

Night Flare



DraconicWolf Studios
Version 1.35c

Contributors:

Michael Wong, Ryan Hartanto,
Joshua Agerton, Christian Segumalian

Version: 1.35c

Table of Contents

| | |
|-----------------------------|---|
| Overview | 3 |
| Minimum Value Product | 4 |
| Core Mechanics..... | 4 |
| Core Interfaces | 7 |

Contributors:

Michael Wong, Ryan Hartanto,
Joshua Agerton, Christian Segumalian

Version: 1.35c

Overview

Night Flare will be a 2D top-down, wave-based survival game that places the player in an arena where they fight off waves of enemies and try to survive for as long as they can. Our game will incorporate basic player movement and combat functionality, as well as a basic wave system and enemy AI functionality.

Our team, or our Team Lead in this case, decided to base our game off Call of Duty: Zombies, in which the player is also placed into an arena-type map and must fight wave after wave of zombies, trying to survive to achieve a high score. To distinguish our game from other wave-based survival games and Call of Duty: Zombies, as a team, we avoided incorporating a hi-score system due to time constraints and agreed to incorporate an interface aspect derived from Darkwood.

Initially, our game was not meant to be a horror game, but we wanted to make our game unique by challenging the player to fight what they can't see immediately. In Darkwood, the player is limited by sight, where anything in their sight-range is dynamic and appears to change to the player, whereas everything else outside the player's sight-range is static and nothing can be seen. We force our player to be fast when it comes to making decisions on where to look to fight because where they look will be what changes onscreen, and those changes affect how the player will react.

One of the most distinguishing features about Night Flare is that our game has a defined beginning and end. Instead of adopting Call of Duty: Zombies' infinite wave-loop, our game will have a maximum number of waves that, should the player survive the last wave successfully, allows for completion of our game. As a team, we agreed on this not only because of the given time and resources available, but because it makes more sense for our player to be able to complete our game. We didn't want our player to be bored through the constant repetition of surviving difficult waves, and by integrating this core mechanic into our game, we believe the player will have an actual goal they play to achieve.

Contributors:

Michael Wong, Ryan Hartanto,
Joshua Agerton, Christian Segumalian

Version: 1.35c

Minimum Value Product

With the provided resources and time for our team, our goal for the Fall 2018 academic quarter is to develop, test, and incorporate the most important mechanics and interfaces necessary for our game to run. By doing this, we assure that our game will be playable, and while it may not be as perfect as we envisioned, it is a product we can display and have others play for feedback and enjoyment.

Core Mechanics

These are the mechanics that we need in our game above all other mechanics. These are necessary for actually playing our game and allowing others to playtest our game.

Player movement

The player will be able to freely-move in infinitely-many directions.

- the player can move around
 - the player can move in all 360-degree directions.
 - the player can walk or sprint.
 - walking is the player's default movement at a fixed speed.
 - coordinate-direction of movement is controlled by key binds, and free-direction of movement is controlled by mouse-aim (refer to Key Binding).
 - running is the player's limited-utility movement at a greater speed than walking.
 - limited by an adrenaline system; sprinting drains the player's adrenaline (refer to adrenaline below).
 - coordinate-direction of movement is controlled by key binds, and free-direction of movement is controlled by mouse-aim (refer to Key Binding).

Player combat

The player can attack and retaliate against enemies and is susceptible to death.

- the player can attack enemies
 - the player's default attack is a ranged, projectile-based attack.
 - projectile factors:

Contributors:

Michael Wong, Ryan Hartanto,
Joshua Agerton, Christian Segumalian

Version: 1.35c

- velocity
 - projectile size
 - pellet count
 - spread
- the player has infinite ammo.
 - however, the player has limited clip size in their weapon; they must take a fixed time to reload before attacking again.
- there is a cooldown-period in-between the player's attacks.
- the player does not require complete animations or charge time to attack.
- the player can only attack in the direction they are facing.
- projectiles damage enemies by a fixed amount per pellet.
- the player can attack either by repeatedly pressing or holding the attack key bind (refer to Key Binding).
 - the cooldown-period in-between attacks remains the same for both cases.
- the player has health
 - total health decreases upon taking damage from enemies.
 - if the player's total health falls below 75% and they haven't taken damage for a fixed time, they start regenerating health at a rate of 1 health/ 0.05 seconds.
 - if the player's health drops to 0, they die.
- the player has adrenaline
 - the player uses adrenaline to sprint.
 - sprinting drains adrenaline at a rate of 10 adrenaline/ second.
 - if the player's adrenaline drops below 100%, they will regenerate 1 adrenaline/ 0.05 seconds after a fixed time.

Enemy combat

Enemies can attack and retaliate against the player and are susceptible to death.

- enemies can attack the player.
 - two forms for enemies: melee-form and ranged-form:
 - melee-form enemies primarily use close-ranged attacks.
 - there is a cooldown-period in-between enemy attacks.

Contributors:

Michael Wong, Ryan Hartanto,
Joshua Agerton, Christian Segumalian

Version: 1.35c

- successful attacks require undergoing a complete animation and landing on the player following the last frame of animation.
- melee-form enemies must attack in the direction of the player.
- successful attacks damage the player by a fixed amount.
- ranged-form enemies primarily use projectile-based, mid-long-ranged attacks.
 - projectile factors:
 - velocity
 - projectile size
 - pellet count
 - spread
 - there as a cooldown-period in-between enemy attacks.
 - ranged-form enemies require no charge time nor complete animations to attack.
 - enemies can only fire in the direction of the player.
 - projectiles damage the player by a fixed amount per pellet.
- enemies have health
 - total health decreases upon taking damage from the player.
 - if an enemy's health drops to 0, they die.

Enemy movement

Enemies can freely-move in infinitely-many directions.

- enemies can move infinitely-many directions.
 - limited by the fact enemies are always going to chase the player if they are melee-form or target them if they are ranged-form.
 - if the enemy is melee-form, they will try to close in on the player to attack.
 - if the enemy is ranged-form, they will only stop within a certain distance from the player to attack.

Wave system

The player must survive wave after wave of progressively-difficult enemies and do their best to survive.

Contributors:

Michael Wong, Ryan Hartanto,
Joshua Agerton, Christian Segumalian

Version: 1.35c

- the player starts at Wave 1 whenever they start a round.
- enemies become stronger with every wave survived.
 - enemies gain the following stat boosts with each new wave:
 - damage increases
 - health increases
- when a wave ends, the player will be given two options:
 - a room opens for the player to enter, which will take them to a break room where they can catch their breath for 60 seconds.
 - when the break ends, they will be forced back into the arena and are given 5 seconds to prepare themselves for the next wave.
 - the player, given the option to take a break, declines the offer, and is given 5 seconds to prepare themselves for the next wave.
- if the player dies during a wave, they will lose that wave.
 - the wave counter does not reset upon death.
 - the player has two choices:
 - the player can retry the wave again from where they left off.
 - the player can return to the Main Menu screen (refer to Main Menu scene).

The End-Game

Our game has an ending, and the player can complete our game, but they must play to reach it.

- the player must survive 5 waves to complete our game.
 - upon successfully surviving Wave 5, the player has completed the game.
 - after completing the game, the player is given two options:
 - replay the game again, starting back at Wave 1.
 - return to the Main Menu screen (refer to Main Menu scene).

Core Interfaces

These are the user interface aspects that must be incorporated in our game for navigation, display, and transition around our game. These interface aspects are **NOT** necessary for playtesting our game. The location of each scene's items will not be mentioned here; they will be displayed in a separate, illustrative documents.

Main Menu scene

Contributors:

Michael Wong, Ryan Hartanto,
Joshua Agerton, Christian Segumalian

Version: 1.35c

This will be the first screen that appears to the player upon starting-up our game. From here, the player can start the game, adjust in-game settings, access our game's instructions, and exit out of the game.

- Start Game control
 - starts the game and immediately transports the player to the Active Level scene.
- Instructions control
 - opens a panel on the Main Menu scene that shows the instructions on how to play our game.
 - all other textual information disappears.
- Options control
 - opens a panel on the Main Menu scene that shows the adjustable game options for the game.
 - all other textual information disappears.
- Credits control
 - displays all the contributors of the game on the Main Menu scene.
 - all other textual information disappears.
- Quit control
 - exits out of the game.

Contributors:

Michael Wong, Ryan Hartanto,
Joshua Agerton, Christian Segumalian

Version: 1.35c