**Final Project Documentation.**

Santa Wants his presents back!

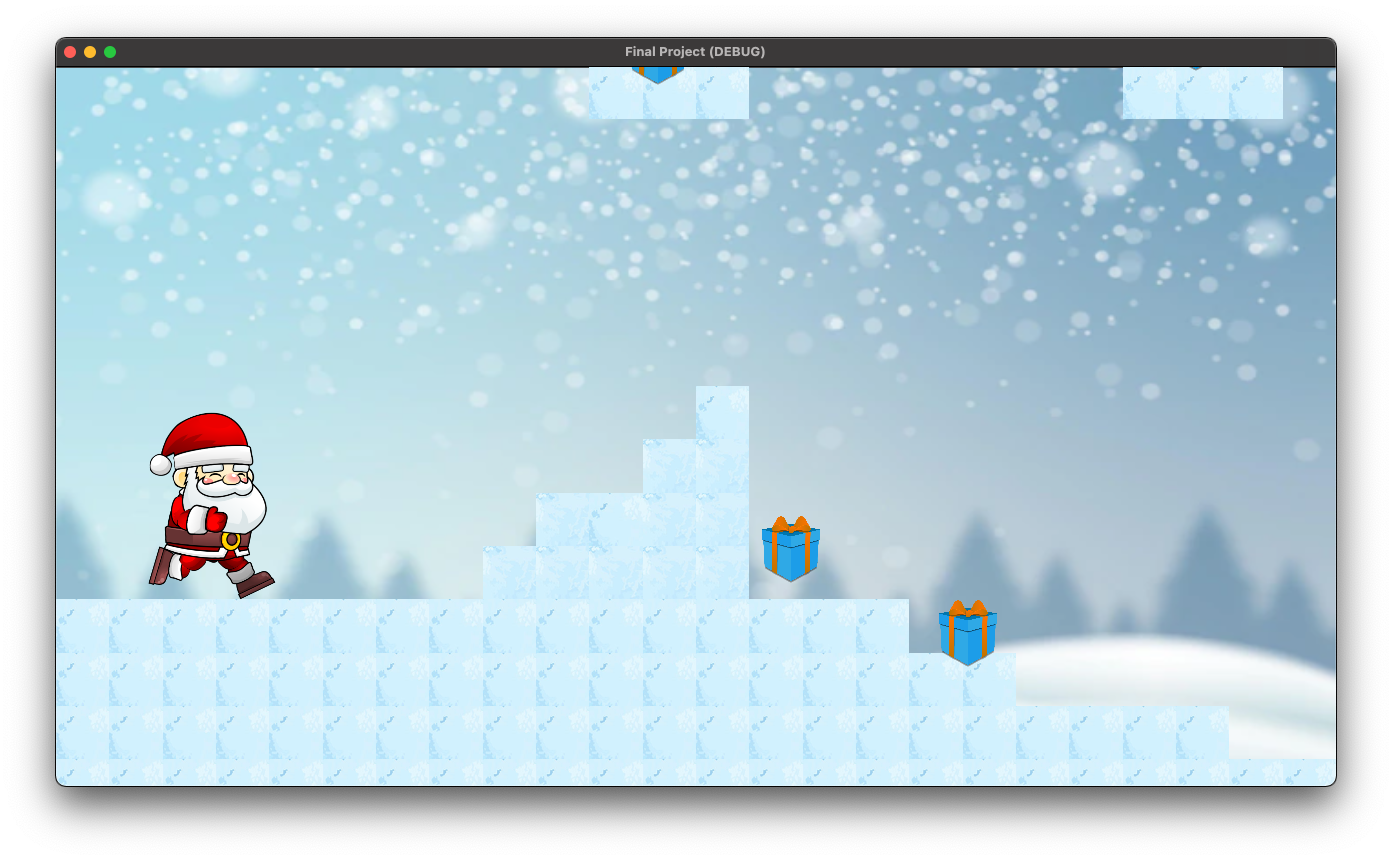


Figure : Game Demo

**Goal:**

Creating a demonstration based on acquired knowledge from this class to build a final project. The project was the result of numerous lessons learned from previous projects and lectures.

**Game Description:**

The following game uses the Godot game engine. The game engine has the tools needed for managing the background, timing, sprites, motion, collision detection, boundary detection and so much more. The game depicts Santa Claus in the north pole. Santa must go through an obstacle course to take back all his presents. The code is divided by sprite and gifts. For the icy surface, the “tileMap” tool was used to place blocks on top of the scene. The tiles also have collision detection enabled. The sprites collision detection shape is a somewhat square shape. Instead of just surrounding the parameter of Santa’s entire body, I went with this shape because it makes jumping easier and makes the sprite less prone to becoming stuck. The sprite collision can be seen in figure 2. Santa can move thanks to the player node being a Kinematic body.

extends KinematicBody2D

class\_name Actor

const FLOOR\_NORMAL: = Vector2.UP

export var speed: = Vector2(400.0, 500.0)

export var gravity: = 3500.0

var \_velocity: = Vector2.ZERO

func \_physics\_process(delta: float) -> void:

\_velocity.y += gravity \* delta

A picture containing text, monitor, indoor, electronics

Description automatically generated

Figure : Sprite Collision and Godot Window.

extends Actor

export var stomp\_impulse: = 600.0

func \_physics\_process(delta: float) -> void:

var is\_jump\_interrupted: = Input.is\_action\_just\_released("jump") and \_velocity.y < 0.0

var direction: = get\_direction()

\_velocity = calculate\_move\_velocity(\_velocity, direction, speed, is\_jump\_interrupted)

\_velocity.y = move\_and\_slide(\_velocity, FLOOR\_NORMAL).y

func get\_direction() -> Vector2:

return Vector2(

Input.get\_action\_strength("move\_right") - Input.get\_action\_strength("move\_left"),

-Input.get\_action\_strength("jump") if is\_on\_floor() and Input.is\_action\_just\_pressed("jump") else 0.0

)

func calculate\_move\_velocity(

linear\_velocity: Vector2,

direction: Vector2,

speed: Vector2,

is\_jump\_interrupted: bool

) -> Vector2:

var out: = linear\_velocity

out.x = speed.x \* direction.x

if direction.y != 0.0:

out.y = speed.y \* direction.y

if is\_jump\_interrupted:

out.y = 0.0

return out

func calculate\_stomp\_velocity(linear\_velocity: Vector2, stomp\_impulse: float) -> Vector2:

var stomp\_jump: = -speed.y if Input.is\_action\_pressed("jump") else -stomp\_impulse

return Vector2(linear\_velocity.x, stomp\_jump)

The player.gd script, as seen above controls movements such as velocity and direction. The Coin.gd script describes the process of picking up a coin and contains an accumulator to count score of the player as seen below.

extends Area2D

var score = 1;

onready var anim\_player: AnimationPlayer = $AnimationPlayer

func \_on\_body\_entered(body: PhysicsBody2D) -> void:

picked()

func picked() -> void:

anim\_player.play("picked")

score += 1

print(score)

**State Diagram:**

SCORE += 1

GIFT PICKED ()

**User Instructions:**

Please download the Godot game engine. Then simply download the files from GitHub here <https://github.com/esaini1/2D-FINAL-PROJECT.git> and run the “project.godot” file to start the game.

**Lessons Learned:**

Thanks to the knowledge that was accumulated through the many assignments in this class, I was able to demonstrate my acquired knowledge from this class by building this final project. The project was the result of numerous lessons learned from previous projects and lectures. The process required learning multiple new languages and facing plenty of bugs and frustration along the way. Godot really helped me gain an appreciation of the complexity of game engines available today. The game engine we built feel relatively minuscule compared to the sheer power of Godot and other platform like it.