments |
| Graphics, Sound FXs, Music  Must have at least two textures.  Textures must look reasonable and be appropriate to the game.  Textures created or modify by you.  Must have at least two different sound fxs. | 15% | We used different sound effects for different actions such as swinging the sword and enemy deaths. We also have a snazzy soundtrack.  15/15 |
| Originality in game  If the game is based on an existing game, at least two novel game play elements are required.  If the game is based on an existing game, at least one unique interface or control element is required to support the item above.  Game must exhibit good flow, and levels should build on each other to create a game experience with a beginning middle, and end. | 15% | We use Zelda-esque combat mechanics, but with a new wave system, and three dimensional movement. In addition, we have added more of an incentive to stay alive, by using the players score as his means of healing himself. Our levels each get progressively harder, and all show unique areas.  15/15 |
| Stability, playability  bug free  smooth  reasonable performance | Pass/Fail | Pass – Our game runs pretty well, and we haven’t run into any problems with it. |
| Easy to play  A player should be able to play the game without (significant) training.  Any key combinations should be simple and appropriate to established conventions, e.g., do not remap WASD to YHNJ.  Keyboard and must should be used for appropriate functions, e.g., do not use a mouse to pick out letters from a menu when the keyboard is available. | 10% | Our controls are very simple and we have a nice splash screen to explain them at the beginning of the game. We use the traditional arrow keys and do not switch between the mouse and keyboard.  10/10 |
| Quality (simple, fun, easy to understand)  The game must play with an organic feel: all the element must work together to create an entertaining experience.  The rules must be clear and simple, so that the player can concentrate on the game, not learning the rules.  The game must exhibit good design principles, such as flow.  While “fun” is hard to quantify, the game should be fun to play.  Game performance is good: frame rates are kept up, response to controls are fast, objects move on the screen as expected (without hiccups or stutters), etc. | 50% | We think the game is pretty fun. While we would like to add more elements for the player to purchase, we decided it would be better to focus on the core gameplay. The game has simple and clear objectives. Our game runs efficiently and does not lag.  50/50 |
| Report | 10% | See Text above  10/10 |