



## Teacher Guide: Code Commander

<https://www.tynker.com/hour-of-code/codecommander>

**Time: 60 minutes**

**Grades: 6+**

**Difficulty: Intermediate**



Students complete a set of 20 coding puzzles that introduce them to all the basic coding concepts they need to code each member of their army. After completing the puzzles, they can play in the multiplayer arena and battle against other players.

**Activity Requirements:** This activity requires computers with a Web browser and an Internet connection. Headphones recommended.

### Programming Activities

#### Solve 20 Coding Puzzles (45 minutes)

Students solve a set of 20 coding puzzles to learn how to program different members of their army. The first puzzles are very simple and introduce basic concepts. As students progress through the set, they unlock new characters that they can program and the puzzles become more challenging. They learn and apply computational thinking concepts like sequencing, pattern recognition, and automation. At the end of this activity, students understand basic programming concepts like conditional logic, loops, and sequencing.

#### Battle in Multiplayer Mode (15 Minutes)

After completing at least 5 puzzles, students will unlock the “Battle with Friends” option. They can click on “My Army” from the main screen to code each member of their army for battle. Encourage your students to continue solving puzzles to unlock more characters or they will be at a significant disadvantage in battle. Remind them that losing a battle can help them if they pay attention to the other army’s strategy. Coming soon, students will be able to use the points they earn in the game to buy better armor and weapons to equip their army.



#### Hour of Code Certificate

Be sure to download a personalized certificate for your students when they complete this activity.

### Standards Mapping

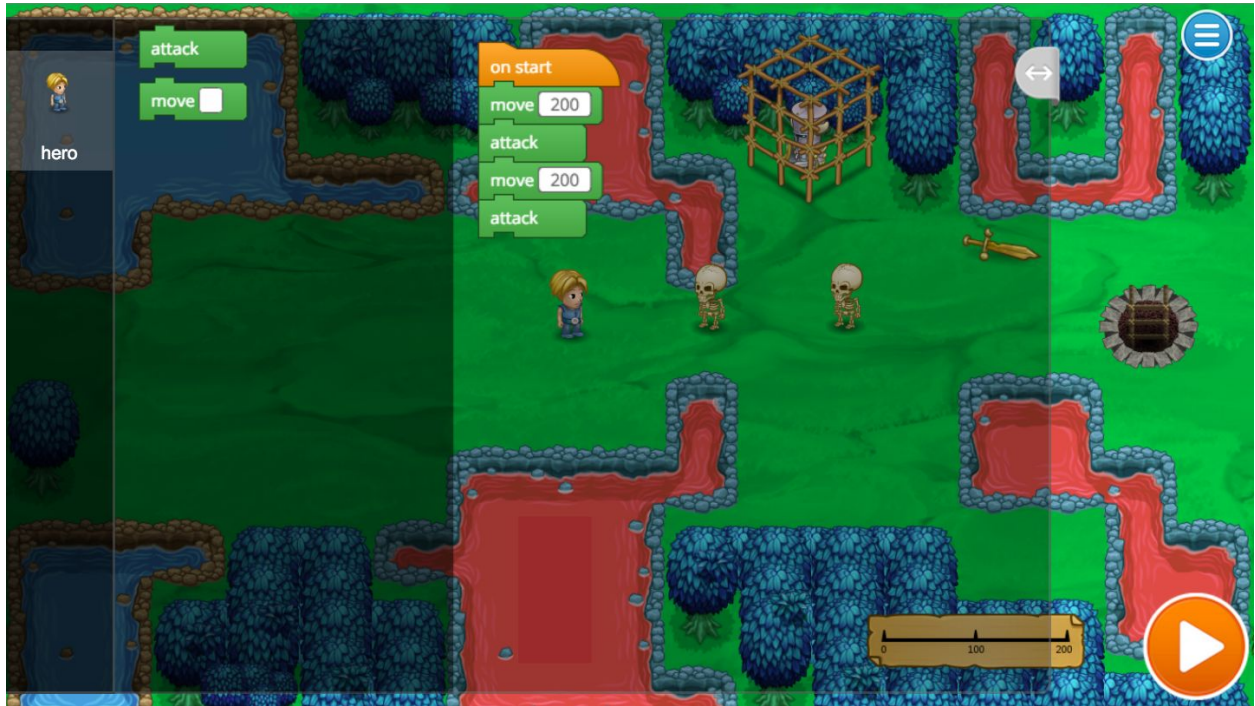
CCSS ELA: RI.3.3, W.3.6, RI.4.5, RI.4.3, RI.5.10, RST.6-8.4, RST.6-8.7, RST.9-10.5, RST.11-12.3

CCSS Math: MP.3.2, MP.3.8, MD.4.5, NF.4.7

CSTA: L1:6.CT.1, L1:6.CPP.5, L1:6.CPP.6, L2:9.CT.1, L2:9.CT.3, L2:9.CT.5, L2:9.CT.12, L2:9.CPP.3, L2:9.CPP.5

## Puzzle Solutions

Puzzle 2



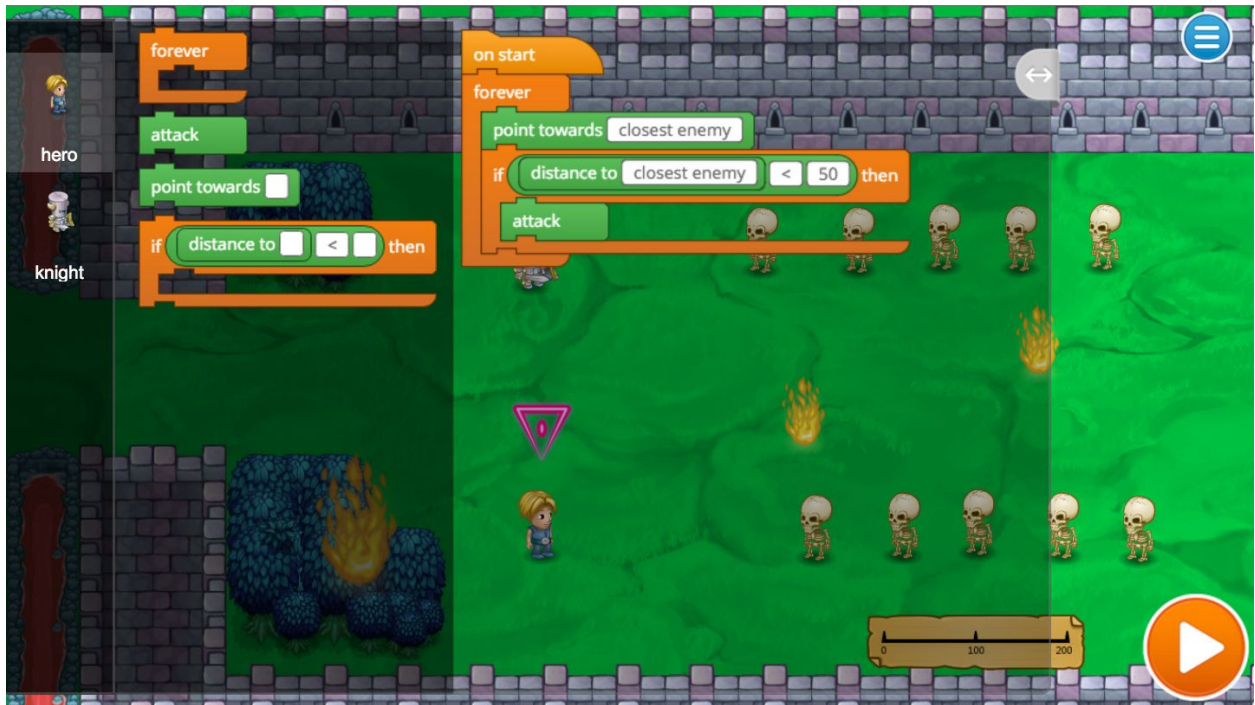
Puzzle 3



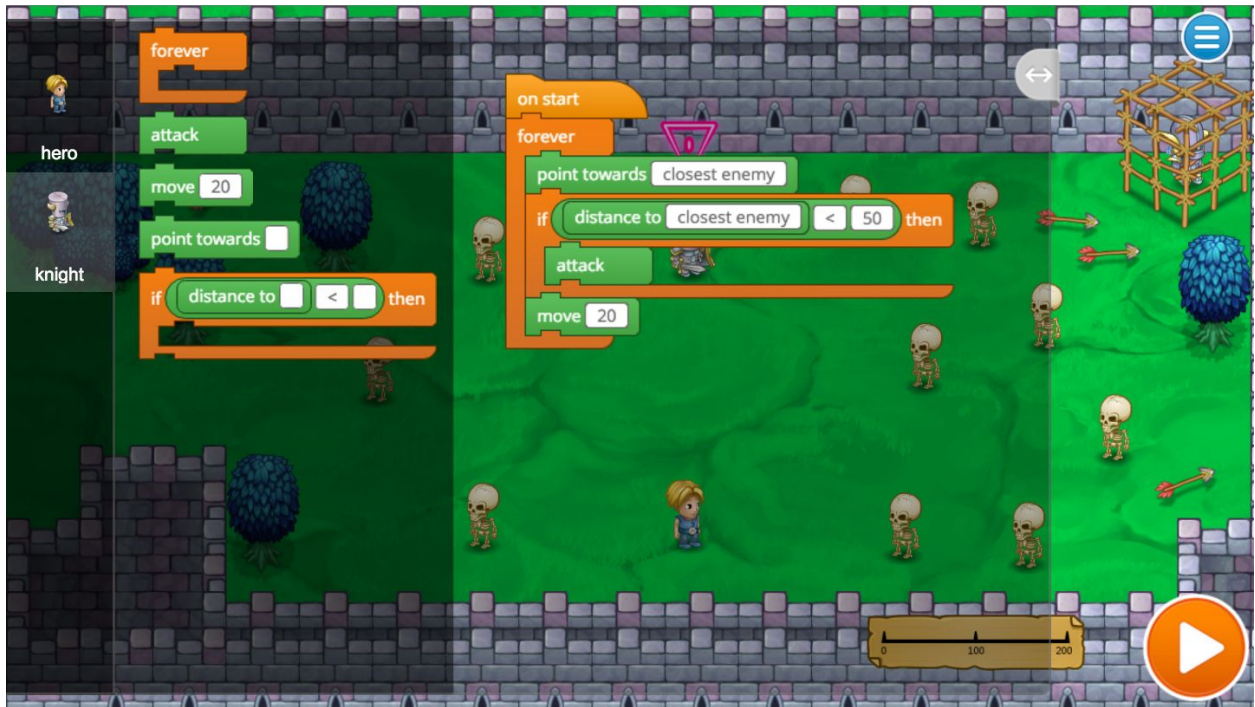
## Puzzle 4



## Puzzle 5



## Puzzle 6



## Puzzle 7



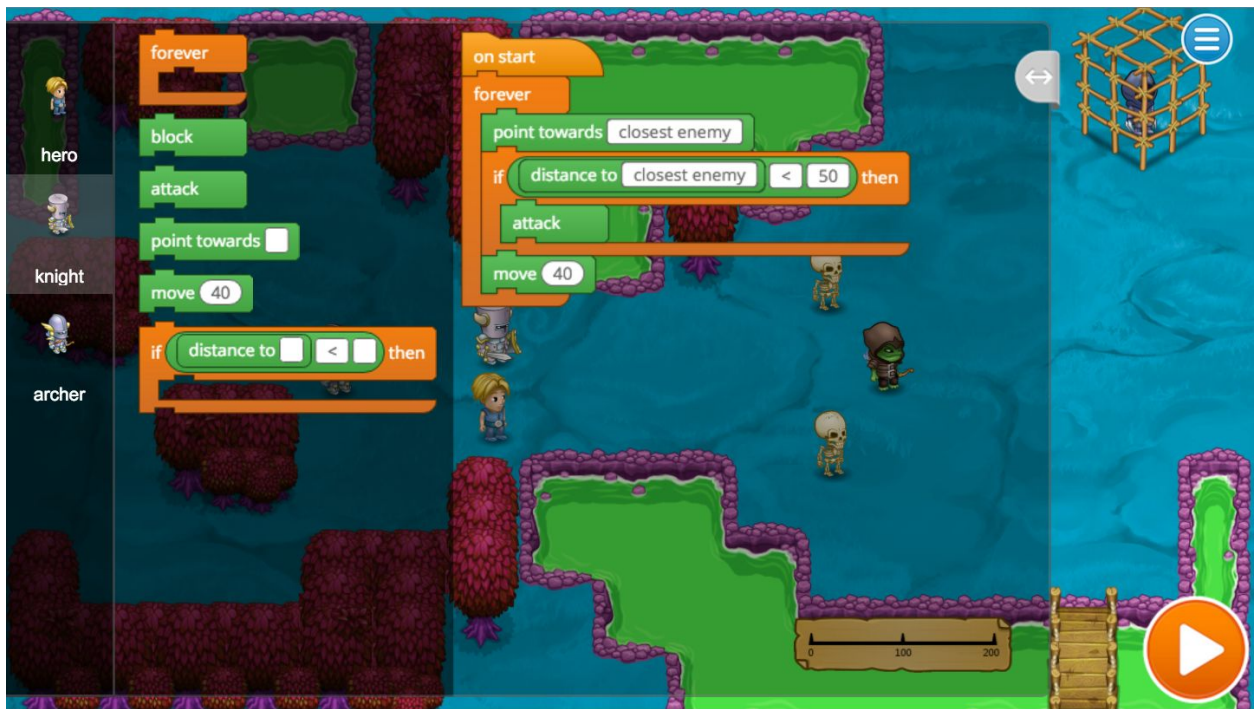
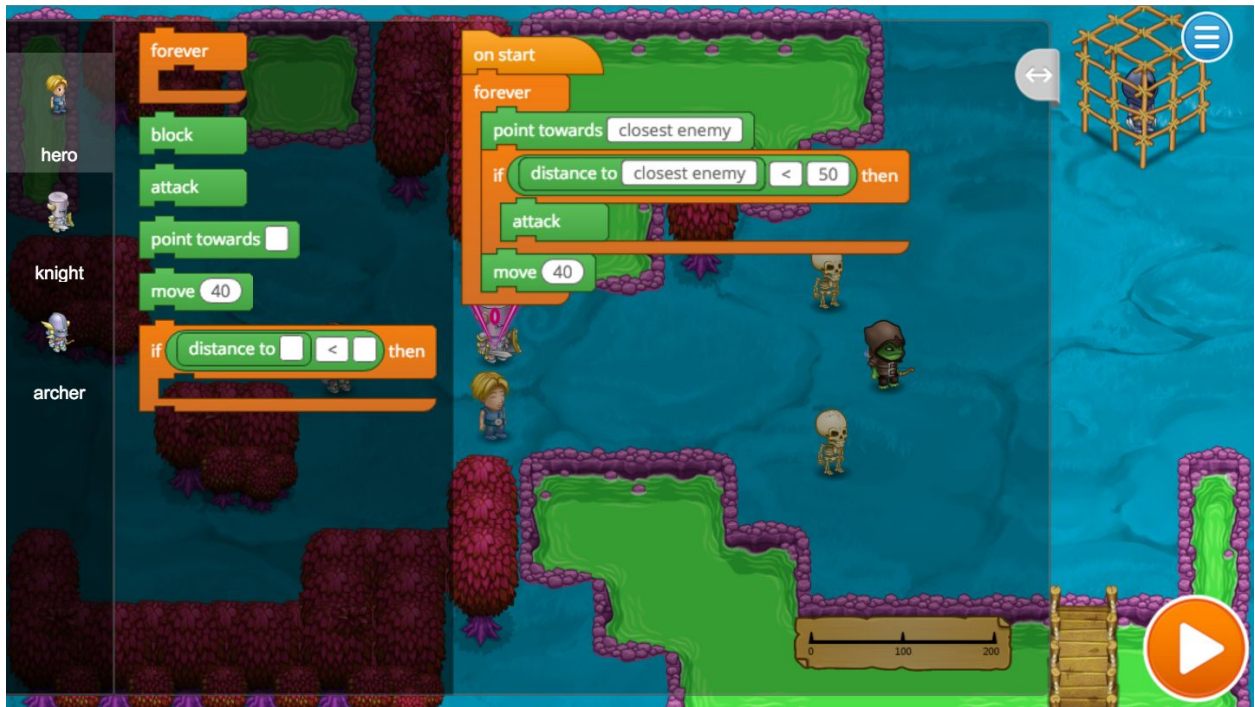
## Puzzle 7 Cont.



## Puzzle 8

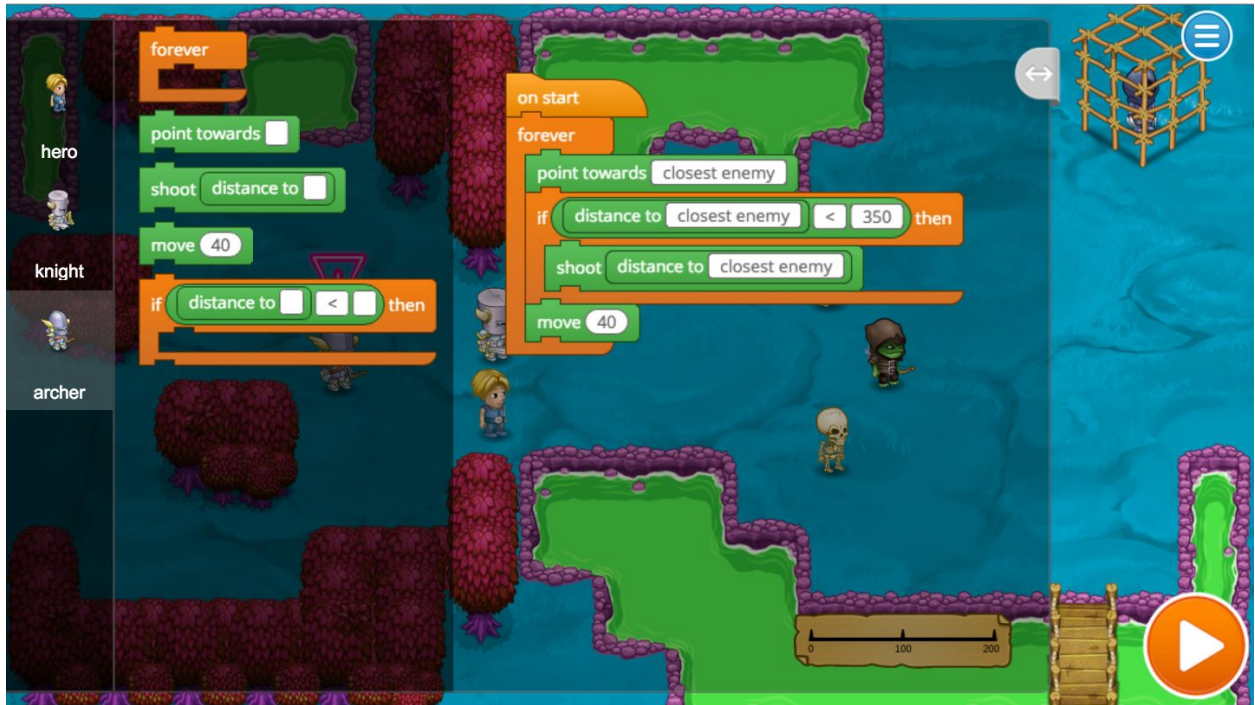


## Puzzle 9

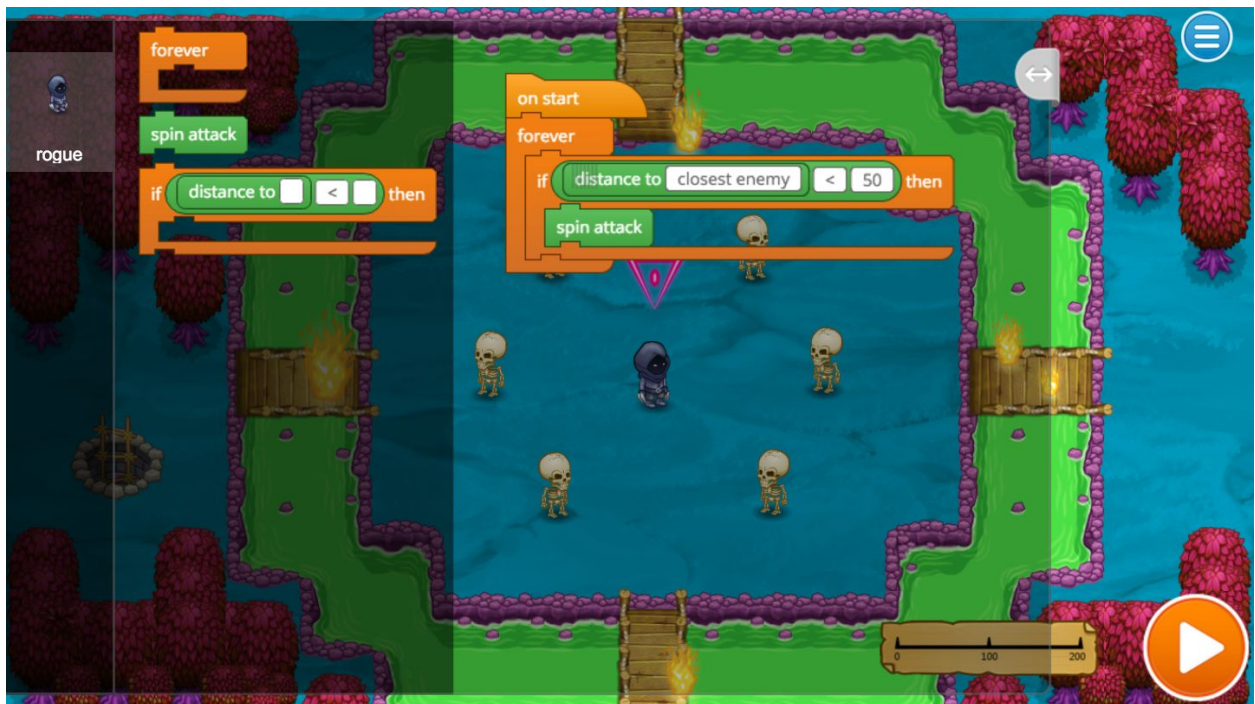




## Puzzle 9 Cont.



## Puzzle 10



## Puzzle 11



## Puzzle 12



Puzzle 12 Cont.

The screenshot shows the Tynker game editor interface. On the left, there are three character slots: hero, knight, and archer. The hero slot contains a 'forever' loop with an 'if' block: 'if distance to closest enemy < 60 then' followed by 'point towards closest enemy' and 'attack'. The knight slot contains a 'forever' loop with 'move 40' and 'point towards closest enemy'. The archer slot contains 'point in direction 90' and 'attack'. On the right, the 'on start' block contains a 'forever' loop with an 'if' block: 'if distance to closest enemy < 60 then' followed by 'point towards closest enemy' and 'attack', and an 'else' block with 'point in direction 90' and 'move 40'. The game scene shows a hero character, several enemy characters, and a treasure chest on a blue, rocky terrain. A scale bar at the bottom indicates 0, 100, and 200 units. A play button is visible in the bottom right corner.

The screenshot shows the Tynker game editor interface. On the left, there are three character slots: hero, knight, and archer. The hero slot contains a 'forever' loop with an 'if' block: 'if distance to closest enemy < 100 then' followed by 'point towards closest enemy' and 'shoot distance to closest enemy'. The knight slot contains a 'forever' loop with 'move 40' and 'point towards closest enemy'. The archer slot contains 'point in direction 90' and 'shoot distance to closest enemy'. On the right, the 'on start' block contains a 'forever' loop with an 'if' block: 'if distance to closest enemy < 100 then' followed by 'point towards closest enemy' and 'shoot distance to closest enemy', and an 'else' block with 'point in direction 90' and 'move 40'. The game scene shows a hero character, several enemy characters, and a treasure chest on a blue, rocky terrain. A scale bar at the bottom indicates 0, 100, and 200 units. A play button is visible in the bottom right corner.

## Puzzle 13

The screenshot shows the Tynker game editor interface for Puzzle 13. On the left, a character selection panel lists 'hero', 'knight', 'archer', and 'rogue'. The main workspace displays a desert-themed game environment with a blue pond, a wooden cage, and several enemy characters. The logic script is as follows:

- on start** block containing a **forever** loop:
  - point towards** closest enemy
  - if** distance to closest enemy < 60 **then**:
    - attack**
  - else**:
    - move** 40

The 'hero' character is currently selected in the left panel. A play button is visible in the bottom right corner.

This screenshot is identical to the one above, showing the same Tynker game editor interface for Puzzle 13. The logic script and game environment are the same. The 'hero' character is selected in the left panel. A play button is visible in the bottom right corner.

## Puzzle 13 Cont.

The screenshot shows the Tynker game editor interface. On the left, a character selection panel lists 'hero', 'knight', 'archer', and 'rogue'. The main workspace displays a script for the 'hero' character. The script consists of a 'forever' loop with an 'if' condition: 'if distance to closest enemy < 300 then'. Inside the 'if' block, there is a 'shoot distance to closest enemy' block. Outside the 'if' block, there is an 'else' block containing a 'move 40' block. The game scene shows a hero character in a cage, several enemy characters, and a body of water with a bridge.

```

    forever
      if distance to closest enemy < 300 then
        shoot distance to closest enemy
      else
        move 40
  
```

The screenshot shows the Tynker game editor interface. On the left, a character selection panel lists 'hero', 'knight', 'archer', and 'rogue'. The main workspace displays a script for the 'knight' character. The script consists of an 'on start' block followed by a 'forever' loop. Inside the 'forever' loop, there is a 'point towards closest enemy' block, followed by an 'if' condition: 'if distance to closest enemy < 60 then'. Inside the 'if' block, there are 'spin attack' and 'attack' blocks. Outside the 'if' block, there is an 'else' block containing a 'move 40' block. The game scene is identical to the first screenshot, showing a knight character in a cage, enemy characters, and a body of water with a bridge.

```

    on start
      forever
        point towards closest enemy
        if distance to closest enemy < 60 then
          spin attack
          attack
        else
          move 40
  
```

## Puzzle 14

The screenshot shows Puzzle 14 in Tynker. On the left, a 'mage' character is selected. The code editor contains the following blocks:

- forever loop containing:
  - heal area
  - point towards [ ]
  - if distance to [ ] < [ ] then

In the main scene, a 'forever' loop is attached to the mage character, containing an 'on start' block and a 'heal area' block. The scene depicts a cave with a group of goblins and several characters. A scale bar at the bottom indicates 0, 100, and 200 units.

## Puzzle 15

The screenshot shows Puzzle 15 in Tynker. On the left, a 'mage' character is selected. The code editor contains the following blocks:

- forever loop containing:
  - cast fireball
  - point towards [ ]
  - if distance to [ ] < [ ] then

In the main scene, a 'forever' loop is attached to the mage character, containing an 'on start' block and another 'forever' loop. The inner 'forever' loop contains:

- point towards closest enemy
- if distance to closest enemy < 350 then
  - cast fireball

The scene depicts a cave with a group of goblins and several characters. A scale bar at the bottom indicates 0, 100, and 200 units.

## Puzzle 16

The screenshot shows a puzzle environment with a mage character. The code blocks are as follows:

```

on start
  forever
    if distance to closest enemy < 100 then
      point towards closest enemy
      cast fireball
    else
      if distance to my commander < 200 then
        heal area
      else
        point towards my commander
        move 20
  
```

The environment features a mage character, a green goblin enemy, and a commander character. The scene includes a river, trees, and stone walls with spikes.

## Puzzle 17

The screenshot shows a puzzle environment with a hero character. The code blocks are as follows:

```

on start
  move 1000
  turn right by 90
  move 350
  turn left by 90
  move 200
  forever
    point towards closest enemy
    if distance to closest enemy < 60 then
      attack
    else
      move 40
  
```

The environment features a hero character, a green goblin enemy, and a commander character. The scene includes stone walls, a dark floor, and a scale bar.

## Puzzle 17 Cont.

The screenshot shows the Tynker interface with a character's code blocks. The left sidebar lists character types: hero, knight, mage, and archer. The main workspace contains the following code blocks:

```

on start
  move 1000
  turn left by 90
  move 350
  turn right by 90
  move 200
  forever
    point towards closest enemy
    if distance to closest enemy < 60 then
      attack
    else
      move 40
  
```

The character is positioned in a dark, stone-walled environment with several green goblin-like enemies. A scale bar at the bottom indicates distances of 0, 100, and 200 units.

The screenshot shows the Tynker interface with a character's code blocks. The left sidebar lists character types: hero, knight, mage, and archer. The main workspace contains the following code blocks:

```

on start
  forever
    if distance to closest enemy < 350 then
      cast fireball
    else
      move 40
  
```

The character is positioned in the same dark, stone-walled environment as in the previous screenshot. A scale bar at the bottom indicates distances of 0, 100, and 200 units.



## Puzzle 17 Cont.

Script for hero:

```

forever
  if distance to closest enemy < 350 then
    shoot distance to closest enemy
  else
    // empty
  
```

Script for knight:

```

on start
  forever
    point towards closest enemy
    if distance to closest enemy < 350 then
      shoot distance to closest enemy
    else
      // empty
    
```

## Puzzle 18

Script for rogue:

```

forever
  attack
  move
  turn left by
  turn right by
  if distance to < < then
  else
  if obstacle ahead then
  else
  
```

Script for knight:

```

on start
  forever
    if distance to closest enemy < 60 then
      attack
      turn right by 90
    else
      if obstacle ahead then
        turn left by 90
      else
        move 40
    
```

## Puzzle 19



## Puzzle 20



Puzzle 20 Cont.

The top screenshot shows the Tynker editor interface. On the left, a vertical menu lists character types: hero, knight, archer, mage, and rogue. The 'knight' character is selected. The code editor shows the following logic:

```

on start
  forever
    point towards closest enemy
    if distance to closest enemy < 50 then
      slash
    else
      move 20
  
```

The game scene on the right shows a knight character on a blue, rocky terrain. A large green ogre boss is on the right, and several smaller enemies are scattered around. A distance scale at the bottom right ranges from 0 to 200. A play button is visible in the bottom right corner.

The bottom screenshot shows the Tynker editor interface with the 'archer' character selected. The code editor shows the following logic:

```

on start
  forever
    point towards closest enemy
    if distance to closest enemy < 300 then
      shoot distance to closest enemy
    else
      move 20
  
```

The game scene on the right is identical to the top screenshot, showing the knight character and the ogre boss on the blue terrain. A distance scale at the bottom right ranges from 0 to 200. A play button is visible in the bottom right corner.

Puzzle 20 Cont.

Hero script:

```

forever
  if distance to closest enemy < 50 then
    cast fireball
  else
    heal area
  
```

Knight script:

```

on start
  forever
    point towards closest enemy
    if distance to closest enemy < 50 then
      cast fireball
    else
      move 20
  
```

Hero script:

```

forever
  if distance to closest enemy < 100 then
    spin attack
  else
    slash
  
```

Knight script:

```

on start
  forever
    point towards closest enemy
    if distance to closest enemy < 100 then
      spin attack
    else
      move 20
  
```



## Teacher Guide to Tynker Hour of Code

Tynker's activities combine structured and open-ended components to support multiple learning styles. This experience emphasizes that programming requires not only knowledge of how to use a language, but also creativity and critical thinking to figure out how to build projects. Tynker is offering a wide variety of activities appropriate for all grades and experience levels.

### What Tynker Provides

- Self-contained, game-based activities that students can complete with minimal support
- A combination of structured and open-ended activities that teach and allow students to create
- Puzzle solutions for all of our puzzles so you can give hints to any students who get stuck
- Common Core alignment for all activities
- A customized Hour of Code certificate for each activity that will show up in the student dashboard when a student completes an hour of programming

### Why Children Love Tynker

- Tynker puzzles use game-based learning to teach programming and computational thinking concepts in a fun way
- Tynker tutorials guide students through all the steps to create storytelling projects, games, animations, and much more
- The Tynker Workshop allows students to create anything they can imagine with code
- Tynker's built-in Physics Engine makes it easy to create exciting projects
- Tynker's high quality media assets give students tons of creative options

### Recommended Setup and Logistics

- Ideal environment: a computer lab, library, or classroom with your class
- Students can work individually or in pairs
- Students should have headphones if possible, but if not, you can turn the computer volume down
- Set up a free teacher account on [tynker.com](https://tynker.com) prior to the activity and add your students so you can track their progress and share a class showcase—and so students can continue working at home! (Note: Creating a teacher account is optional. You can complete your Hour of Code with Tynker without creating an account.)

We hope you take a look at all of our Hour of Code activities to figure out which one is right for your class. Join the global movement and host your Hour of Code with Tynker!