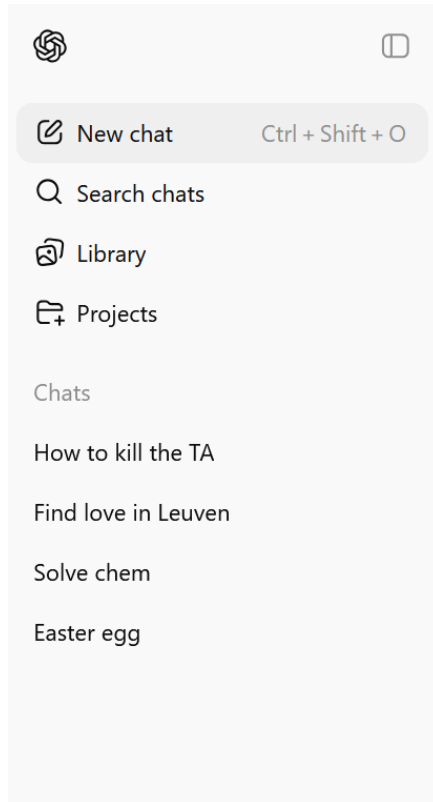


Non-complex example: Game in Godot game engine with Claude 4.7 opus and VScode

The case



ChatGPT ▾

✦ Get Plus

What can I help with?

 Assignment_programming_techniques...
PDF

+ Plz solve



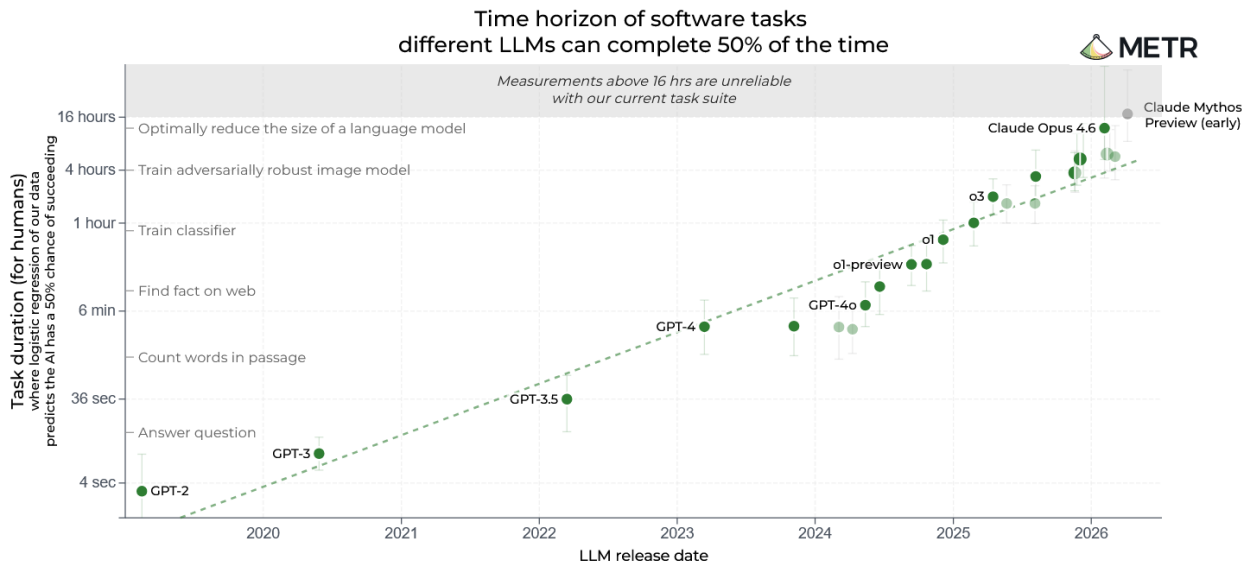
The case

One year ago:

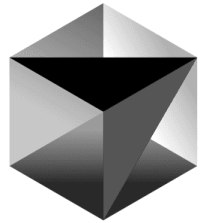
- Agentic AI brand new
- Claude 4 with 200K tokens

Now:

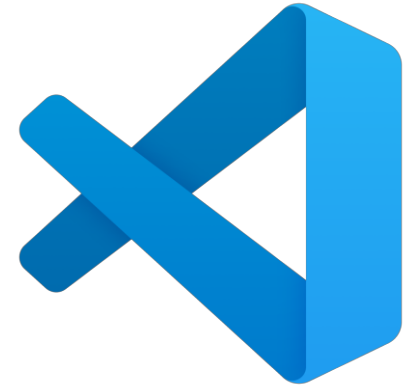
- Claude 4.7 opus with 1M tokens
- Few minutes to 10+ hours of independent work
- Run and troubleshoot in bash



The case

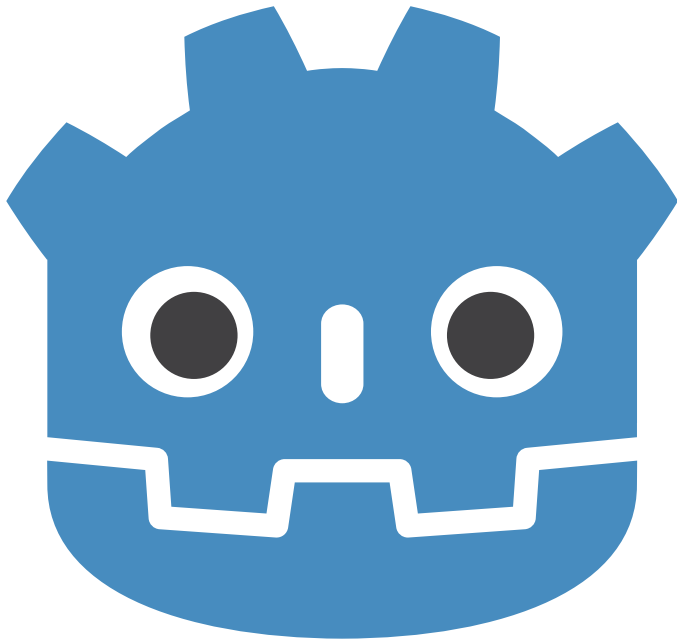


CURSOR



Claude

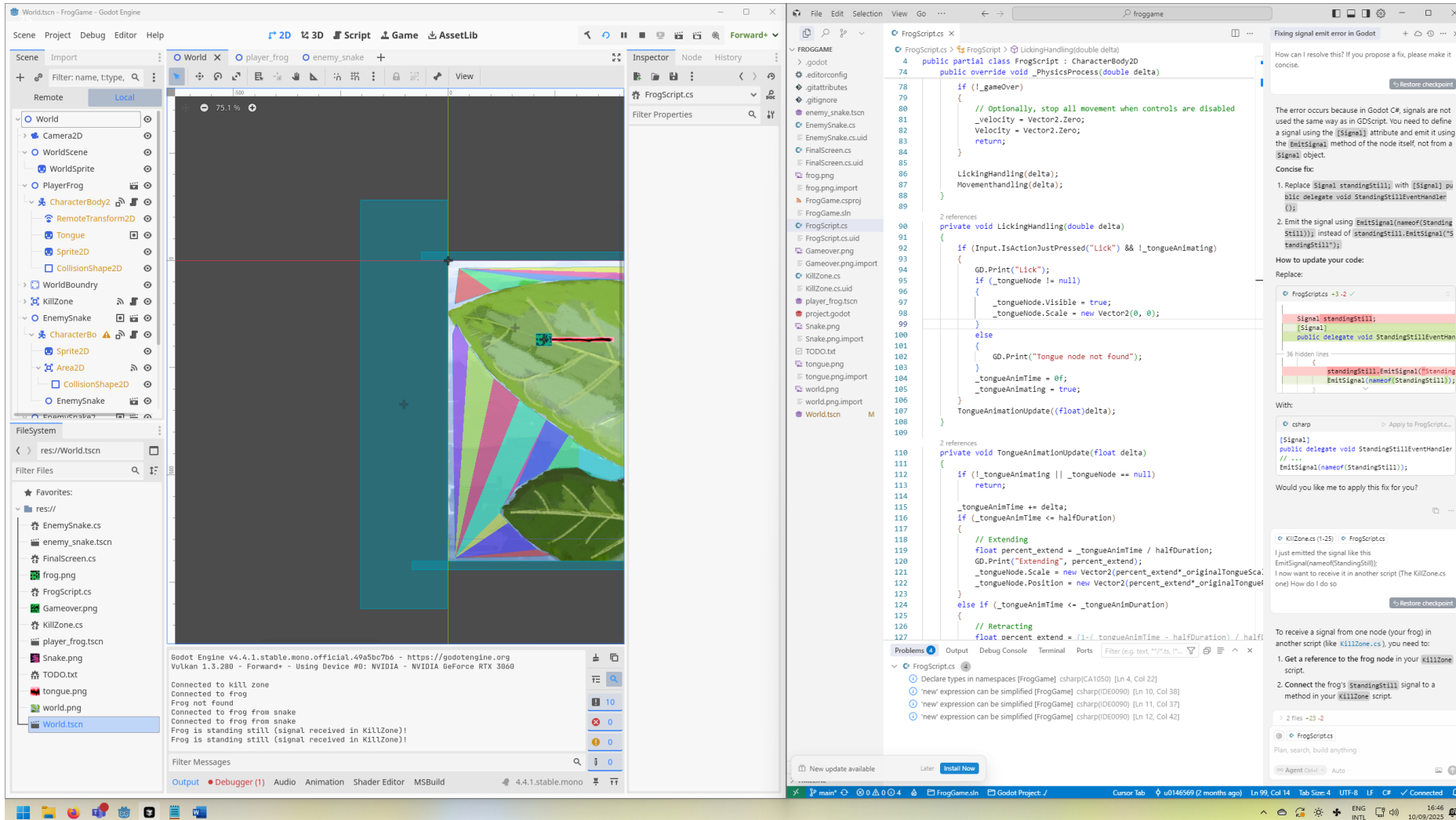
The case



GODOT

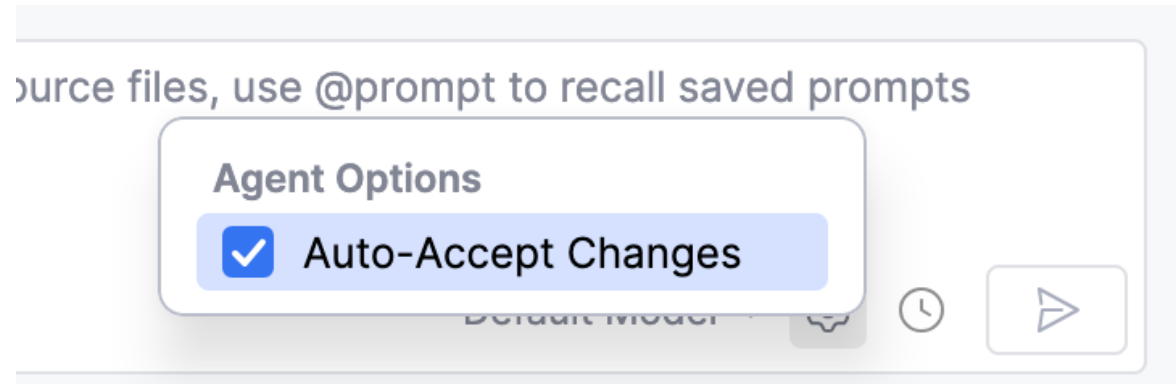
Game engine

The case



The set-up

Vibe coding



Step 1: add a tongue

- “Give me code to add a tongue to the frog”

Step 1: add a tongue



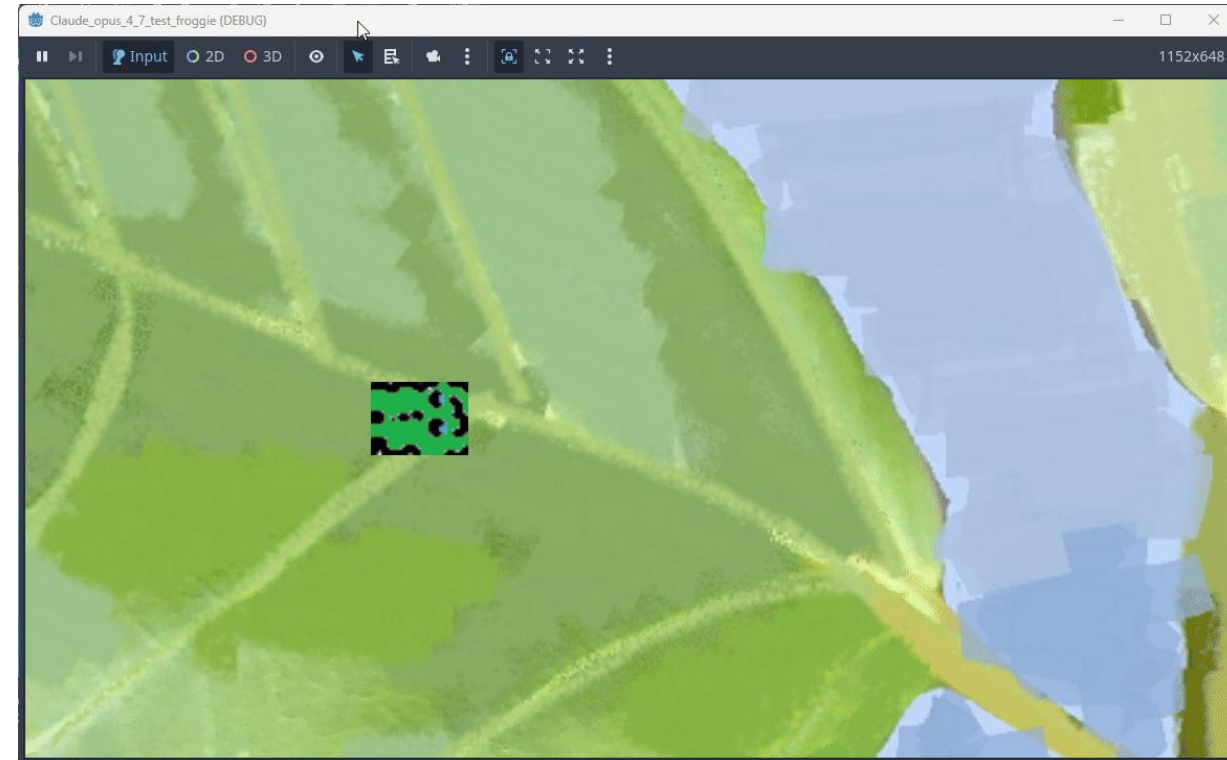
Step 2: animation

- We would like a smooth animation
- Took a couple of minutes and many prompts to itself.
- But it worked!



Step 1&2: update

- "Great, now add a tongue animation. It will extend slowly when pressing q and retract after a bit. You can use the tongue.png file in assets"
- Why not? Later



Step 1&2: update

- "It doesn't seem to work"
- It works!



Step 3: Turning and jumping

- In a similar way I asked it to make jumping logic and turning logic
 - Problem brewing in the background
 - Will not turn during jump



Step 3: update

- "In my godot frog game Add this to my frog protagonist a jumping up with space, leaping with W (variable depending on how long it keeps being pressed in mid-air), turning like a tank with A and D, and finally slowly slithering back with S."



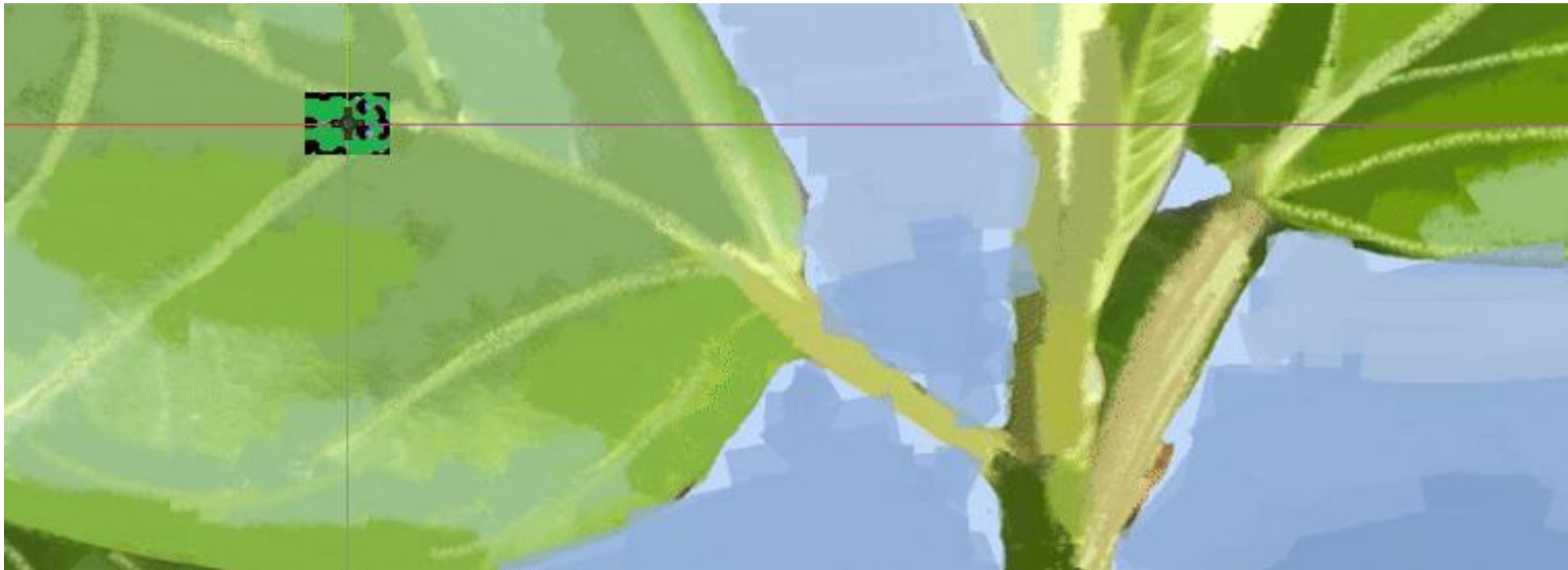
Step 3: update 2

- "In my godot frog game Add this to my frog protagonist a jumping up with space, leaping with W (variable depending on how long it keeps being pressed in mid-air), turning like a tank with A and D, and finally slowly slithering back with S."



Step 3: update 3

- "Please remove the gravity, this game is topdown, not a 2D scroller, so turning should be like looking top down on a tank and turning that way, The size of the frog should also increase to imply going up and reduce to imply going down."



Step 3: update 3

What went wrong:

- Created a new file instead of continuing with the file I reattached to the node...
- After manually cleaning it works
- W does work but only when airborne
- We have already reached 15% of daily tokens.
- Other things revealed later.

Step 4: Immobilise

Immobilise the frog upon touching the snake.

- Doesn't work
- Permanently immobile
- Disables only turning
- Every new prompt breaks a function
- Making all four steps work was unsuccessful after 6 prompts



Step 5: further steps

Kill zones

Camera logic

Snake movement and teleportation

Tongue hitbox

=> Impossible



Step 4&5: update

"Add collision logic so that when the frog touches the KillZone Movement is disabled and the GameOversprite is shown as an overlay"

"Add collision logic so that when the frog touches the KillZone Movement is disabled and the GameOversprite is shown as an overlay"

"Make it so that the kill only occurs if the frog isn't airborne"

Step 4&5: update

"Add collision logic so that when the frog touches the KillZone Movement is disabled and the GameOversprite is shown as an overlay"

"Add collision logic so that when the frog touches the KillZone Movement is disabled and the GameOversprite is shown as an overlay"

"Make it so that the kill only occurs if the frog isn't airborne"



Claude 4.0 problems

Why?

- Different input methods
- No structure
- It doesn't understand its own logic
- Spaghetti

```
private void LickingHandling(double delta)
{
    if (Input.IsActionJustPressed("Lick") && !_tongueAnimating)
    {
        2 references
        GD.Print("Lick");
        if (_tongueNode != null)
        {
            _tongueNode.Visible = true;
            _tongueNode.Scale = new Vector2(0, 0);
        }
        else
        {
            GD.Print("Tongue node not found");
        }
        _tongueAnimTime = 0f;
        _tongueAnimating = true;
    }
    TongueAnimationUpdate((float)delta);
}
```

```
public void GetInput()
{
    if (!_gameOver)
    {
        _rotationDirection = 0;
        return;
    }
    _rotationDirection = Input.GetAxis("SlidePort", "SlideStarboard");
}
```

```
if (Input.IsActionJustPressed("JumpForward"))
{
    _jumpfinished = false;
    _isJumpingForward = true;
    _jumpForwardPressTime = 0f;
    _oldRotation = Rotation;
}
```

```
if (Input.IsActionJustPressed("JumpUp"))
{
    _isJumpingUp = true;
}
```

Claude 4.0 problems

Why?

- Different methods with overlapping purpose
- Methods overstep their purpose
- Overlap causes confusion
- More prompts add to the problem

```
public void GetInput()
{
    if (!_gameOver)
    {
        _rotationDirection = 0;
        return;
    }
    _rotationDirection = Input.GetAxis("SlidePort", "SlideStarboard");
}
```

```
if (Input.IsActionJustPressed("JumpForward"))
{
    _jumpfinished = false;
    _isJumpingForward = true;
    _jumpForwardPressTime = 0f;
    _oldRotation = Rotation;
}
```

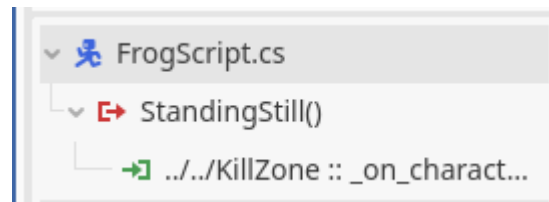
```
private void LickingHandling(double delta)
{
    if (Input.IsActionJustPressed("Lick") && !_tongueAnimating)
    {
        2 references
        GD.Print("Lick");
        if (_tongueNode != null)
        {
            _tongueNode.Visible = true;
            _tongueNode.Scale = new Vector2(0, 0);
        }
        else
        {
            GD.Print("Tongue node not found");
        }
        _tongueAnimTime = 0f;
        _tongueAnimating = true;
    }
    TongueAnimationUpdate((float)delta);
}
```

```
if (Input.IsActionJustPressed("JumpUp"))
{
    _isJumpingUp = true;
}
```

Claude 4.0 problems

Why?

- THREE communication methods!
 - Simple exported variable
 - GetNode (deprecated)
 - Signals and slots
 - Mix of two



```
public void SetControlsEnabled(bool enabled)
{
    _gameOver = enabled;
}
```

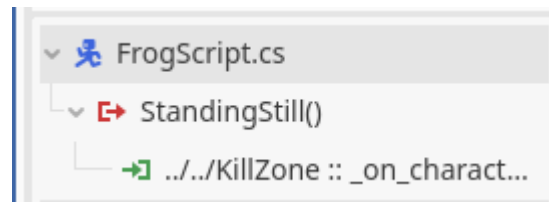
```
private void OnBodyEntered(Node body)
{
    GD.Print("Body entered", body.Name);
    if (body.Name == "PlayerFrog" || body.GetParent()?.Name == "PlayerFrog")
    {
        GD.Print("SNAKE!!!");
        frog.SetControlsEnabled(false);
        EmitSignal(nameof(FrogKilled));
    }
}
```

```
public override void _Ready()
{
    var frog = GetNode<KillZone>(KillZonePath);
    if (frog != null)
    {
        GD.Print("Connected to kill zone");
        frog.Connect("FrogKilled", new Callable(this, nameof(OnFrogKilled)));
    }
    else
    {
        GD.Print("kill zone not found");
    }
}
```

Claude 4.0 problems

Why?

- THREE communication methods!
- Behind the scenes:
 - It confused itself and called the Killzone a frog, connected the wrong signal to it
 - It used the gameOver variable inconsistently
 - etc



```
public void SetControlsEnabled(bool enabled)
{
    _gameOver = enabled;
}
```

```
private void OnBodyEntered(Node body)
{
    GD.Print("Body entered", body.Name);
    if (body.Name == "PlayerFrog" || body.GetParent()?.Name == "PlayerFrog")
    {
        GD.Print("SNAKE!!!");
        frog.SetControlsEnabled(false);
        EmitSignal(nameof(FrogKilled));
    }
}
```

```
public override void _Ready()
{
    var frog = GetNode<KillZone>(KillZonePath);
    if (frog != null)
    {
        GD.Print("Connected to kill zone");
        frog.Connect("FrogKilled", new Callable(this, nameof(OnFrogKilled)));
    }
    else
    {
        GD.Print("kill zone not found");
    }
}
```

Claude 4.7 problems

Code is still unstructured but it works

- Messy main loop
- Flexible responsibility

```
public override void _PhysicsProcess(double delta)
{
    float dt = (float)delta;
    Vector2 velocity = Velocity;

    float turn = 0f;
    if (Input.IsPhysicalKeyPressed(Key.A)) turn -= 1f;
    if (Input.IsPhysicalKeyPressed(Key.D)) turn += 1f;
    Rotation += turn * TurnSpeed * dt;

    Vector2 forward = new Vector2(Mathf.Cos(Rotation), Mathf.Sin(Rotation);
    bool airborne = _height > 0f;

    bool jumpHeld = Input.IsPhysicalKeyPressed(Key.Space);
    if (jumpHeld && !_jumpHeldLastFrame && !airborne)
        _heightVelocity = JumpUpVelocity;
    _jumpHeldLastFrame = jumpHeld;

    bool wHeld = Input.IsPhysicalKeyPressed(Key.W);
    bool wJustPressed = wHeld && !_wHeldLastFrame;
    _wHeldLastFrame = wHeld;

    if (wJustPressed && !airborne)
    { ...
    }

    if (_leapHolding)
    { ...
    }

    if (airborne || _heightVelocity > 0f)
    { ...
    }
    airborne = _height > 0f;

    if (Input.IsPhysicalKeyPressed(Key.S) && !airborne)
    { ...
    }
    else if (!airborne)
    { ...
    }

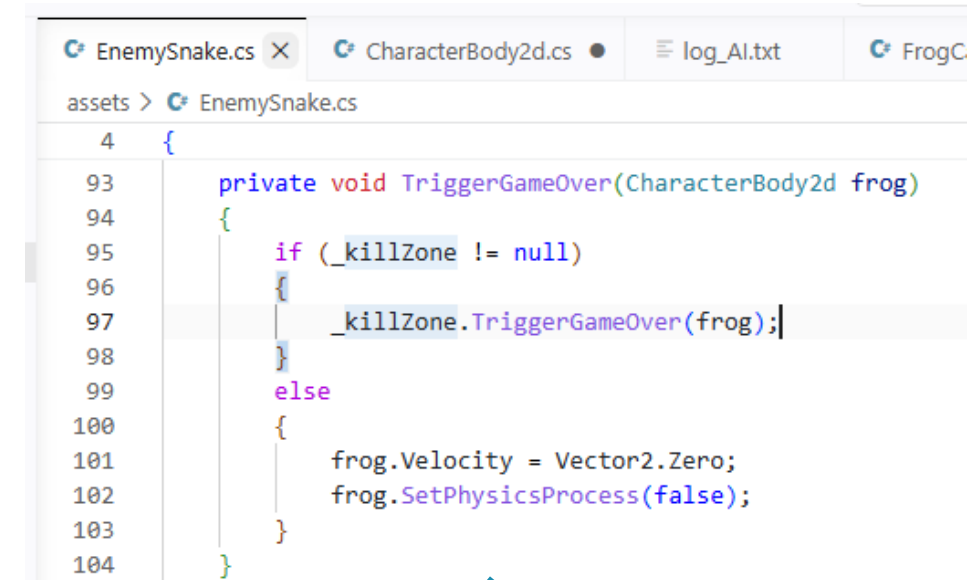
    UpdateTongue(dt);

    float heightMultiplier = 1f + _height * HeightScaleFactor;
```

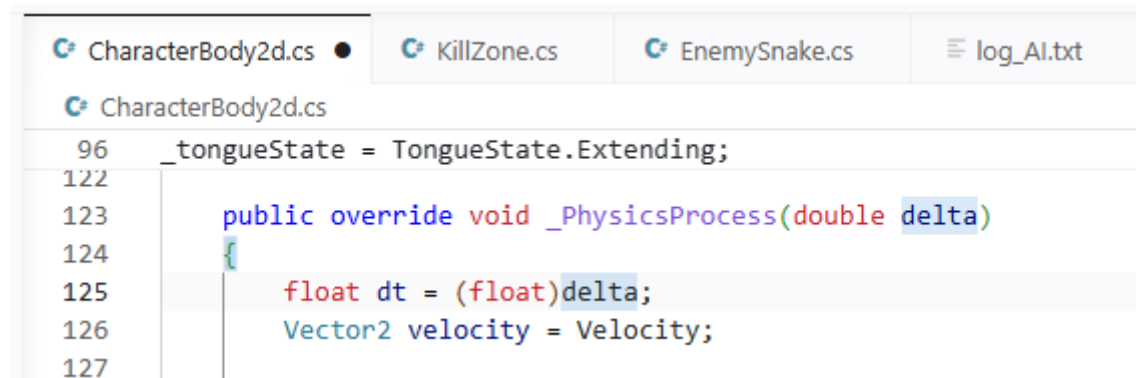
Claude 4.7 problems

Code is still unstructured but it works

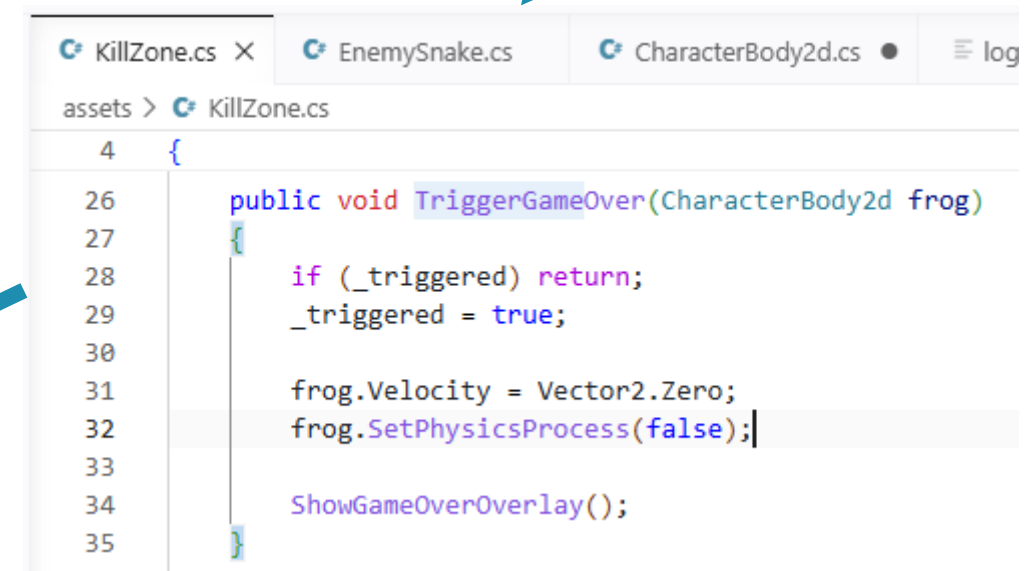
- Messy main loop
- Flexible responsibility
- Spaghetti



```
EnemySnake.cs
4 {
93     private void TriggerGameOver(CharacterBody2d frog)
94     {
95         if (_killZone != null)
96         {
97             _killZone.TriggerGameOver(frog);
98         }
99         else
100         {
101             frog.Velocity = Vector2.Zero;
102             frog.SetPhysicsProcess(false);
103         }
104     }
```



```
CharacterBody2d.cs
96     _tongueState = TongueState.Extending;
122
123     public override void _PhysicsProcess(double delta)
124     {
125         float dt = (float)delta;
126         Vector2 velocity = Velocity;
127     }
```



```
KillZone.cs
4 {
26     public void TriggerGameOver(CharacterBody2d frog)
27     {
28         if (_triggered) return;
29         _triggered = true;
30
31         frog.Velocity = Vector2.Zero;
32         frog.SetPhysicsProcess(false);
33
34         ShowGameOverOverlay();
35     }
```

Claude 4.7 problems

Mistakes occur but are found

- Expensive
- Slow

```
_tongueSprite = new Sprite2D
{
    Texture = tex,
    Centered = true,
    Offset = new Vector2(tex.GetWidth() / 2f, 0f),
    Position = new Vector2(mouthX, 0f),
    Scale = new Vector2(0f, 1f),
    ZIndex = 1,
};
```

It doesn't seem to work



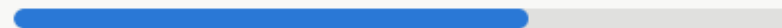
Claude

Most likely cause: `ZIndex = -1` is putting the tongue *behind the world background sprite*. In Godot 4 with `ZAsRelative = true` (the default), the tongue's effective global Z becomes `-1`, while the world background draws at Z `0` — so the world is painted on top of the tongue, hiding it completely.

Plan usage limits Pro

Current session

Starts when a message is sent



66% used

Claude 4.7 problems and possible causes

Problems:

- Deprecated/non-preferred logic
- Unstructured code
- Flexible responsibility
- Spaghetti
- Small and big mistakes
- Hidden problems
- Very costly

Causes?

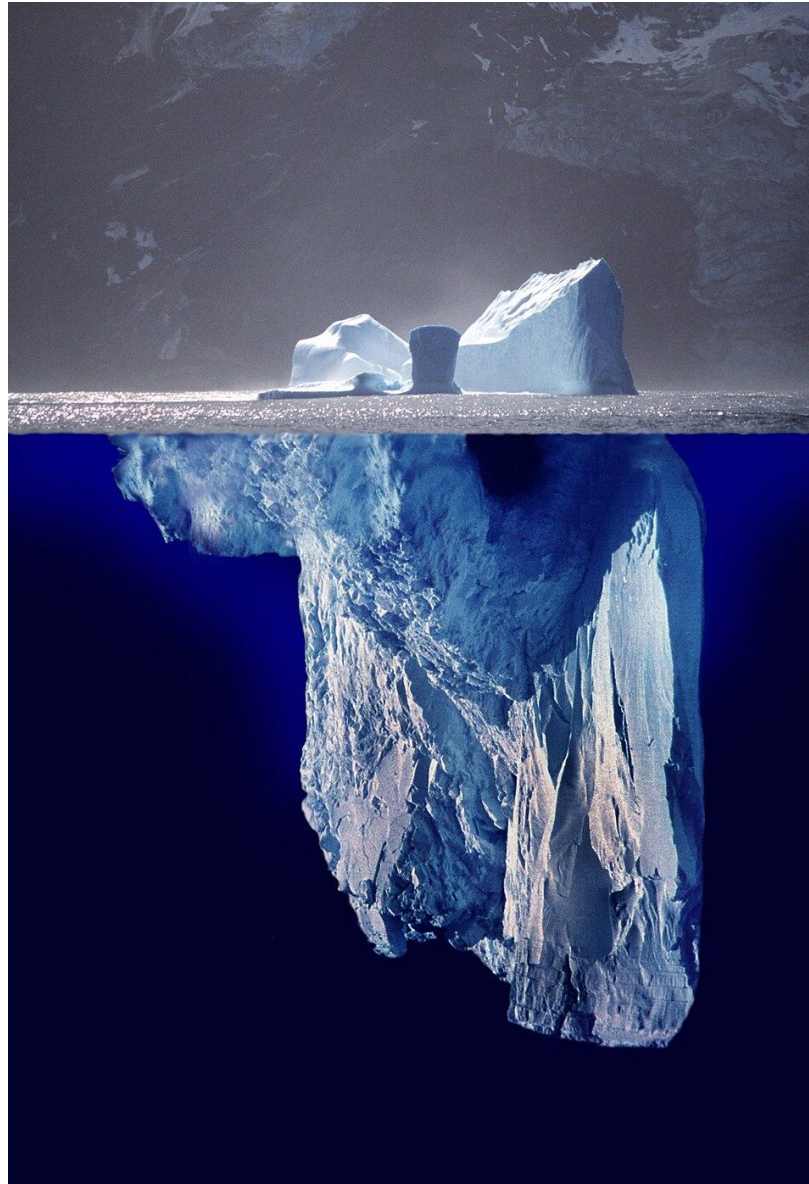
- No feedback
- Unstructured, unsafe, bad code still works, for now...
- The model's code output is not as good as the code it was trained on

Conclusion

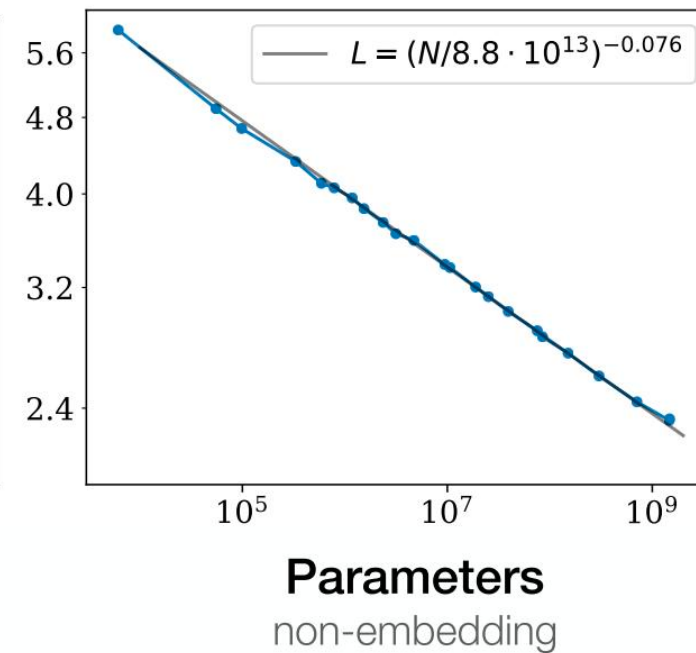
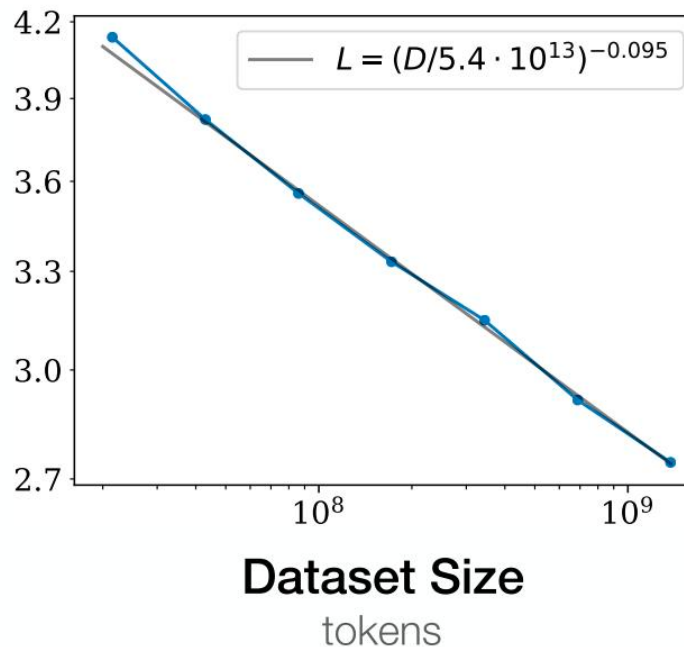
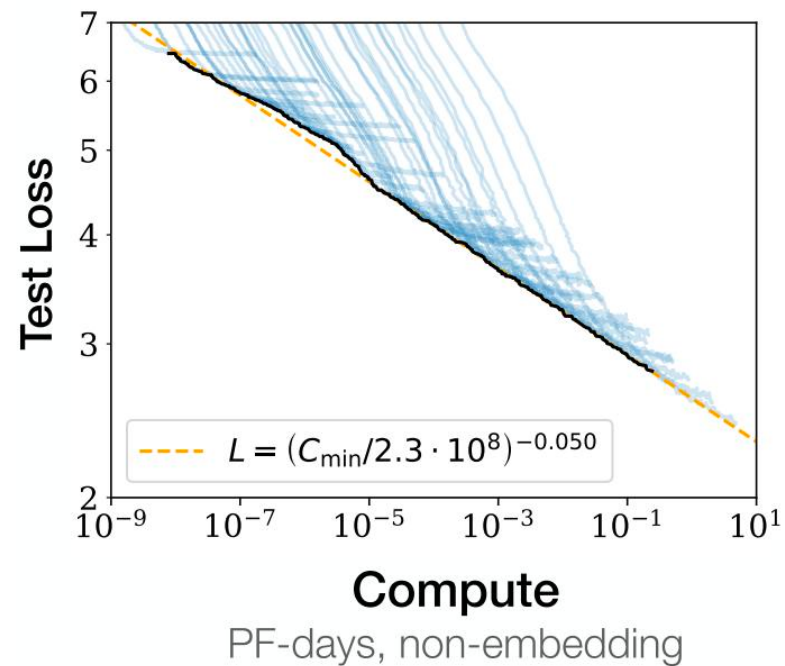
Advice:

- AI is very capable
- Use vibe coding where cleanliness, performance, or safety are unimportant
- Give it proper direction and context window
- Pick an appropriate model to avoid cost and time issues
- Check the code frequently
- Do not blindly follow AI, double check
- Use git! (Or build in versioning control)

Limitations

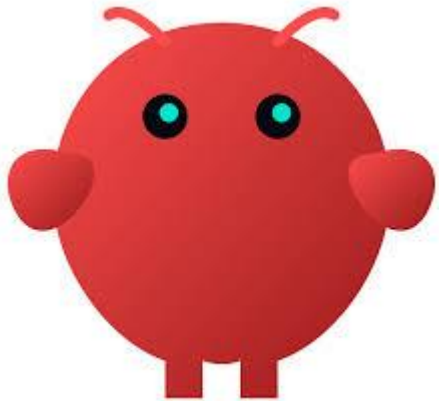
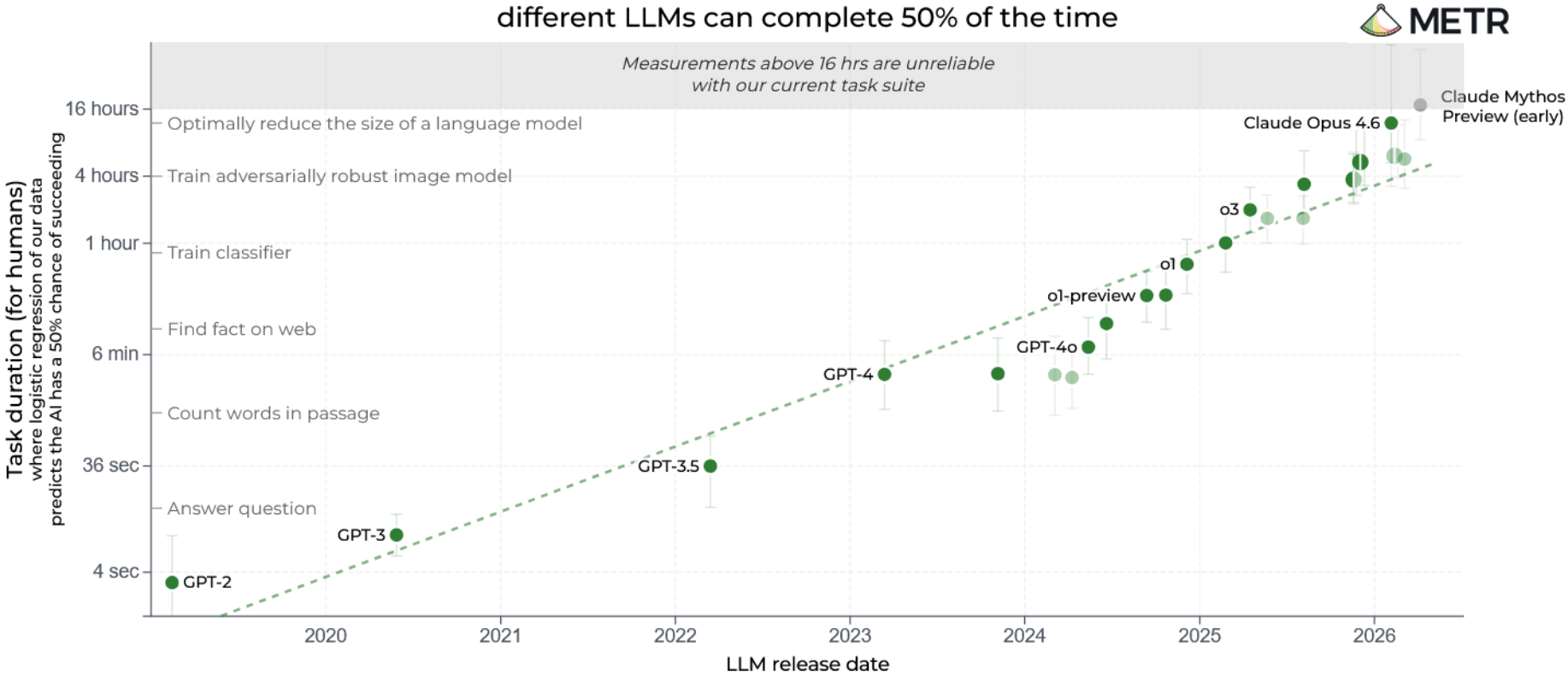


Limitations



Limitations

Time horizon of software tasks
different LLMs can complete 50% of the time



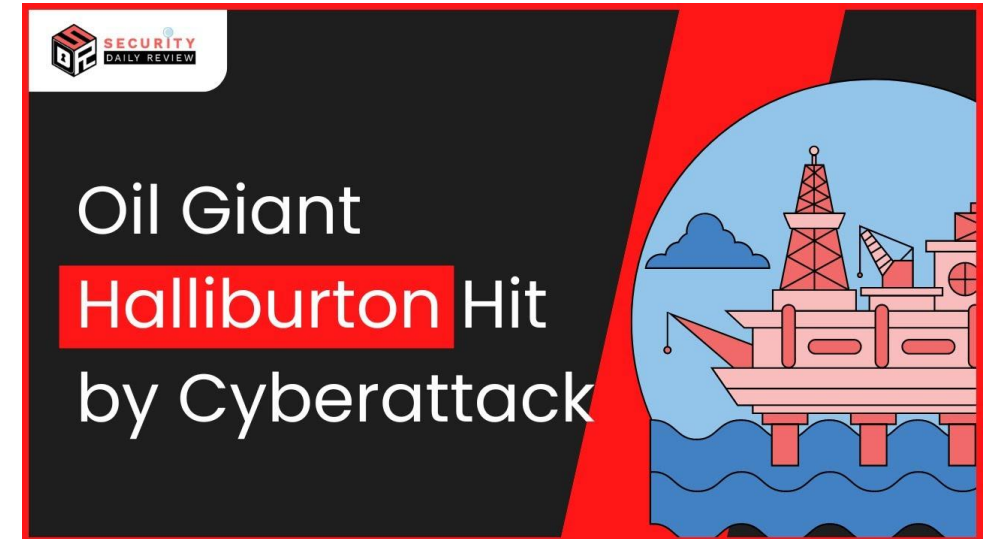
Ending remarks



Purpose: This SAFO encourages the development of training and line-operations policies which will ensure that proficiency in manual flight operations is developed and maintained for air carrier pilots.

FAA 2007: Manual Flight Operations Proficiency

CHANGE HEALTHCARE



Require Parts 121, 135, and 91K operators and Part 141 pilot schools to develop and conduct training that incorporates stalls that are fully developed, are unexpected, involve autopilot disengagement, and include airplane-specific features, such as a ref speeds switch (A-10-22);

US CONGRESS.GOV AVIATION SAFETY: ONE YEAR AFTER THE CRASH OF FLIGHT 3407