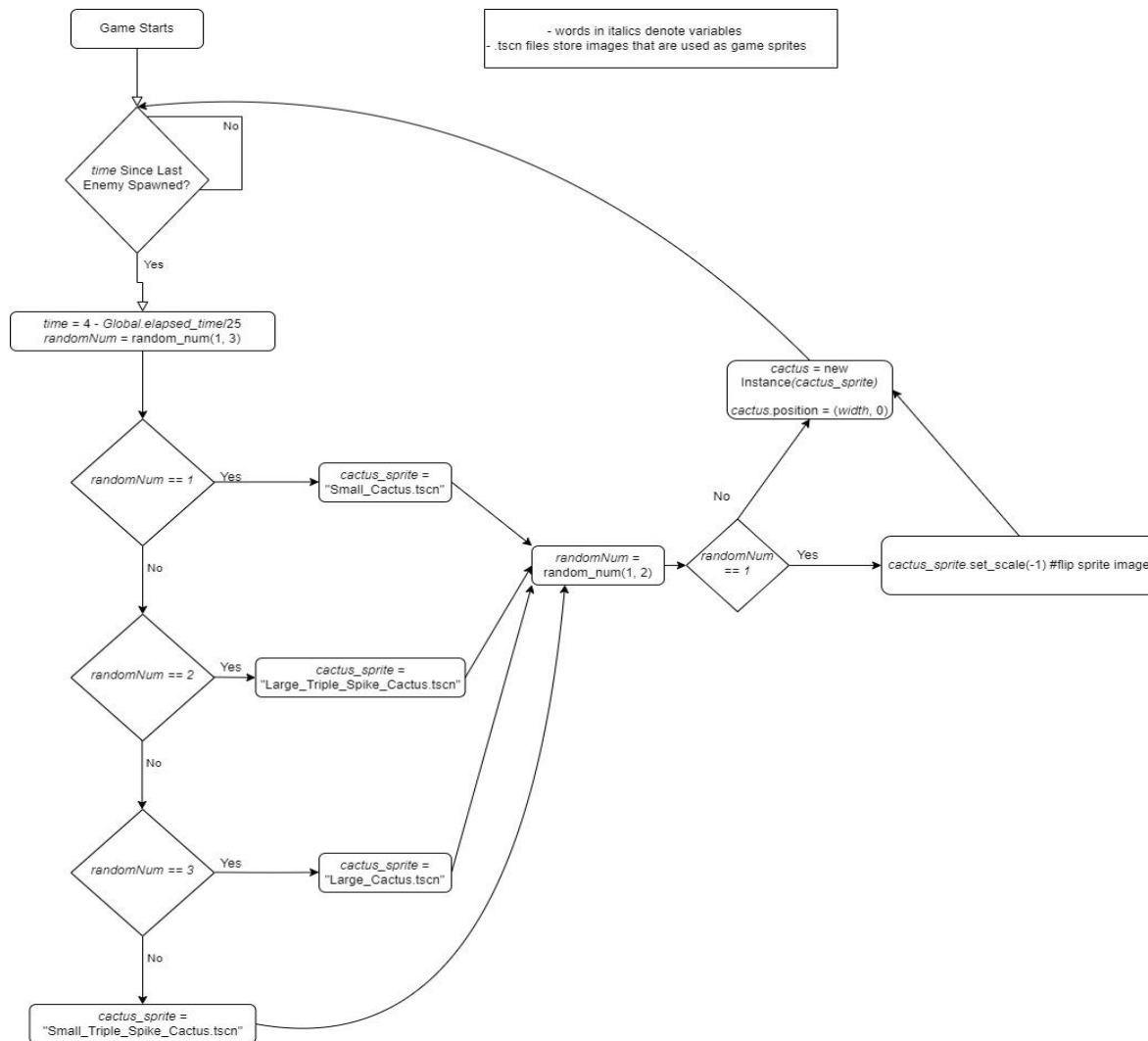


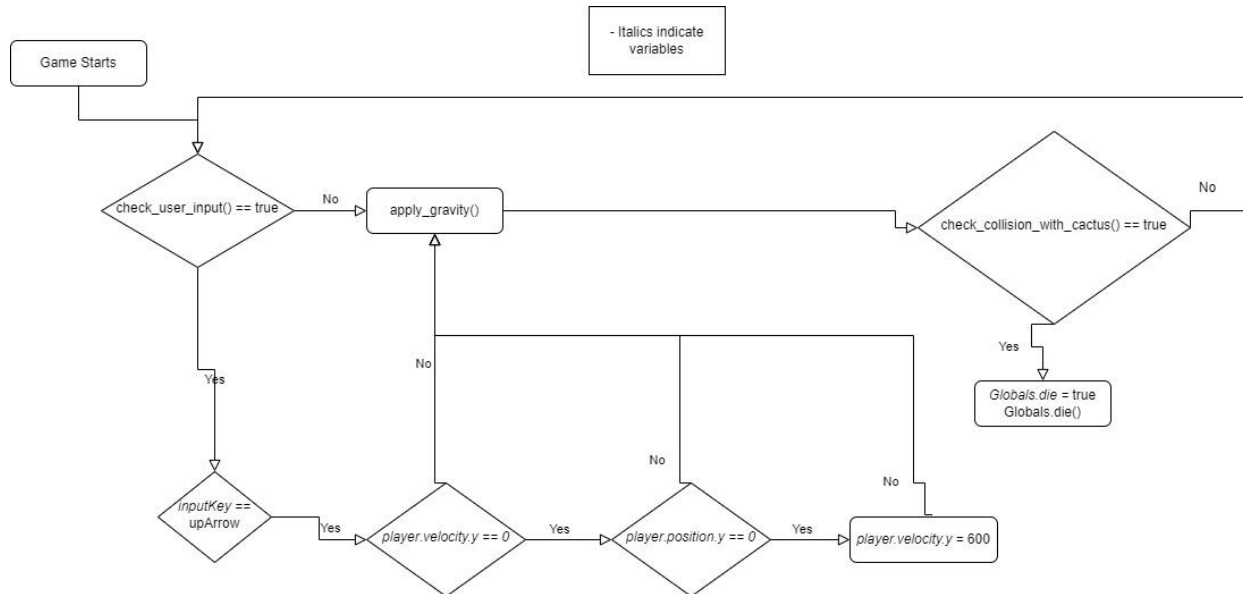
Criterion B: Design

I decided to use Godot's recommended strategy of designing games broken into classes. This structure would have one main class (which I called Ground) that would have every other class (Player, Enemy, etc) be a child of it. Every text label would be its own class and display on screen if a game condition were met (ex: instruction label appearing before the game started). There would be two modules every class could use (AudioStreamPlayer and Global), which contain methods and variables every class could access (ex: the score and the time since the game started). The game loop would be run in the background by the engine. The UML Diagram can be found in Appendix E.

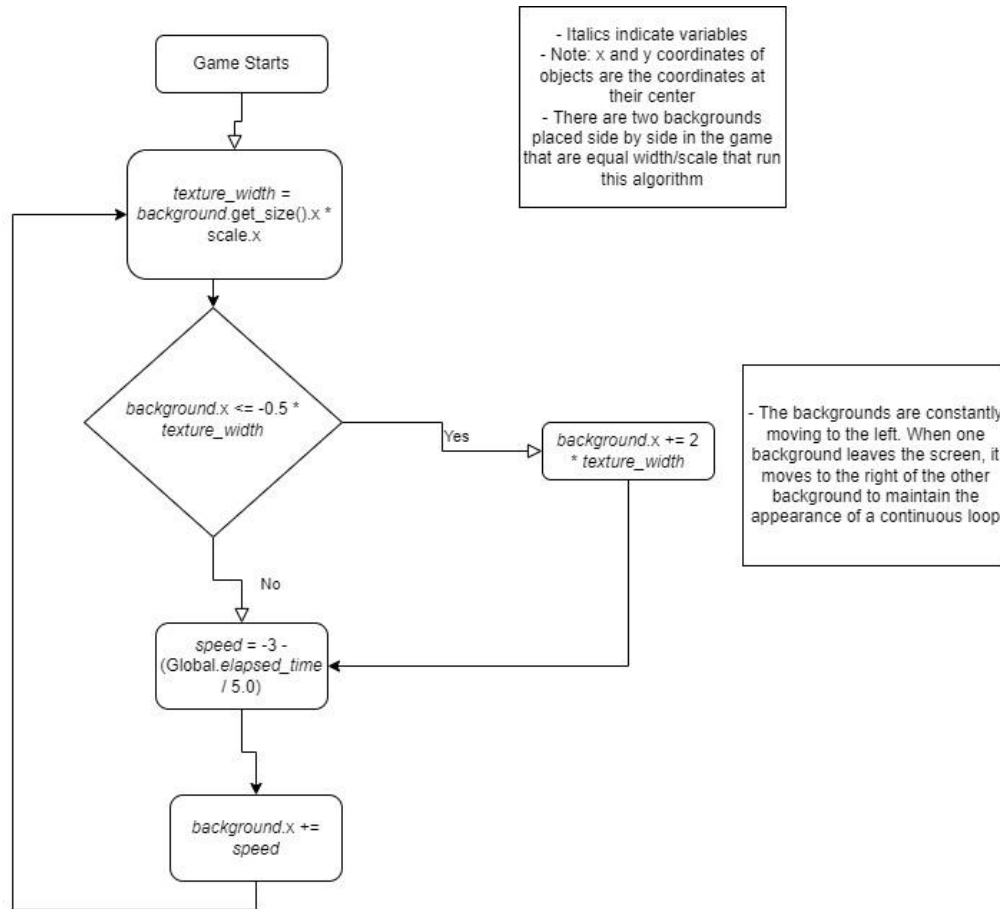
One of the most important algorithms determines when to spawn new cacti and what sprite it should use.



Another one of the main algorithms determines player movement and whether or not a collision has occurred with an enemy cactus.



The final major algorithm I used was to make the background appear as if it was continuously looping and increasing in speed as the game progressed.



Feature to Test	Test Strategy
<ul style="list-style-type: none"> Scoring System 	<ul style="list-style-type: none"> Verify that the score doesn't change on the menu screen or when the game is paused Ensure the score goes up once per second with a timer
<ul style="list-style-type: none"> Audio 	<ul style="list-style-type: none"> Run game and verify: <ul style="list-style-type: none"> Startup sound plays on startup Jump sound plays when the player jumps Death sound plays when the player loses
<ul style="list-style-type: none"> Player Losing the Game 	<ul style="list-style-type: none"> Run game and ensure that the player dies only when the dinosaur hits a cactus
<ul style="list-style-type: none"> Game Available Online 	<ul style="list-style-type: none"> Go to the the website where the game

	<p>is hosted (https://anonymoustank.github.io/DinosaurGame/index.html) on a different computer and ensure it runs on the major browsers (Firefox, Safari, Chrome, Opera, Brave)</p>
<ul style="list-style-type: none"> • Storing High Scores Persistently 	<ul style="list-style-type: none"> • Achieve a new high score and then close the game • Reopen the game and ensure the high score still appears on the screen
<ul style="list-style-type: none"> • Endless Runner Game 	<ul style="list-style-type: none"> • Play the game for as long as possible without dying and ensure that the game keeps going
<ul style="list-style-type: none"> • Looping Background 	<ul style="list-style-type: none"> • Ensure there are no visible breaks in the background
<ul style="list-style-type: none"> • Random Cactus Spawning 	<ul style="list-style-type: none"> • Ensure that every cactus (after the first) is randomly selected from the 4 sprites available
<ul style="list-style-type: none"> • Keyboard Input 	<ul style="list-style-type: none"> • Verify: <ul style="list-style-type: none"> ○ Enter/Space keys starts the game ○ The R key restarts the game ○ The Escape key pauses the game ○ The Up/W keys make the player jump