



# Game Development 1

## Python

*Learn to create your own games which you can share with your friends.*

## Table of Contents

1. Over the Garden Wall
2. Click Gait
3. Hero's Journey
4. A-maze-ing
5. Gemtacular
6. Vorpal Mouse

- 7. Crushing It
- 8. Give and Take
- 9. Army Training
- 10. Ranger Danger
- 11. Hedge Magic
- 12. Forest Incursion
- 13. Throwing Fire
- 14. Them Bones
- 15. Behavior Driven Development
- 16. Time To Live
- 17. Seeing is Believing
- 18. Persistence Pays
- 19. Tabula Rasa

# #1. Over the Garden Wall

## Level Overview and Solutions

### Intro

This is the first game development level.

You will learn how to make your own games!

You need to spawn 4 more "fence" s (for a total of 8, including the ones given to you) to protect these villagers from marauding ogres:

```
game.spawnXY("fence", 39, 6)
game.spawnXY("fence", 39, 10)
```

### Default Code

```
# Use game.spawnXY() to build a fence around the farm!

game.spawnXY("fence", 39, 16)
# Spawn 2 more fences 4 meters apart...
# Spawn a "fence" at y 20

# Spawn a "fence" at y 24

game.spawnXY("fence", 39, 28)
game.spawnXY("fence", 39, 32)
# Spawn 2 more fences 4 meters apart...
# Spawn a "fence" at y 36

# Spawn a "fence" at y 40

game.spawnXY("fence", 39, 44)
```

## Overview

You should `spawnXY` fences at the same x coordinate ( 39 ), but vary the y coordinate.

Each fence segment should be 4 meters more than the previous "fence" s y position.

## Over the Garden Wall Solution

```
# Use game.spawnXY() to build a fence around the farm!
```

```
game.spawnXY("fence", 39, 16)
# Spawn 2 more fences 4 meters apart...
# Spawn a "fence" at y 20
game.spawnXY("fence", 39, 20)
# Spawn a "fence" at y 24
game.spawnXY("fence", 39, 24)

game.spawnXY("fence", 39, 28)
game.spawnXY("fence", 39, 32)
# Spawn 2 more fences 4 meters apart...
# Spawn a "fence" at y 36
game.spawnXY("fence", 39, 36)
# Spawn a "fence" at y 40
game.spawnXY("fence", 39, 40)
game.spawnXY("fence", 39, 44)
```

## #2. Click Gait

# Level Overview and Solutions

### Intro

To play your game, click the Test Level button under the code window.

In this level, we've added the code required to make this mini-game work.

All you need to do is click Test Level, then move the hero around the house by clicking with the mouse.

To win, you have to move to all the red X marks!

### Default Code

```
# Don't change any code. Just click Test Level!
# Move to all the red X marks to win.

game.spawnHeroXY("knight", 5, 8)
game.addMoveGoalXY(10, 18)
game.addMoveGoalXY(10, 38)
game.addMoveGoalXY(48, 41)
```

### Overview

*Coming soon!*

## Click Gait Solution

```
# Don't change any code. Just click Test Level!
# Move to all the red X marks to win.
```

```
game.spawnHeroXY("knight", 5, 8)
game.addMoveGoalXY(10, 18)
game.addMoveGoalXY(10, 38)
game.addMoveGoalXY(48, 41)
```

---

## #3. Hero's Journey

# Level Overview and Solutions

### Intro

Every game needs a goal for the player.

One type of goal is a move goal - somewhere the player needs to get to:

```
game.addMoveGoalXY(10, 15)
```

### Default Code

```
# Each game must have a hero and a goal.  
  
# Use game.spawnHeroXY("captain", x, y)  
# to add a hero to your game:  
  
# Use game.addMoveGoalXY(x, y)  
# to add a movement goal to your game:  
  
# If you want to, use spawnXY("fence", x, y)  
# to make a maze with fences...  
  
# Then, click "Test Level" to try your first playable game!
```

### Overview

*Coming soon!*

## Hero's Journey Solution

```
# Each game must have a hero and a goal.  
  
# Use game.spawnHeroXY("captain", x, y)  
# to add a hero to your game:  
game.spawnHeroXY("captain", 9, 9)  
# Use game.addMoveGoalXY(x, y)  
# to add a movement goal to your game:  
game.addMoveGoalXY(53, 45)  
# If you want to, use spawnXY("fence", x, y)  
# to make a maze with fences...  
  
# Then, click "Test Level" to try your first playable game!
```

---

## #4. A-maze-ing

# Level Overview and Solutions

### Intro

Now you can use `game.spawnXY()` to spawn a "forest" tile, and a "chest" of gems!

Also, you can add a goal to collect all gems with `game.addCollectGoal()`.

### Default Code

```
# Alejandro likes a challenge!
# Add more "forest" objects to the level to create a long maze.

# Set up the game's goal.
game.addCollectGoal()
# Spawn a hero and chest to collect.
game.spawnHeroXY("duelist", 9, 59)
game.spawnXY("chest", 8, 14)

game.spawnXY("forest", 26, 51)
# Add 2 more "forest" objects. Make sure not to block the gems completely!
```

### Overview

*Coming soon!*

## A-maze-ing Solution

```
# Alejandro likes a challenge!
# Add more "forest" objects to the level to create a long maze.

# Set up the game's goal.
game.addCollectGoal()
# Spawn a hero and chest to collect.
game.spawnHeroXY("duelist", 9, 59)
game.spawnXY("chest", 8, 14)

game.spawnXY("forest", 26, 51)
# Add 2 more "forest" objects. Make sure not to block the gems completely!
game.spawnXY("forest", 34, 22)
game.spawnXY("forest", 50, 22)
```

## #5. Gemtacular

# Level Overview and Solutions

### Intro

Add a collection goal, then spawn some gems for the player to collect!

### Default Code

```
# Anya is searching for gems!
# Add gems to the level for the player to find.
# You must be able to beat your level to continue.

game.spawnHeroXY("captain", 9, 18)
# Add a goal to collect the gems using game.addCollectGoal()

game.spawnXY("gem", 28, 28)
# Add 3 more gems across the level for the player to collect:
```

### Overview

*Coming soon!*

### Gemtacular Solution

```
# Anya is searching for gems!
# Add gems to the level for the player to find.
# You must be able to beat your level to continue.

game.spawnHeroXY("captain", 9, 18)
# Add a goal to collect the gems using game.addCollectGoal()
game.addCollectGoal()
game.spawnXY("gem", 28, 28)
# Add 3 more gems across the level for the player to collect:
game.spawnXY("gem", 69, 10)
game.spawnXY("gem", 9, 36)
game.spawnXY("gem", 45, 44)
```

## #6. Vorpal Mouse

### Level Overview and Solutions

#### Intro

When playing a game, you can click on the ground to move and an ogre to attack it!

This level, all you need to do is click PLAY, then defeat all the ogres!

#### Default Code

```
# When playing, click an ogre to attack!
# No need to type any code for this level, just click Test Level!

# The following code spawns a playable hero:
game.spawnHeroXY("guardian", 40, 40)

# These add goals to the level:
game.addDefeatGoal()
game.addSurviveGoal()

# This spawns some enemies to fight:
game.spawnXY("munchkin", 40, 25)
game.spawnXY("munchkin", 15, 15)
game.spawnXY("munchkin", 65, 15)

# This spawns a maze for the player:
game.spawnXY("forest", 30, 54)
game.spawnXY("forest", 50, 54)
game.spawnXY("forest", 30, 48)
game.spawnXY("forest", 50, 48)
game.spawnXY("forest", 30, 40)
game.spawnXY("forest", 50, 40)
game.spawnXY("forest", 30, 32)
game.spawnXY("forest", 50, 32)
game.spawnXY("forest", 30, 26)
game.spawnXY("forest", 50, 26)
game.spawnXY("forest", 30, 10)
game.spawnXY("forest", 50, 10)
game.spawnXY("forest", 22, 26)
game.spawnXY("forest", 58, 26)
game.spawnXY("forest", 14, 26)
game.spawnXY("forest", 66, 26)
```

## Overview

*Coming soon!*

## Vorpal Mouse Solution

```
# When playing, click an ogre to attack!
# No need to type any code for this level, just click Test Level!

# The following code spawns a playable hero:
game.spawnHeroXY("guardian", 40, 40)

# These add goals to the level:
game.addDefeatGoal()
game.addSurviveGoal()

# This spawns some enemies to fight:
game.spawnXY("munchkin", 40, 25)
game.spawnXY("munchkin", 15, 15)
game.spawnXY("munchkin", 65, 15)

# This spawns a maze for the player:
game.spawnXY("forest", 30, 54)
game.spawnXY("forest", 50, 54)
game.spawnXY("forest", 30, 48)
game.spawnXY("forest", 50, 48)
game.spawnXY("forest", 30, 40)
game.spawnXY("forest", 50, 40)
game.spawnXY("forest", 30, 32)
game.spawnXY("forest", 50, 32)
game.spawnXY("forest", 30, 26)
game.spawnXY("forest", 50, 26)
game.spawnXY("forest", 30, 10)
game.spawnXY("forest", 50, 10)
game.spawnXY("forest", 22, 26)
game.spawnXY("forest", 58, 26)
game.spawnXY("forest", 14, 26)
game.spawnXY("forest", 66, 26)
```

## #7. Crushing It

# Level Overview and Solutions

### Intro

Practice making a simple ogre-fighting game:

Remember to

1. Add a hero for the player to control.
2. Add a goal to defeat the ogres.
3. Spawn some munchkins!

### Default Code

```
# If you forget any commands, look below the code window!

# Spawn a hero using game.spawnHeroXY(type, x, y).

game.addSurviveGoal()
# Add a goal to defeat ogres with game.addDefeatGoal()

game.spawnXY("munchkin", 40, 10)
# Spawn at least 3 more munchkins.
```

## Overview

*Coming soon!*

## Crushing It Solution

```
# If you forget any commands, look below the code window!

# Spawn a hero using game.spawnHeroXY(type, x, y).
game.spawnHeroXY("champion", 20, 20)
game.addSurviveGoal()
# Add a goal to defeat ogres with game.addDefeatGoal()
game.addDefeatGoal()

game.spawnXY("munchkin", 40, 10)
# Spawn at least 3 more munchkins.
game.spawnXY("munchkin", 34, 49)
game.spawnXY("munchkin", 51, 40)
game.spawnXY("munchkin", 64, 21)
```

## #8. Give and Take

# Level Overview and Solutions

### Intro

To make your games a little more interesting, try adding some "fire-trap"s and "potion-small"s!

```
game.spawnXY("fire-trap", 40, 20)
game.spawnXY("potion-small", 25, 20)
```

### Default Code

```
# Move to all the X-marks.
# Those fire-traps hurt!

game.spawnHeroXY("samurai", 40, 50)
game.addSurviveGoal()
game.addMoveGoalXY(25,40)
game.spawnXY("fire-trap", 25, 40)
game.addMoveGoalXY(55,40)
game.spawnXY("fire-trap", 55, 40)
game.addMoveGoalXY(40,20)
game.spawnXY("fire-trap", 40, 20)

# Spawn some "potion-small" objects to heal the player!
```

### Overview

*Coming soon!*

## Give and Take Solution

```
# Move to all the X-marks.  
# Those fire-traps hurt!  
  
game.spawnHeroXY("samurai", 40, 50)  
game.addSurviveGoal()  
game.addMoveGoalXY(25,40)  
game.spawnXY("fire-trap", 25, 40)  
game.addMoveGoalXY(55,40)  
game.spawnXY("fire-trap", 55, 40)  
game.addMoveGoalXY(40,20)  
game.spawnXY("fire-trap", 40, 20)  
  
# Spawn some "potion-small" objects to heal the player!  
game.spawnXY("potion-small", 30, 39)  
game.spawnXY("potion-small", 40, 25)
```

---

## #9. Army Training

# Level Overview and Solutions

### Intro

Spawn new unit types:

```
game.spawnXY("thrower", 10, 10) # Ogre Spear Thrower  
game.spawnXY("soldier", 10, 10) # Human Soldier  
game.spawnXY("archer", 10, 10) # Human Archer
```

### Default Code

```
# Spawn a hero and add a goal.  
game.spawnHeroXY("champion", 40, 15)  
game.addDefeatGoal()  
  
# Spawn at least 2 "munchkin"s.  
  
# Spawn at least 2 "thrower"s.  
  
# Spawn at least 2 "soldier"s.  
  
# Spawn at least 2 "archer"s.
```

### Overview

*Coming soon!*

## Army Training Solution

```
# Spawn a hero and add a goal.  
game.spawnHeroXY("champion", 40, 15)  
game.addDefeatGoal()  
  
# Spawn at least 2 "munchkin"s.  
game.spawnXY("munchkin",32,32)  
game.spawnXY("munchkin",29,43)  
  
# Spawn at least 2 "thrower"s.  
game.spawnXY("thrower",21,34)  
game.spawnXY("thrower",23,46)  
  
# Spawn at least 2 "soldier"s.  
game.spawnXY("soldier",52,31)  
game.spawnXY("soldier",51,41)  
  
# Spawn at least 2 "archer"s.  
game.spawnXY("archer",62,35)  
game.spawnXY("archer",59,42)
```

---

# #10. Ranger Danger

## Level Overview and Solutions

### Intro

Archers are powerful ranged units! Use them instead of soldiers to deal lots of damage.

Be careful, though, as archers don't have as much health as soldiers do!

### Default Code

```
# Archers are powerful ranged units!
# While soldiers have health, archers provide damage.

# Too many soldiers, not enough archers!
game.spawnXY("soldier", 20, 40)
game.spawnXY("soldier", 20, 30) # Δ Remove this line.
game.spawnXY("soldier", 20, 20) # Δ Remove this line.

game.spawnXY("archer", 10, 30)
# Add 2 more archers:
```

### Overview

When adding units to your level, remember there are different types of units which have different stats!

Archers are long-ranged damage dealers, but they have low health.

Soldiers are close-ranged tanky units, with lots of health.

In this level, compare the brawler's health between 3 soldiers and 1 archer, versus 1 soldier and 3 archers!

## Ranger Danger Solution

```
# Archers are powerful ranged units!
# While soldiers have health, archers provide damage.

# Too many soldiers, not enough archers!
game.spawnXY("soldier", 20, 40)

game.spawnXY("archer", 10, 30)
# Add 2 more archers:
game.spawnXY("archer", 20, 30)
game.spawnXY("archer", 20, 20)
```

## #11. Hedge Magic

# Level Overview and Solutions

### Intro

Spawn a maze with:

```
game.spawnMaze(1)
```

Change the number 1 to a different number, and you'll get different mazes.

The maze for each number will always be the same!

### Default Code

```
# Spawn a maze. Change the number for a different maze!
game.spawnMaze(1)

# Spawn a hero with spawnHeroXY(type, x, y)

# Add at least one goal!
```

### Overview

*Coming soon!*

## Hedge Magic Solution

```
# Spawn a maze. Change the number for a different maze!
game.spawnMaze(1234567890)
```

```
# Spawn a hero with spawnHeroXY(type, x, y)
game.spawnHeroXY("duelist", 12, 13)
# Add at least one goal!
game.addMoveGoalXY(60, 59)
```

---

## #12. Forest Incursion

# Level Overview and Solutions

### Intro

When you call `spawnHeroXY()`, it returns a hero object for you to modify.

You can then change properties of the hero, like `maxSpeed`, `maxHealth`, `health`, and `attackDamage`!

```
hero = game.spawnHeroXY("goliath", 20,20)
hero.maxSpeed = 12
hero.maxHealth = 300
hero.health = 300
hero.attackDamage = 9000
```

### Default Code

```
# Okar needs to stomp out these annoying little munchkins!
# Unfortunately he is slow, and his attacks do little damage.
# Fortunately, as a game developer, you have full control over the world!
# Set Okar's properties to buff him up for ogre slaying!

game.addDefeatGoal()
game.addSurviveGoal()
hero = game.spawnHeroXY("goliath", 12, 10)
# Increase the hero's maxSpeed, so he runs faster.
hero.maxSpeed = 25
# Increase the hero's maxHealth, so he lasts longer.

# Increase the hero's attackDamage, so he can quickly take down the ogres.
```

## Overview

*Coming soon!*

## Forest Incursion Solution

```
# Okar needs to stomp out these annoying little munchkins!
# Unfortunately he is slow, and his attacks do little damage.
# Fortunately, as a game developer, you have full control over the world!
# Set Okar's properties to buff him up for ogre slaying!

game.addDefeatGoal()
game.addSurviveGoal()
hero = game.spawnHeroXY("goliath", 12, 10)
# Increase the hero's maxSpeed, so he runs faster.
hero.maxSpeed = 25
# Increase the hero's maxHealth, so he lasts longer.
hero.maxHealth = 2000
# Increase the hero's attackDamage, so he can quickly take down the ogres.
hero.attackDamage = 99
```

## #13. Throwing Fire

### Level Overview and Solutions

#### Intro

Some game objects can be configured by setting their properties.

A "fire-spewer" shoots a bunch of fireballs!

By default it shoots them "horizontal" ly.

You can change it to shoot "vertical" ly like this:

```
spew = game.spawnXY("fire-spewer", 40, 40)
spew.direction = "vertical"
```

#### Default Code

```
# Game objects can be configured by setting properties  
  
# Don't change this, it sets up the game.  
player = game.spawnHeroXY("knight", 40, 10)  
game.addCollectGoal()  
game.addSurviveGoal()  
  
game.spawnXY("gem", 32, 55)  
game.spawnXY("gem", 51, 55)  
  
fs1 = game.spawnXY("fire-spewer", 12, 25)  
fs2 = game.spawnXY("fire-spewer", 70, 30)  
fs3 = game.spawnXY("fire-spewer", 12, 35)  
fs4 = game.spawnXY("fire-spewer", 70, 40)  
  
# Change fs1.direction to "vertical":  
fs1.direction = "horizontal"  
  
# Now set fs2.direction to "vertical":  
  
# Do the same for fs3 and fs4:  
  
# Now play the game and collect the gems!
```

## Overview

*Coming soon!*

## Throwing Fire Solution

```
# Game objects can be configured by setting properties

# Don't change this, it sets up the game.
player = game.spawnHeroXY("knight", 40, 10)
game.addCollectGoal()
game.addSurviveGoal()

game.spawnXY("gem", 32, 55)
game.spawnXY("gem", 51, 55)

fs1 = game.spawnXY("fire-spewer", 12, 25)
fs2 = game.spawnXY("fire-spewer", 70, 30)
fs3 = game.spawnXY("fire-spewer", 12, 35)
fs4 = game.spawnXY("fire-spewer", 70, 40)

# Change fs1.direction to "vertical":
fs1.direction = "vertical"

# Now set fs2.direction to "vertical":
fs2.direction = "vertical"

# Do the same for fs3 and fs4:
fs3.direction = "vertical"
fs4.direction = "vertical"

# Now play the game and collect the gems!
```

## #14. Them Bones

### Level Overview and Solutions

#### Intro

A "generator" is a building that spawns enemies every so often.

By default, the "generator" spawns "skeleton"s, a powerful enemy that will attack both humans and ogres!

But "skeleton"s have a weakness to "lightstone"s.

If the player picks up a "lightstone", skeletons will flee from the player, buying enough time to destroy the "generator".

#### Default Code

```
# Generators spawn enemies over time.  
# Skeletons are afraid of lightstones.  
  
player = game.spawnHeroXY("champion", 15, 35)  
player.attackDamage = 60  
player.maxSpeed = 8  
  
game.addSurviveGoal()  
game.addDefeatGoal()  
  
# Spawn a "generator"  
  
# Spawn a "lightstone"  
  
# Now beat your game!
```

## Overview

You can configure the behavior of a "generator" by setting its properties.

```
generator = game.spawnXY("generator", 20, 20)  
  
generator.spawnType = "skeleton"  
generator.spawnDelay = 5
```

`generator.spawnType` is a string, representing the type of enemy to spawn.

`generator.spawnDelay` is a number, representing the delay in seconds between spawns.

## Them Bones Solution

```
# Generators spawn enemies over time.  
# Skeletons are afraid of lightstones.  
  
player = game.spawnHeroXY("champion", 15, 35)  
player.attackDamage = 60  
player.maxSpeed = 8  
  
game.addSurviveGoal()  
game.addDefeatGoal()  
  
# Spawn a "generator"  
game.spawnXY("generator", 45, 35)  
  
# Spawn a "lightstone"  
game.spawnXY("lightstone", 25, 45);  
  
# Now beat your game!
```

# #15. Behavior Driven Development

## Level Overview and Solutions

### Intro

You can change a unit's behavior by setting its `behavior` property.

`unit.behavior = "AttacksNearest"` will attack the nearest enemy.

`unit.behavior = "Defends"` will attack only if an enemy comes close.

`unit.behavior = "Scampers"` will make the unit move around randomly.

`unit.behavior = "RunsAway"` will make the unit run away from the player.

### Default Code

```
# Assign behaviors to units with the behavior property.
```

```
skeleton1 = game.spawnXY("skeleton", 60, 50)
skeleton2 = game.spawnXY("skeleton", 60, 40)
skeleton3 = game.spawnXY("skeleton", 60, 30)
skeleton4 = game.spawnXY("skeleton", 60, 20)
skeleton5 = game.spawnXY("skeleton", 60, 10)
```

```
skeleton1.behavior = "Scampers"
skeleton2.behavior = "Scampers"
skeleton3.behavior = "Scampers"
# Assign "Scampers" to skeleton4.behavior
```

```
# Assign "Scampers" to to skeleton5.behavior
```

```
ogre1 = game.spawnXY("ogre", 70, 50)
ogre2 = game.spawnXY("ogre", 70, 30)
ogre3 = game.spawnXY("ogre", 70, 10)
```

```
ogre1.behavior = "AttacksNearest"
# Assign "AttacksNearest" to ogre2.behavior
```

```
# Assign "AttacksNearest" to ogre3.behavior
```

```
archer1 = game.spawnXY("archer", 10, 30)
# Assign "Defends" to archer1.behavior
```

```
# Don't need to change anything below here.
# But feel free to take a look!
player = game.spawnHeroXY("raider", 20, 30)
player.attackDamage = 10
game.addSurviveGoal()
game.addDefeatGoal()
```

```
game.spawnXY("forest", 40, 10)
game.spawnXY("forest", 40, 18)
game.spawnXY("forest", 40, 26)
game.spawnXY("forest", 40, 42)
game.spawnXY("forest", 40, 50)
game.spawnXY("forest", 40, 58)
```

```
game.spawnXY("lightstone", 30, 45)
```

```
game.spawnXY("lightstone", 30, 20)
game.spawnXY("lightstone", 30, 55)
game.spawnXY("lightstone", 30, 10)

game.spawnXY("potion-medium", 10, 50)
```

## Overview

*Coming soon!*

## Behavior Driven Development Solution

```
# Assign behaviors to units with the behavior property.

skeleton1 = game.spawnXY("skeleton", 60, 50)
skeleton2 = game.spawnXY("skeleton", 60, 40)
skeleton3 = game.spawnXY("skeleton", 60, 30)
skeleton4 = game.spawnXY("skeleton", 60, 20)
skeleton5 = game.spawnXY("skeleton", 60, 10)

skeleton1.behavior = "Scampers"
skeleton2.behavior = "Scampers"
skeleton3.behavior = "Scampers"
# Assign "Scampers" to skeleton4.behavior
skeleton4.behavior = "Scampers"
# Assign "Scampers" to skeleton5.behavior
skeleton5.behavior = "Scampers"

ogre1 = game.spawnXY("ogre", 70, 50)
ogre2 = game.spawnXY("ogre", 70, 30)
ogre3 = game.spawnXY("ogre", 70, 10)

ogre1.behavior = "AttacksNearest"
# Assign "AttacksNearest" to ogre2.behavior
ogre2.behavior = "AttacksNearest"
# Assign "AttacksNearest" to ogre3.behavior
ogre3.behavior = "AttacksNearest"

archer1 = game.spawnXY("archer", 10, 30)
# Assign "Defends" to archer1.behavior
archer1.behavior = "Defends"

# Don't need to change anything below here.
# But feel free to take a look!
player = game.spawnHeroXY("raider", 20, 30)
player.attackDamage = 10
game.addSurviveGoal()
game.addDefeatGoal()

game.spawnXY("forest", 40, 10)
game.spawnXY("forest", 40, 18)
game.spawnXY("forest", 40, 26)
game.spawnXY("forest", 40, 42)
game.spawnXY("forest", 40, 50)
game.spawnXY("forest", 40, 58)

game.spawnXY("lightstone", 30, 45)
```

```
game.spawnXY("lightstone", 30, 20)
game.spawnXY("lightstone", 30, 55)
game.spawnXY("lightstone", 30, 10)

game.spawnXY("potion-medium", 10, 50)
```

---

## #16. Time To Live

### Level Overview and Solutions

#### Intro

`game.addSurviveGoal()` sets up a goal for the player to survive until all other goals are complete.

You can pass a number as an argument like: `game.addSurviveGoal(20)` to configure it so that the player has to survive for 20 seconds.

#### Default Code

```
# Pass an argument to addSurviveGoal() to specify a time.  
  
# This means the player must survive for 20 seconds.  
game.addSurviveGoal(20)  
  
# Spawn a generator with spawnXY  
# Remember to assign the spawned generator to a variable!  
# Use the variable to configure the generator below.  
  
# Set the generator's spawnType to "munchkin"  
  
# Use spawnHeroXY to spawn a hero for the player.  
# Remember to assign the spawned hero to a variable!  
  
# Set the hero's maxHealth to 100  
  
# Set the hero's attackDamage to 10  
  
# Play the game!
```

## Overview

*Coming soon!*

## Time To Live Solution

```
# Pass an argument to addSurviveGoal() to specify a time.

# This means the player must survive for 20 seconds.
game.addSurviveGoal(20)

# Spawn a generator with spawnXY
generator = game.spawnXY("generator", 60, 40)
# Set the generator's spawnType to "munchkin"
generator.spawnType = "munchkin"
# Use spawnHeroXY to spawn a hero for the player.
player = game.spawnHeroXY("knight", 15, 15)
# Set the hero's maxHealth to 100
player.maxHealth = 100
# Set the hero's attackDamage to 10
player.attackDamage = 10

# Play the game!
```

# #17. Seeing is Believing

## Level Overview and Solutions

### Intro

Use the `ui.track` function to display an object's property for the player.

Be sure to beat up 10 munchkins before the 20-second survival ends.

### Default Code

```
# Players like seeing score, so use ui.track()!
# It will create a user-interface element for them to see.
hero = game.spawnHeroXY("samurai", 20, 20)

spawner = game.spawnXY("generator", 50, 50)
spawner.maxHealth = 9001
spawner.spawnType = "munchkin"

# ui.track() displays an object's property for players to see!
ui.track(game, "time")

game.addSurviveGoal(20)

# Use ui.track to track game's "defeated" property:

# Change the hero's attackDamage and maxSpeed:

# Add more spawners for more enemies on the field:

# Press play and defeat 10 munchkins or skeletons!
```

## Overview

The `ui` object contains helpful methods to control the player's user-interface. One particular method `track()` is used to display an object's properties in real time for the player to observe.

For example, if you want to show the user how long they've been playing the game, the `game` object has a `time` property, so use:

```
ui.track(game, "time") # Will display the elapsed time in seconds
```

For this level you should add a `ui.track` for "defeated" and defeat 10 munchkins in under 20 seconds!

## Seeing is Believing Solution

```
# Players like seeing score, so use ui.track()!
# It will create a user-interface element for them to see.
hero = game.spawnHeroXY("samurai", 20, 20)

spawner = game.spawnXY("generator", 50, 50)
spawner.maxHealth = 9001
spawner.spawnType = "munchkin"

# ui.track() displays an object's property for players to see!
ui.track(game, "time")

game.addSurviveGoal(20)

# Use ui.track to track game's "defeated" property:
ui.track(game, "defeated")
# Change the hero's attackDamage and maxSpeed:
hero.attackDamage = 100
hero.maxSpeed = 20
# Add more spawners for more enemies on the field:
spawner = game.spawnXY("generator", 40, 50)
spawner.spawnType = "munchkin"
spawner = game.spawnXY("generator", 30, 50)
spawner.spawnType = "munchkin"
# Press play and defeat 10 munchkins or skeletons!
```

## #18. Persistence Pays

### Level Overview and Solutions

#### Intro

A database can store information between plays of your game.

`db.set("defeated", game.defeated)` sets the "defeated" number in the database to the value of `game.defeated`.

`db.add("plays", 1)` adds 1 to the "plays" number stored in the database.

Then use `ui.track(db, "plays")` to display the "plays" count!

#### Default Code

```
# You can use a database to store persistent data.  
# Persistent data stays the same between plays of your game!  
  
generator = game.spawnXY("generator", 60, 40)  
generator.spawnType = "munchkin"  
player = game.spawnHeroXY("raider", 15, 15)  
player.maxHealth = 80  
player.attackDamage = 10  
game.addSurviveGoal(20)  
  
# db stands for database  
# db.add(key, value) increments a value stored in the database.  
# This adds 1 to the "plays" key in the database.  
db.add("plays", 1)  
  
# Show the value of the "plays" key in the db  
ui.track(db, "plays")  
  
# Show the value of the "defeated" property of the game object  
  
# The code below will run when the player wins the game.  
def onVictory(event):  
    pass  
    # Use db.add(key, value) to add the value of  
    # game.defeated to the database with the key "defeated"  
  
game.on("victory", onVictory)
```

## Overview

*Coming soon!*

## Persistence Pays Solution

```
# You can use a database to store persistent data.  
# Persistent data stays the same between plays of your game!  
  
generator = game.spawnXY("generator", 60, 40)  
generator.spawnType = "munchkin"  
player = game.spawnHeroXY("raider", 15, 15)  
player.maxHealth = 80  
player.attackDamage = 10  
game.addSurviveGoal(20)  
  
# db stands for database  
# db.add(key, value) increments a value stored in the database.  
# This adds 1 to the "plays" key in the database.  
db.add("plays", 1)  
  
# Show the value of the "plays" key in the db  
ui.track(db, "plays")  
  
# Show the value of the "defeated" property of the game object  
ui.track(game, "defeated")  
  
# The code below will run when the player wins the game.  
def onVictory(event):  
    # Use db.add(key, value) to add the value of  
    # game.defeated to the database with the key "defeated"  
    db.add("defeated", game.defeated)  
  
game.on("victory", onVictory)
```

## #19. Tabula Rasa

# Level Overview and Solutions

### Intro

Now it's time to build your own game!

Be creative - combine all the commands you've learned into a fun mini-game.

When you're done, you can share your game with your friends!

### Default Code

```
# Create your own game!
# Spawn a hero with spawnHeroXY(type, x, y)
# Add at least one goal!
# Spawn objects into the game with spawnXY(type, x, y)
```

### Overview

*Coming soon!*

## Tabula Rasa Solution

*Coming soon!*

---