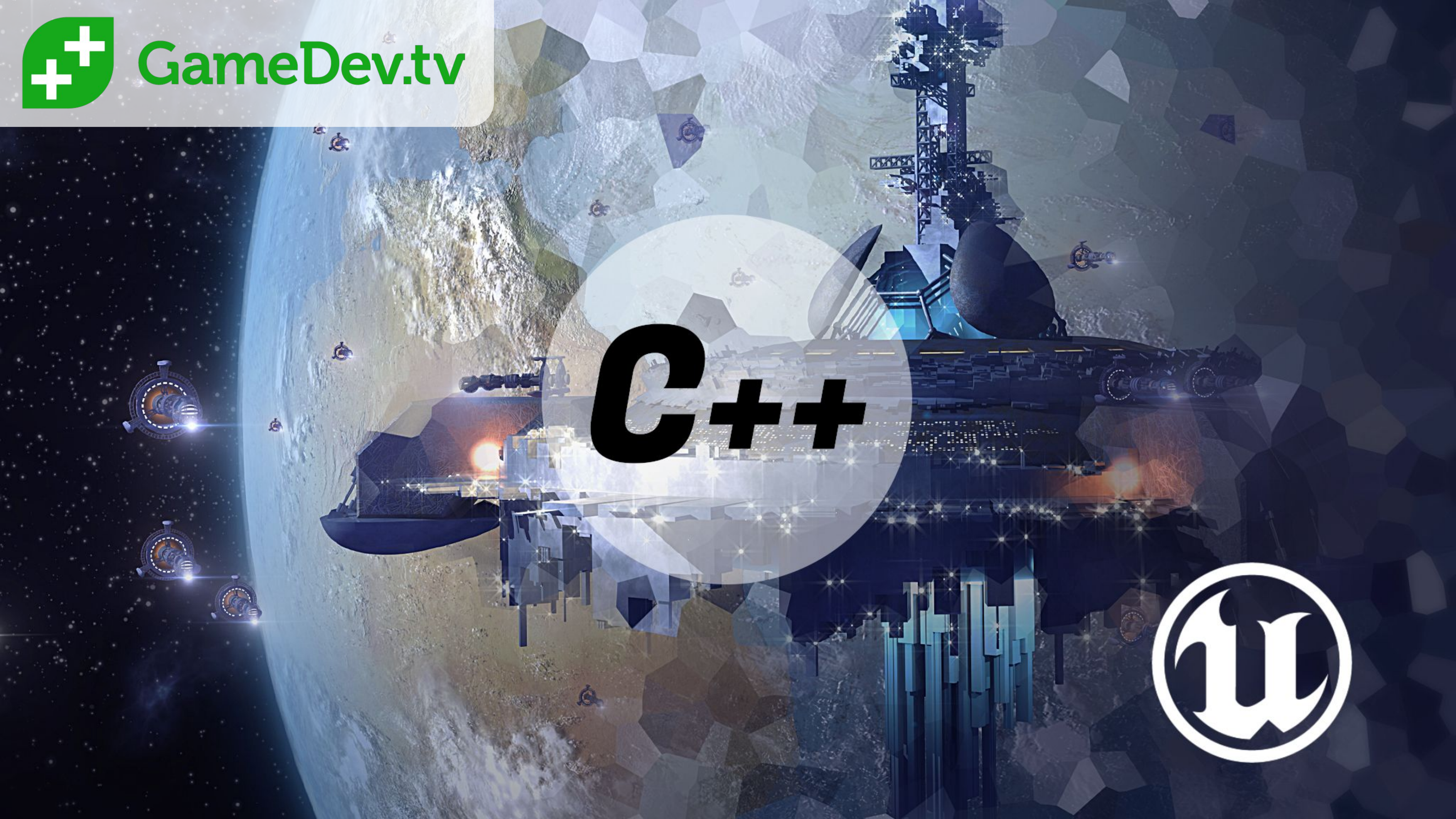
GameDev.tv



GameDev.tv

Section 3: Bull Cow Game

gdev.tv/uc2slides

GameDev.tv

Introduction To Bull Cow Game

gdev.tv/uc2slides

Introducing The Bull Cow Project

- Understand the game we are going to create.
- It’s a terminal game, but within Unreal.
- Allows us to use Unreal specific C++.

How Does The Game Work?

- Word guessing game.
- Guess the Isogram...
- Letter in guess and right place is a “Bull”.
- Letter in guess, but wrong place is a “Cow”.

Mechanics Of The Game

- Hidden word pulled from a list.
- Game will let us know the number of lives we have.
- Player will make a guess.
- Lose a life if you get it wrong.

Mechanics Of The Game

- Display how many bulls and cows.
- If the correct word is guessed then the game is won.
- If the lives run out, the game is over.

GameDev.tv

Importing Bull Cow Game

gdev.tv/uc2slides

Download the Bull Cow Assets

- Checking you have UE 4.22 or greater.
- Download assets from Udemy resources.
- Make sure the project works.

Open Bull Cow Game

- Make sure you are using Unreal Engine 4.22
- Download the BullCowGame-starter-kit.zip
- Extract the Files
- Check you can launch the game.

GameDev.tv

Importing Bull Cow Game (GameDev.tv)

gdev.tv/uc2slides

Download the Bull Cow Assets

- Checking you have UE 4.22 or greater.
- Download BullCow.
- Make sure the project works.

Open Bull Cow Game

- Make sure you are using Unreal Engine 4.22
- Download the BullCowGame-starter-kit.zip
- Extract the Files
- Check you can launch the game.

GameDev.tv

Game Module X Could Not Be Loaded

gdev.tv/uc2slides

GameDev.tv

Helping Us Help You

gdev.tv/uc2slides

Asking For Help

- Give us your project log.
- Help answer other students questions.
- Be specific.
- Tell us what you have tried.
- Let us know when you have solved your issue.

GameDev.tv

A Look Around Unreal

gdev.tv/uc2slides

Playing With The Interface

- Play with the interface.
- Understand how to set it up just right for your setup.

Get Comfortable With Unreal

- Dock and Undock some windows.
- Practice reorganising the layout.
- Remember you can reset the layout under Window > Reset Layout.

GameDev.tv

Controlling The Viewport

gdev.tv/uc2slides

Viewport Control

- Create multiple viewports.
- Use LMB RMB and MMB to navigate.
- User RMB and WASD, EQ to move around the level and the scroll wheel adjusts the speed.

Viewport Control

- Get comfortable moving around your level.
- Share other movement techniques.

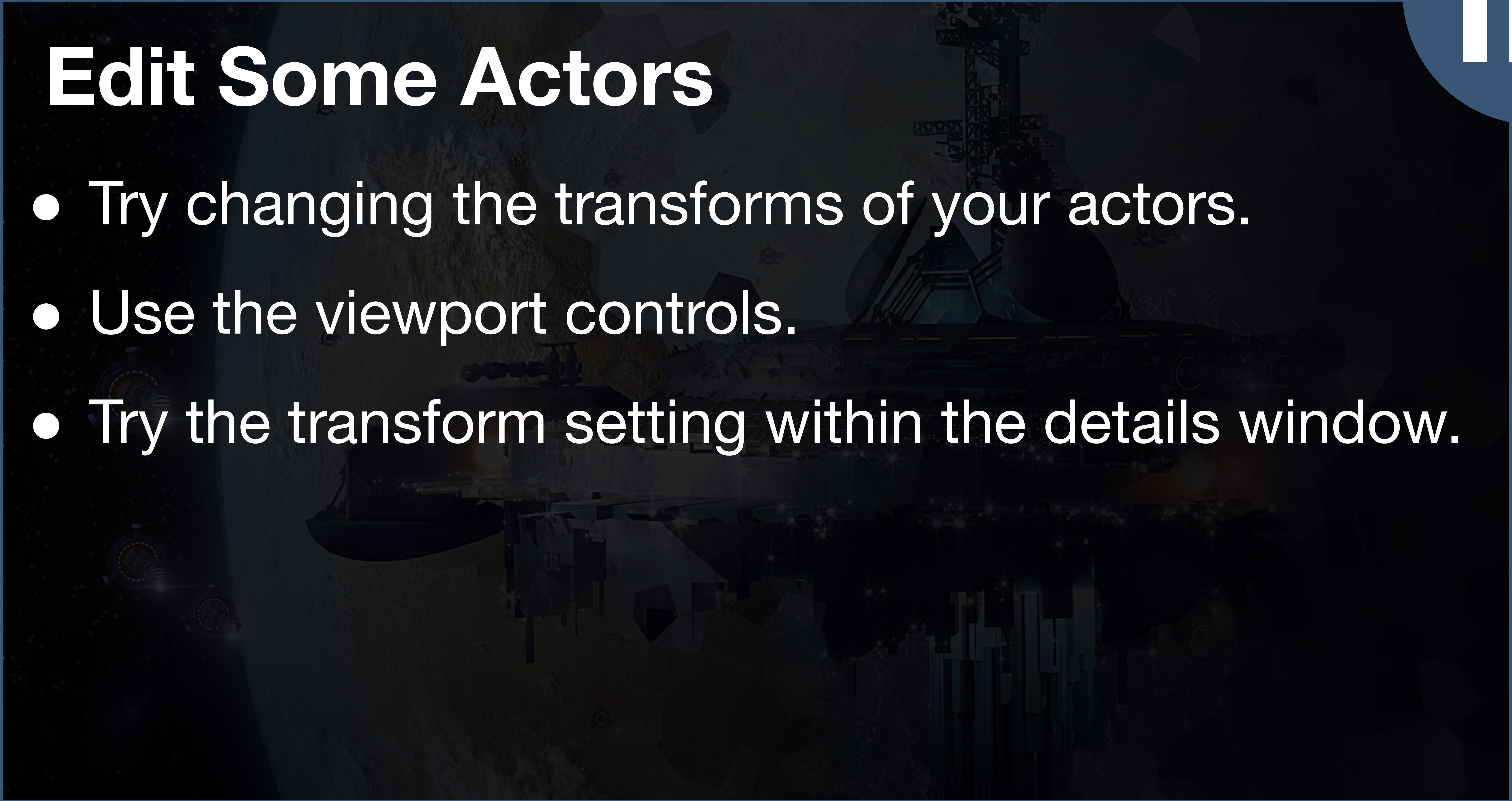
GameDev.tv

Editing Actors In Our Level

gdev.tv/uc2slides

Actors

- Any object that can be placed in your level is called an Actor.
- They all have a transform.
- There are several different types of Actor.



Edit Some Actors

- Try changing the transforms of your actors.
- Use the viewport controls.
- Try the transform setting within the details window.

GameDev.tv

Adding Actors To The Level

gdev.tv/uc2slides

Adding Actors

- Duplication.
- Content Browser window.
- Default Actors in the Modes window.

GameDev.tv

Editing the Landscape

gdev.tv/uc2slides

Editing The Landscape

- Explore the different tools available to modify the landscape.

Make The World Your Own

- Edit the landscape.
- Add some additional actors.
- Have fun, you should find yourself much more comfortable in Unreal now.

GameDev.tv

Setting Up VS Code In Unreal

gdev.tv/uc2slides

Setting Up Your IDE In UE4

- Setup UE4 and your IDE.
- Create VS Code project.
- Open and check the project.

Get Your IDE Setup

- Setup your IDE as the default editor in UE4.
- Make sure you can open up:

```
BullCowCartridge.h

BullCowCartridge.cpp
```

GameDev.tv

Actors And Components

gdev.tv/uc2slides

Actors And Components

- An Actor is a container that can have many components.
- There are many different component types.
- We have a special component called the “BullCowCartridge”.

Terminal and Cartridge

Add The BullCowCartridge

- Add The BullCowCartidge component to the Terminal Actor.
- Double check it is there otherwise the code you write will not work.

GameDev.tv

Using The In Game Terminal

gdev.tv/uc2slides

Using The In Game Terminal

- Show you how to print to our terminal in game using `PrintLine("");`.
- Introduce you to `ClearScreen();`.

Clear The Terminal

- Change our current “HI THERE” to a Bull Cows welcome message, welcoming the player.
- Print another line asking the player to input something and press enter.
- In the `OnInput` function, call `ClearScreen();`.

GameDev.tv

Unreal’s Types - FString

gdev.tv/uc2slides

Unreal Has It’s Own C++ Types

- `std::cout << “Welcome”;`
- `std::string Word = “Welcome”;`
- For strings we need to use `FString` instead.
- Later on instead of `int`, we will use `int32`.

The HiddenWord

- Within the `OnInput` function...
- Declare and initialise a `FString` called `HiddenWord`.
- Assign it an isogram of your choosing.

GameDev.tv

The TEXT Macro

gdev.tv/uc2slides

The TEXT Macro

- Encodes our `FString` allowing it to work across multiple platforms.

Wrap Your Strings

- Make sure your strings are using the TEXT() macro.
- `PrintLine(TEXT(“Your String Here”));`
- `FString HiddenWord = TEXT(“cake”);`

GameDev.tv

Bull Cow Basic Game Loop

gdev.tv/uc2slides

What’s A Game Loop?

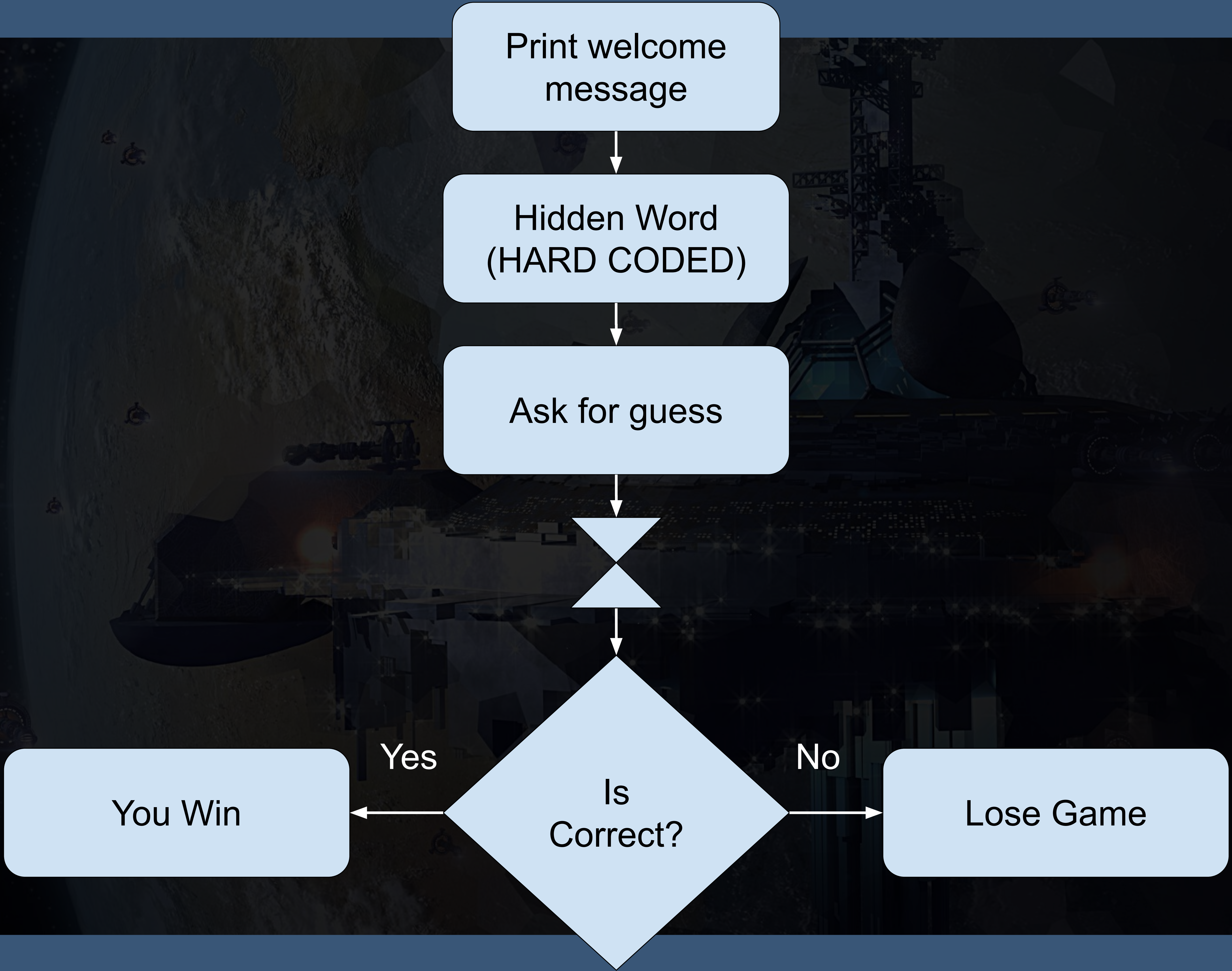
- Process the player will go through.
- There will usually be many loops with in a larger game.
- What is our player’s experience from the moment they interact with the terminal.

Why Write It Down?

- Visualises the process before you start coding. Check your logic.
- Describes your intended behaviour, does your code match?
- Defines the scope of your game.

Write Your Basic Loop

- Sketch the most basic game loop.
- Start game, set hidden word, prompt for guess, is guess correct and win or lose.


Print welcome message
Hidden Word (HARD CODED)
Ask for guess
Is Correct?
Yes
You Win
No
Lose Game

GameDev.tv

Accessing Player Input

gdev.tv/uc2slides

GameDev.tv

Coding The Basic Game Loop

gdev.tv/uc2slides

Medium Challenge

- Write all of your code in the `OnInput` function.
- What do you think about having all our code here, can you foresee any issues?
- Write code that evaluates whether the `Input` is equal to the `HiddenWord`.
- HINT: Use an if, else to check.

GameDev.tv

Instance Variables

gdev.tv/uc2slides

Scope Across Our Class

- We will cover classes in more detail later on.
- `HiddenWord` needs to be used across multiple functions in the `BullCowCartridge` Class.
- “`BCC`” Class creates an instance of “`BCC`”.
- `HiddenWord` can then be used across that instance.

Declare and Initialisation

- Declare `HiddenWord` in the Header file.
- Initialise `HiddenWord` in the `BeginPlay()` function.

GameDev.tv

Solving Slow Compile Times

gdev.tv/uc2slides

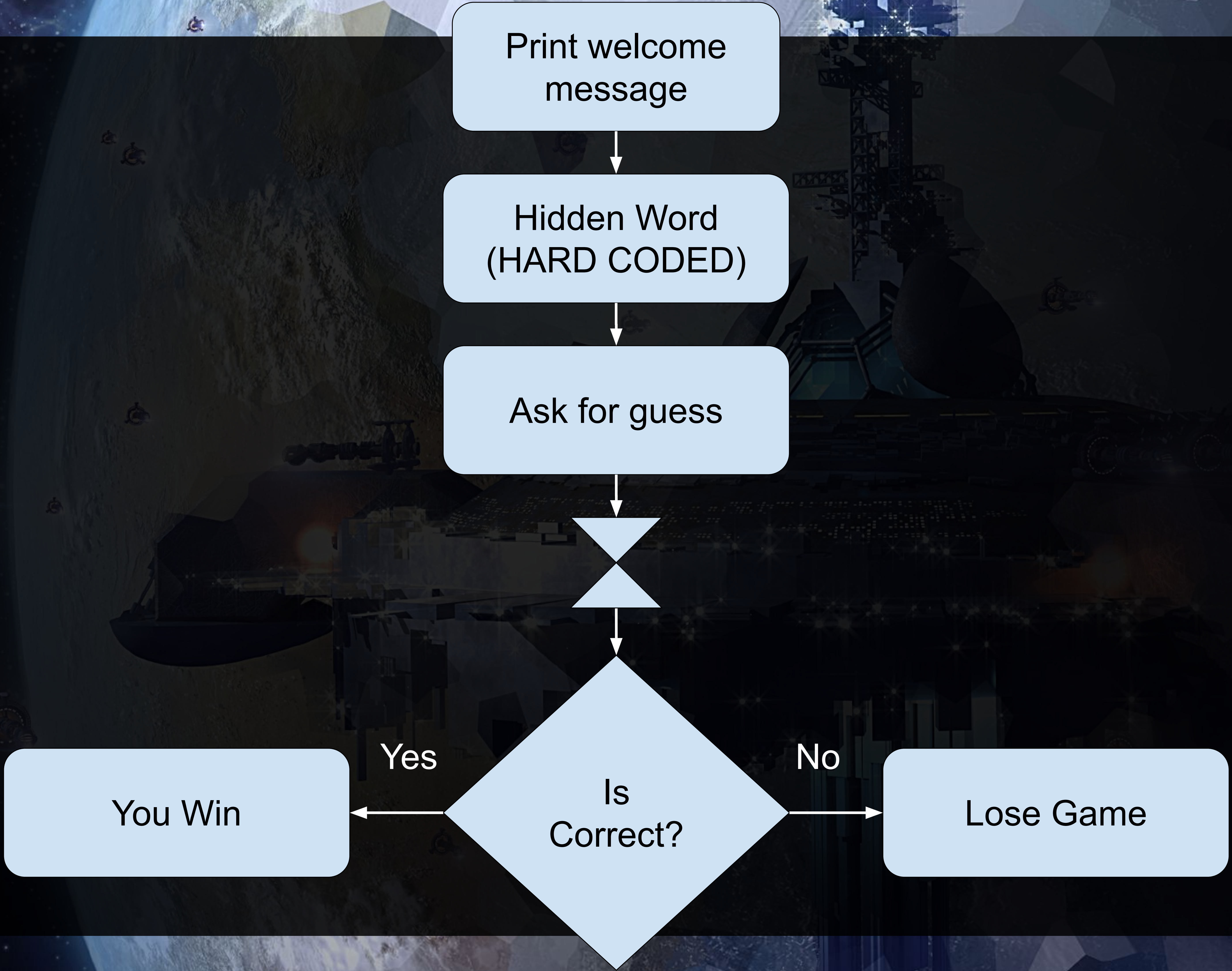
GameDev.tv

The Full Game Loop

gdev.tv/uc2slides

A More Complete Game

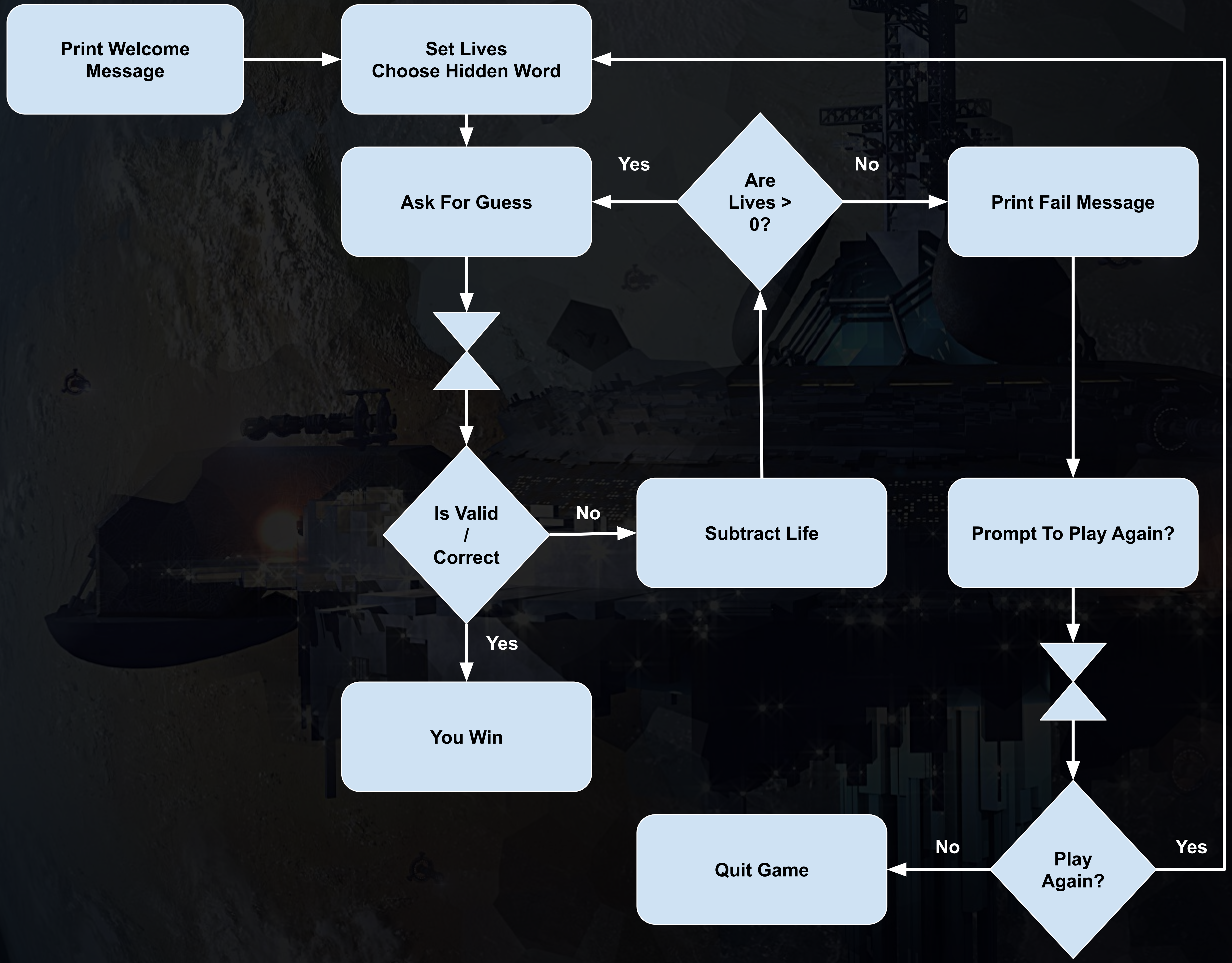
- We have our basic functionality working.
- Let's think about the rest of the game and map out how it will work and it's overall behaviour.



Print welcome message
Hidden Word (HARD CODED)
Ask for guess
Is Correct?
Yes
You Win
No
Lose Game

Additional Game Components

- We need lives.
- Player’s input needs checking.
- Wrong answer should deprecate the lives.
- Lives can run out, what happens then?
- Take your original flow chart, adapt or make a new one incorporating these new elements.


Print Welcome Message
Set Lives
Choose Hidden Word
Ask For Guess
Yes
Are Lives > 0?
No
Print Fail Message
Is Valid / Correct
No
Subtract Life
Prompt To Play Again?
Yes
You Win
Quit Game
No
Play Again?
Yes

GameDev.tv

Pseudo Coding

gdev.tv/uc2slides

Laying Out Our Coding Structure

- Complements the flowchart approach really well.
- Allow you to put down ideas, and keep track of progress.
- Easier to catch logical errors early.
- Detailed, easy to read template for the rest of our code.

Pseudo Code The Game

- Include details, almost like a checklist.
- Add questions if you are unsure about something.
- Run through your code, does it make sense.
- Cross reference it with your flow chart.

GameDev.tv

Code Formatting

gdev.tv/uc2slides

Indentation Rules

- Code blocks are areas between {}
- Code blocks can be with-in other blocks
- Each block is indented one level more
- Use tabs or spaces but be consistent!

Find My Errors

- Go line by line
- Where are the code blocks?
- How should they be indented?

GameDev.tv

Creating Our First Function

gdev.tv/uc2slides

Do Only One Thing

- Functions work best when they are specific.
- The name of the function tells us what it does.
- If you are having trouble naming your functions it might be doing too much.
- `void UBullCowCartridge::InitGame()`

Declare And Initialise Lives

- Declare the member variable `Lives`.
- Initialise Lives in the `InitGame()` function.
- `Lives` will be an integer. Not `int` but `int32`.
- Test your game.

GameDev.tv

Finding And Replacing In VS Code

gdev.tv/uc2slides

Where Is It?

- Find and replace.
- Find and replace across multiple files.
- Use with caution.

GameDev.tv

C++ Dot Operator

gdev.tv/uc2slides

Compare HiddenWord To Input

- Add a check that the number of letters the player types, `Input`, is the same as the `HiddenWord`.
- This will create a nested if statement.
- Equal to is `==`.
- Not Equal to is `!=`.

GameDev.tv

Formatting FStrings

gdev.tv/uc2slides

Inserting Data

- Insert data into our strings using a format specifier.
- We should use the `FString::Printf()` function to do this.
- Format specifiers: `%s` for strings, `%i` for integers.

Static Member Function

- `FString::Printf()`.
- `Printf()`. Doesn’t require an instance of an `FString`.
- `FString::Printf()` does not print anything.
- Still need `PrintLine()` to output to the terminal.

Correct Formatting

- With just a `string`:

```
PrintLine(TEXT("Hello!"));
```

- When inserting values:

```
PrintLine(FString::Printf(TEXT("Hello! %s"), *HiddenWord));
```

Remove The Magic Numbers

- Replace the magic numbers in our code with an actual number.
- Use the hidden words length as the argument.
- You'll need to use `%i`.
- Tidy up your code.

GameDev.tv

Booleans

gdev.tv/uc2slides

True Or False

- Nothing checking whether the game is won.
- Boolean variables in Unreal are prefixed with a “b”.
- Example: `bool bGameOver`.

Better Behaving Code

- Create an `EndGame()` function we can call.
- Get it to set `bGameOver` to true and instruct the player to press enter to continue.
- Implement an if statement checking if the game is over or not and run our existing code if not.
- Get the game to welcome the player again.

GameDev.tv

Pre vs Post Increment / Decrement

gdev.tv/uc2slides

Increment and Decrement Operators

- Pre decrement `++Lives --Lives`.
- Post decrement `Lives++ Lives--`.
- Most of the time you will pre decrement.
- Pre will do the operation and reference the result.
- Post takes a copy, increments the value, but returns the copy from before.

Implement Lives

- Make `Lives` equal the `HiddenWord` length.
- Show number of lives at beginning of the game.
- Take a life when the player guesses incorrectly.
- Let them know they have lost a life.
- Check if lives are greater than 0, if not EndGame().

GameDev.tv

Parameters And Arguments

gdev.tv/uc2slides

Common Confusion

- They are often used interchangeably.
- Parameters are used when defining a function.
- Arguments are the actual values used in the function when called.
- Let’s implement a new guess checking function.

GameDev.tv

Early Returns

gdev.tv/uc2slides

Simpler If Statements

- When running through a series of checks you can return early.
- This helps prevent dense nested If, Else statements.
- This makes our code simpler and easier to read.

Optimising `ProcessGuess()`

- Code your additional checks, using early returns.
- Use code / pseudo code for the remaining checks.
- Where in the process you will decrement a life?
- Use the earlier pseudocode and flowcharts for reference.
- Move pseudo code into the appropriate place.

GameDev.tv

Structure Of An FString

gdev.tv/uc2slides

Strings And Arrays

- A string is a group of characters.
- Strings are varying in length.
- A String is an Array of Characters.
- An `FString` is a `TArray` of `TCHAR`.

Strings And Arrays

- E.g. `FString HiddenWord = TEXT(“cakes”);`
- Array length would be 6, you start from 0!

  `{‘c’,‘a’,‘k’,‘e’,‘s’,‘\0’}`
- `HiddenWord[3]` is “e”
- HiddenWord[5 or greater] will crash Unreal.
- `const TCHAR ArrayOfChars[] = TEXT("cakes");`

Pseudo Coding `IsIsogram()`

- Create the `IsIsogram(something?)` function.
- It will return a boolean.
- Pseudo Code the process that we will go though to check the `FString` characters against each other.

GameDev.tv

Const Member Functions

gdev.tv/uc2slides

Safety First

- By using `const` you protect yourself.
- If a function doesn’t change any member variables of the class, make it `const`.
- Makes sure your functions aren’t modifying your classes when you don't intend them too.
- Also referred to as a “const function”.

`const` Functions

- `const` functions can't call non-const functions.
- `const` objects can only call const functions.

GameDev.tv

Loops In C++

gdev.tv/uc2slides

Looping In C++

- `while`.
- `do while`.
- `for`.
- `for` and `while` are essentially the same.
- `do while`, has a slightly different execution flow.
- Finally, range `for`. A range-based for loop.

While

- `while (condition)`

  ```
  { //Code to be executed }
  ```
- While loops will always check the condition *first*.
- Use these when the amount of loops is unknown.

For

- `for ( init; condition; increment )`

  `{ // Code to be executed. }`

- Use these when you know how many times the loop will be run.

Do While

-
  ```
  do { // Code to be executed }
  while (condition);
  ```
- The code will be executed *once* regardless of the condition.

Print Out Each Character

- Create a for loop that prints out the characters of the `HiddenWord` and `Input` on a new line in our in-game terminal.
- Set the number of iterations to the words length.
- Test one at a time.
- Use the `IsIsogram()` function for this code.

GameDev.tv

Checking Characters Part 1

gdev.tv/uc2slides

Are 2 Characters The Same?

- Create the if statement needed to check the first character against all the others.

GameDev.tv

Checking Characters Part 2

gdev.tv/uc2slides

Are 2 Characters The Same?

- Create a nested for loop to go through each letter and compare it against the others.
- If any of the characters are the same then it fails the test.
- Rewrite it for the practice!
- HINT: 1st loop `Index` then, loop `Comparison`.

Hint 2

```
for Index
{     for Comparison
      {     if ( Condition)
            {
            return something;
            }
      }
}
return true;
```

GameDev.tv

TArray Of Hidden Words

gdev.tv/uc2slides

List Of Words

- We have looked at a TArray before…
- Formatting:

```
const TArray<type> Name =
{ element1, element2... };
```

Create A Hidden Word List.

- Include the HiddenWordList.h in the BullCowCartridge.cpp.
- Include 5 Words for testing.
- Remember the TEXT macro is required.

GameDev.tv

Advanced Cursor Usage In VSCode

gdev.tv/uc2slides

GameDev.tv

TArray Functions .Num()

gdev.tv/uc2slides

.Num()

- `.Num()` is similar to function `.Len()`.
- Returns the number of elements in the array.
- `Words.Num();`.

The First 5 Words

- Write a for loop that prints out the first 5 words from `Words`.

GameDev.tv

Logical Operators

gdev.tv/uc2slides

NOT, AND and OR

● NOT	!
● AND	&&
● OR	||

No Limits...

- You can nest if loops indefinitely.
- You can string together logical operators.
- This main issue becomes readability of your code.

Check For Length Of Word

- Use your for loop and create an if statement that checks words from the `Words` TArray.
- Check the first 10 words.
- If the word is >= 4 characters and <= 8 characters print them out.

Check For Length Of Word

- `if (!/*Condition*/)`

  `{/*Code To Execute*/}`
- `if (/*Condition*/ && /*Condition*/)`

  `{/*Code To Execute*/}`
- `if (/*Condition*/ || /*Condition*/)`

  `{/*Code To Execute*/}`

GameDev.tv

TArray Functions Adding And Removing

gdev.tv/uc2slides

TArray Functions

- `.Add()` New element to the end of the array.
- `.Emplace()` New element to the end of the array.
- `.Remove()` Removes matching elements.
- `.RemoveAt()` Removes element at index.
- Many more functions for `TArrays` in resources.

Create A Valid Hidden Word List.

- Create a new function called `GetValidWords()` that takes in `Words`.
- Put all of our checking code in there.
- Check for Isograms as well.
- Nested `if` or &&.
- Remember the function needs to return something.

GameDev.tv

Range-Based For Loop

gdev.tv/uc2slides

A New Loop

- Designed for iterating through collections.
- A TArray is a collection!
- Will make our code easier to read.

-
  ```
  for (type TempVar : Data)
  {/*Code To Execute*/}
  ```

Use A Range-Based For Loop

- Swap out our current for loop in `GetValidWords()`.
- Replace with a range-based for loop.
- Check your code as it’ll need changing.
- HINT

GameDev.tv

A Quick Look At Memory

gdev.tv/uc2slides

Primitive Types (A Typical Machine)

- `int32`: a 32 bit integer , 4 Bytes.
- `bool`: a 1 bit, but takes up 1 Byte.
- `char`: 1 byte per character.
- `string`: dynamic at creation = `chars` + ‘`\0`’.
- `float`: 32bit, again 4 Bytes.

Memory Example (0x0000001F)

0x00	0x01	0x02	0x03	0x04	0x05	0x06	0x07
08	00	00	00				
0x08	0x09	0x0A	0x0B	0x0C	0x0D	0x0E	0x0F
	01						
0x10	0x11	0x12	0x13	0x14	0x15	0x16	0x17
20	00	00	00				
0x18	0x19	0x1A	0x1B	0x1C	0x1D	0x1E	0x1F

Memory Example (0x0000001F)

0x00	0x01	0x02	0x03	0x04	0x05	0x06	0x07
08	00	00	00				
0x08	0x09	0x0A	0x0B	0x0C	0x0D	0x0E	0x0F
	01	c	a	k	e	s	\0
0x10	0x11	0x12	0x13	0x14	0x15	0x16	0x17
20	00	00	00				
0x18	0x19	0x1A	0x1B	0x1C	0x1D	0x1E	0x1F

GameDev.tv

Introduction To References

gdev.tv/uc2slides

Why Copy When You Don’t Need To?

- We are currently copying around data we don’t need to copy. This is called passing by value.
- We can look at the originating data directly. This is called passing by reference.
- Students will often get in a muddle with pointers and references.

Key Points

- The symbol we are using is &.
- It is an address in memory.
- & can have other meanings...
- You must initalise a reference!
- You cannot reassign a reference….

Challenge

- Rewrite the `GetValidWords()` function to take in the original word list by reference.
- Are there any other functions that you can pass values in by reference?
- Is the parameter you are passing in `const`?

GameDev.tv

Random And The UE4 Docs

gdev.tv/uc2slides

Many Ways…

- We will be using `RandRange()`.
- But why this one. How would you know?

Challenge

- Use `RandRange()`.
- Set the `HiddenWord` to a random word from `ValidWords`.

GameDev.tv

Out Of The Frying Pan

gdev.tv/uc2slides

Into The Fire...

- We called a function, for just a number.
- Let’s have a look at our code...

Challenge

- Create a member variable `Isograms` that is the `ValidWords` `TArray`.
- Use that instead when initialising `HiddenWord`.

GameDev.tv

Out Parameters

gdev.tv/uc2slides

Good Bad And The Ugly

- We need to count our Bulls and Cows.
- We could use a struct or USTRUCT, or member variables.
- We are going to use out parameters.
- Unreal uses them extensively.
- Be able to identify them and use them effectively.

Challenge

- Create the new `for` loop to iterate around the rest of the letters of the `HiddenWord`.
- If you find one that's the same increment the `CowCount`.

GameDev.tv

Break Out Of A Loop

gdev.tv/uc2slides

Getting Out Of Loops

- We have seen `continue`.
- Let's look at `break`.

Example

- HiddenWord is “sink”, Player guesses “iced”.
- i == s // not bull, goes to next for.
- i == i // increments Cows and should stop here.
- i == n // wasted effort.
- i == k // wasted effort.

GameDev.tv

Structs

gdev.tv/uc2slides

Structs Are Structures

- A struct is a user defined data type.
- Structs are somewhere to store data, possibly different types.
- By default their data is public.
- You will notice a similarity with classes- other than default visibility they are functionally the same.

Struct Syntax

```
struct FStructureName {
type StructMember1;
type StructMember2;
};
```

Challenge

- Re-write your code to use a struct for counting the Bulls and Cows.
- Remember you will need to return something from `GetBullCows()`.

GameDev.tv

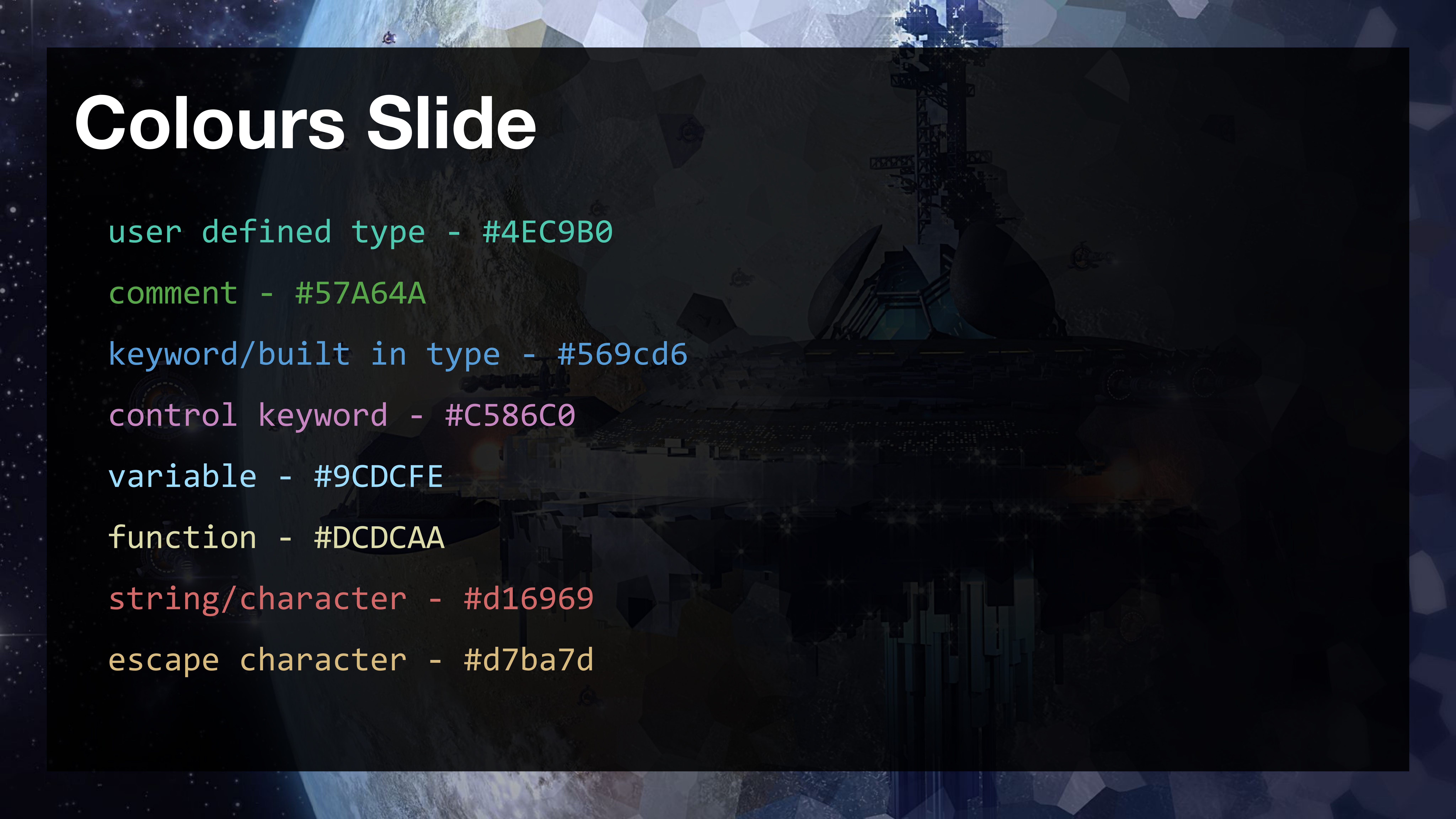
Bull Cow Extras

gdev.tv/uc2slides

GameDev.tv

Bull Cow Game Wrap Up

gdev.tv/uc2slides



Colours Slide

```
user defined type - #4EC9B0

comment - #57A64A

keyword/built in type - #569cd6

control keyword - #C586C0

variable - #9CDCFE

function - #DCDCAA

string/character - #d16969

escape character - #d7ba7d
```