Corpse Run Design Document

2/10/2015

Fundamental Gameplay Functions

* Character must wander around a graveyard, performing exorcisms and avoiding monsters until caught. Each exorcism gives points.
* Basic character capabilities:
  + Move
  + Attack
  + Exorcise Gravestones
    - Stay within an area for a continuous duration of time
    - Protect a tape recorder chanting an exorcism
    - Possessed gravestones spawn one at a time
* Survival mode – monsters keep coming until character is dead
  + Scoring system arcade style – notes your high score at the end
* Obstacles:
  + Monsters will attempt to kill you
    - Rate of monster spawning speeds up as game continues
    - Type of monsters gets stronger as game continues
  + Physical obstacles, i.e. gravestones, gates, collision

Additional Features

* Hit point system for player and monsters
* Item pickups:
  + Major pickups spawn upon exorcising a grave
  + Minor pickups spawn on a timer
* Combat System:
  + Multiple weapons that can be switched between
    - Base weapon is an unlimited melee shovel
    - Pick up guns as you play – major pickups
    - Ammo is a minor pickup
      * Ammo is generic
      * Ammo pickups are semi-random
* Temporary status effects for monsters and players
* Multiple difficulty levels
  + Potentially difficulty-locked content

Necessary Code Components

//Ben: This is all v.0.1.1 written by somebody who has really no idea how Unity works or what can be done in it, as a programmer thinking about how this would be done on a base level. So yeah, this is all subject to change once it’s looked over by someone with their shit together.

Map Data

Map class contains data for both the underlying start variables of a given game and the current game state.

*Constants generated for each game:*

* Difficulty
  + Has an effect on other variables such as spawn rates, # and placement of graves
* Monster spawn rate constants
* Constants for exorcism
* Pickup spawn rates

*During-play variables:*

* Existance/locations of characters, pickups, and environment objects
* Timer and current spawn rate data
* Score
* List for any ongoing effects that effect the whole map and not individual characters

*Functions:*

* Spawn monsters
* Control haunting/exorcism data
* Spawn pickups
* Track score
* End the game when player HP reaches 0

Score Data

Real easy one, just a file that stores the score data and an associated name as returned by the map data at the end of the game.

Character Data

Generic character class used to represent any active character in the game. Will not ACTUALLY be used in game, just a superclass for monster and player subclasses

*Members:*

* Character Location
* Hit point maximum and hit point total
* Movement speed and movement type(can it move through solid objects? Flight?)
* List of ongoing effects that modify current members or methods

*Methods:*

* Mainly just input to map data re:movement and attack usage
* Take damage
* Kill-switch

Two subclasses of Character should exist: Player and Monster

Player Data

Subclass of Character with variables and functions specific only to the player.

*Members:*

* List of currently available weapons
* Currently selected weapon
* Amount of each ammunition type

*Methods:*

Pass data to map data:

* Keyboard input & controls
  + Movement
  + Attack
  + Start exorcism
* Pickup item

Alter own variables:

* Change weapon
* Pickup item
* Attack(alter ammo value)

Monster Data

Subclass of Character with variables and functions specific only to monsters.

*Members:*

* Data for individual attack

*Methods:*

* Artificial intelligence(input to map data)
* Attack functions

Weapon Data

Class for the individual weapon types a player can pick up.

*Members:*

* Reference to the player in order to access variables
* Ammo type/cost
* Existence/duration of a cooldown

*Methods:*

* Fire weapon

Pickup Data

Class for consumable pickups/powerups that may spawn on the map.

//Ben: This could be implemented as a single class, or as a superclass with a subclass for each individual pickup.

*Members:*

* Effects upon pickup – which player stats it modifies, which effects it adds to the player or to the map

*Methods:*

* Pickup

Effect Data:

//Ben: Effects are a REALLY broad category and I need to clean this up.

Class for temporary effects applied to the player, monsters, or map itself.

*Members:*

* Duration timer

*Methods:*

* Initiation: what happens when the effect is applied
* Kill-switch: what happens when the effect ends