

Lesson 5: Game Outcome

Teacher: Claire Lin	Grades: 3-5	Time Required: 90 min
Subject: Game Design	Teaching objectives: Students will 1) Develop the skills to design game mechanism and design the outcomes in the racing game. 2) Know how to improve the gaming experience of the games designed by themselves.	
Teaching Materials		New Concepts
Teacher: Codey, a computer (installed with the latest version of mBlock 5), the lesson plan and the slide.	Students: Codey, a computer (installed with the latest version of mBlock 5)	Game mechanism Rules and Outcome Coding blocks: Variables, Condition, Operators
The Teaching Procedure		
Introduction: <p style="text-align: center;">The teacher will explain to students the concept of Game Outcome.</p> <p>(Objectives)Students are able to master a basic understanding of: a. what the concept of Game Outcome is; b. what the common game outcomes are; c. how to design outcomes.</p> <p>1) The teacher can tell students: “Our game now has a variety of fun elements. We have rules. And we make the rules complicated by designing moving obstacles in the game. But have you ever noticed one thing? The game has no winning state. Then how should we design a Winning state for our game?”</p>		

2) The teacher starts playing the game demo: “We’ve already designed a **Losing** state for our game: The game will be over when the racing car runs outside the track or hits obstacles. But how to **Win** the game? We haven’t set the rule yet. So, for players to gain the sense of achievement coming from winning the game, we need to design a set of rules that define the winning state.”

3) When we are designing rules that define the winning state and losing state of the game, we can say we are designing the **Game Outcomes**. In our daily life, each single choice might lead to one certain outcome. Games are in the same case. Players make choices and show different behaviors in games. These choices and behaviors will lead to either a winning outcome or a losing outcome.

Guided Practice:

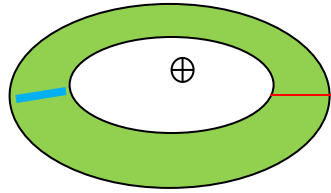
1)Brainstorming: (Objective) The students will consider what game outcomes fit in the racing game.

The teacher asks students: “Can you think of any game outcomes? What’s the situation for winning the game? And what’s the situation for losing the game? (**The teacher shows examples included in the slide**) Then what kind of game outcomes do you think fit in the racing game? You can discuss with your desk mates.” Leave enough time for students to have a discussion (5-10 min). And help students take a player-centric approach to design the outcomes.

The teacher shares his or her design concept with students: “All of your ideas are brilliant! And each of your outcome settings is different, which distinguishes your game from other games. I’ve also had a design: **When the car accomplishes one circle (successfully gets to the blue finishing line), the car will show a sign and make a sound, telling you that you win the game!**”

2)Prototyping: (Objective) Have students use mBlock 5 to design the game outcomes.

Physical prototype:



Digital prototype:

Have students start coding with mBlock 5: “First design the stage and sprites. Then connect Codey to mBlock 5. Next you can start programming! You’ll need to use the Broadcast block if you are going to use the Codey to control the racing car.”(More details can be found in the slide)

Outcomes:

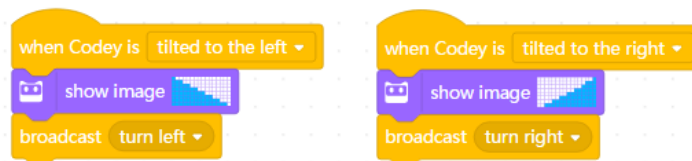
Win: The racing car can accomplish one running lane.

Lose: The car has to go back to the starting point when it runs outside the track or hits obstacles.

Questions:

How does the program tell whether the car accomplishes one running lane or not?

The program for controlling Codey:



The program for controlling the bat(nothing changed):

```
when I receive game starts
  switch costume to Car1
  go to x: pick random -200 to 200 y: 150
  forever
    change y by -5
    if touching edge ? then
      go to x: pick random -200 to 200 y: 150
    if touching Car1 ? then
      switch costume to crashed bat
      wait 1 seconds
      switch costume to bat2-b
      go to x: pick random -200 to 200 y: 165
```

The program for controlling the racing car:

The image displays two versions of a Scratch script, one on the left and one on the right, with a third script block centered below them. The left script starts with a 'when I receive' event block set to 'turn right'. It includes a 'stop other scripts in sprite' block, followed by a 'forever' loop. Inside the loop, it turns 1 degree, moves 2 steps, and checks if it is touching a yellow color. If so, it thinks 'Ahya!', changes to the next costume, waits 1 second, goes to x: -79, y: -104, points in direction 90, switches to costume 'Car1', and stops the script. If not touching yellow, it checks if touching a blue color. If so, it says 'Win!', starts the 'Meow' sound, and stops the script. The right script is identical but starts with a 'when I receive' event block set to 'turn left'. The centered script block starts with a 'when I receive' event block set to 'hit me', followed by going to x: -79, y: -104 and pointing in direction 90.

3) Playtesting: (Objective) Students will have a better understanding of Playtesting - a pivotal step in game design.

The teacher asks students: “Anyone wants to experience this game and shares with us how you feel about the game? Is there anything that you think should improve?”

4) Iterating & Implementing:(Objective)Students will know how to reiterate the game prototype and implement the game.

The teacher will make modifications based on the feedback from students.

Independent Practice:

It's Your Turn!

1) Selecting a solution: (The teacher saying to students) “We’ve just discussed how to design game outcomes. You came up with so many ideas, but now you have to pick your own ultimate design.

After that, you can add game rules and goals to your games.”

2) Prototyping: (The teacher saying to students) “Draw a draft or create a prototype based on your idea. Or you can use mBlock 5 to write programs straightaway.”

3) Playtesting: (The teacher saying to students) “When your game is ready, put your hands up. Then, invite some target players to experience your prototype. Of course, you can invite me to try it.”

4) Iterating & Implementing: (The teacher saying to students) “Fix bugs based on the players’ feedback. Perfect your game.”

Share:

Presentation

Allow each student to share his or her own game project with the class and tell them to invite classmates to experience their game.

At the end of this session, let students vote for the best design of the day.

Differentiation & Modification

For advanced students	For struggling students	Strategies to maximize engagement
Complicated rules: 1) the car	Simple rules:	Have students vote for the Best

<p>should accomplish one running lane within a set period of time;</p> <p>2) should accomplish several lanes; 3)should gets to the red line.</p>	<p>1) Accomplish one running lane. There's no limit on time.</p>	<p>Game of the Day</p>
<p>Comments:</p>		
<p>Teachers' Reflections:</p>		