＊すぐに授業ができちゃうセット＊

プログラミング教育
単元プランシート
〔第４学年〕

第４学年　プログラミング教育授業シート

【単元名】	「ライトボット」でプログラミングー！
時　数	全６時間（総合的な学習の時間）

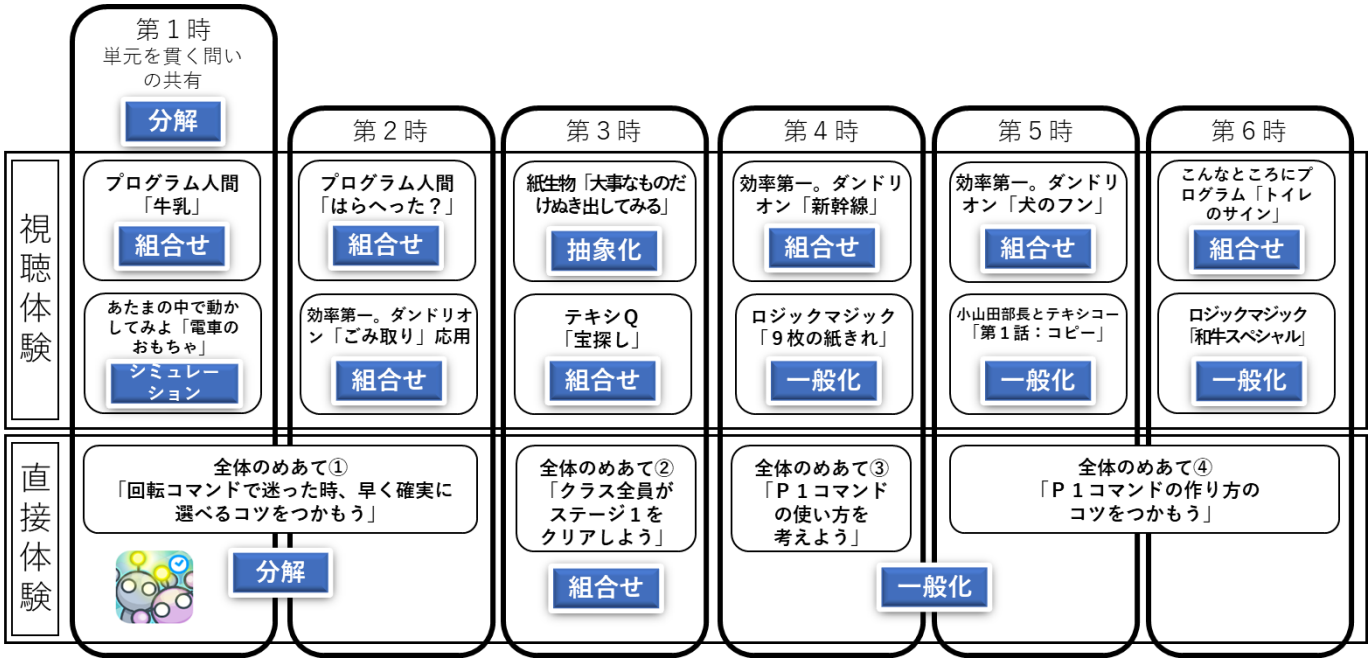
【プログラミング教育のねらい】

知識及び技能	思考力、判断力、表現力等	学びに向かう力、人間性
身近な生活でコンピュータが活用されていることや問題の解決には必要な手順があることに気付くこと。	発達の段階に即して、「プログラミング的思考（自分が意図する一連の活動を実現するために、どのような動きの組合せが必要であり、一つ一つの動きに対応した記号を、どのように組み合わせたらいいのか、記号の組合せをどのように改善していけば、より意図した活動に近づくのか、といったことを論理的に考えていく力）」を育成すること。	発達の段階に即して、コンピュータの働きを、よりよい人生や社会づくりに生かそうとする態度を涵養すること。

働かせる見方・考え方：
プログラミング的思考

分解：小さく分けて考える　**組合せ**：手順の組み合わせを考える
一般化：パターンを見つける　**抽象化**：大事なものだけ抜き出して考える
シミュレーション：頭の中で手順をたどる

【単元構想】



視聴体験で活用した「テキシコー」は以下の通りです。

第１時	第２時	第３時	第４時	第５時	第６時
＃４　Scene６	＃５　Scene６	＃４　Scene４	＃７　Scene４	＃４　Scene３	＃５　Scene５
＃１　Scene １・２	＃１　Scene３	＃７　Scene３	＃１　Scene ４・５	＃６　Scene６	＃10　Scene ４・５・６

【活用コンテンツの紹介】

NHK for School「テキシコー」

「テキシコー」は、魅力的な映像やアニメーションを使って、プログラミング的思考（テキシコー）の面白さを伝え、その思考を育むプログラミング教育番組です。

https://www.nhk.or.jp/school/sougou/texico/

「テキシコー」では、コンピューテーショナルシンキングという概念やプログラミング的思考についての先行研究などを参考に「プログラミング的思考」を５つに分解・整理しています。【掲示物②　テキシコー】

←「テキシコー」にある配信リストから、映像やアニメーションを選んで、視聴することができます。成果物のなかに「テキシコー」の内容一覧も作成しています。ぜひ、ご参照ください。

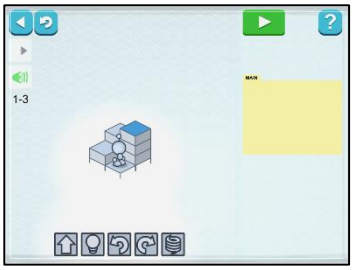
プログラミング学習アプリ「Lightbot」

「Lightbot」は、無料でダウンロードすることができます。

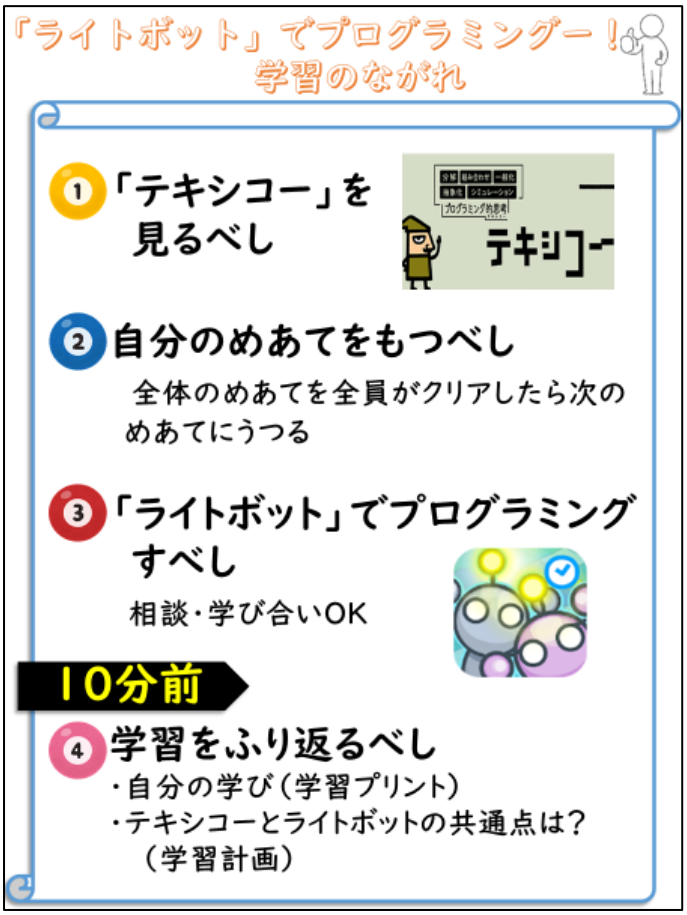
https://www.lightbot.com/flash.html

コマンドを組み合わせてロボットを動かし、青いタイルに電気をつけていくゲームです。ステージ１.基本（８コース）、ステージ２.プロシージャ（６コース）、ステージ３.ループ（６コース）の 20 コースがあります。

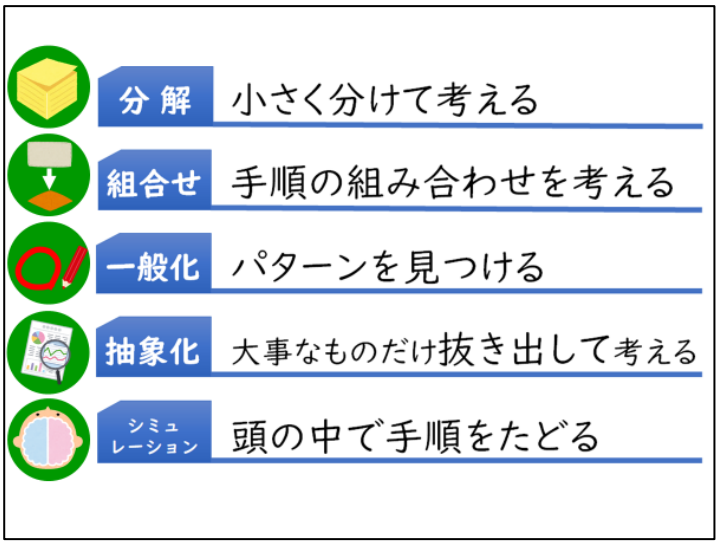
＊ダウンロードの方法は https://any-planet.com/?p=3267 を参照ください。



【掲示物①　学習の流れ】



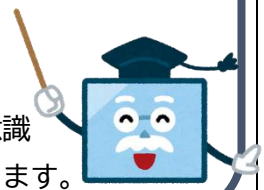
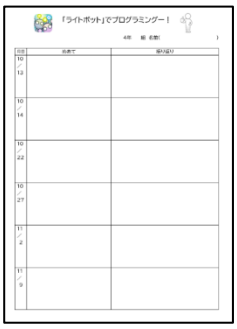
【掲示物②　テキシコー（プログラミング的思考）】



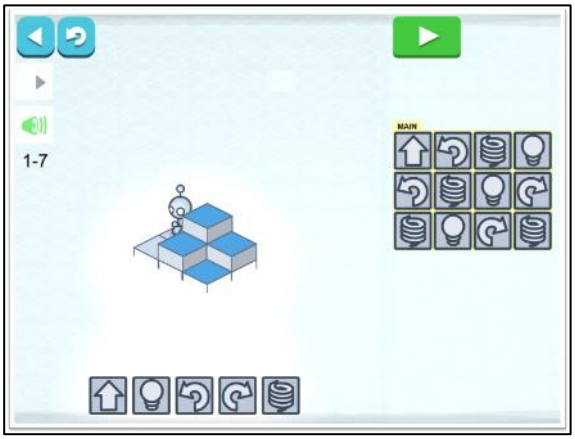
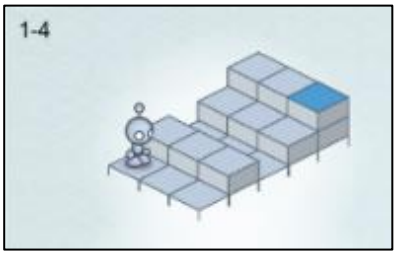
総合的な学習の時間（全６時間）の学習計画〔第１時〕

段階	学習活動と子どもの反応（◎）	学習を支える教師の働きかけ	プログラミング的思考
導入	１　単元のめあてと学習計画を確認する ◎問題の解決の仕方にはいろいろあるんだな。 ◎どんな問題にであっても、解決できるようになりたいな。	・問題には、「正解が１つのもの」（例：1+1）と「いくつかの選択肢から適したものを選ぶもの」（例：忘れ物をした時どうする？）があることを知らせ、問題解決の仕方について単元を通して考えていくという問題意識をもてるようにする。	
	単元のめあて　問題をどのように解決すればよいか考えよう。		
展開	２　「テキシコー」を視聴する ◎「プログラミング人間」って、１つ１つ命令して、その通りに動くんだ。 ◎「シミュレーション」は、予想していた動きとは違ってびっくりした。 ３　「ライトボット」を使って、プログラミングを体験する	・【掲示物①　学習の流れ】を提示し、説明を加えながらやってみることで、自分のめあてをもって各自学習を進め、振り返るというサイクルをつかませ、次時から主体的に取り組めるようにする。 ・映像やアニメーションと結びつけながら用語（プログラム＝命令、プログラミング＝プログラムをつくること、プログラミング的思考＝よいプログラムにするための考え方）を説明することで、プログラミングの意味をつかめるようにする。	組合せ シミュレーション
	めあて⓪　できだけたくさんクリアしよう。		
	  ◎やり方を説明してくれるから、自分のペースで進められる。 ◎クリアして進んでいくのは楽しい。 ◎回転コマンドで左右を間違ってしまう。	・立ち歩きＯＫ相談ＯＫ、全員が達成できたら次のめあてにレベルアップする、というルールを確認することで、学び合いを促す。 ・アプリに進め方の説明があるので、授業者は説明せずに「やってごらん」と子どもに委ねることで、分からない時は友だちに聞こうとし始めるなど、協働的な学びへとつなげることができる。	分解 組合せ
まとめ	４　本時の学習を振り返る ◎ライトボットを動かす時に、前、右・左回転、ジャンプと１つ１つ命令していくところと、「プログラミング人間」の命令が同じ。 ◎「シミュレーション」をつかって、ロボットがどんな動きをするか、頭の中で動かしてみた。	・「テキシコーとライトボットの共通点は？」という振り返りの観点を与えることで、視聴体験（「テキシコー」の視聴）と直接体験（プログラミング体験）を関連づけられるようにする。 ・学習計画を印刷した画用紙に毎時間の学習プリントを貼り付けたポートフォリオを作成することで、既習を振り返りながら見通しをもって学習に臨めるようにする。	

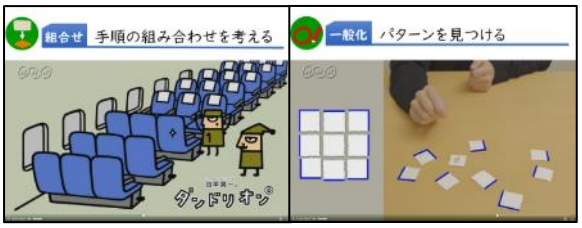
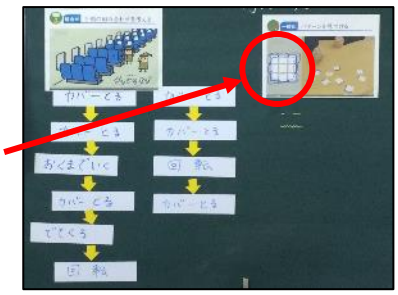
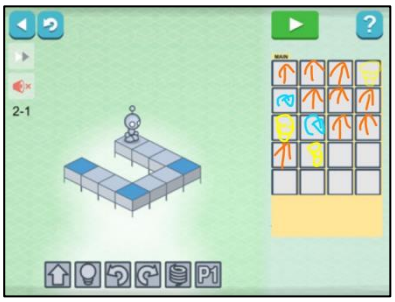
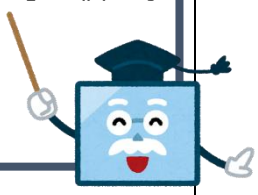
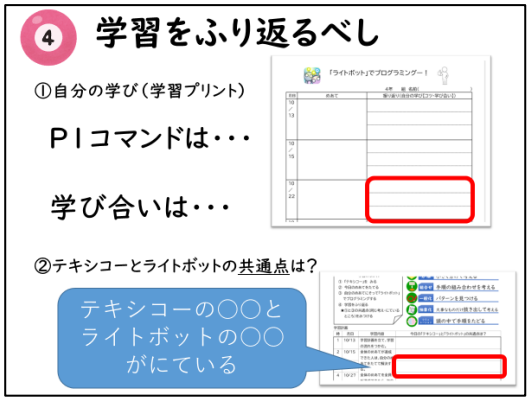
〔第２時〕

段階	学習活動と子どもの反応（◎）	学習を支える教師の働きかけ	プログラミング的思考
導入	１　「テキシコー」を視聴する ◎このプログラムだと「プログラム人間」はどう動くのかな。 ◎「組合せ」って、どんな順番ですればいいかを考えることだ。	・映像やアニメーションに合わせたカードやフローチャートを提示することで、プログラミング的思考を活用する方法に慣れることができるようにする。 ・映像やアニメーションを一時停止することで子どもが予想したり考えたりできる時間を確保する。	分解 組合せ
展開	２　自分のめあてをもつ **めあて①　回転コマンドで迷った時、早く確実に選べるコツをつかもう。** ◎私と同じところで困っていた人がたくさんいたんだ。 ◎コツはつかんだから、今日は、ステージ１をクリアすることをめあてにしよう。 ３　「ライトボット」をつかって、プログラミングを体験する ◎頭の中でロボットを矢印の方に動かして確かめたよ。 ◎あっているかどうか再生して動きを確かめればいいよ。 ◎自分では思いつかないから、〇〇さんに聞いてみよう。	・回転コマンドで困ったという子どもの感想を紹介し、問題意識を共有してから、めあてを提示する。 ・めあて①が達成できている子どもには、自分なりのめあてを記入させることで、問題解決の手順を繰り返し体験できるようにする。 「Lightbot」では、１つ１つのコマンドに分けて考えることが、「分解」になります。ということは、子どもたちは常に「分解」を使って考えているはずです。 無意識に行っていたことを意味づけてあげることで意識的に活用できるようになります。 	分解 組合せ
まとめ	４　本時の学習を振り返る ◎「ライトボット」も「ダンドリオン」みたいに、いろいろ試して、よい方法をみつければいい。 ◎「プログラミング人間」みたいに、命令を１つずつ分けているから、その１つずつの命令ごとに確かめるといい。	 ・学習プリントの振り返り欄に見つけたコツを書かせることで、子どもたちの問題解決の状況を把握する。 ・コツには唯一の正解はないので、子どもたちの発見を認め、交流により、よりよいコツ（解決法）の共有を図る。 ・めあてが全員達成できていたらめあて②に進み、まだの場合はめあて①を継続する。	

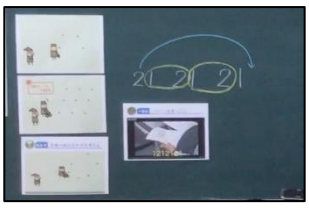
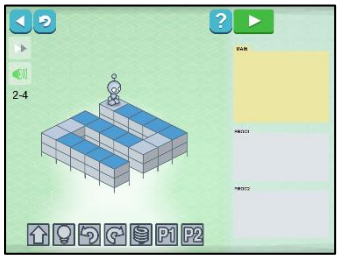
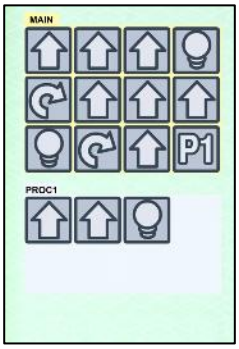
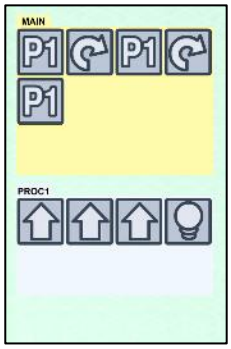
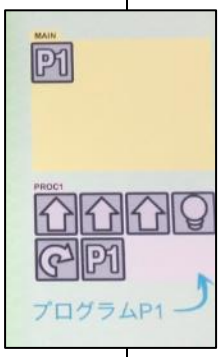
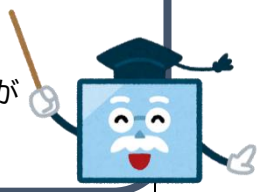
〔第３時〕＊めあて①の達成がまだの場合はめあて②のところをめあて①にする。

段階	学習活動と子どもの反応（◎）	学習を支える教師の働きかけ	プログラミング的思考
導入	１　「テキシコー」を視聴する ◎「ゲゾタラズ」さんの足とか体ではなく磁石だけを取り出して、動きを考えることを「抽象化」というんだな。 ◎いろいろな方法があるけど、簡単で早くできる方法を考えるのは楽しいな。	・視聴の前に、【掲示物②　テキシコー（プログラミング的思考）】を提示し、「今日のテキシコーはどれかな？」と発問することで、映像やアニメーションをプログラミング的思考に着目して観ることができるようにする。 ・めあての達成状況に応じて、全体交流の内容やタイミングを調整することで、子どもが自力解決を図れるようにする。	抽象化 組合せ
展開	２　自分のめあてをもつ めあて②　クラス全員がステージ１をクリアしよう。 ◎もう少しでクリアできそう。 ◎もうクリアしているから、まだの人にコツを教えたいな。 ３　「ライトボット」をつかって、プログラミングを体験する ◎コマンドが入らなくてクリアできなかったコースも、道順を変えたらクリアできた！  	・黒板にネームカードを貼るなど、達成した人が分かるようにすることで、まだの人にアドバイスしたり、クリアしている人に聞きにいったりと協働的な学びを促すことができるようにする。 ステージ１のポイントは２つです。 ①１－４で、何通りかのプログラムを提示し、「道筋（プログラム）が違ってもクリアできる」ことを実感させます。 ②１－７で、「近道（コマンド数の少ないプログラム）をみつける」ことです。  	分解 組合せ
まとめ	４　本時の学習を振り返る ◎「ダンドリオン」と同じで、効率のいいプログラムがあるんだな。 ◎「テキシＱ」で→と←は同じことだから消して手順を減らしていたところと、「ライトボット」で無駄な動きを減らすところが似ている。	・ステージ１を全員クリアできることで、ステージ２から新しく加わるプロシージャの問題を共有できるようにする。	

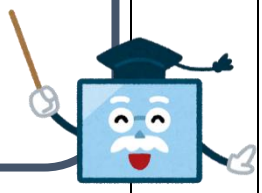
〔第４時〕

段階	学習活動と子どもの反応（◎）	学習を支える教師の働きかけ	プログラミング的思考
導入	１　「テキシコー」を視聴する  ◎もっと早く楽に仕事を進めるには、どんな順序で行えばいいかな。 ◎同じところを丸で囲むとパターンがみつけやすい。	・映像の中の同じところを丸で囲んでみせることで、同じところを括っていくとパターンが見つけやすいことに気づかせる。  ・ステージ２から新しく出てくるプロシージャ（PROC）をどう使うかという問題意識を共有し、めあてにつなぐ。	組合せ 一般化
展開	２　自分のめあてをもつ **めあて③　Ｐ１コマンドの使い方を考えよう。**		
	  ◎プロシージャ１の命令は、Ｐ１コマンド１個で動かせるってことかな。	 ・まずは P1 を使わずにプログラムを考え（左図は「ロイロノート・スクール」を活用。写真の上から手書きで↑を書き入れている）、コマンド数が多すぎて MAIN の枠をはみ出してしまうことを実感させることで P1 のよさや意味に気づかせる。	分解 組合せ 一般化
	３　「ライトボット」をつかって、プログラミングを体験する ◎PROC１にどんなコマンドを入れたらいいのかな。 ◎P1 コマンドで PROC１にワープできるみたい。 ◎繰り返しのある動きをみつけて、PROC１に入れれば、P1 コマンドを繰り返し使える。	「プロシージャ」は、PROC１に入れた複数のコマンドを MAIN の「P１コマンド」１個で呼び出せる手続きのことです。 これがあると、コマンドの数を少なくすることができます。 	
まとめ	４　本時の学習を振り返る ◎「ロジックマジック」もプロシージャも同じパターンのところを丸で囲むと分かりやすい。 ◎「ダンドリオン」で手順を少なくしたことと、「ライトボット」の P1 コマンドでコマンドの数を減らしたことが似ている。	 	

〔第５時〕

段階	学習活動と子どもの反応（◎）	学習を支える教師の働きかけ	プログラミング的思考
導入	１　「テキシコー」を視聴する ◎ページを入れ替えなくても全体で見たら１・２のグループができているから、最初の２を後ろにもっていけばいい。 ◎第４時のように、同じところを丸で囲むとパターンがみつけやすい。	・映像の中の同じところを丸で囲んでみせることで、同じところを括っていくとパターンが見つけやすいことに気づかせる。  ・めあて③が達成できている子どもには、自分なりのめあてを記入させることで、問題解決の手順を繰り返し体験できるようにする。	組合せ 一般化
展開	２　自分のめあてをもつ **めあて③　Ｐ１コマンドの作り方のコツをつかもう。** ◎コマンド数を少なくするためには、P1 コマンドをどう使えばいいのかな。 ◎なんとなくわかるけど、言葉で説明するのが難しい。 ◎P1 コマンドを何回か使えば、コマンドを少なくできるから、同じ動きをみつけて PROC１に入れるのかな。 ３　「ライトボット」をつかって、プログラミングを体験する  ◎PROC１と PROC２の２つがでてきた。２つの動きのパターンがあるのかな。	・プログラムの例  A　ワープ方式  B　パターン方式  上記のプログラムの例は、どれもクリアできます。しかし、A のワープ方式では２－４からはクリアできません。 同じ動きのパターンをみつけて PROC１と PROC２にまとめることが必要になります。 	分解 組合せ 一般化
まとめ	４　本時の学習を振り返る ◎「小山田部長とテキシコー」の１・２というパターンをみつけるのと、「ライトボット」の同じ動きをみつけるのが同じ。	・頭の中だけで考えることが苦手な子どもには、書き込んで考えられるようにライトボットのコースを印刷しておき、いつでも活用できるようにしておく。 ・それぞれのペースで進んでいいことを伝え、クリアできないことによる苦手意識が生じないようにする。	

〔第６時〕

段階	学習活動と子どもの反応（◎）	学習を支える教師の働きかけ	プログラミング的思考
導入	１　「テキシコー」を視聴する ◎プログラミングの考えは、トイレでも使われているんだ。 ◎トランプの置き方と指をさしたトランプのマークの位置が同じところが「パターン」になっている。	・身の回りの生活のなかにプログラムやプログラミング的思考が活用されている例を提示することで、身近にあるプログラミングを探したい、これからもプログラミング的思考を使いたいという気持ちをふくらませることができるようにする。 	組合せ 一般化
展開	２　自分のめあてをもつ **めあて③　Ｐ１コマンドの作り方のコツをつかもう。** ◎コツはつかんだから、今日は、「P2 コマンドの作り方のコツをつかもう」にしよう。 ３　「ライトボット」をつかって、プログラミングを体験する ◎プロシージャ１にどんなコマンドを入れたらいいのかな。 ◎Ｐ１コマンドで、PROC１の動きができるから、繰り返しのある動きをみつければできそう。 ◎繰り返している動きを丸で囲んでみるとプロシージャに入れるコマンドがわかるかも。	・すでにコツをつかんでいる子どもには、自分のめあてをたてるよう促すことで、発展的に問題解決に取り組めるようにする。 ・学習のまとめの時間なので、プログラミング体験の時間をできるだけ長く確保することで、これまでのコツを活用できるようにする。 同じ動きのパターンをみつけてプロシージャにまとめることが苦手な人もいます。どうしても同じ動きのパターンがみえにくい場合は、ステージ２をすべてクリアできなくても、ステージ３．ループに挑戦するよう促してもいいですね。一番大事なのは、プログラミング体験を楽しめることではないかと思います。 	分解 組合せ 一般化
まとめ	４　本時の学習を振り返る ◎「ロジックマジック」もプロシージャも同じところを丸で囲むと分かりやすい。別の手品でも同じように考えると、たねが分かるかも。 ◎「トイレのサイン」みたいに、身の回りのプログラムを見つけてみたい。	・これまでの振り返りの記述を読み返す時間を設定することで、単元を通して身についた力をまとめることができるようにする。	