

Documentație
Metode de Dezvoltare Software

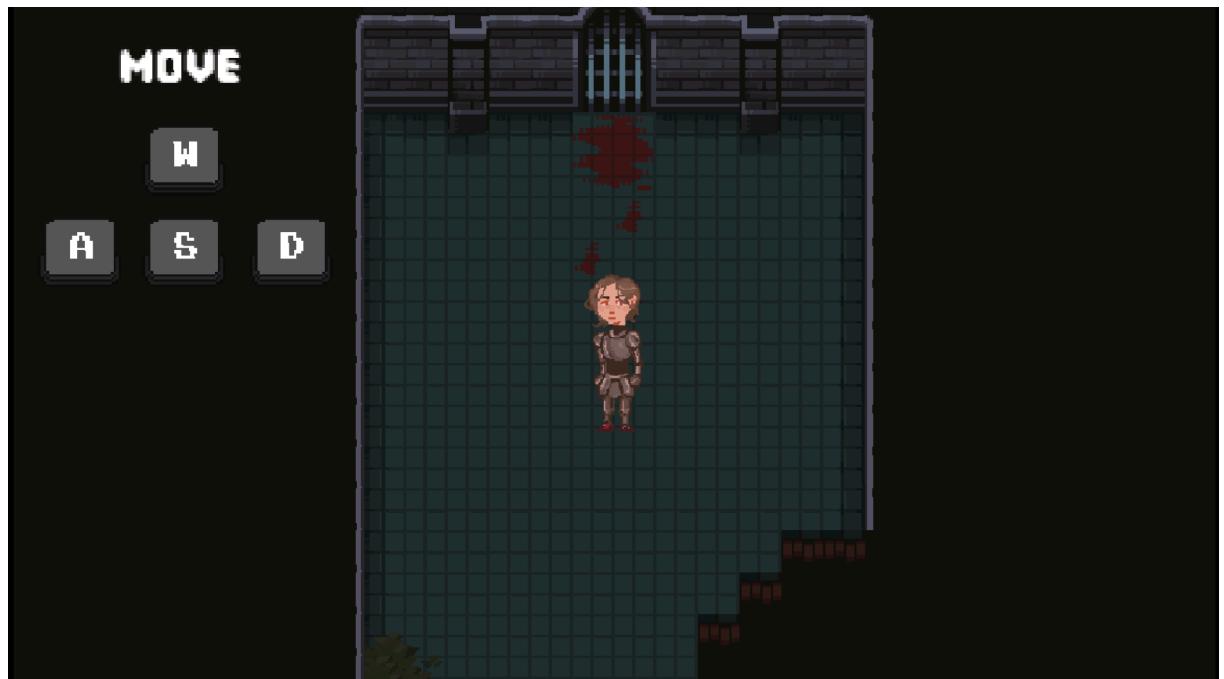
RUNAWAY
-joc realizat în Godot-

Cuprins:

A. Demo Offline:	2
B. Procesul de dezvoltare software	3
1. User stories + backlog creation	3
2. Diagrame	5
3. Source control cu git	7
4. Teste automate	8
5. Raportare bug și rezolvare cu pull request	8
6. Comentarii cod și respectarea code standards	8
7. Design patterns	9
8. Prompt engineering	9

A. Demo Offline:

<https://youtu.be/AVCOGII08Bw>

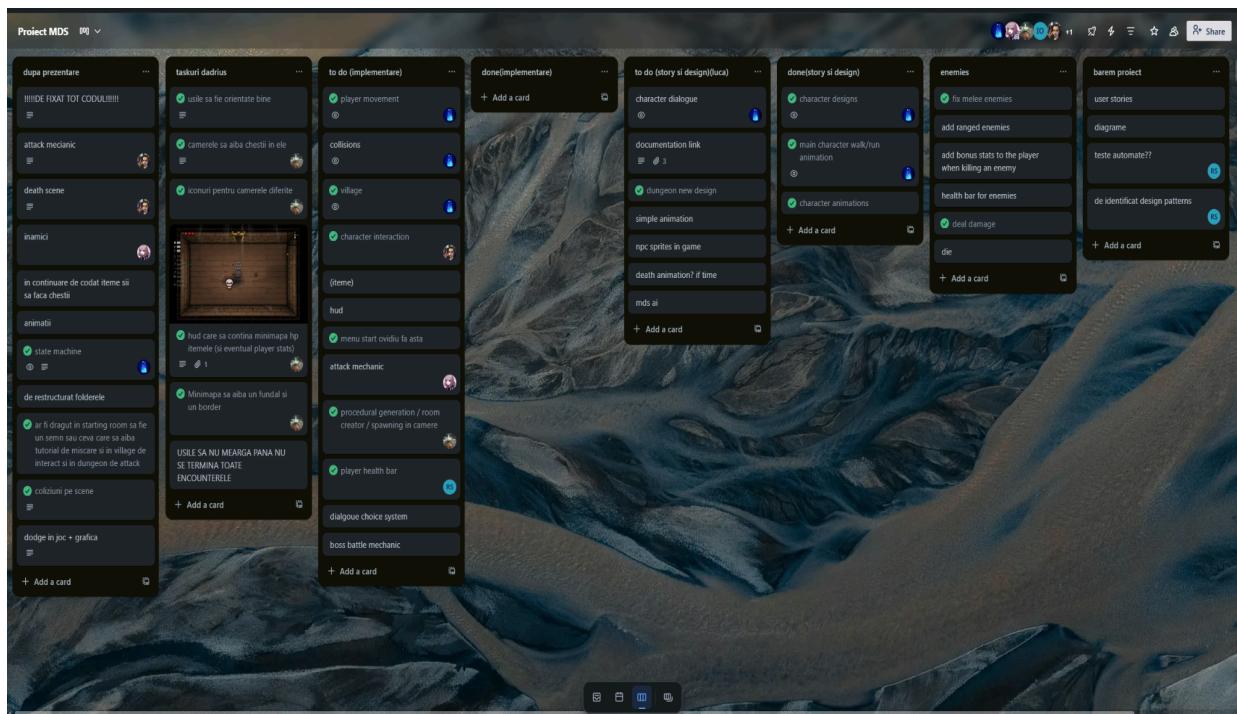


B. Procesul de dezvoltare software

1. User stories + backlog creation

As a	I want	So that
Player	An intuitive tutorial	I can easily get familiarized with the game's mechanics and controls
	To interact with the characters through dialogue	I can explore the game's lore, get to know their stories and personalities better and advance in the game
	A fighting area, like a dungeon	I can progress in the game and test my mechanical abilities
	To fight multiple types of enemies, for example melee and ranged	I am challenged to do my best and not get bored
	An attack system, for example with bullets	I have a chance to clear through the enemies
	An inventory system that can hold a variety of items	I can get bonus stats, health regeneration and advance through stages that require me getting a certain item
	A map while I'm in the dungeon	I know where I am and what is left to explore in the dungeon
	A fair boss fight	I don't want to easily win and neither get easily frustrated and have a rewarding ending to the game
	A new dungeon layout every playthrough	Every time I play the game, I don't get bored
	A health bar system	I know my limits when fighting enemies and plan out my strategy
	A pause menu	I can pause the game and return to it whenever I want to

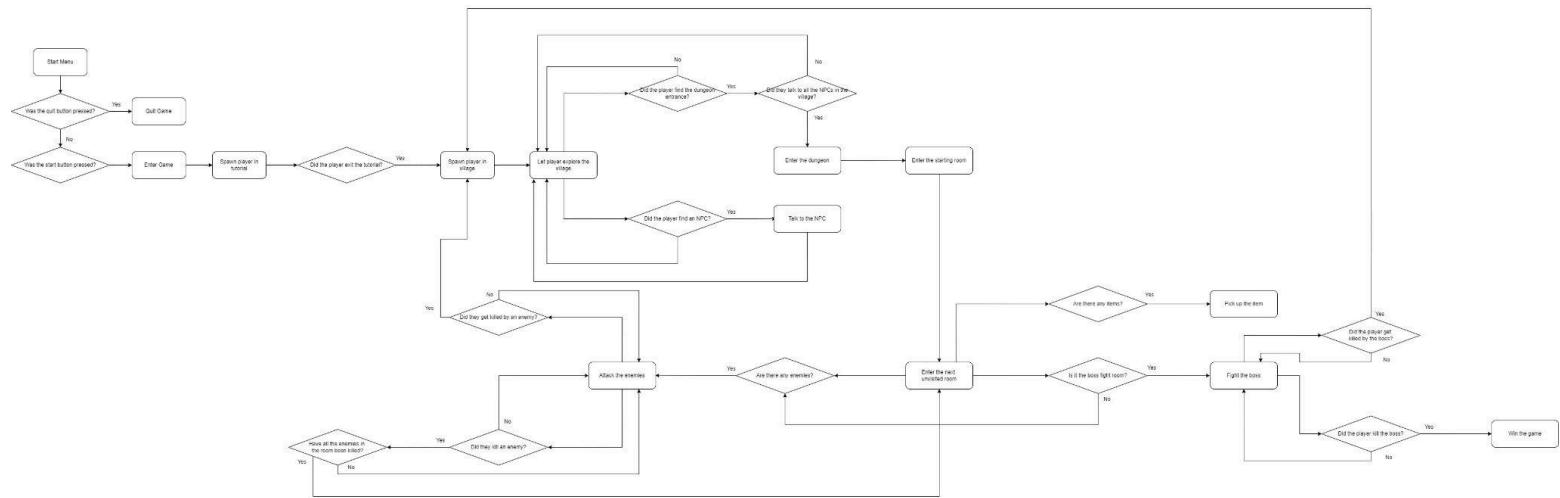
Trello: <https://trello.com/b/Lv38hj65/project-mds>



2. Diagramme

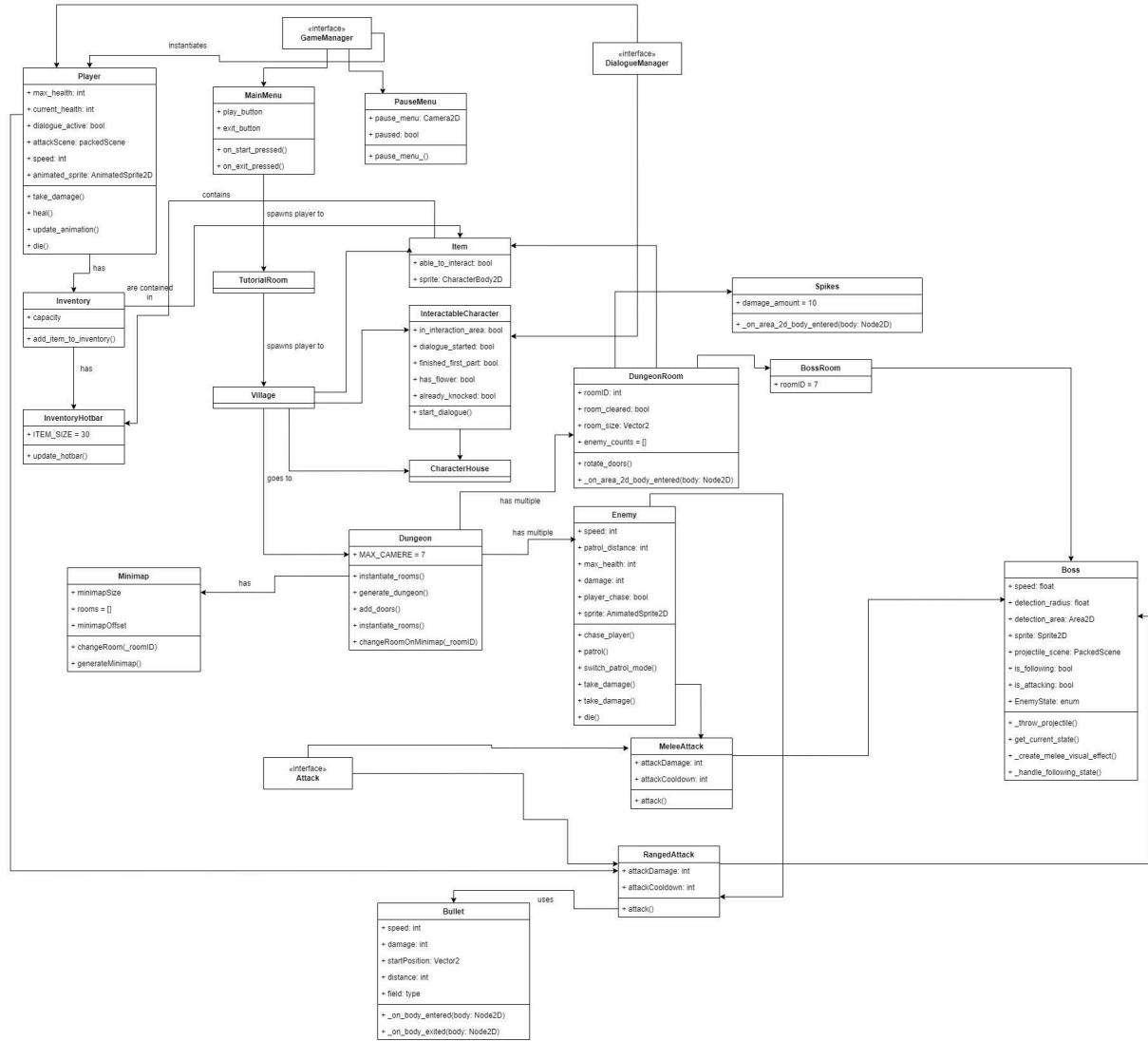
Workflow:

state diagram.jpg



UML:

 uml v2.jpg



3. Source control cu git

(branch creation, merge/rebase, pull requests, minim 10 commits)

The screenshot shows the GitHub repository 'ProjectMDS' (Public). At the top, there are buttons for 'Pin' and 'Unwatch'. Below that, a dropdown shows 'main' with 33 branches and 0 tags. A search bar says 'Go to file' with a 't' icon. There are buttons for 'Add file' and 'Code'. A list of recent commits is shown:

- elenacurecheriu** Update README.md (933b94f - 4 hours ago) 117 Commits
- Backgrounds (duplicate of Darius-DungeonRooms) (last month)
- Dialogues (fixed dialogue bugs, new dialogue balloon and fonts) (13 hours ago)

The screenshot shows the GitHub Pull Requests page for the 'ProjectMDS' repository. The 'Pull requests' tab is selected. A search bar at the top right says 'Type [] to search'. Below it, there are filters: 'Filters' (set to '0 Open, 27 Closed'), 'Labels' (9), 'Milestones' (0), and a 'New pull request' button. A link to 'Clear current search query, filters, and sorts' is also present. The main list of pull requests is shown:

- Boss fight** #33 by imnofuckingfun was merged 13 hours ago
- dialog and dungeon updates** #32 by imnofuckingfun was merged 13 hours ago
- Terminat tot ce inseamna story scene** #31 by Darius-Beldi was merged 16 hours ago
- Gui Hotbar in tot jocul** #30 by Darius-Beldi was merged 16 hours ago
- Started implementing the dialogue** #29 by Darius-Beldi was merged 17 hours ago
- Gui items** #28 by Ivan-Ovidiu was closed 16 hours ago
- Enemies** #27 by elenacurecheriu was merged yesterday
- Robert teste** #26 by SirbuRobert was closed yesterday
- Added a button to open doors** #25 by Darius-Beldi was merged yesterday
- Added attack cooldown** #24 by mihaiardanu11 was merged 2 days ago

4. Teste automate

Ce sunt și de ce le folosim?

Testele automate verifică că funcționalitățile jocului nostru funcționează corect după fiecare modificare. În loc să testezi manual player stats, dungeon generation și sistemul de interacțiuni de fiecare dată, testele fac asta automat în 10 secunde.

Framework: GUT (Godot Unit Test)

Testele sunt implementate folosind **GUT** - un framework de testing dedicat pentru Godot. GUT funcționează ca un plugin care:

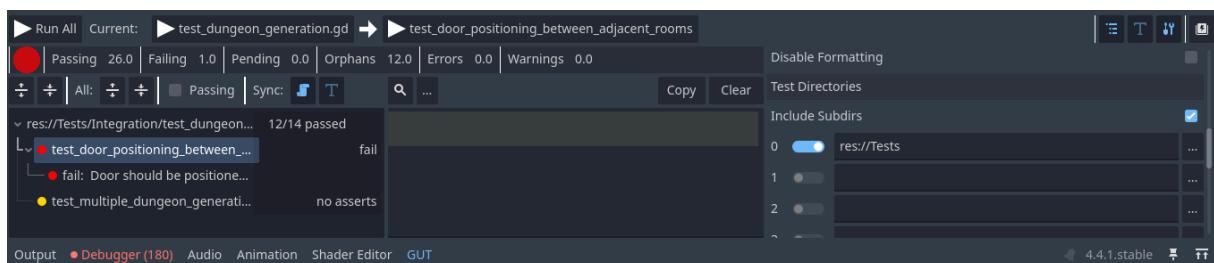
- Rulează scripturi care extind [GutTest](#)
- Oferă funcții de verificare ([assert_eq](#), [assert_true](#), etc.)
- Raportează rezultatele în timp real
- Se integrează direct în editorul Godot

Structura testelor

```
Tests/
├── Unice/                                # Teste pentru componente
    └── individuale
        ├── test_player.gd                 # Player stats, health, inventory
        ├── test_game.gd                  # Dungeon generation logic
        └── test_interacting_component.gd  # Interaction system
    └── Integration/                      # Teste pentru workflow-uri complete
        ├── test_dungeon_generation.gd   # Generare + UI + navigare
        └── test_player_interaction.gd   # Player + items + feedback
```

Interpretarea rezultatelor

- **PASSED** - Funcționalitatea merge corect
- **FAILED** - Bug detectat, needs fixing
- **RISKY** - Test fără verificări, probabil incomplet



```
Finished

=====
= Run Summary
=====

res://Tests/Integration/test_dungeon_generation.gd
- test_door_positioning_between_adjacent_rooms
  [Failed]: Door should be positioned between rooms
  at line 169
- test_multiple_dungeon_generations_are_different
  [Risky]: Did not assert

---- Totals ----
Scripts          6
Tests            28
  Passing        26
  Failing         1
  Risky/Pending   1
Asserts          95
Time             0.795s

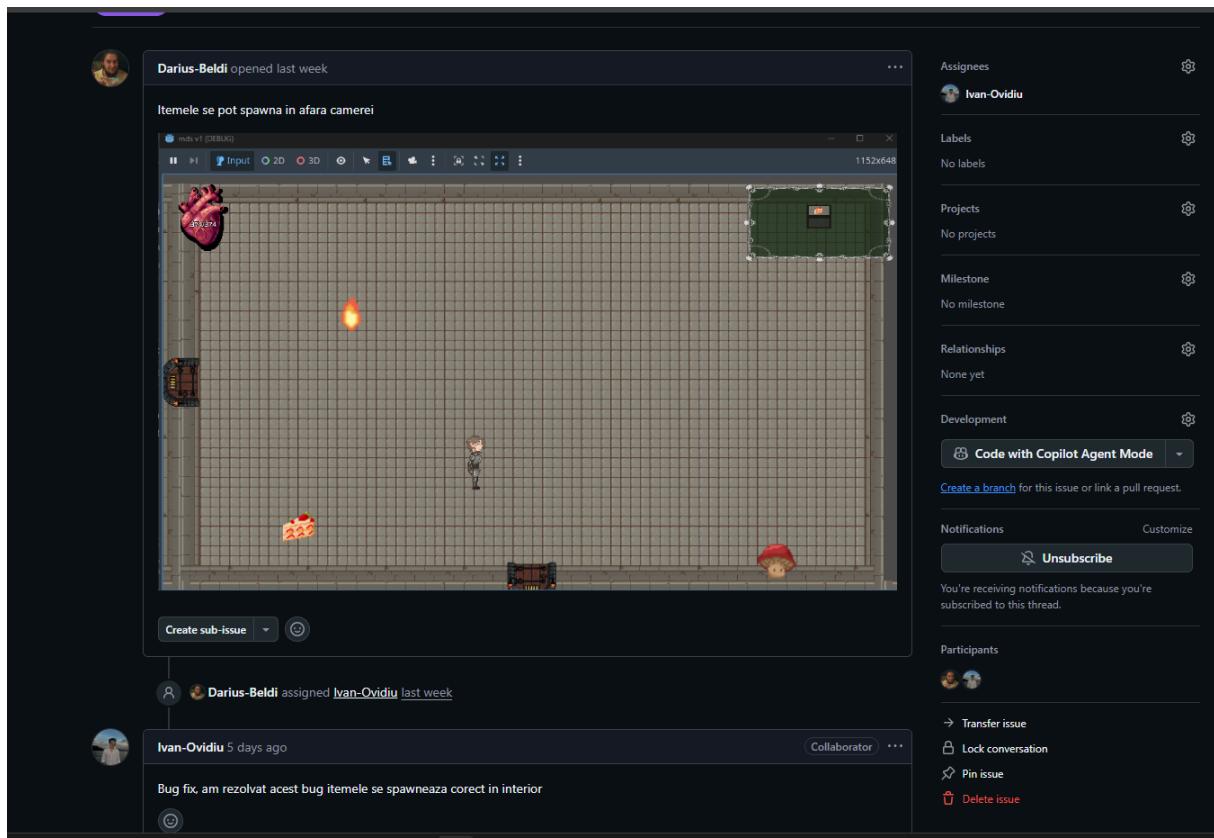
[Orphans]: Total orphans in run 6.0
Note: This count does not include GUT objects that will be freed upon exit.
      It also does not include any orphans created by global scripts
      loaded before tests were ran.
Total orphans = 12 0
Tests  100% res://Tests
Scripts 100% test_suite.gd

Compact
```

Se gasesc pe repo-ul de Github: <https://github.com/elenacurecheriu/ProjectMDS/tree/main/Tests>

5. Raportare bug și rezolvare cu pull request

<https://github.com/elenacurecheriu/ProjectMDS/issues?q=is%3Aissue%20state%3Aclosed>



6. Comentarii cod și respectarea code standards

Exemplu: <https://github.com/elenacurecheriu/ProjectMDS/blob/main/Scripts/game.gd>; https://github.com/elenacurecheriu/ProjectMDS/blob/main/assets/attack_bullets/bullet.gd

7. Design patterns

Singleton: nod “manager” (Game), nu se instantiaza de mai multe ori (Player, GUI)

Component: player-ul si inamicii au health, componente de atac

Factory: instantieri de scene pentru iteme, camere de dungeon, inamici, etc.

State: schimbari de animatie pentru player

Metoda template: “ActiveInteraction”

MVC: separarea jocului in scene, scripturi si date

Prototype: utilizarea unor elemente prefabricate (spikes, items, rooms)

Observer: pentru dialogul cu Red Panda(boss-ul) : red_panda_relay.gd verifica daca red_panda.gd emite semnalul startBoss si emite la randul lui semnalul startBossRelay care este prins de game.gd

8. Prompt engineering

documentarea folosirii toolurilor de AI in timpul dezvoltarii software (d. ex. GitHub Copilot, chatGPT, Microsoft Copilot)

Exemplu 1:

Raluca - Utilizat CLAUDE în ajutarea creării unor blending-modes shaders

Prompt: in godot 4 im trying to make some shaders for color rect with blending modes, so far i have this but it doesnt apply the base to the view itself, how can i fix it to work like a blending mode in photoshop?

R: I can see that you're working on creating a ColorRect shader in Godot 4 that applies blending modes similar to Photoshop. The issue you're facing is that your shader isn't properly applying the base image to the view.

Looking at your fragment() function, I can see a couple of issues:

1. You're trying to use SCREEN_TEXTURE both as a sampler2D and as a color vector in the overlay function
2. Your code isn't correctly separating the base layer from the blend layer

Let me help you fix the shader to work more like Photoshop blending modes:

The main issue with your original shader is that you're not properly separating the base layer (what's behind your ColorRect) from the blend layer (your ColorRect with tint applied). In Photoshop-style blending, you need two distinct layers.

Here's what I changed in the shader:

1. Added a default value for the tint uniform: `uniform vec3 tint = vec3(1.0, 1.0, 1.0)`

2. Added a uniform to select the blending mode: `uniform int blend_mode = 6` (default is overlay)
3. Fixed the fragment shader to:
 - o Get the screen color (base layer) from `SCREEN_TEXTURE`
 - o Use the tint as the blend layer
 - o Apply the selected blending mode
 - o Output the result

To use this shader with your `ColorRect`:

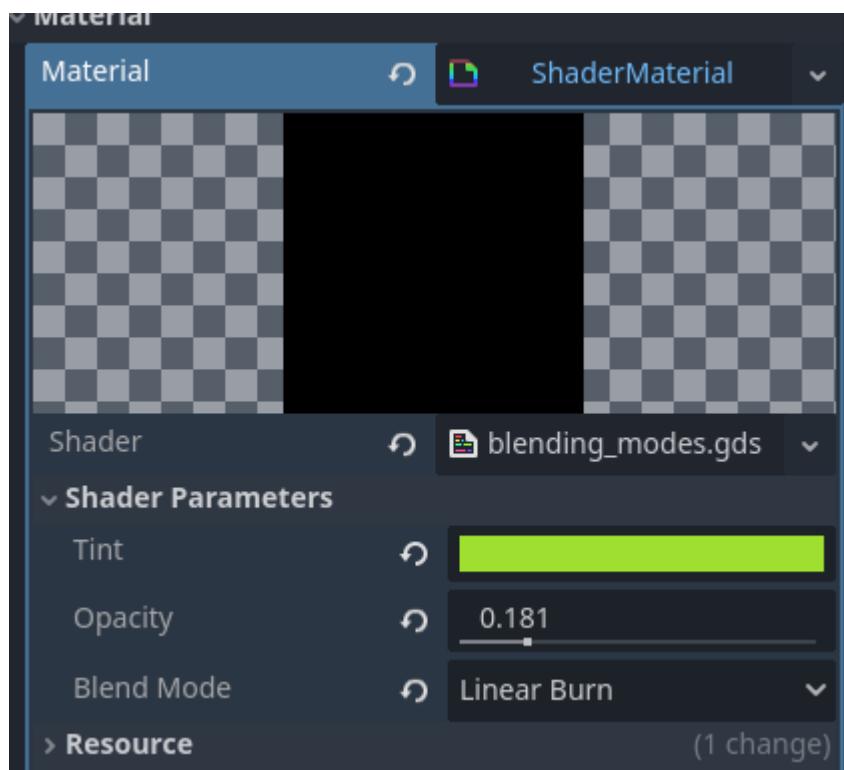
1. Create a `ColorRect` node in your scene
2. Assign this shader to the `ColorRect`'s material
3. Adjust the tint and `blend_mode` uniforms in the Inspector

You can also expand the shader to include alpha handling if needed, but this basic version should work for most cases.

For even more control, you could create a custom shader material with two textures (base and blend) if you want to blend between two specific textures rather than blending with the screen.

Would you like me to explain any specific part of the shader in more detail or adapt it for a different use case?

- cu ajutorul acestui prompt am reușit să înțeleg cum funcționează Materialele și Shaderele în această aplicație și am implementat cu succes blending modes



Exemplu 2:

my enemy has a detection area to chase the player, but it doesn't chase them. what are the possible issues?

If your enemy has a **detection area** (likely an `Area2D` with a `CollisionShape2D`) but **doesn't chase the player**, here are the most common issues to check:

✓ 1. Signal Not Connected

Make sure the `Area2D` is connected to a signal like:

```
gdscript
func _on_detection_area_body_entered(body):
    if body.name == "Player":
        chasing = true
```

⚠ Check: Go to the `Area2D` node → Node tab → Ensure `body_entered` or `area_entered` is connected to a script.

i have fixed the chasing issue, but when 2 enemies touch each other, they overlap; i want them when they touch to go in the opposite direction

Awesome! Glad chasing works now. 😊

Now let's fix **enemy-on-enemy overlapping** by making them **repel** each other when they touch.

✓ Goal: Enemies push away from each other on contact

⚠ Best Approach: Use `move_and_slide()`'s collisions to repel nearby enemies

Update `_physics_process()` to detect collisions like this:

```
gdscript
func _physics_process(delta: float) -> void:
    # [ ... your existing logic ... ]

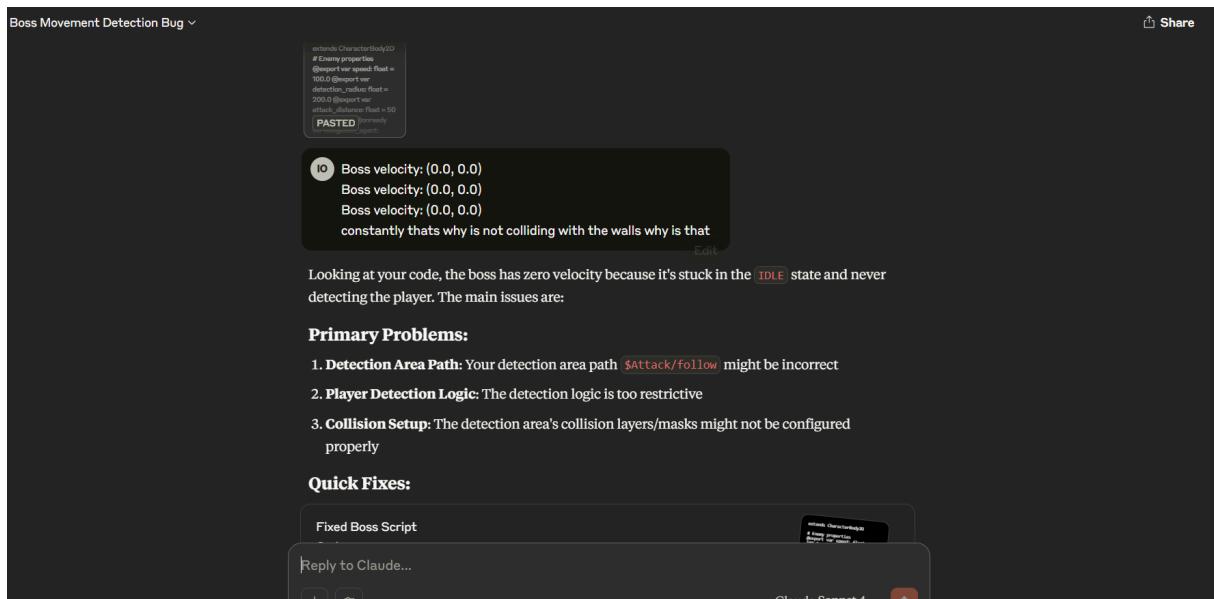
    move_and_slide()

    # Repel other enemies after moving
    repell_from_other_enemies()
```

Ask anything

A
G

Exemplu 3:



Am utilizat Claude pentru a reusi sa fac boss-ul sa nu mai treaca prin peretii dungeon-ului:

Looking at your code, the boss has zero velocity because it's stuck in the **IDLE** state and never detecting the player. The main issues are:

Primary Problems:

1. **Detection Area Path:** Your detection area path **\$Attack/follow** might be incorrect
2. **Player Detection Logic:** The detection logic is too restrictive
3. **Collision Setup:** The detection area's collision layers/masks might not be configured properly

Quick Fixes:

Fixed Boss Script

Code

```
extends CharacterBody2D # Enemy properties @export var speed: float = 100.0 @export var detection_radius: float = 200.0 @export var attack_distance: float = 50 # References @onready var navigation_agent: NavigationAgent2D = $NavigationAgent2D @onre
```

Key Changes Made:

1. **Better Detection Area Handling:** Added fallback paths and manual detection
2. **Fixed Player Detection:** More flexible **_is_player()** method

3. **Manual Detection Fallback:** If Area2D fails, uses distance-based detection
4. **Collision Setup:** Automatically sets collision mask to 1
5. **Added Safety Checks:** Prevents crashes from missing nodes
6. **Better State Debugging:** Shows current state in physics process

Things to Check in Your Scene:

1. **Detection Area Setup:**
 - Make sure the Area2D node exists and is properly named
 - Set its collision mask to match your player's collision layer (usually 1)
 - Ensure the CollisionShape2D child has a proper shape
2. **Player Setup:**
 - Add your player to a group called "player":
`player.add_to_group("player")`
 - Or ensure the player node is named "Player"

Scene Structure should look like:

```
Boss (CharacterBody2D)
└── Sprite2D
└── CollisionShape2D (for boss collision)
└── DetectionArea (Area2D)
    └── CollisionShape2D (large circle for detection)
└── NavigationAgent2D
    3. └── AttackTimer
```

The boss should now move towards the player once detected!

Exemplu 3:

Am folosit GitHub Copilot pentru a ma ajuta sa intreleg mai usor documentatia de la Addon-ul de dialog.

Am creat un Space nou cu readme-urile din documentatie si apoi i-am dat query urile

The screenshot shows a GitHub Space interface. On the left, there is a form for creating a new space named 'dialog'. The right side displays a search result for 'how do i do a dialogue that keeps going without user inputs'.

Space Creation Form (Left):

- Name:** dialog
- Description (optional):** Memorable name that helps you find your space.
- Icon:** Pick a color to make your space more recognizable. (Auburn selected)
- Instructions (optional):** Changes how Copilot responds on specific questions or tasks.
- Attachments (optional):** A list of files from the Godot Dialogue Manager documentation folder, including .md files like gdignore, 2to3.md, API.md, Basic_Dialogue.md, Conditions_Mutations.md, FAQ.md, CSharp.md, Example_Balloons.md, Settings.md, Translations.md, and Using_Dialogue.md.

Search Result (Right):

Query: how do i do a dialogue that keeps going without user inputs

Result Preview:

Nathan: Hello there! [next=2]
Coco: Meow! [next=1]
Lilly: Let's go! [next=auto]

Text:

- [next=2] waits for 2 seconds before continuing.
- [next=auto] automatically determines an appropriate delay based on the length of the line.

Example:

How it works:

- The example balloon (`<show_example_dialogue_balloon>`) and the official `DialogueLabel` node both support `[next=...]` and will proceed when the timer expires.
- If you use a custom balloon, make sure your code respects the `[next=...]` tag (see the [API documentation](#) and [Basic Dialogue docs](#)).

Summary:

Just add `[next=N]` or `[next=auto]` to your dialogue lines, and the dialogue will keep going without the user doing anything.

Relevant documentation:

- [Basic Dialogue](#)
- [API](#)

Let me know if you need a script/code example!

The screenshot shows a code editor with a file named 'boss' containing a dialogue script for Red Panda.

```
1 | start
2 | Red Panda: Let me help you jog your memory.[next=auto]
3 | Red Panda: You had just fought over something stupid, in hindsight. [next=auto]
4 | Red Panda: You hoped a drive in the woods and getting something sweet would help. [next=auto]
5 | Red Panda: Things calmed down, somewhat. They made a passing remark at the argument, and it was the ember that was needed to reignite the fire again. [next=auto]
6 | Red Panda: Fighting while driving at night. A blinding headlight, a wrong turn. Wrong place, wrong night. [next=auto]
7 | Red Panda: One day and your life is changed forever. You know what happened, you saw it all right before you came here. [next=auto]
8 | Red Panda: [color=red][wave]Do you remember now?[/wave][/color][next=auto]
9 | Red Panda: [color=red]I guess it's a bit too much to take in when your head's concussed in the ditch of a random forest inside of a wrecked up car.[/color][next=auto]
10 | Red Panda: You've been here a while. But this is only the beginning. [next=auto]
11 | Red Panda: This, everything you went through here, is merely to [color=red]help you accept the reality you had faut in creating[/color]. [next=auto]
12 | Red Panda: Grief will always follow in your shadow - it's a nasty little parasite like that. [next=auto]
13 | Red Panda: Do you think you can ever forget the look on their face before they took their last breath? [next=auto]
14 | Red Panda: But don't feel too lonely. We will always be here, lurking in your subconscious, whispering in your ear. [next=auto]
15 | Red Panda: What, did you really think that you, a measly kid, are some knight in shining armour, sent to defeat the evils roaming in the dungeons of some whacky to
16 | Red Panda: Well, my dear knight, your target was a bit misplaced. I wouldn't call us evil, but WE are the ones you'll wish to get rid of. [next=auto]
17 | Red Panda: Blows of Denial, Depression, fits of Anger at yourself and the world, they will forever poison your mind. [next=auto]
18 | Red Panda: You'll keep wishing you could Forget it all or Change the past. For the Built you will feel will haunt your every waking moment and make you see blood o
19 | Red Panda: Time doesn't heal. It just helps you forget the details.[next=auto]
20 | Red Panda: But [color=red]you will never forget this. [/color][next=auto]
21 | Red Panda: You will never forget them, or this cursed day. [next=auto]
22 | Red Panda: And you shall never forgive yourself. For if you do, we will be waiting for you, to teach you all the lessons, again and again. [next=auto]
23 | Red Panda: But we won't be as patient, and we won't be as nice, so try to make sure that doesn't happen. [next=auto]
24 | Red Panda: Go on up now. I expect you're in the hospital by this time. Someone's waiting for you to wake up, kid.[next=auto]
25 |
26 | => END
```