

University of Regina

ENSE 400/477

Project Night Terror System & Object Design

Version 2

Author:
Evan Geissler
200331033

Advisor:
Dr. Christine Chan



University
of Regina

Last Modified
March 26, 2019

Revision History

Revision Version	Date
Version 1	February 24, 2019
Version 2	March 26, 2019

Table of Contents

1	Introduction.....	1
1.1	Purpose.....	1
1.2	System & Component Overview.....	1
2	Class Diagrams.....	1
2.1	Purpose.....	1
2.2	Characters.....	2
2.3	Pickups.....	3
2.4	Weapons.....	4
3	Flow Chart Structures.....	4
3.1	Purpose.....	4
3.2	General Game Flow.....	4
3.3	Demon Behaviour State Flow.....	5
3.4	Ghost Behaviour State Flow.....	6
3.5	Saving & Loading.....	7
4	Perk Trees.....	8
4.1	Purpose.....	8
4.2	Health.....	9
4.3	Flashlight.....	10
4.4	Pistol.....	11
4.5	Rifle.....	12
5	Blend Spaces.....	12
5.1	Purpose.....	12
5.2	General Blend Space.....	13
6	State Machines.....	14
6.1	Purpose.....	14
6.2	Main Character.....	14
6.3	Demon Characters.....	17
6.4	General Ghost.....	18
6.5	Main Ghost.....	18

List of Figures

Figure 1.2-1.	Component Overview.....	1
Figure 2.2-1.	Character Class Diagram.....	2
Figure 2.3-1.	Pickup Class Diagram.....	3
Figure 2.4-1.	Weapon Class Diagram.....	4
Figure 3.2-1.	Basic Flowchart of the Game.....	5
Figure 3.3-1.	Demon Flowchart.....	6
Figure 3.4-1.	Main Ghost Flowchart.....	7
Figure 3.4-2.	General Ghost Flowchart.....	7
Figure 3.5-1.	Loading Flowchart.....	7
Figure 3.5-2.	Saving Flowchart.....	8
Figure 4.1-1.	Perk Tree Example	8
Figure 4.2-1.	Health Perk Tree.....	9
Figure 4.3-1.	Flashlight Perk Tree.....	10

Figure 4.4-1. Pistol Perk Tree.....	11
Figure 4.5-1. Rifle Perk Tree.....	12
Figure 5.2-1. General Blend Space Graph.....	13
Figure 6.2-1. Main Character Main State Machine.....	15
Figure 6.2-2. Main Character Walk-Run Sub State Machine.....	15
Figure 6.2-3. Main Character Rifle & Pistol Sub State Machine.....	16
Figure 6.2-4. Main Character Cross & Knife Sub State Machine	17
Figure 6.3-1. Demon State Machine.....	17
Figure 6.4-1. General Ghost State Machine.....	18
Figure 6.5-1. Main Ghost State Machine.....	18

List of Tables

Table 5.2-1. General Blend Space Points.....	14
--	----

1 Introduction

1.1 Purpose

The purpose of this document is to outline and summarize the systems, structures, and objects of Project Night Terror. It is important to note that as a game, there are very many structures involved. In response, this document will outline major classes, structures, and objects. It is also important to note that an Unreal Engine 4 Blueprint is a type of representation for classes and the classes will be created through blueprints instead of C++ text code.

1.2 System & Component Overview

The purpose of the diagram in Figure 1.2-1 is to show how different parts of the game relate and connect. This will help aid in understanding where some of the many figures are located and/or work together.

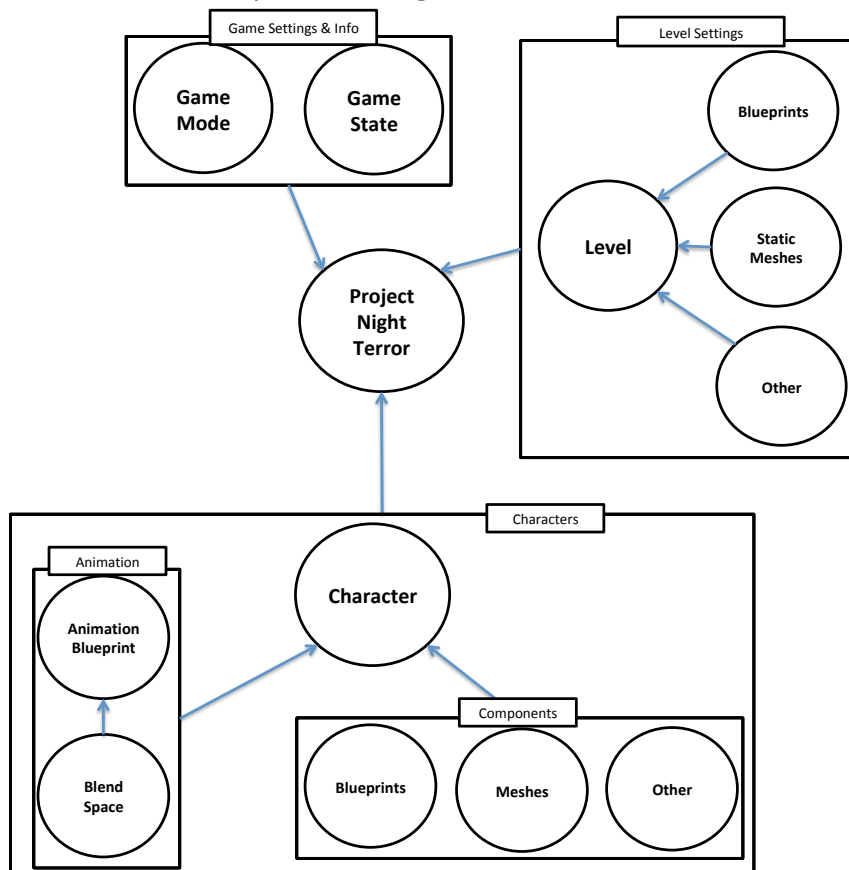


Figure 1.2-1. Component Overview

2 Class Diagrams

2.1 Purpose

The purpose of the class diagrams below is to reduce the amount of unneeded code duplication; increase the use and ease of modularity by utilizing inheritance; and to assist in decreasing the time that is required to make new features and classes. Unreal Engine 4 greatly assists and makes it easier to create and use child classes by having a quick command to create them. The engine also makes it very easy to

change parent classes as any changes are immediately and seamlessly added to the children with no additional work. Accessing the parent classes also helps to reduce unneeded code. For example, instead of needing to call the fire function specifically from the rifle class or the pistol class, the parent base *Gun* can be called instead with no regard to what weapon is actually being used.

2.2 Characters

Characters are any class that is humanoid or has AI behaviour to it. Most of the characters in the game are interactive and can interact with the player or other characters in some way. Demons can attack and be attacked by the player and a specific human character; the specific human character can attack the demons; and finally the player themselves who can attack or be attacked by demons. Along with these characters there are non-interactive characters that do not have the same abilities and will just roam the environments. The only exception to this is the main ghost who will, indirectly, guide the player character.

Each of the characters can move and have specific movement animations associated with that specific character. Each character also has a specific set of components and skeletal mesh. Interactive characters have additional functionality most commonly with their attacks and their variable health.

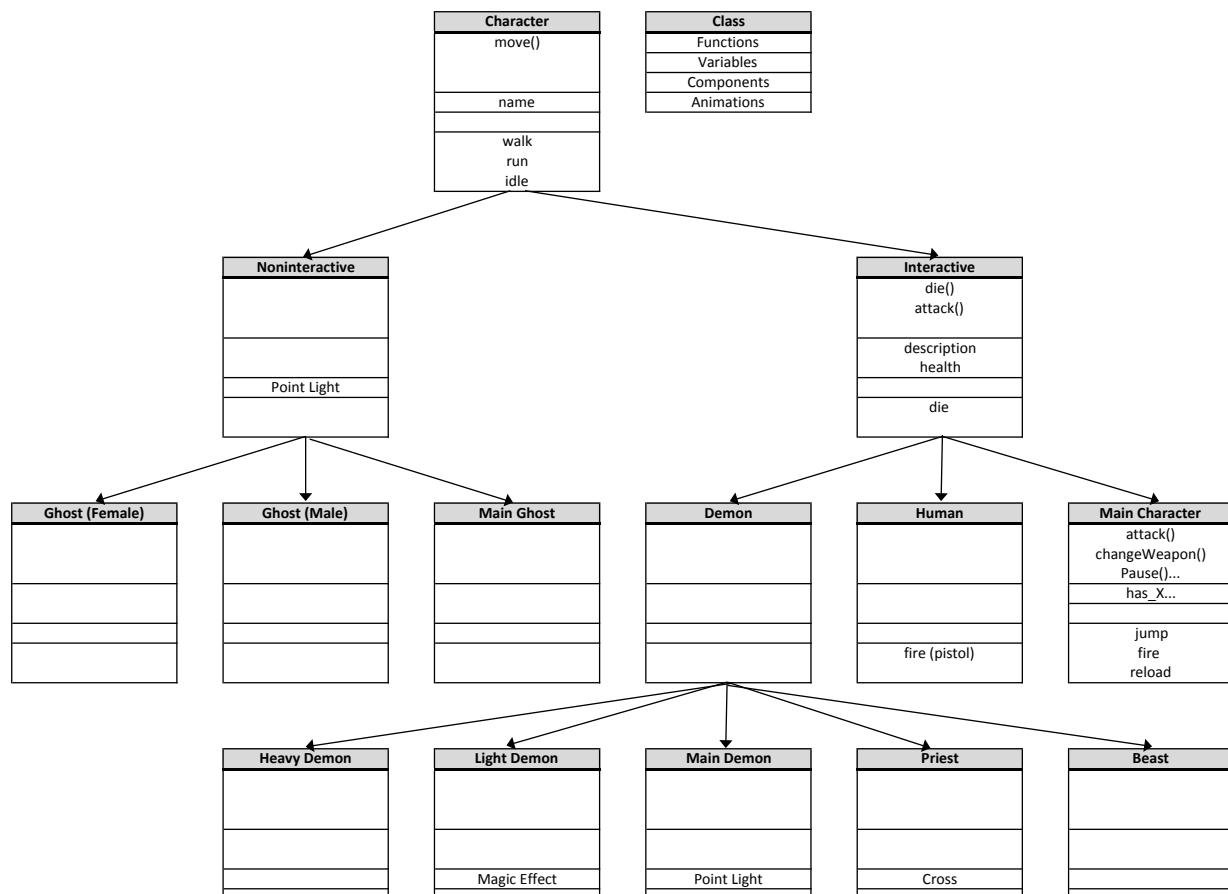


Figure 2.2-1. Character Class Diagram

2.3 Pickups

Pickup classes are any form of pickup that the user can encounter in the game including story pickups, ammo, and health. Aside from each pickup having a name, each pickup has a similar destroy function that will remove it from the level once the player character has picked it up. The main difference between common leaf nodes is the model used for specific classes. Ammo pickups increase the player's current ammo for a specific weapon; health pickups increase the player's health; story pickups include notes, artifacts, audio logs, and upgrades.

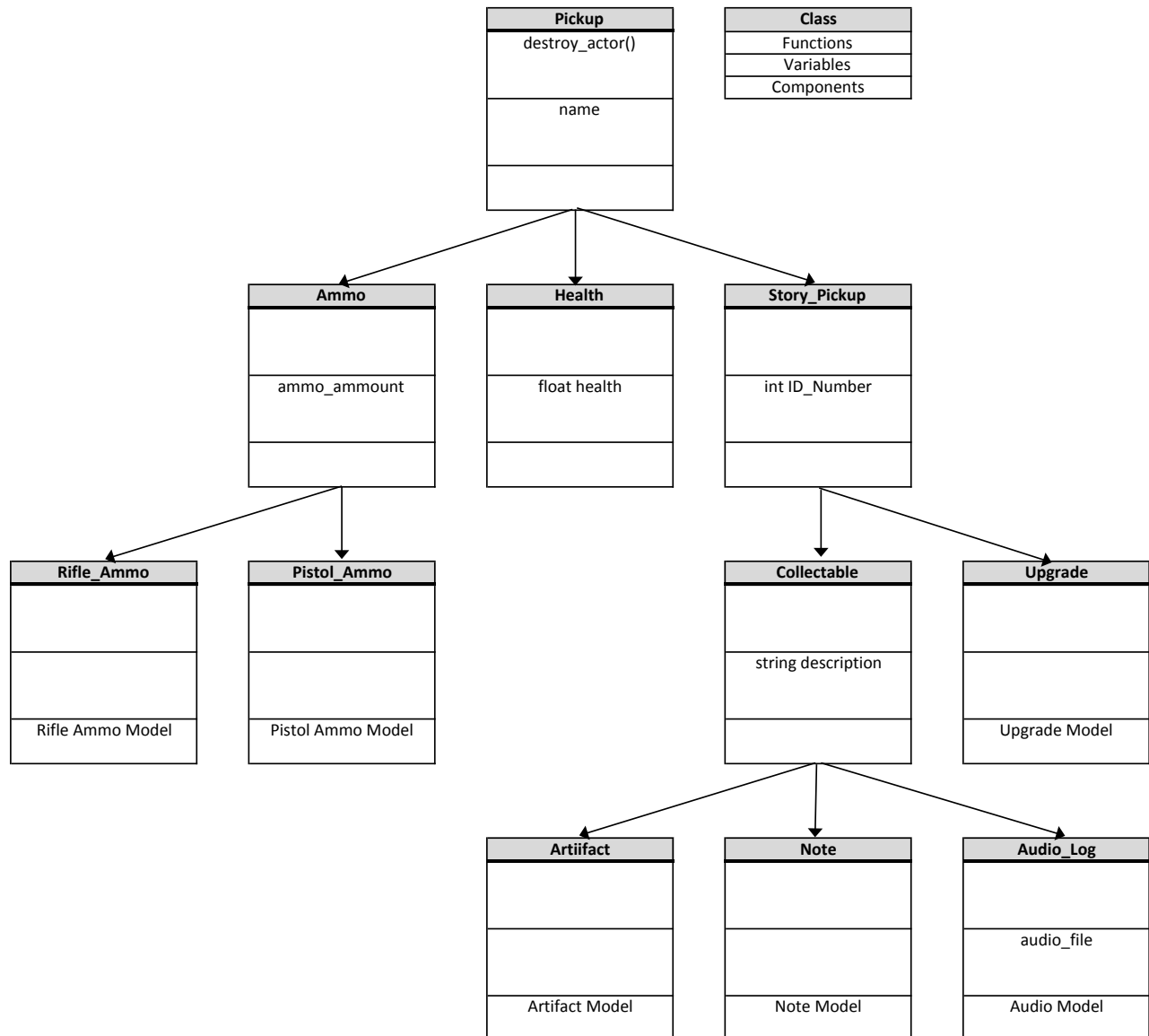


Figure 2.3-1. Pickup Class Diagram

2.4 Weapons

Weapon classes are split into two main categories, guns and melee. All weapons require their own set of animations when used by the player and also their own static mesh. Gun classes require a fire and reload functions and their own variables such as magazine size, accuracy reload speed. Melee weapons are weapons that the player uses in close combat with the demon characters. These require an attack range, as the demons should only take damage when they are a certain range from the player and when they are struck by the specific melee class. Currently, the difference between gun leaf nodes and between melee leaf nodes do not have any specific difference other than the ones already mentioned.

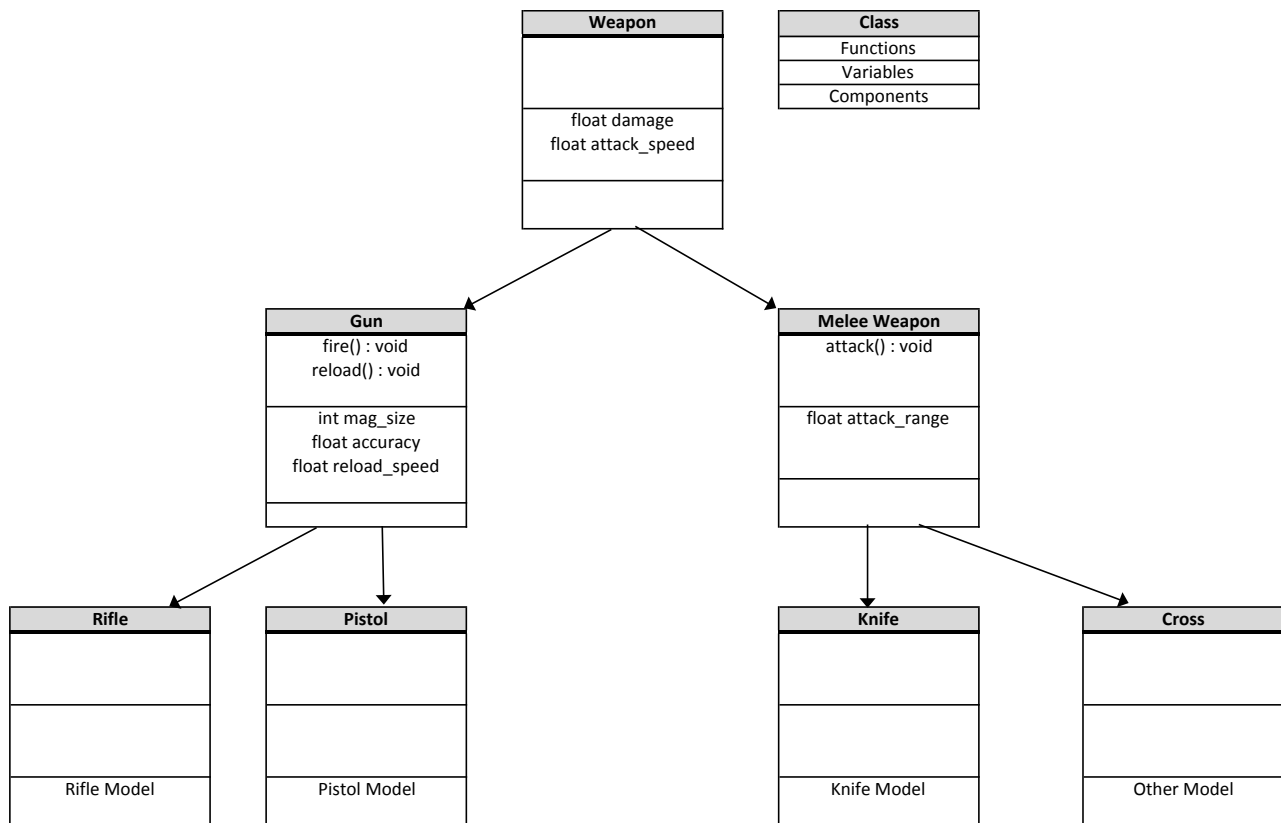


Figure 2.4-1. Weapon Class Diagram

3 Flowchart Structures

3.1 Purpose

Flowcharts help to assist in visualizing how the game runs from start up to closing and to also visualize the flow of AI behaviour from one state to another and the in workings of a state itself. This helps to clarify workflow and to assist in debugging/testing purposes.

3.2 General Game Flow

The most general, but always used flow is the overview of the game shown in Figure 3.2-1 below. All players will have to start by opening the game, which brings them to

the main menu. From there, the player has the option to delete saves, load a save, exit the game, or start the game from the beginning. Once in normal gameplay, the player has similar options, but will usually return to the same normal state. When a level is completed the game is saved and the next level is loaded. Once the game is finished, the player will be shown the credits, and will finally be sent back to the main menu.

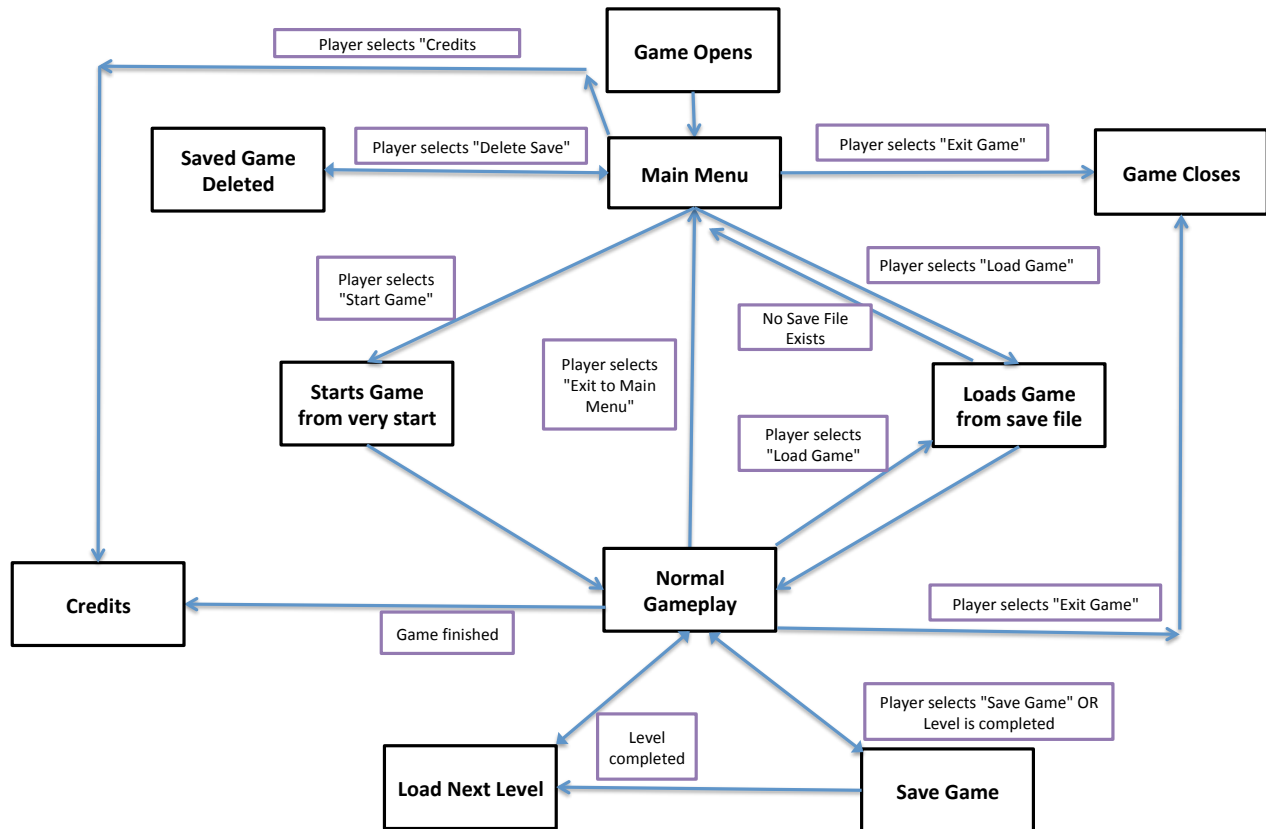


Figure 3.2-1. Basic Flowchart of the Game

3.3 Demon Behaviour State Flow

All demon characters follow the same general flow. Upon begin play, the demon will either stand idle or will randomly walk to a randomly picked location within a navigation-bounded area. Upon seeing the player, the demon will move so they are in range with the player; if the demon is a ranged character they will stop further away from the player and will stop closer if they are a close combat demon. For example, light demons, the main demon, and the priest all have ranged attacks while the heavy demon and beast characters have short ranged attacks. Many of the checks to attack and to move are updated with each event tick.

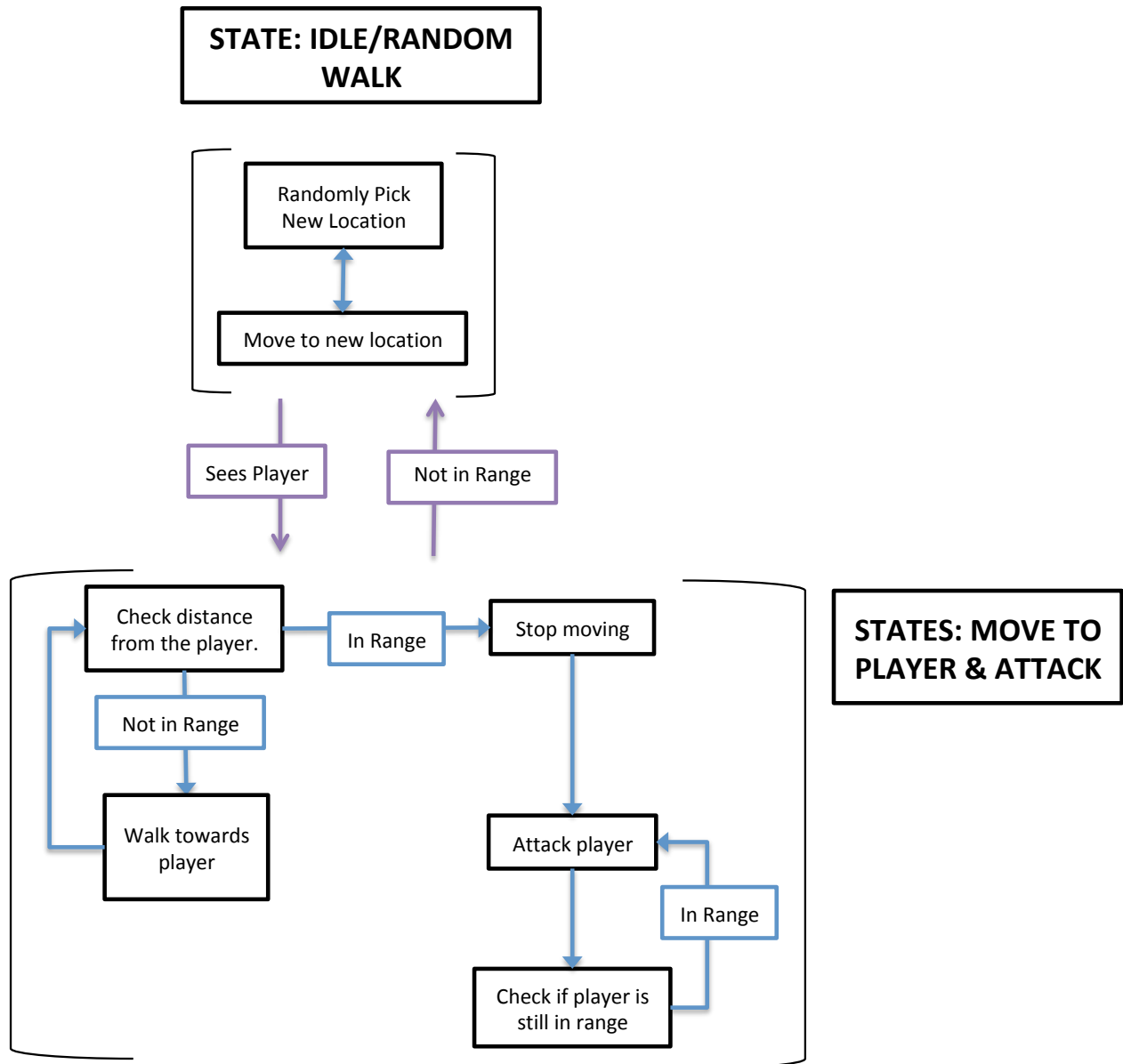


Figure 3.3-1. Demon Flowchart

3.4 Ghost Behaviour State Flow

Both the general ghost characters and the main ghost character follow a very similar flow within their states. Both sets of characters will have a location that they will either currently be at or one that they need to randomly select or have preselected and move to. This will continue indefinitely for the general ghost characters and will stop for the main ghost when she has reached her final predefined location.

STATE: IDLE/RANDOM WALK

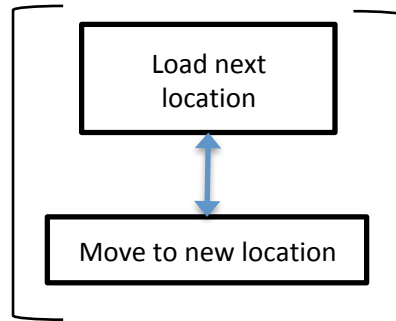


Figure 3.4-1. Main Ghost Flowchart

STATE: IDLE/RANDOM WALK

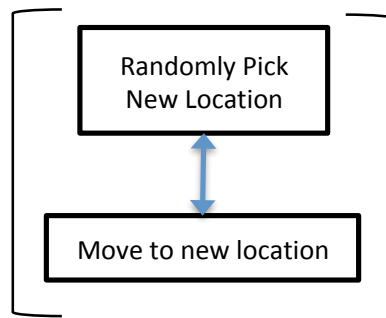


Figure 3.4-2. General Ghost Flowchart

3.5 Saving & Loading

Loading the game is very simple. The loading function checks to see if there is a save file and will do nothing if there is no file. If the function does find one, then the save is opened, and the variables in that file are stored into the current game variables.

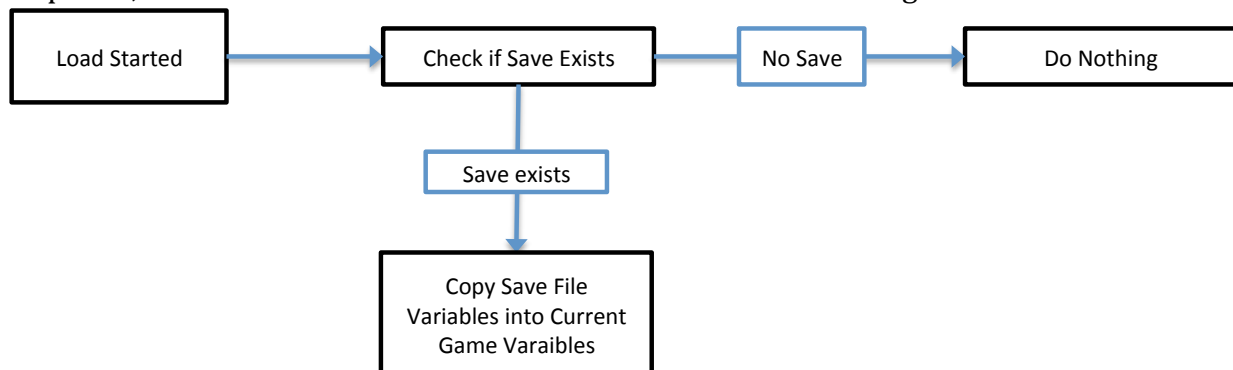


Figure 3.5-1. Loading Flowchart

Similar to loading, the save function for the game is very simple. The saving function checks to see if there is a save file and will create a new one if there is no file. If the function does find one, then the player will be asked if they want to overwrite that current save. If the user selects no then the game is not saved and nothing happens. If the user chooses yes then the save is opened and the current in game variables are saved to the save file.

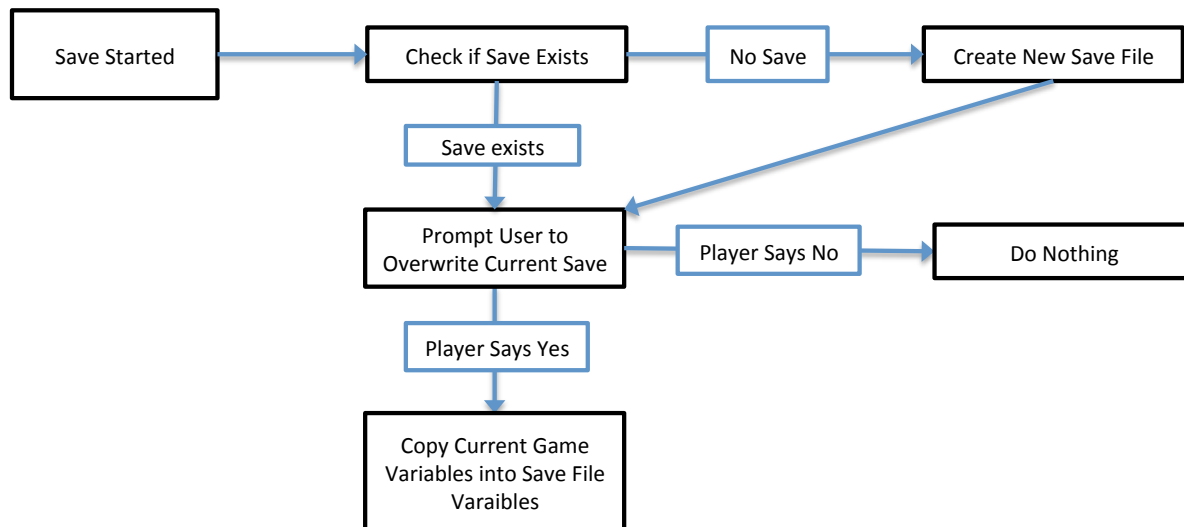


Figure 3.5-2. Saving Flowchart

4 Perk Trees

4.1 Purpose

Throughout the game players collect upgrades to enhance their character in different categories. The different areas are: health, flashlight, pistol, rifle, cross, and knife. The player can select any perk they want as long as all prerequisite parent nodes have been unlocked. Initial nodes in the tree can be chosen with no prerequisite nodes required. For example, in Figure 4.1-1 node 1 and node 2 can be chosen right away. However, node 3 can only be chosen once node 1 has been previously chosen. Node 4 can only be chosen once node 1 and node 2 have been previously chosen.

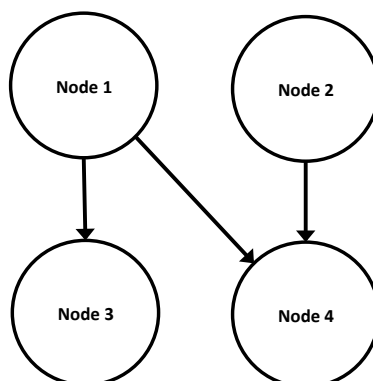


Figure 4.1-1. Perk Tree Example

For the scope of this project, only perks for health, the flashlight, the pistol, and the rifle will be implemented. Perks for the cross and knife will be created or implemented currently.

4.2 Health

The health perk tree consists of 5 nodes shown in Figure 4.2-1. These nodes include permanent health increases to the player and also increases to the health gained from health pickups. *Health Increase 1* and *Health Increase 2* increases the player's maximum health by 25 and *Health Increase 3* increases it by another 50. *Health Pickup 1* and *Health Pickup 2* increases the amount of health gained by 10 each.

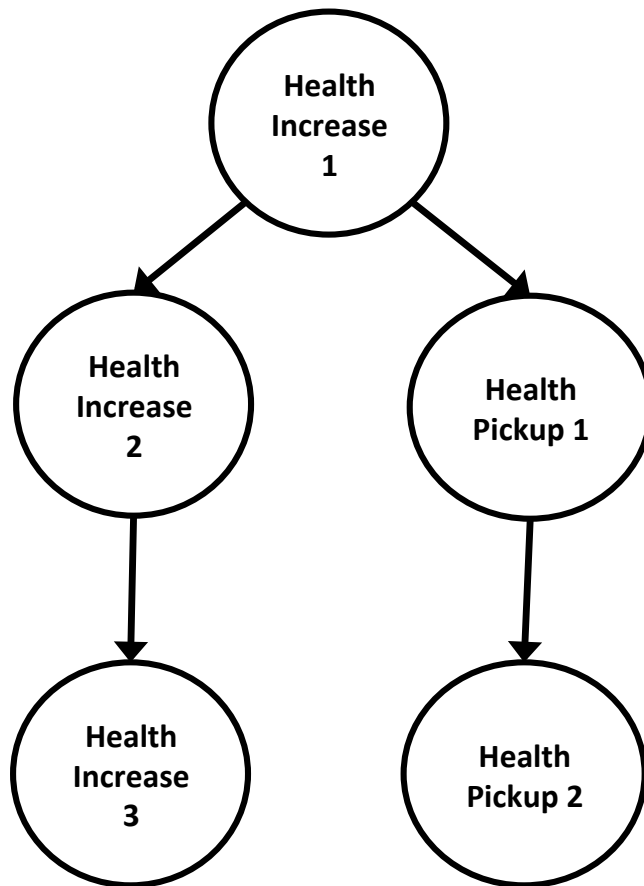


Figure 4.2-1. Health Perk Tree

4.3 Flashlight

The flashlight perk tree consists of 6 nodes shown in Figure 4.3-1. *Battery Capacity 1*, *Battery Capacity 2*, and *Battery Capacity 3* increases the flashlight maximum charge by 25, 25, and 50 respectively. *Efficient Batteries 1* and *Efficient Batteries 2* decrease the speed that the flashlight runs out of power by 0.25s each. Finally, *Limitless Power* allows the user to have the flashlight on for as long as they want without having to recharge it.

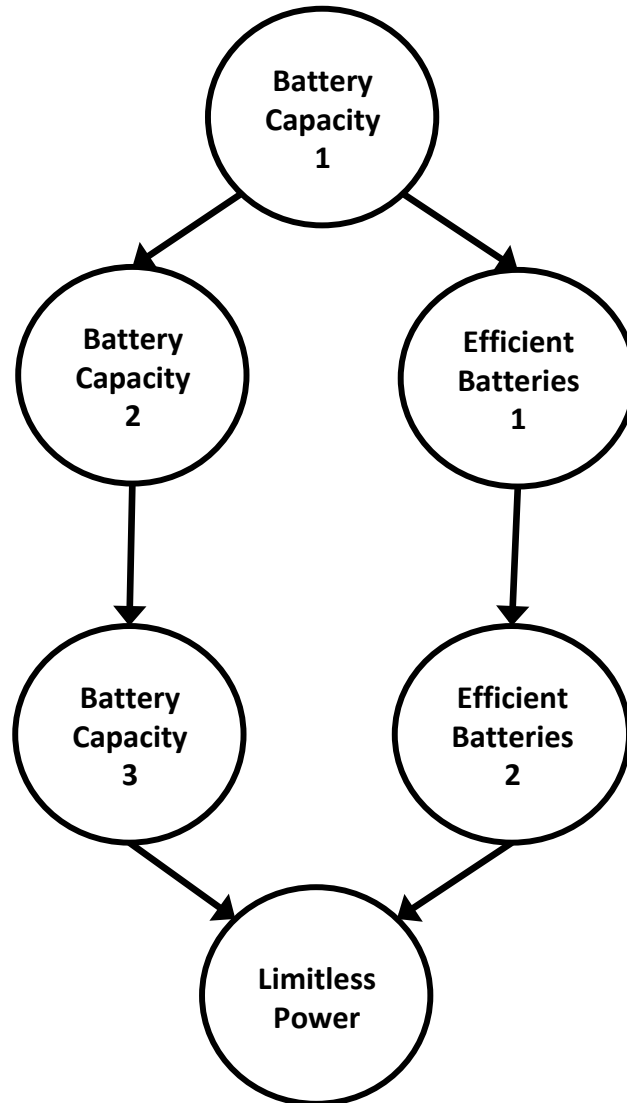


Figure 4.3-1. Flashlight Perk Tree

4.4 Pistol

The pistol perk tree consists of 6 nodes shown in Figure 4.4-1. *Pistol Damage 1*, *Pistol Damage 2*, and *Pistol Damage 3* increases the damage dealt by the pistol by 10% each. *Pistol Magazine 1*, *Pistol Magazine 2*, and *Pistol Magazine 3* each increase the magazine size of the pistol by 2.

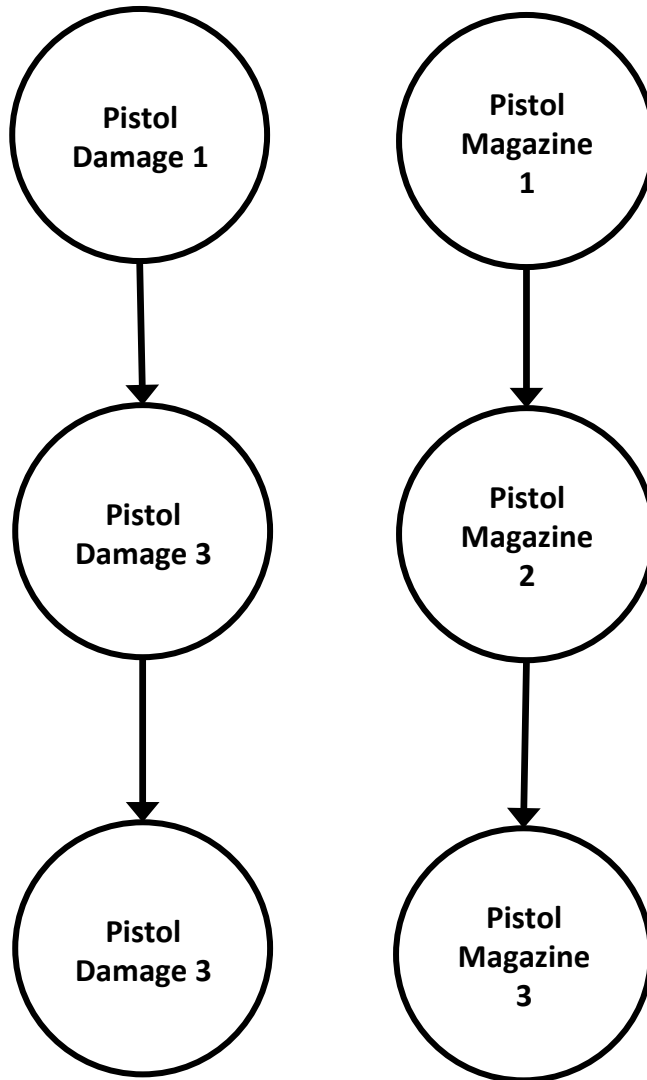


Figure 4.4-1. Pistol Perk Tree

4.5 Rifle

The rifle perk tree consists of 6 nodes shown in Figure 4.5-1. *Rifle Damage 1* and *Rifle Damage 2* increases the damage dealt by the rifle by 5% each. *Rifle Reload 1* and *Rifle Reload 2* decreases the time it requires to reload the rifle by 5% each. *Rifle Magazine 1* and *Rifle Magazine 2* each increase the magazine size of the pistol by 2.

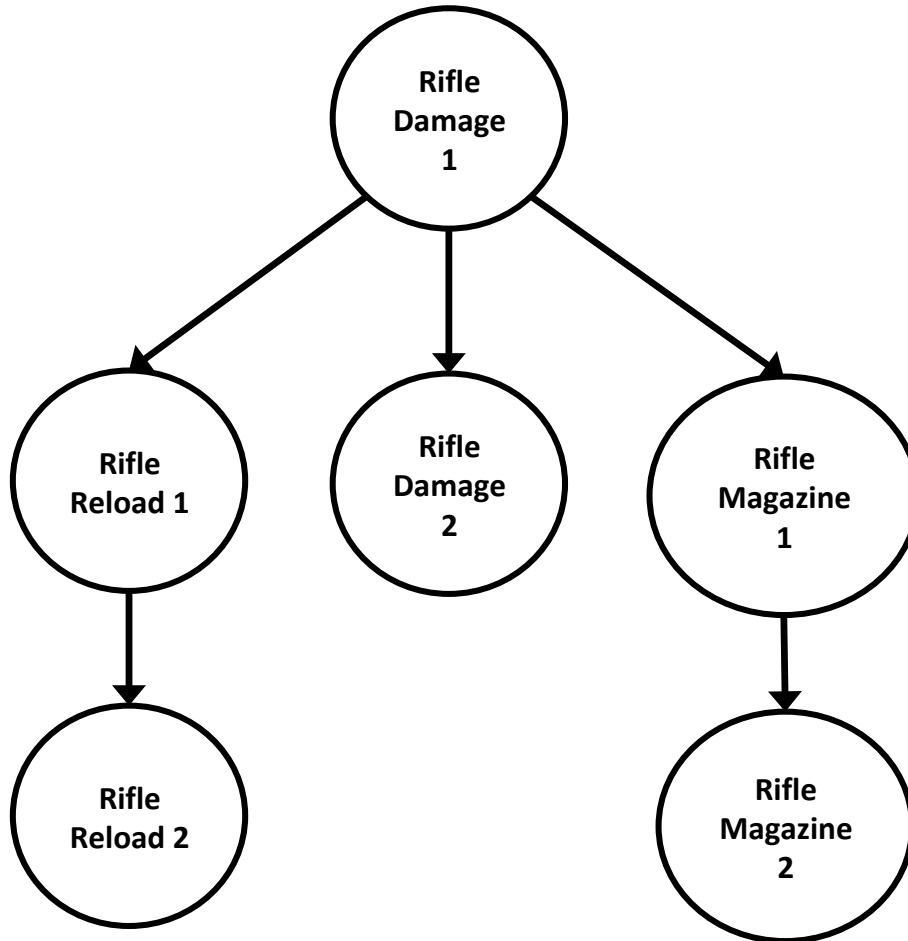


Figure 4.5-1. Rifle Perk Tree

5 Blend Spaces

5.1 Purpose

A blend space is used to create smooth animation transitions for characters when certain variables are constantly being changed or updated. In this application, all transitions are based on the character's speed and direction. The blend spaces are used alongside state machines that, given the character's direction and speed, will play a specific animation. Animations are added to discrete points on the blend space graph so the state machine knows which animation to use based on the variable input described above. If the variables are not on one of those specific points then the animation that is used will be a blend of nearby animations that creates a smoother transition.

5.2 General Blend Space

All character blend spaces follow the same variables, coordinate system, and general animation set up. The animations will be slightly different depending on what the character is currently doing and who the character is. For example, the main player character will have multiple blend spaces for each item they have equipped, as the animations will be different. The player will still have the basic animations in the same locations (walk, walk left, walk right, etc.), but the animations will reflect the item they have such as holding a rifle would use rifle walk, rifle walk left, rifle walk right, etc. in place of walk, walk left, walk right, etc.

The blend space is set up as shown below in Figure 5.2-1 and it is important to note that the top of the speed axis is the top speed of the character and is not 20. This value was used to show the points better. When the speed is 0, the character is not moving and all is idle. When they reach a speed of 10, the character is wholly using a walk animation either forward, right, left, or backwards given their direction. Finally, if the character is moving at full speed, they will be running in similar directions as they would if they were walking. If the character is at any point other than the ones shown discretely below, then the animation that is used will be a blend of surrounding animations points. The discrete points are outlined in Table 5.2-1.

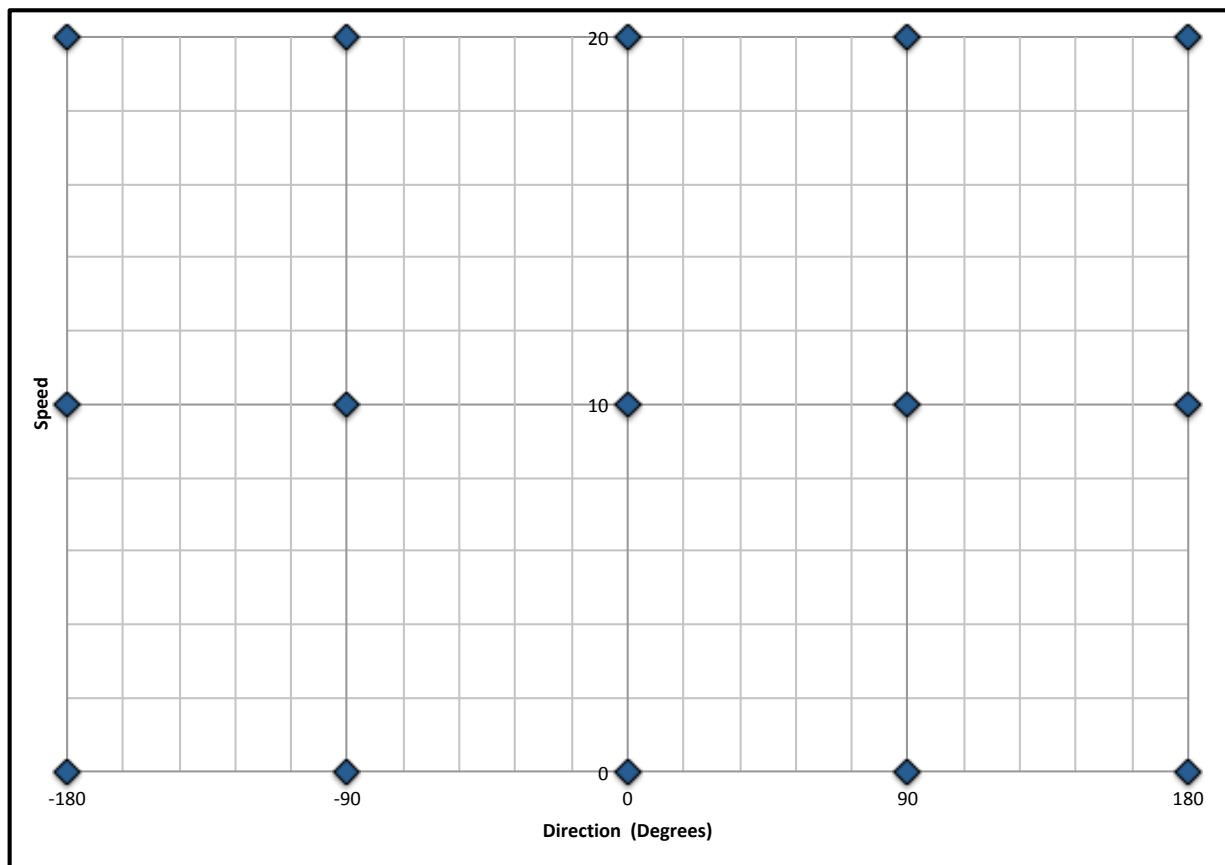


Figure 5.2-1. General Blend Space Graph

Table 5.2-1. General Blend Space Points

Direction	Speed	Animation
-180	0	Idle
-180	10	Walking Backwards
-180	20	Running Backwards
-90	0	Idle
-90	10	Walking Left
-90	20	Running Left
0	0	Idle
0	10	Walking Forwards
0	20	Running Forwards
90	0	Idle
90	10	Walking Right
90	20	Running Right
180	0	Idle
180	10	Walking Backwards
180	20	Running Backwards

6 State Machines

6.1 Purpose

The state machines described below allow the characters to move from one animation state to another. The state machines allow for better and smoother transitions created by blend spaces. This also allows separation and use of unique animation subsets such as normal movement blend spaces versus movement with a weapon blend spaces.

6.2 Main Character

The main player character has multiple sub state machines that make up one large machine to reflect the items that they are currently using. The main state machine, only showing the sub state machines and the transitions is shown in Figure 6.2-1. The transitions between these sub states are controlled by an equipped integer variable that updates when the player swaps between items or has no item equipped. It is important to note that all sub state machines can enter the *Die* state at any point and does not need to transfer to other states first. It is also very important to note that although not planned or intended, the main state machine is a pentagram; this is very fitting for the feel and mood of the game. Laudate satanico sint status machine.

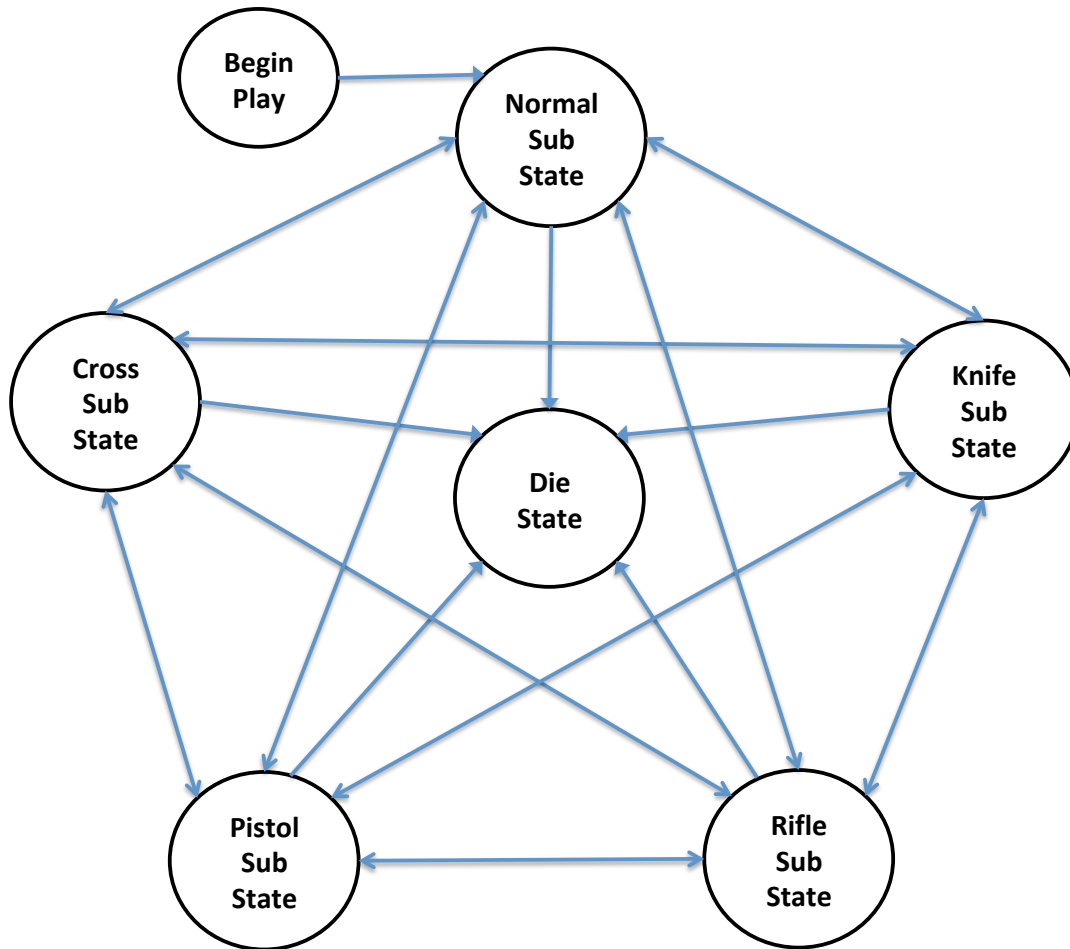


Figure 6.2-1. Main Character Main State Machine

When no item is equipped, the player uses the default Walk-Run sub state machine shown in Figure 6.2-2. This allows the game to display the correct animations while the user is walking, running, or idle.

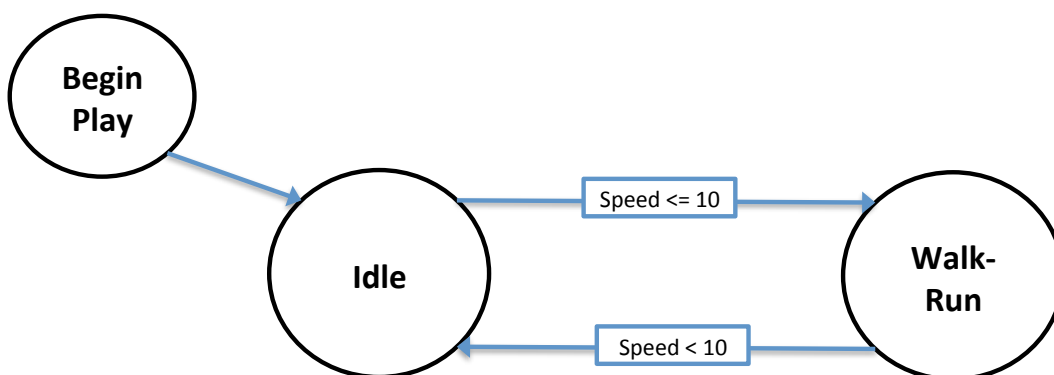


Figure 6.2-2. Main Character Walk-Run Sub State Machine

When the rifle is equipped, the player uses the rifle sub state machine shown in Figure 6.2-3. This allows the game to display the correct animations while the user

is walking, running, or idle with a rifle. The rifle also utilizes reload and fire states for additional animations. These two states do not have blend spaces associated with them as character speed and direction will not impact the animations. A very similar state machine is used for the pistol since the actions the player goes through are the same. The difference between the pistol sub state and the rifle sub state are the blend spaces and animations used. The guns can go directly from the idle or the walk-run states into the reload or fire states and vice versa back into the defaults when no action is taken place. The function finished conditions shown are to get back to idle and walk-run states.

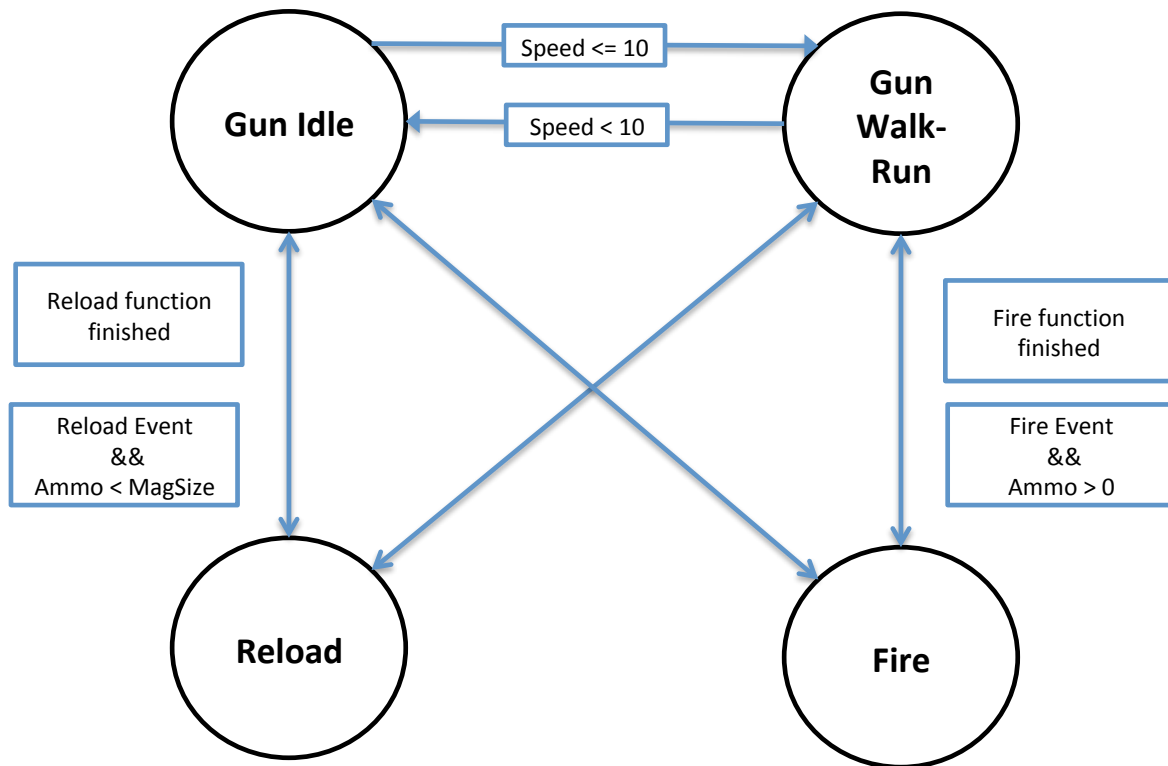


Figure 6.2-3. Main Character Rifle & Pistol Sub State Machines

When the cross is equipped, the player uses the cross sub state machine shown in Figure 6.2-4. This allows the game to display the correct animations while the user is walking, running, or idle with a cross. The cross also utilizes an attack state for an additional animation. This state does not have a blend space associated with it as character speed and direction will not impact the animation. A very similar state machine is used for the knife since the actions the player goes through are the same. The difference between the cross sub state and the knife sub state are the blend spaces and animations used. The attack state can be accessed from either idle or walk-run states and access is bidirectional.

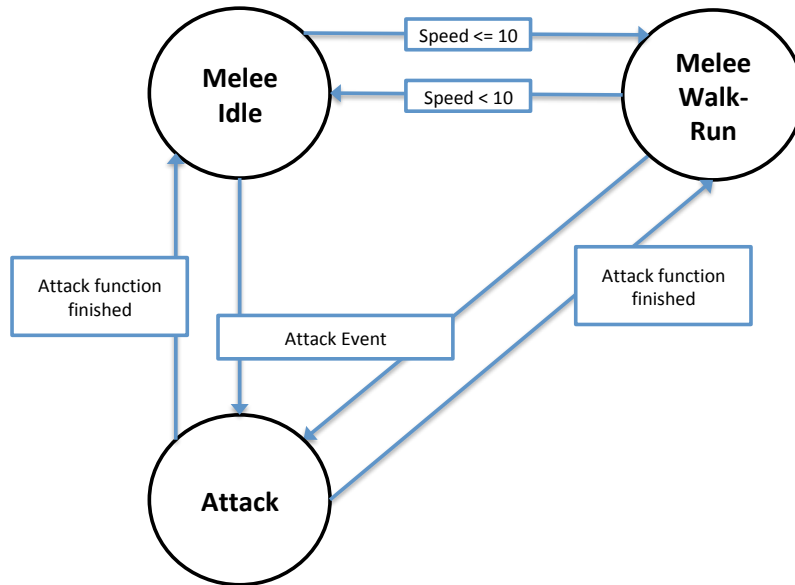


Figure 6.2-4. Main Character Cross & Knife Sub State Machines

6.3 Demon Characters

All demon characters follow the same general state machine. The demons will stand idle when finding a new location, walk when the location is found, move to the player when the player is close enough, begin to attack the player, or die if their health drops equal to or below 0. These five specific states are shown below in Figure 6.3-1.

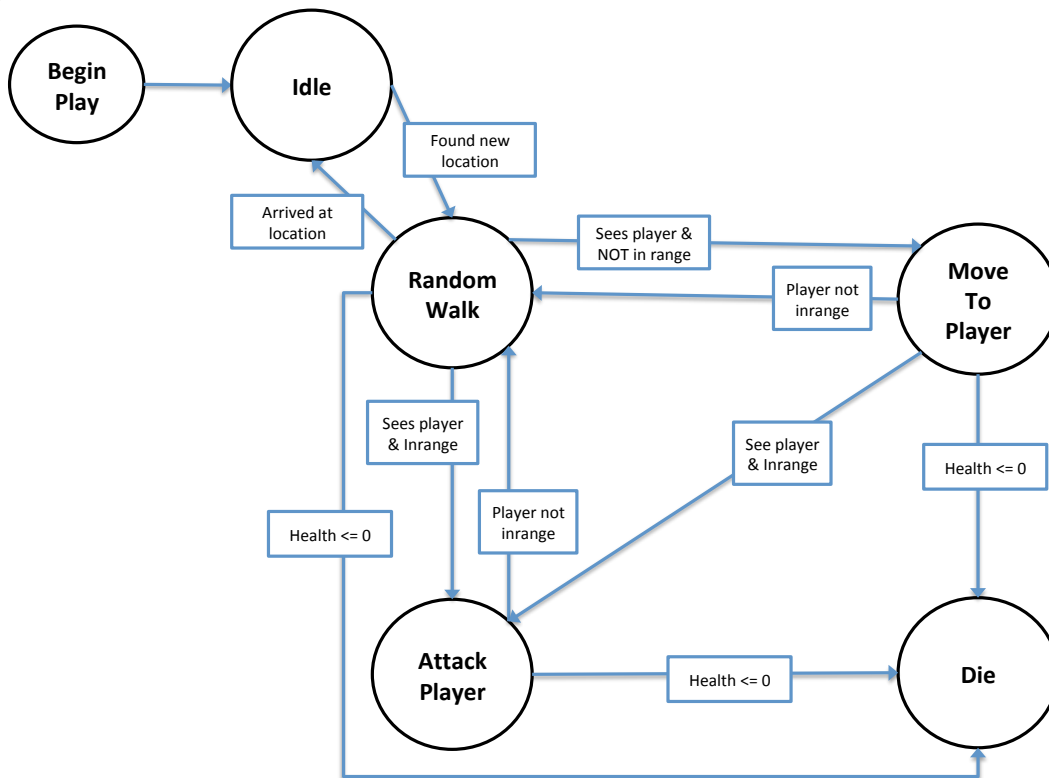


Figure 6.3-1. Demon State Machine

6.4 General Ghost

All general ghosts follow the same state machine and is only two states large. This is due to the ghosts not interacting directly with the player or with other characters. The two states are an idle state where the character will pick a new location to move to and a random walk state where the character will move to the random location. This will go on indefinitely.

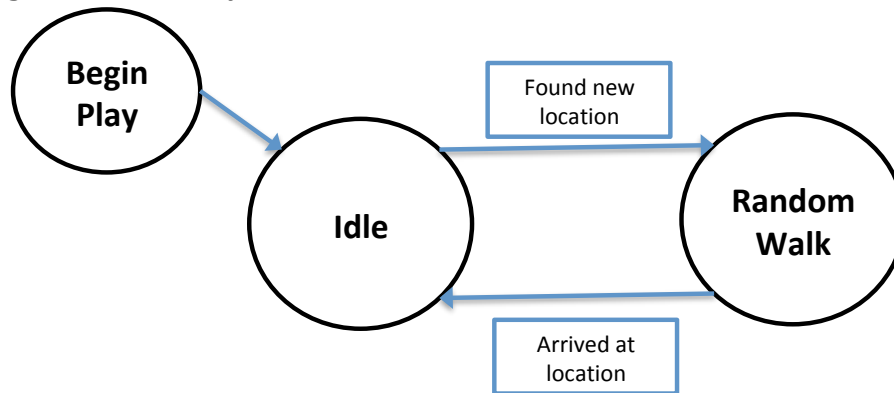


Figure 6.4-1. General Ghost State Machine

6.5 Main Ghost

The main ghost's state machine is only two states large. This is due to the ghost not interacting directly with the player or with other characters. The two states are an idle state where the character will pick a new location to move to based on predefined locations and a walk state where the character will move to that location. This will continue until the character has reached its final destination.

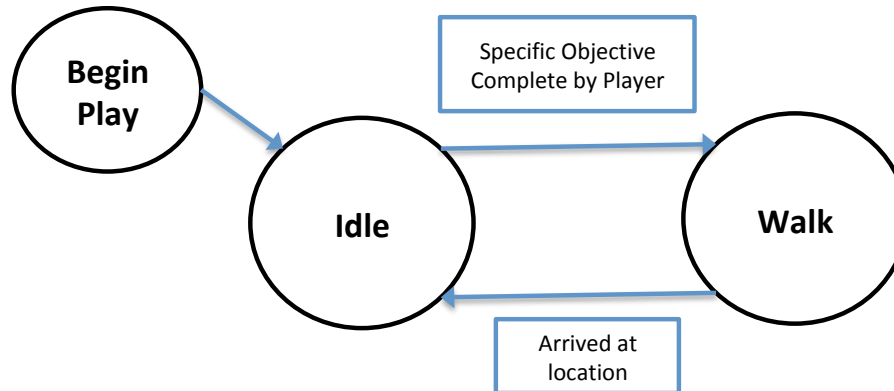


Figure 6.5-1. Main Ghost State Machine