Beyond Code Generation

LLM-supported Exploration of the Program Design Space

J.D. Zamfirescu-Pereira • EPIC Advance • October 7, 2024

Joint work with **Eunice Jun • Michael Terry • Qian Yang • Björn Hartmann**

Under review, please don't share!

What's the relationship between these variables?

Oh wait, there's data missing!

Let's consider these other variables instead?

Can I triangulate to this relationship of interest?

Who are the main characters?

Does this fit with the storyline’s goals?

How do these characters interact?

Playtesting, balancing, storytelling.

What's the primary educational goal?

How do I maintain my child's attention & focus?

How cute are these characters???

Is this storyline meaningfully engaging?

Viewing **Programming** as a **Design Activity**

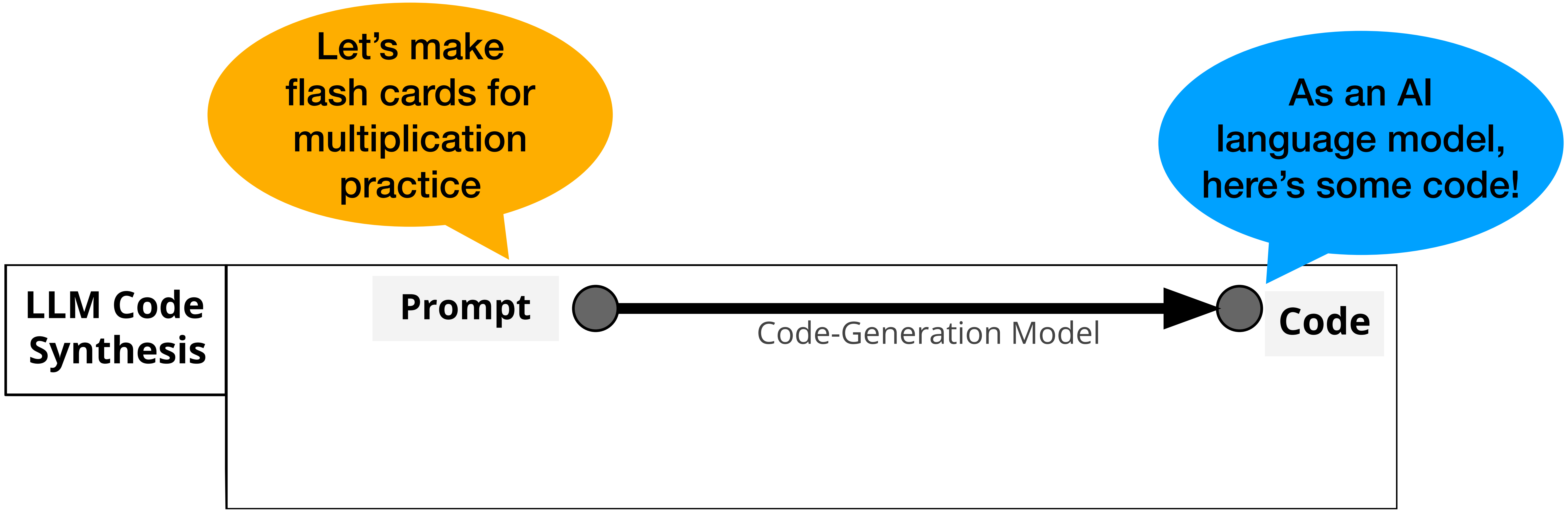
GITHUB COPILOT: CHAT
parse_expenses.py
addresses.rb
sentiments.ts
GitHub Copilot
Ask a question or type '/' for commands

Hi @monalisa, how can I help you?

I'm powered by AI, so surprises and mistakes are possible. Make sure to verify any generated code or suggestions, and share feedback so that we can learn and improve.

```python
import datetime
```

30
25
20
15
10
5
0
y
−1
0
1
2
3
4
5
6
7
8
9
10
x
•••
Python
Markdown
⌘ + K to switch modes
Edit ^
How would you like to edit this cell?



Let’s make flash cards for multiplication practice

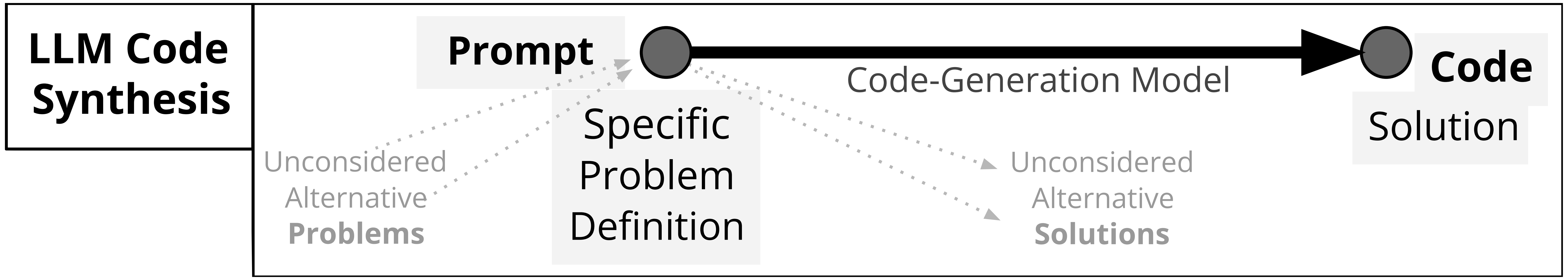
As an AI language model, here’s some code!

LLM Code Synthesis

Prompt

Code-Generation Model

Code



LLM Code Synthesis

Prompt

Specific Problem Definition

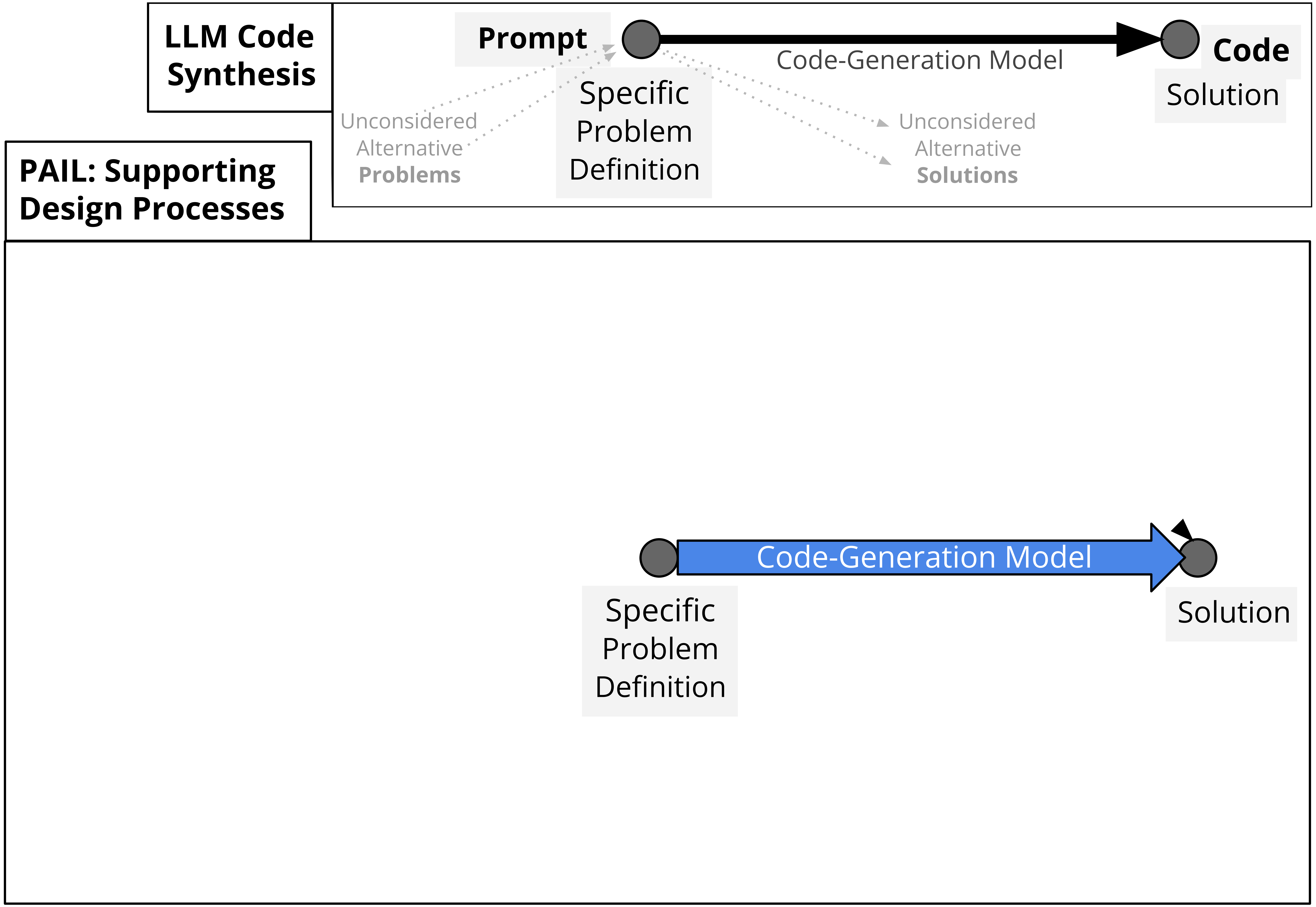
Code-Generation Model

Code

Solution

Unconsidered Alternative **Problems**

Unconsidered Alternative **Solutions**



LLM Code Synthesis

Prompt

Code-Generation Model

Code

Solution

Specific Problem Definition

Unconsidered Alternative **Problems**

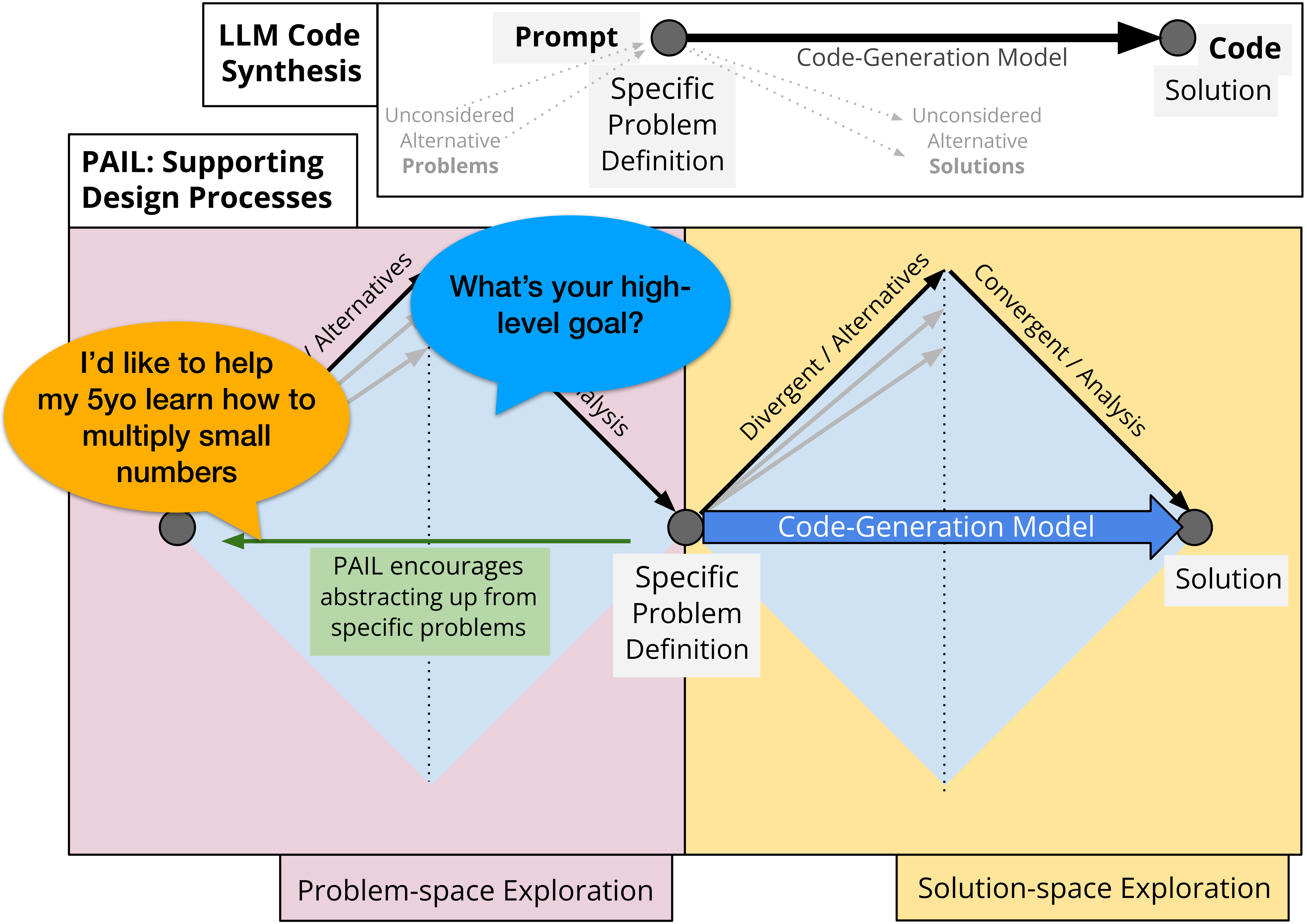
Unconsidered Alternative **Solutions**

PAIL: Supporting Design Processes

Code-Generation Model

Specific Problem Definition

Solution



LLM Code Synthesis

Prompt

Code-Generation Model

Code

Solution

Specific Problem Definition

Unconsidered Alternative **Problems**

Unconsidered Alternative **Solutions**

PAIL: Supporting Design Processes

What's your high-level goal?

I'd like to help my 5yo learn how to multiply small numbers

PAIL encourages abstracting up from specific problems

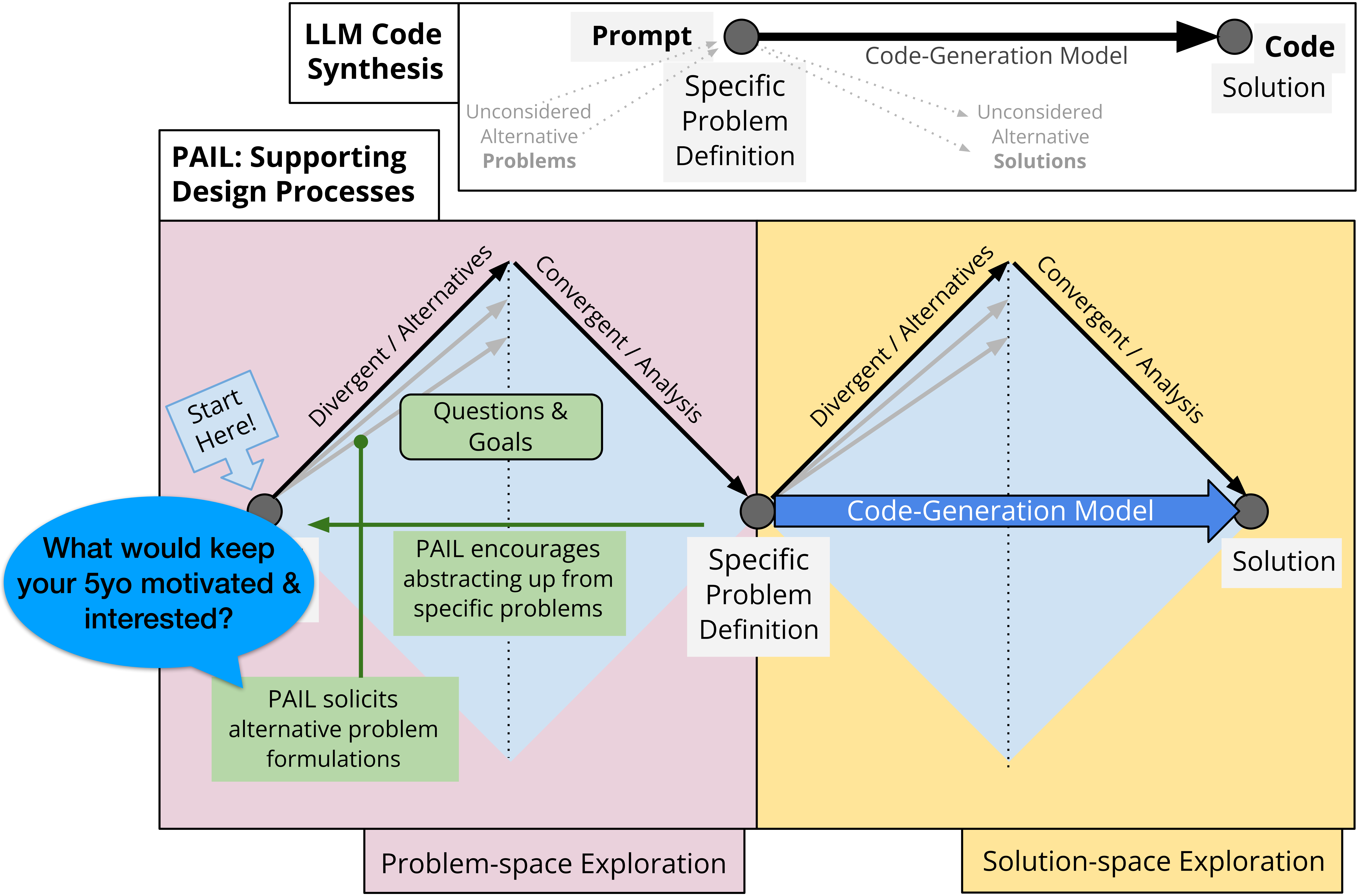
Specific Problem Definition

Code-Generation Model

Solution

Problem-space Exploration

Solution-space Exploration



LLM Code Synthesis

Prompt

Specific Problem Definition

Code-Generation Model

Code

Solution

Unconsidered Alternative **Problems**

Unconsidered Alternative **Solutions**

PAIL: Supporting Design Processes

Start Here!

Divergent / Alternatives

Convergent / Analysis

Questions & Goals

PAIL encourages abstracting up from specific problems

PAIL solicits alternative problem formulations

What would keep your 5yo motivated & interested?

Specific Problem Definition

Code-Generation Model

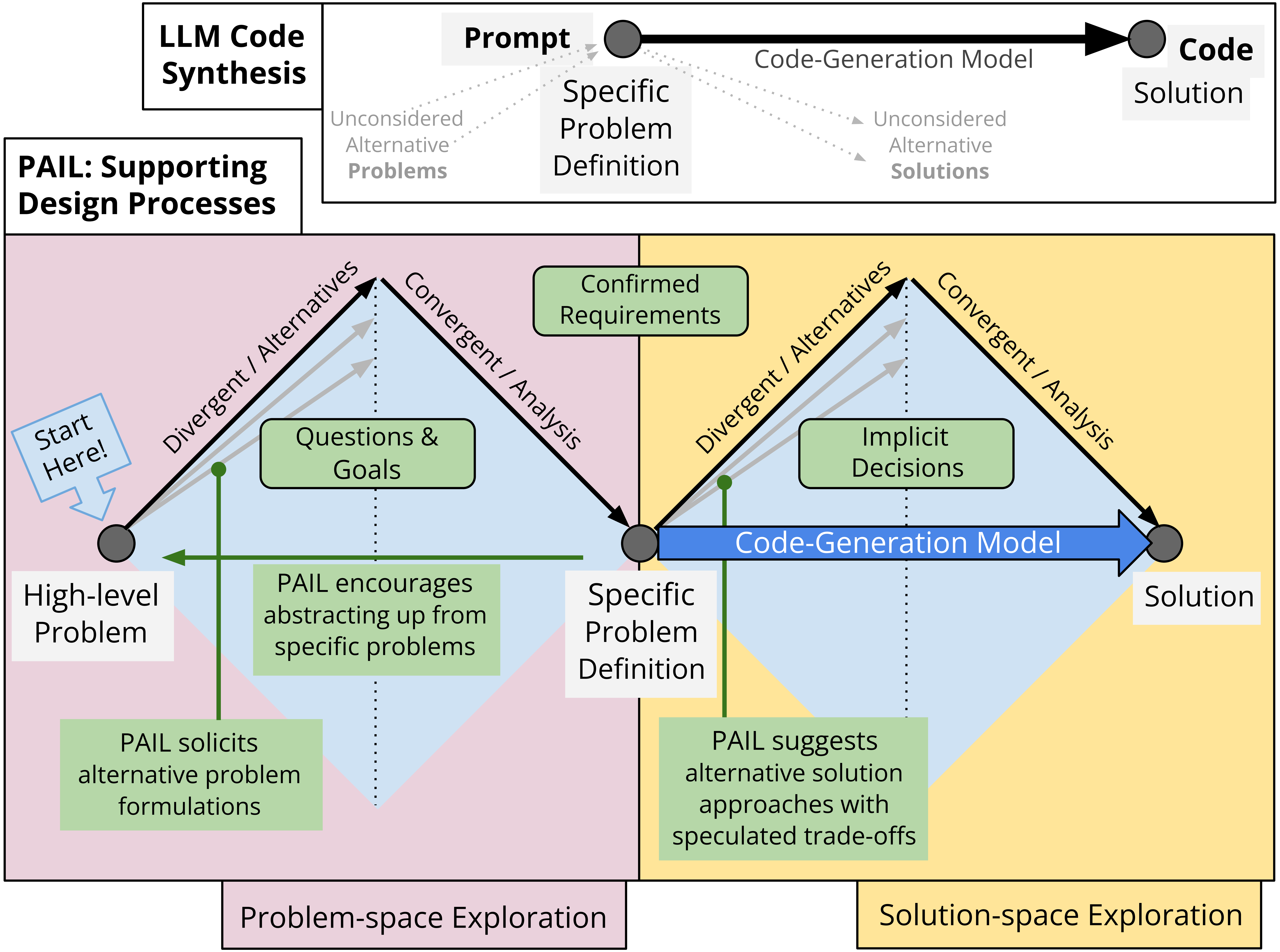
Solution

Divergent / Alternatives

Convergent / Analysis

Problem-space Exploration

Solution-space Exploration


LLM Code Synthesis
Prompt
Code-Generation Model
Code
Solution
Unconsidered Alternative Problems
Specific Problem Definition
Unconsidered Alternative Solutions
PAIL: Supporting Design Processes
Start Here!
Divergent / Alternatives
Convergent / Analysis
Confirmed Requirements
Divergent / Alternatives
Convergent / Analysis
Questions & Goals
Implicit Decisions
High-level Problem
PAIL encourages abstracting up from specific problems
Specific Problem Definition
Code-Generation Model
Solution
PAIL solicits alternative problem formulations
PAIL suggests alternative solution approaches with speculated trade-offs
Problem-space Exploration
Solution-space Exploration

Questions

- How do we encourage and support broader exploration of the design space?
  - Generate alternatives?
  - Better support for iteration?
- *Research-through-Design* point solution:
  - Is explicit design support useful? How & when?
  - What challenges emerge?

PAIL

PAIL • Project
multiply
Show changes since
× Show latest
Console
Type a message...
Status: ready

PAIL • Project
multiply
Show changes since
× Show latest
Console
Type a message...
Status: ready

Let's make a game to help a child learn multiplication

That sounds like a great idea! Let's start by defining the goals and desired outcomes for this game.

Goals and Desired Outcomes

1. **Educational Goal**: Help children practice and improve their multiplication skills.
2. **Engagement Goal**: Make the learning process fun and interactive to keep the child engaged.
3. **Usability Goal**: Ensure the game is easy to understand and play for children.

Target Users and Needs

1. **Primary Users**: Children aged 6-10 who are learning multiplication.
2. **Secondary Users**: Parents and teachers who might use the game as a teaching tool.

Needs

1. **Educational Content**: The game should present multiplication problems for the child to solve.
2. **Feedback Mechanism**: Provide immediate feedback on whether the answer is correct or incorrect.
3. **Progress Tracking**: Optionally, track the child's progress over time.
4. **Engagement**: Use visuals, sounds, and possibly rewards to keep the child motivated.

Does this align with your vision for the game? Would you like to add or modify any of these points?

PAIL · Project
multiply
Show changes since
× Show lates
Type a message...
Status: ready

Design Questions & Goals

- 👉 What specific multiplication skills should the game focus on? (e.g., single-digit, double-digit)
  - *Maybe:* Single-digit multiplication? Try
- What kind of feedback mechanism will be most effective for children?
- How can we make the game engaging and fun for children?

+ Add

Confirmed Requirements

- The game should help children practice and improve their multiplication skills.
- 👉 The game should be fun and interactive to keep the child engaged.
  - *Maybe:* The game should include a storyline or adventure? Try
- The game should be easy to understand and play for children.

+ Add

ign Questions & Goals

What specific multiplication skills should the game focus on? (e.g., gle-digit, double-digit)
Maybe: Single-digit multiplication? Try
at kind of feedback mechanism will be most effective for children?
v can we make the game engaging and fun for children?
Add

firmed Requirements

game should help children practice and improve their ltiplication skills.
The game should be fun and interactive to keep the child engaged.
Maybe: The game should include a storyline or adventure? Try
game should be easy to understand and play for children.
Add

licit Decisions

Add

ful Abstractions

Player: the child who is playing the game.
Maybe: The AI Tutor: an in-game character providing guidance? Try
Multiplication Problems: the questions presented to the player to ve.
dback Mechanism: the system that provides immediate feedback the player's answers.
Add

PAIL · Project
multiply
Show changes since
× Show lates
1

Design Questions & Goals

👉 What specific multiplication skills should the game focus on? (e.g., single-digit, double-digit)
- *Maybe:* Single-digit multiplication? Try

What kind of … ildren?

How can we

Simpler for younger children; may not challenge older children enough.

+ Add

Confirmed Requirements

The game should help children practice and improve their multiplication skills.

👉 The game should be fun and interactive to keep the child engaged.
- *Maybe:* The game should include a storyline or adventure? Try

The game should be easy to understand and play for children.

+ Add

ign Questions & Goals

What specific multiplication skills should the game focus on? (e.g., gle-digit, double-digit)
Maybe: Single-digit multiplication? Try
at kind of … ildren?
v can we
Simpler for younger children; may not challenge older children enough.
Add

firmed Requirements

game should help children practice and improve their ltiplication skills.
The game should be fun and interactive to keep the child engaged.
Maybe: The game should include a storyline or adventure? Try
game should be easy to understand and play for children.
Add

licit Decisions

Add

ful Abstractions

Player: the child who is playing the game.
Maybe: The AI Tutor: an in-game character providing guidance? Try
Multiplication Problems: the questions presented to the player to ve.
dback Mechanism: the system that provides immediate feedback the player's answers.

Add

PAIL • Project
multiply
Show changes since
× Show latest
1
Console
Revert
Type a message...
Status: ready
+ Add
Try

In Design Questions & Goals: What specific multiplication skills should the game focus on? (e.g., single-digit, double-digit) → Single-digit multiplication

Great, focusing on single-digit multiplication will help narrow down the scope and make the game more manageable for younger children.

Updated Design Questions & Goals

- **Single-digit multiplication**: The game will focus on helping children practice and improve their single-digit multiplication skills.

Confirmed Requirements

- The game should help children practice and improve their single-digit multiplication skills.
- The game should be fun and interactive to keep the child engaged.
- The game should be easy to understand and play for children.

Next Steps

1. **Game Mechanics**: Let's define how the game will work. For example, will it be a quiz format, a timed challenge, or something else?
2. **Feedback Mechanism**: Decide how to provide feedback. Should it be visual (e.g., green for correct, red for incorrect), auditory, or both?
3. **Engagement**: Think about how to make the game engaging. This could include animations, sounds, and rewards.

Do you have any preferences for the game mechanics, feedback mechanism, or engagement features?

Exploring: Single-digit multiplication

Design Questions & Goals

👉 ~~What specific multiplication skills should the game focus on? (e.g., single-digit, double-digit)~~

👉 What kind of feedback mechanism will be most effective for children?

How can we make the game engaging and fun for children?

Confirmed Requirements

~~The game should help children practice and improve their multiplication skills.~~

The game should help children practice and improve their single-digit multiplication skills.

👉 The game should be fun and interactive to keep the child engaged.
- *Maybe:* The game should include a storyline or adventure?

The game should be easy to understand and play for children.

Implicit Decisions

Useful Abstractions

The Player: the child who is playing the game.
- *Maybe:* The AI Tutor: an in-game character providing guidance?

👉 ~~Multiplication Problems: the questions presented to the player to solve.~~

Multiplication Problems: single-digit multiplication questions presented to the player to solve.

Feedback Mechanism: the system that provides immediate feedback on the player's answers.

PAIL • Project
multiply
Show changes since
× Show latest
1
Console

advance.

3. **Resource Management**:
   - **Mechanic**: The player earns resources (e.g., seeds, water) for each correct answer, which are needed to plant and grow the crops.
   - **Example**: Correct answers provide seeds, and additional challenges might require watering the plants by solving more problems.
4. **Visual and Auditory Feedback**:
   - **Mechanic**: Provide immediate feedback with visual cues (e.g., plants growing) and sounds (e.g., a chime for correct answers).
   - **Example**: When the player plants the correct number of seeds, the plants grow and flourish, accompanied by a cheerful sound.

Next Steps

1. **Storyline Integration**: Define a simple storyline where the player is a farmer or gardener working to grow a beautiful garden.
2. **Game Design**: Sketch out the levels, starting with simple plots and gradually increasing complexity.
3. **Feedback Mechanism**: Design visual and auditory feedback to make the game rewarding and engaging.

Would you like to proceed with this planting mechanic? If so, we can start by designing the initial setup and the first level.

Exploring: The game should include a storyline or adventure

Design Questions & Goals

👉 What kind of feedback mechanism will be most effective for children?
- *Maybe:* Immediate feedback with explanations? Try

How can we make the game engaging and fun for children?

What mechanics can make multiplication integral to the gameplay?

Confirmed Requirements

The game should help children practice and improve their single-digit multiplication skills.
- *Maybe:* The game should help children practice and improve their math skills through real-world scenarios? Try

👉 The game should include a storyline or adventure to keep the child engaged.

The game should be easy to understand and play for children.
- *Maybe:* The game should have adjustable difficulty levels? Try

Implicit Decisions

Useful Abstractions

The Player: the child who is playing the game.
- *Maybe:* The AI Tutor: an in-game character providing guidance? Try

👉 Multiplication Problems: single-digit multiplication questions presented to the player to solve.

Feedback Mechanism: the system that provides immediate feedback on the player's answers.

PAIL • Project
multiply
Show changes since
× Show latest
1
Console
Revert
Type a message...
Status: ready
+ Add
Try

3. **Resource Management**:
   - **Mechanic**: The player earns resources (e.g., seeds, water) for each correct answer, which are needed to plant and grow the crops.
   - **Example**: Correct answers provide seeds, and additional challenges might require watering the plants by solving more problems.
4. **Visual and Auditory Feedback**:
   - **Mechanic**: Provide immediate feedback with visual cues (e.g., plants growing) and sounds (e.g., a chime for correct answers).
   - **Example**: When the player plants the correct number of seeds, the plants grow and flourish, accompanied by a cheerful sound.

Next Steps

1. **Storyline Integration**: Define a simple storyline where the player is a farmer or gardener working to grow a beautiful garden.
2. **Game Design**: Sketch out the levels, starting with simple plots and gradually increasing complexity.
3. **Feedback Mechanism**: Design visual and auditory feedback to make the game rewarding and engaging.

Would you like to proceed with this planting mechanic? If so, we can start by designing the initial setup and the first level.

Exploring: The game should include a storyline or adventure

Design Questions & Goals

👉 What kind of feedback mechanism will be most effective for children?
- *Maybe:* Immediate feedback with explanations? Try

How can we make the game engaging and fun for children?

What mechanics can make multiplication integral to the gameplay?

How can we integrate the planting mechanic into the storyline?

Confirmed Requirements

The game should help children practice and improve their single-digit multiplication skills.
- *Maybe:* The game should help children practice and improve their math skills through real-world scenarios? Try

👉 The game should include a storyline or adventure to keep the child engaged.

The game should be easy to understand and play for children.
- *Maybe:* The game should have adjustable difficulty levels? Try

Implicit Decisions

The game will use a planting mechanic where players calculate the area of a rectangular plot to determine the number of seeds to plant.

The game will provide immediate visual and auditory feedback for correct answers.

The game will have level progression with increasing plot sizes and complexity.

Useful Abstractions

The Player: the child who is playing the game.
- *Maybe:* The AI Tutor: an in-game character providing guidance? Try

👉 Multiplication Problems: single-digit multiplication questions presented to the player to solve.

Feedback Mechanism: the system that provides immediate feedback on the player's answers.

Plot of Land: a rectangular area where the player plants seeds based on multiplication calculations.

Resources: items like seeds and water that the player earns and uses to grow plants.

PAIL • Project
multiply
Show changes since
× Show latest
Plant 72 pla
Console
Revert
Type a message...
Status: ready

```diff
+let startX, startY, endX, endY;
+let dragging = false;
+let requiredPlants;
+
 function setup() {
   createCanvas(800, 600);
   textSize(32);
   textAlign(CENTER, CENTER);
   generateProblem();
 }
-
-let length, width, answer;
 
 function generateProblem() {
-  length = floor(random(1, 10));
-  width = floor(random(1, 10));
-  answer = length * width;
+  requiredPlants = floor(random(1, 10)) * floor(ran
 }
 
 function draw() {
   background(200);
   fill(0);
-  text(`Plot size: ${length} x ${width}`, width / 2
-  text('How many seeds to plant?', width / 2, heigh
+  text(`Plant ${requiredPlants} plants`, width / 2,
+  if (dragging) {
+    fill(100, 200, 100, 150);
+    rect(startX, startY, mouseX - startX, mouseY -
+    let plotWidth = abs(mouseX - startX) / 40;
+    let plotHeight = abs(mouseY - startY) / 40;
+    text(`${floor(plotWidth)} x ${floor(plotHeight)
+  }
 }
 
-function keyPressed() {
-  if (keyCode === ENTER) {
-    let userAnswer = int(prompt('Enter your answer:
-    if (userAnswer === answer) {
+function mousePressed() {
+  startX = mouseX;
+  startY = mouseY;
+  dragging = true;
+}
+
+function mouseReleased() {
+  dragging = false;
+  endX = mouseX;
+  endY = mouseY;
+  let plotWidth = abs(endX - startX) / 40;
+  let plotHeight = abs(endY - startY) / 40;
+  let plants = floor(plotWidth) * floor(plotHeight)
+  if (plants === requiredPlants) {
     alert('Correct!');
     generateProblem();
```

```
      alert('Try again!
    }
  }
}

```

```

function mouseReleased(
  dragging = false;
  endX = mouseX;
  endY = mouseY;
  let plotWidth = abs(e
  let plotHeight = abs(
  let plants = floor(pl
  if (plants === requir
    alert('Correct!');
    generateProblem();
  } else {
    alert('Try again!')
  }
}
```

The initial drag-and-drop interface for creating plots is now implemented. Here's what it does:

1. **Drag to Plant**: The player can drag to create a rectangular plot of land.
2. **Visual Feedback**: The plot being created is highlighted, and the dimensions and total number of plants are displayed.
3. **Validation**: When the player releases the mouse button, the plot dimensions are checked against the required number of plants, and feedback is provided.

Next Steps

1. **Enhance Visual Feedback**: Improve the visual representation of the plot and the plants.
2. **Storyline Integration**: Add elements of the storyline to make the game more engaging.
3. **Level Progression**: Implement multiple levels with increasing complexity.

Would you like to proceed with enhancing the visual feedback, or do you have any other specific features in mind?

Exploring: The game should include a storyline or adventure

Design Questions & Goals

👉 What kind of feedback mechanism will be most effective for children?
- *Maybe:* Immediate feedback with explanations? Try

How can we make the game engaging and fun for children?

What mechanics can make multiplication integral to the gameplay?

How can we integrate the planting mechanic into the storyline?

+ Add

Confirmed Requirements

The game should help children practice and improve their single-digit multiplication skills.
- *Maybe:* The game should help children practice and improve their math skills through real-world scenarios? Try

👉 The game should include a storyline or adventure to keep the child engaged.

The game should be easy to understand and play for children.
- *Maybe:* The game should have adjustable difficulty levels? Try

The game will use a planting mechanic where players calculate the area of a rectangular plot to determine the number of seeds to plant.

The game will provide immediate visual and auditory feedback for correct answers.

The game will have level progression with increasing plot sizes and complexity.
- *Maybe:* The game will have a story mode with narrative-driven levels? Try

+ Add

Implicit Decisions

~~👉 The game will use a planting mechanic where players calculate the area of a rectangular plot to determine the number of seeds to plant.~~

~~The game will provide immediate visual and auditory feedback for correct answers.~~

~~The game will have level progression with increasing plot sizes and complexity.~~

👉 The game will use a drag-and-drop interface for creating plots.

The game will highlight the plot being created and display the dimensions and total number of plants.

The game will validate the plot dimensions against the required number of plants and provide feedback.

+ Add

Useful Abstractions

Study: Tasks

We asked 10 participants to:

1. Create an interactive feature to go along with an article about the impacts of air conditioning on migration patterns in high-average-temperature areas.

2. Create a game that helps a young child learn how to read or multiply.

3. Create a simulation to show the effects of medical overtesting, e.g., recommending screening for specific rare diseases for more people.

Findings

1. Design support helps broaden design space exploration 🎉
    - Activities span *define*, *discover*, and *develop* phases of design process.
2. Information quantity overwhelming.
    - Focused attention spread thin across code, chat, design panel.
    - Awareness (of code, design), once lost, costly to regain.
3. Expertise matters (still!)—intersects in unexpected ways

Broadening Design Space Exploration

- When prompted to consider the user and their goals…
  - …most participants did, unsurprisingly.

    > “ *This really broke it down in an interesting way, who is your target audience.*

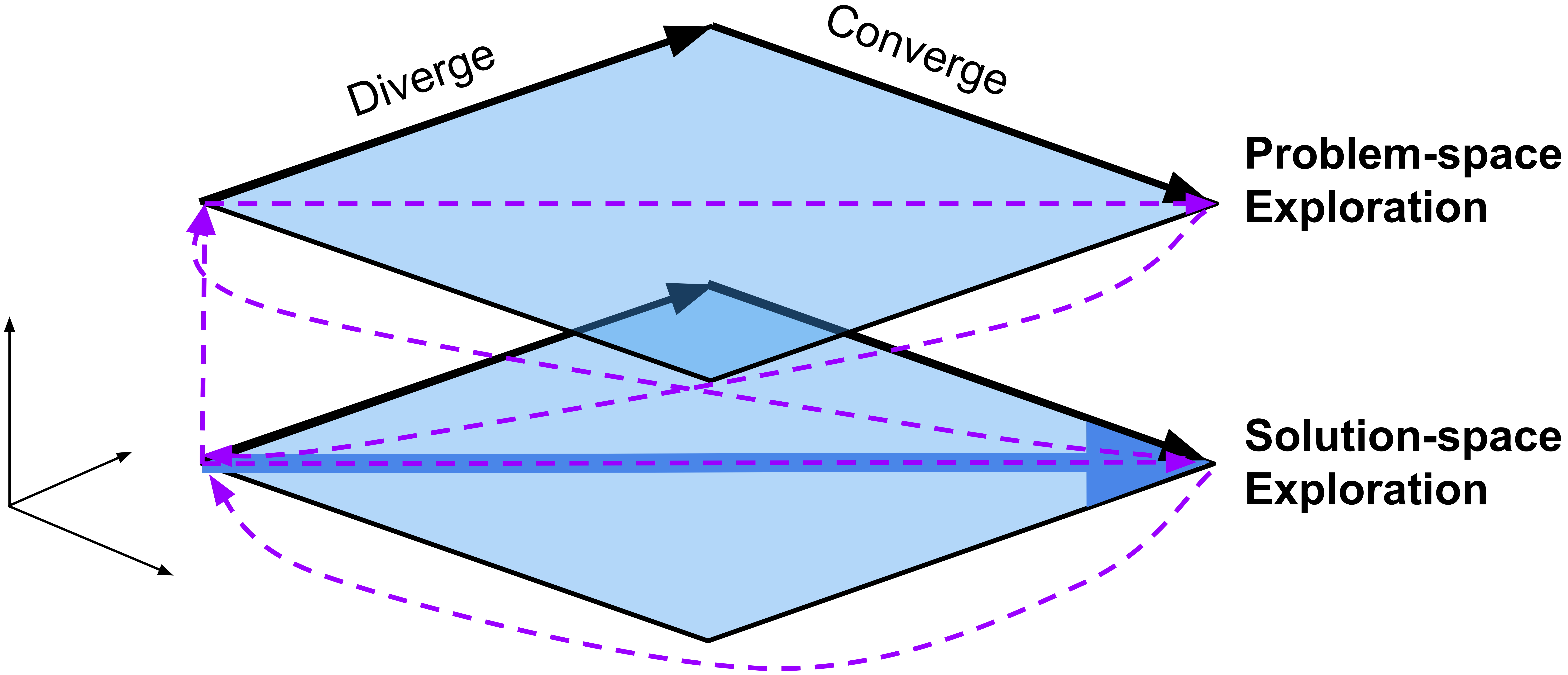
  - …but some **didn’t**, and demanded point solutions up front.

    > “ *I’m not really interested in what the system might tell me, the only thing I’d want to do is try it with [my child]*

Sketches, Prototypes: A State of “Flow”

- Rapid iteration across code, chat, TRY-ing alternatives in the design panel.
  - Activities span *define*, *discover*, and *develop* phases.
  - One user action might serve **multiple goals**:
    - validating parts the defined problem…
    - …simultaneously progressing towards prototype development…
    - …and revealing some new, unanticipated end-user need
- For experts, looking for a “lack of surprises” in code & design updates

Sketches, Prototypes: A State of “Flow”



Challenges

- Too much information:

  > “ Each time I click on something here [in the design panel] [...] I'm like “Ahhh! What part of this is important?”

- Experts found it easier to maintain and regain awareness over code…
  - …but none actually maintained it consistently.
  - *Domain* and *programming* expertise impacted perception of “surprises”

- Does all this AI-generated information **actually *inform*?** Or **mostly *overwhelm*?**

Open Questions & Future Work

- Trade-offs around providing **all the information** and limiting to **AI-identified salient information:**
  - How much initiative should automated systems take in setting direction?
  - How forceful should they be in, e.g., encouraging consideration of end-user needs?

- **How do we manage and/or direct users' scarce attention?**
  - What cues can we provide users to ensure they retain control & agency?