

EXPLORE

THE ENGINEERING DESIGN CYCLE

ASK

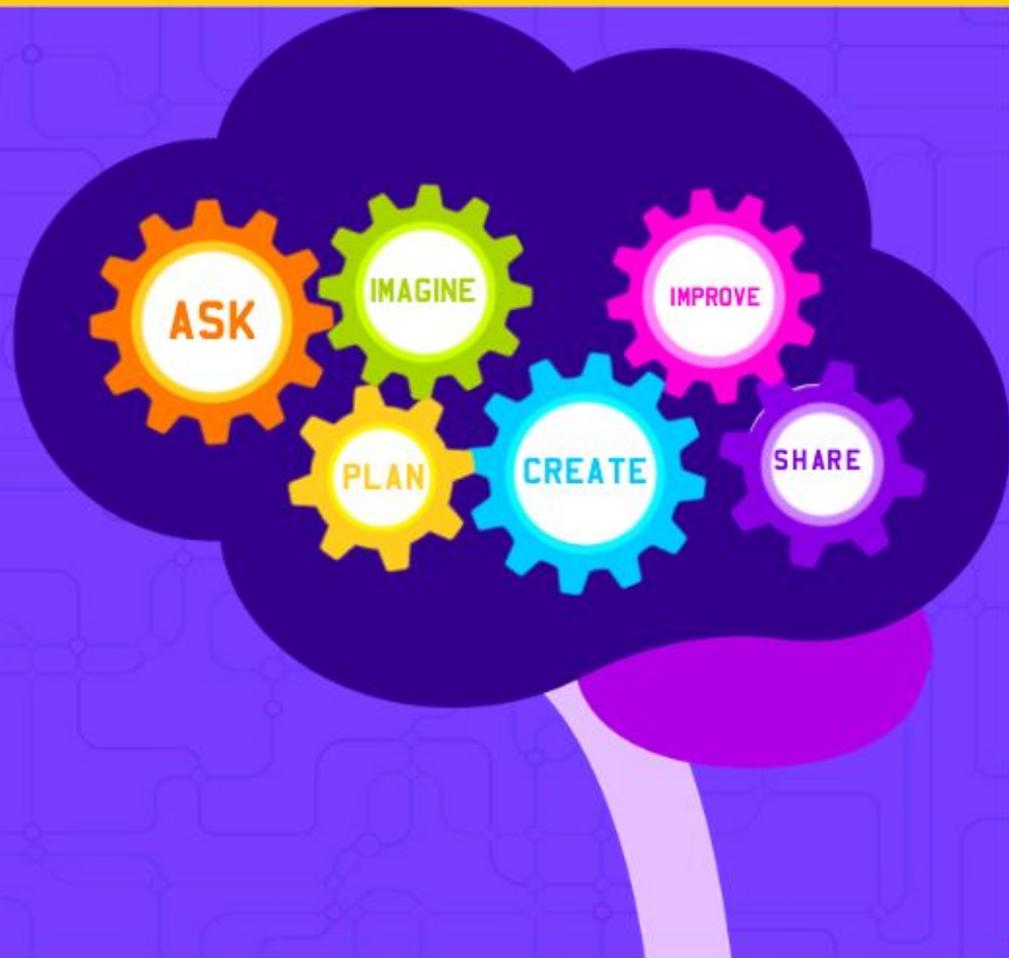
IMAGINE

PLAN

CREATE

IMPROVE

SHARE



EXPLORE

THE ENGINEERING DESIGN CYCLE

ASK

IMAGINE

PLAN

CREATE

IMPROVE

SHARE

ASK – Define the Task

GOT IT!

The first step in the engineering design cycle (EDC) is the “ASK”. This is where we define the task at hand and address the constraints of the project.



GAME DEVELOPMENT CONNECTION

When game developers are instructed to build a game, they have many constraints. Some may include time, money, and resources such as software. They must take this into consideration at the beginning of each project.

ASK

Define the Task

Address the constraints

Time

Money

Resources

EXPLORE

THE ENGINEERING DESIGN CYCLE

ASK

IMAGINE

PLAN

CREATE

IMPROVE

SHARE

IMAGINE - Brainstorm!

GOT IT!

The next step is to **IMAGINE!** This is where all the initial ideas are discussed and considered. Collaboration is an extremely important part of this step!



GAME DEVELOPMENT CONNECTION

It's very rare that games are developed in total isolation, especially large games! Working together to come up with ideas is more effective than isolating your own ideas. No idea is impossible during this stage!

EXPLORE

THE ENGINEERING DESIGN CYCLE

ASK

IMAGINE

PLAN

CREATE

IMPROVE

SHARE

PLAN – Determine Solution

GOT IT!

Next, you must make a **PLAN** based off your ideas during the Imagine step of the EDC. Determine your plan of action and estimate what you'll need and how long it'll take.



GAME DEVELOPMENT CONNECTION

The saying “make a plan or plan to fail” certainly applies to game design! Often times, if a project doesn't get finished or built well, the developers don't get paid! Planning keeps us on track – which makes for a better game!

EXPLORE

THE ENGINEERING DESIGN CYCLE

ASK

IMAGINE

PLAN

CREATE

IMPROVE

SHARE

CREATE – Build the Game

GOT IT!

Now, it's time to bring your ideas to life! In the **CREATE** step of the engineering design cycle, you put your plan to work! In this step you'll find out whether your plan works...or not!



GAME DEVELOPMENT CONNECTION

Building games is typically a time consuming task. Depending on the complexity of the game, a build may take weeks, months, or even years to accomplish. In this step, developers make note of potential improvements.

EXPLORE

THE ENGINEERING DESIGN CYCLE

ASK

IMAGINE

PLAN

CREATE

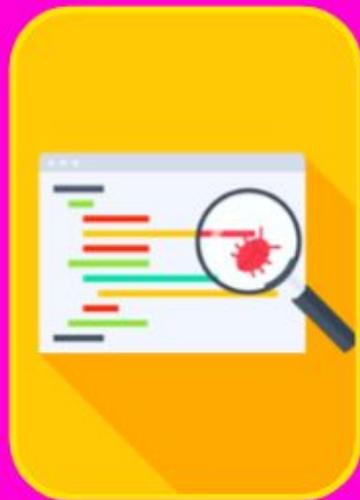
IMPROVE

SHARE

IMPROVE – Make It Better

GOT IT!

During the **IMPROVE** step of the EDC, the goal is to improve upon what was created during the Create step. Think about it this way – no project is ever done, it can always be better!



GAME DEVELOPMENT CONNECTION

Game development relies on the user experience – which leaves little room for error. Luckily, by using the EDC this can be avoided by fixing any bugs prior to launching the game. The more time spent in this step, the better!

EXPLORE

THE ENGINEERING DESIGN CYCLE

ASK

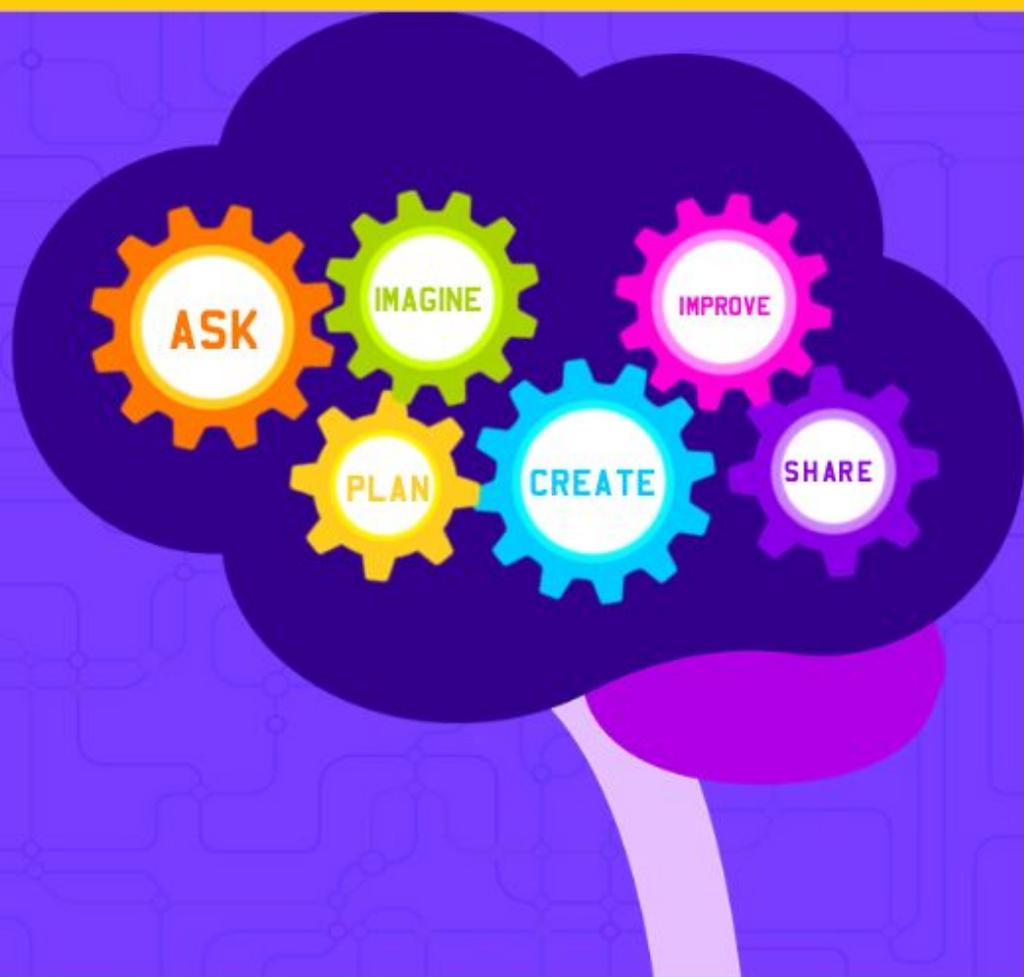
IMAGINE

PLAN

CREATE

IMPROVE

SHARE



EXPLORE

THE ENGINEERING DESIGN CYCLE

ASK

IMAGINE

PLAN

CREATE

IMPROVE

SHARE

SHARE – Play the Game

GOT IT!

The final step of the engineering design cycle is to **SHARE** your game! This can be with a focus group, classmates, or family members. Here you can gain feedback and repeat the cycle!



GAME DEVELOPMENT CONNECTION

By sharing your game with others, you can gain extremely valuable feedback from their gameplay. Large game companies typically hire individuals to test their games before bringing them to the public marketplace.

Abstraction

Last updated: May 21, 2020

What Does Abstraction Mean?

Abstraction is a fundamental principle in some types of computer science. It is a key design aspect of object-oriented programming languages and application programming interfaces. It's also one of the least understood ideas in programming, partially for semantic reasons.

Abstraction is commonly defined as the extraction of relevant information from a larger data set, where utilizing abstraction allows engineers and others to simplify a codebase.

<https://www.techopedia.com/definition/3736/abstraction-computer-science>

Question 7

5 pts

Match the definition on the left with the correct term from the dropdown list.

The measure of an object's ability to increase speed

Acceleration



Isolating a problem, ignoring the irrelevant

Abstraction



Breaking down a task into manageable pieces

Decomposition



The speed and direction of an object

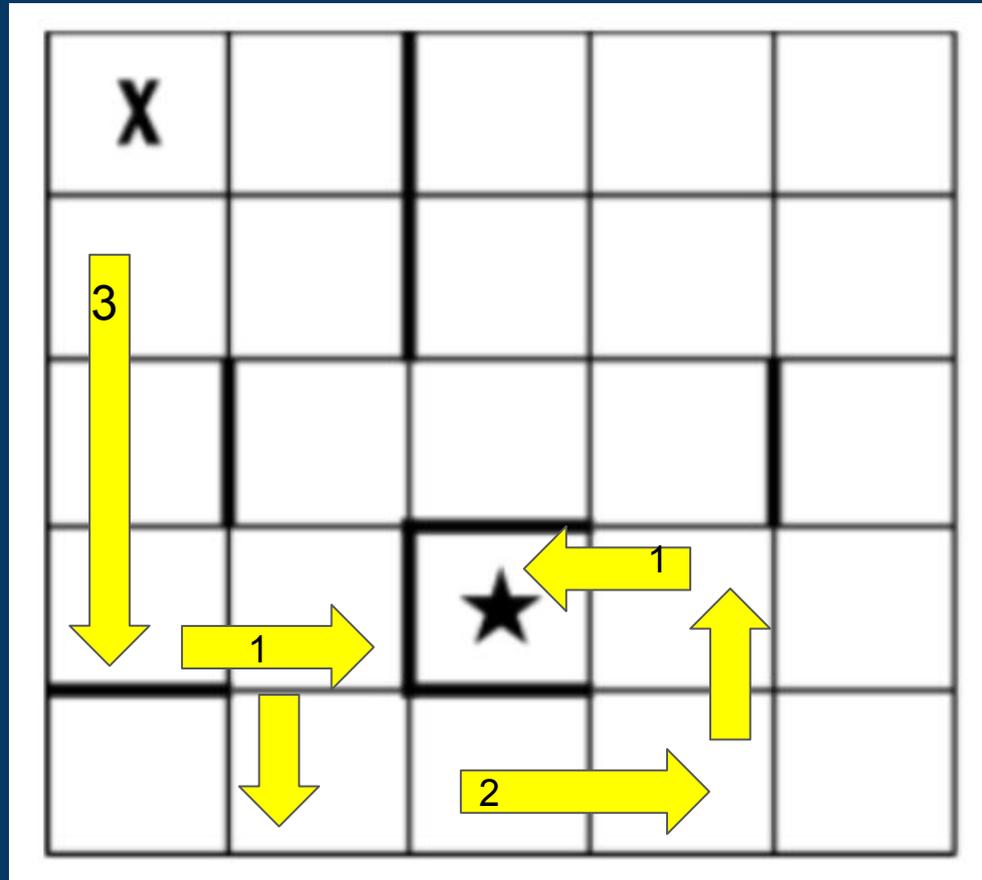
Velocity



A series of steps to solve a problem

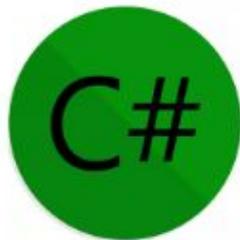
Algorithm





3 down
1 right
1 down
2 right
1 up
1 left

🧙‍♂️ 60% OFF BLACK FRIDAY SALE 650+ books, courses & coding



C# was created by Microsoft. It has many of the same advantages as C++, and it's a particularly good choice for developing 2D and 3D games.

One advantage of C# over C++ is that it's easier to learn. C# is also a cross-platform language, which means that games developed with C# can be run on Windows, macOS, and Linux.

The object-oriented programming language C++ is the most popular for game design. For a 3D first-person shooter, a language like C++ or C# would be useful.

C#, that was developed by Microsoft, can be used to make games that will work on many platforms. It is used for the game engine Unity.

JavaScript is used to develop games that run on web browsers, and would also be fine for a 2D platformer.

Java, is a platform-independent language. Its games can be run on any operating system, including Windows, macOS, and Linux. It is a popular choice for developing mobile games, for iOS devices and Android. This one disadvantage is that it's not as fast as C++.

Python is a scripting language that is fairly simple to learn, can be used to make games, and has other applications in data science and A.I.