

C++ CW2

Jirui Zhang

May 2025

Contents

1	Introduction	3
2	Implementation	3
2.1	Start State (Main Menu)	3
2.1.1	Implemented Function	3
2.1.2	Meet requirements	3
2.2	Intro State (opening cinematic)	4
2.2.1	Implemented Function	4
2.2.2	Meet requirements	4
2.2.3	Background Story	4
2.3	Name State (enter Name)	5
2.3.1	Implemented Function	5
2.3.2	Meet requirements	5
2.4	Loading State(Loading)	5
2.4.1	Implemented Function	5
2.4.2	Meet requirements	6
3	Running State	6
3.1	Background	6
3.1.1	Implemented Function	6
3.1.2	Meet requirements	6
3.2	Save and Load	6
3.2.1	Implemented Function	6
3.2.2	Meet requirements	6
3.3	Displayable objects	7
3.3.1	Player	7
3.3.2	WereWolf	8
3.3.3	AI Logic	9
3.3.4	Knight	9
3.3.5	AI Logic	10
3.3.6	Pet Dog	11
3.3.7	AI Logic	12
3.3.8	Meet the requirements	13

3.4	Tiles	13
3.4.1	ornament	13
3.4.2	Ground	13
3.4.3	Animated and Interactive tiles	14
3.4.4	Meet the Requirements	14
4	Other States:	15
4.1	Pause State	15
4.1.1	Implemented Function	15
4.1.2	Meet requirements	16
4.2	Lose State	16
4.2.1	Implemented Function	17
4.2.2	Meet requirements	17
4.3	Win State (Ending)	17
4.3.1	Implemented Function	18
4.3.2	Meet requirements	18
4.3.3	Endings:	18
5	Code	18

1 Introduction

Skeleton is a C++ real-time action game where the player controls a skeleton hero in three distinct classes: Normal, Archer, and Warrior. Each class offers unique abilities:- Normal Skeleton: balanced speed and melee attack.- Archer Skeleton: ranged bone projectiles with moderate speed.- Warrior Skeleton: slower speed but heavy melee damage and guard ability. The core gameplay loop involves navigating a procedurally generated cemetery level, surviving patrolling werewolves, and defeating the boss knight. Interactive elements include fire hazards that damage the player, water pools that slow movement, and a wizard tile which, upon contact, summons rain to extinguish fires. Players can pause, restart, save and load their progress at any time.

2 Implementation

2.1 Start State (Main Menu)



Figure 1: Start State

2.1.1 Implemented Function

- Click New Game: Start a Game
- Click exit: Quit the Game

2.1.2 Meet requirements

- Handling of program states.

2.2 Intro State (opening cinematic)



Figure 2: Intro State

2.2.1 Implemented Function

- Animated Background.
- Press 'Space' to skip

2.2.2 Meet requirements

- Handling of program states.
- Animated Background

2.2.3 Background Story

Once, he was a celebrated knight of the realm—brave, strong, and true. In a fierce battle on the king's frontier, he fell, his life claimed by the cruel hand of Death. Impressed by the knight's valor and power, the Lord of Shadows refused to let such a warrior slip away forever.

Yet Death had heard tales of another: the king's own champion, reputed to be the finest knight in all the lands. Curiosity and rivalry stirred within Death's dark heart, and so he wove his most forbidden necromancy. He raised the fallen hero from his grave, but in skeletal form—stripped of flesh, yet bound by unbreakable will.

Death spoke his command: “Prove your worth beyond any mortal challenge. Seek out the king's invincible knight and strike him down. Only by overcoming this champion may you earn true life again.” And so the Skeleton Knight set forth, steel in hand and a single, grim purpose: to defeat the greatest knight of the realm or remain forever a servant of Death.

2.3 Name State (enter Name)



Figure 3: Name State

2.3.1 Implemented Function

- Press the keyboard to type.
- Press the 'Delete key' to delete.
- Press enter to confirm.
- the name will be saved as the main character's name

2.3.2 Meet requirements

- Handling of program states.
- Enter Text.

2.4 Loading State(Loading)

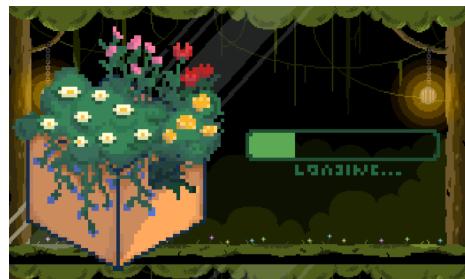


Figure 4: Loading State

2.4.1 Implemented Function

- Animated Background by changing the surface.



Figure 5: Running State

2.4.2 Meet requirements

- Have an animated or changing background by utilising multiple images.
- Handling of program states.

3 Running State

3.1 Background

3.1.1 Implemented Function

- Scrolling background.
- Press 'q', 'e' to scroll the background (Have the boundary)
- Press 'x' to Zoom in and 'z' to Zoom out (must Zoom in First)

3.1.2 Meet requirements

- Implement a scrolling background by manipulating the way that the background image is drawn.
- Correctly implement scrolling and zooming of the foreground, allowing the user to scroll around using keys and/or mouse.

3.2 Save and Load

3.2.1 Implemented Function

- Press 'f5' to save
- Press 'f9' to load (must save before load, and cannot save when fighting with the knight(boss))

3.2.2 Meet requirements

- Input/output features

3.3 Displayable objects

3.3.1 Player



Figure 6: Normal skeleton



Figure 7: Warrior skeleton



Figure 8: Archer skeleton

Implemented Function

- 'w': jump
- 's': crouch
- 'a'/'d': walk
- 'p': Pause
- shift + 'a'/'d': run
- Click the left button: Attack 1
- Click the right button: Attack 2
- '1','2','3': Change State

- health: 3 (Figure 9)

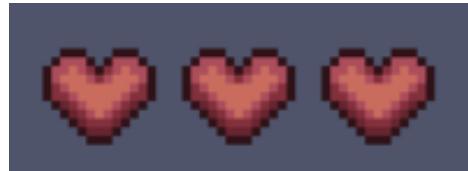


Figure 9: Health

3.3.2 WereWolf



Figure 10: Human



Figure 11: Wolf

Basic Information

- Human Health: 1
- Wolf Health: 5
- Attribute: Hostile

3.3.3 AI Logic

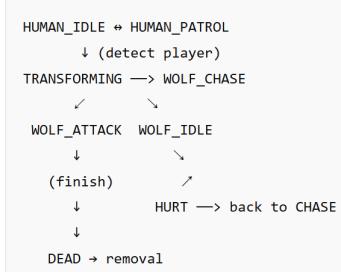


Figure 12: Behavior Tree

Story Born under a blood-red moon in a remote mountain village, he was once a skilled hunter and protector of his people. But one fateful night, he tracked a wounded wolf into an ancient ruin and discovered a hidden altar of dark magic. Desperate to save his dying father, he drank from a chalice of cursed water.

At dawn he awoke changed: in human form he felt sharper senses and boundless strength—but when the moon rose, his humanity slipped away. Fur sprouted, claws tore free, and the beast within took over. Unable to control his transformations, he fled into the wild, torn between guilt for the lives he’d claimed and his desperate need to find a cure.

Now known as the Midnight Fang, he prowls the land, hunting those who traffic in forbidden sorcery. His rage is tempered by the memory of the man he once was—and he carries one hope: that by slaying the true master of the curse he might break his own chains and reclaim his lost soul.

3.3.4 Knight



Figure 13: Knight

Basic Information

- Health: 5
- Attribute: Neutral

3.3.5 AI Logic



Figure 14: BehaviorTree

Story Sir Alaric of Valebrook was raised in the royal court, the son of a humble blacksmith who dreamed his child would one day serve the crown. From boyhood he trained relentlessly in swordplay and tactics under the kingdom's finest masters. His honor and skill earned him a place among the king's personal guard, then promotion to the title of Champion Knight—protector of the realm and hero of countless battles.

When the neighboring duchy rebelled, Alaric rode at the head of the royal host. His blade sang through the enemy ranks, turning back invaders who threatened to burn the capital. Songs were written of his valor; children claimed he bore the strength of ten men. Yet behind his steel mask lay a gentle heart: he tended wounded allies, spared defeated foes, and upheld mercy even in the heat of war.

Now Alaric stands alone in the ruined courtyard of Castle Valebrook, awaiting the Skeleton Knight sent by Death itself. Though his armor is battered and his banner torn, his resolve never wavers. He knows that if he falls, the kingdom will lose its greatest shield—and that only by facing this undead champion can he prove once more that true honor survives beyond the grave.

3.3.6 Pet Dog



Figure 15: Dog

3.3.7 AI Logic

```

Selector (updateAI)
|
└─ [playerIsIdle > 3s]
    └─ SIT
        └─ (if SIT > timeout) → SLEEP
|
└─ [playerIsDead]
    └─ SIT_BESIDE_PLAYER
|
└─ [fireAhead]
    └─ WAIT_FIRE (SIT)
|
└─ [assisting]
    |
    └─ [findEnemyAhead(10)]
        └─ ATTACK
            └─ CATCHUP
    |
    └─ [findEnemyAhead(>10)]
        └─ CHASE
|
└─ [default]
    ↓
    PredictPlayerX → followBehind
        └─ [absDist < NEAR] → IDLE
        └─ [absDist ≥ NEAR]
            └─ if !m_returning → FOLLOW
            └─ if m_returning → CATCHUP

```

Figure 16: Enter Caption

Story Fenrir was once a humble farm dog, loyal companion to a wandering tinker whose caravan rolled through forgotten hamlets. By day he chased foxes in golden fields; by night he slept at the forge, warmed by the glow of hot iron and soot-smudged hands that scratched behind his ears.

One stormy evening, the tinker’s cart broke an axle deep in an ancient graveyard. Lightning cracked the sky and a necromantic wind stirred headstones. Fenrir sensed something wrong—shadows moved without source, and distant howls carried in on the wind. He dashed into the tombs and discovered a restless spirit bound by chains of darkness. Barking furiously, he tore away shroud and shackles, freeing the lost soul.

In gratitude, the ghost whispered a sacred vow: “Because you showed com-

passion to one condemned, you shall guard the living against the dead.” When Fenrir returned, he found his master fallen—poisoned by cursed whispers from the grave. With a bark that shook the stones, Fenrir swore to protect the tinker’s legacy.

Now known as the Shadow Hound, Fenrir follows the Skeleton Knight through haunted ruins and rain-soaked cemeteries. He sniffs out lurking monsters, warns of hidden traps with a low growl, and leaps into battle when darkness descends—because no master deserves to face Death alone.

3.3.8 Meet the requirements

- Displayable object features
- Animate moving objects
- Collision detection

3.4 Tiles

3.4.1 ornament

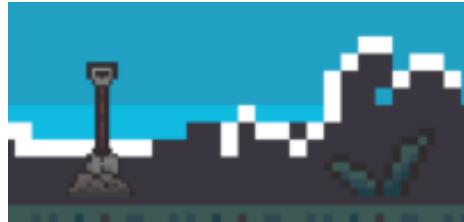


Figure 17: ornament

Implemented Function

- Randomly Generated.
- No Collision.

3.4.2 Ground



Figure 18: Ground

Implemented Function

- Fixed Generated.
- Solid Box Collision.

3.4.3 Animated and Interactive tiles



Figure 19: Magician and Fire

Implemented Function

- Randomly Generated.
- Rain.
- Pixel Collision.
- Hurt the character

3.4.4 Meet the Requirements

- Tile manager usage.

4 Other States:

4.1 Pause State



Figure 20: Day



Figure 21: Morning



Figure 22: Night

4.1.1 Implemented Function

- Different images based on different system times.
- Useful Hints changes in the interval time
- Press 'P' to return to the Running State.

4.1.2 Meet requirements

- Handling of program states.

4.2 Lose State



Figure 23: Knight Death



Figure 24: Wolf Death



Figure 25: Fire Death

4.2.1 Implemented Function

- Different images based on the death reasons.
- Press 'space' to return to the Start State.

4.2.2 Meet requirements

- Handling of program states.

4.3 Win State (Ending)



Figure 26: Win State

4.3.1 Implemented Function

- Animated endings.
- Press 'space' to return to the Start State.

4.3.2 Meet requirements

- Handling of program states.

4.3.3 Endings:

As the final blow struck true, the Champion Knight's proud banner fell limp, and the cursed chains that bound the Skeleton Knight unraveled. Flesh and blood surged into his hollow bones, warmth flooding his veins—he was alive again at last. Victory and relief washed over him as he stood triumphant, sword dripping with the foe's ancient blood.

But across the shattered courtyard, his heart seized. The motionless armor of the Champion Knight stirred. A pale glow seeped from shattered helm and greaves. Bone cracked and knitted itself back together; sinew and ragged cloth reformed until, where the valiant knight had fallen, now rose a skeletal warrior with empty eye sockets blazing like embers. The resurrected Champion—undead and unbowed—raised his blade in silent challenge.

In that dreadful moment, the newly restored hero understood Death's cruel joke: life reclaimed, only to watch his greatest adversary rise as a fellow servant of the grave. Steel met steel once more under the moon's cold gaze, and the true battle for their souls

5 Code

Detailed Explanation Could be Found in Demo Video and in C++ Codes.