

Game Development with Construct 3.



Game development objectives.

- Use of libraries
- Drag and drop
- Geometric properties
- Importing assets
- Object and environment properties
- Interactions, events and actions
- Game play controls
- Triggers
- Collisions
- Using algorithms
- Saving and exporting a digital game





**DIGITAL
SCHOOLHOUSE**
together with



Importing assets.



ukie



PlayStation.

SEGA[®]



Warwickshire
County Council



UBISOFT



Importing assets adding elements such as images, sound and animation into the game environment.





Sprite a (typically) small image that can be used as a game object or character.





Idle animation an animation triggered when a sprite is not moving.



Importing Assets.

Design – students may demonstrate this through the use of an assets table.

Implementation – students may implement this by importing images, sound and/or animations into Construct 3.

Progression – at it's most basic, students can import their assets, for a more advanced demonstration of importing assets students might create multiple animations for different aspects of their sprite objects.



**DIGITAL
SCHOOLHOUSE**
together with



Geometric properties.



ukie



PlayStation.

SEGA[®]



Warwickshire
County Council



UBISOFT



Geometric properties the settings for size and scale of the objects within the game environment or world.

Geometric properties.

Design – students may demonstrate this through including detailed measurements in their layout designs.

Implementation - For example, the size of the player character must be appropriate to the game environment so that they can move through and around the game obstacles and objects.

Progression – at its most basic students will place and resize their images, intermediate progression can be demonstrated through the use of X and Y positions to position their assets, for a more advanced demonstration students may use an objects X and Y position in an Event to move the object as the result of an event.



**DIGITAL
SCHOOLHOUSE**
together with



Use of libraries.



cukie



PlayStation.

SEGA[®]



Warwickshire
County Council



UBISOFT



Libraries a set of pre-built assets such as environments, objects and characters included with the game design software.





Spritefont a grid of every character that can be drawn. By drawing parts of this image in sequence, strings of text can be rendered. This allows complete artistic control over the appearance of text, rather than having to choose from existing fonts





Tilemap a two-dimensional grid made up of rectangular tiles of equal size, each of which can display an image



Libraries.

Design – explaining what pre-built assets students might use in their game.

Implementation – the use of spritefonts and tilemaps in the creating of a game.

Progression - Construct 3 doesn't have asset libraries – the only built in assets are the tilemap and spritefont, therefore students will be required to source their own assets for their work, allowing them demonstrate more advanced progression.



**DIGITAL
SCHOOLHOUSE**
together with



Game play controls.



Cukie



PlayStation

SEGA



**Warwickshire
County Council**



UBISOFT



Game play controls the way in which the player interacts with the game.



PlayStation.



Game play controls.

Design – students will include detailed information about how their game will be played and the game play controls they will be including.

Implementation - examples include keyboard (WASD and arrow keys), touch (drag and drop) and mouse (movement and mouse click).

Progression – at it's most basic students will use Construct's built in behaviors for movement, for a more advanced demonstration of progression students may design their own game play controls.



**DIGITAL
SCHOOLHOUSE**
together with



Drag and drop.



PlayStation.





Drag and drop the ability to add assets to the game environment by dragging them in.

Drag and drop.

Design – this can't be demonstrated in the design phase.

Implementation – students can drag and drop assets straight into Construct 3 and they will automatically be created as objects in the game.

Progression – at the most basic level students will drag and drop single files, a more advanced demonstration of this can be shown through dragging in multiple image files which will automatically create an animated sprite object.



**DIGITAL
SCHOOLHOUSE**
together with



Object and environment properties.



cukie



PlayStation

SEGA



**Warwickshire
County Council**



UBISOFT



Object and environment properties parameters (defined in the game creation software) that control how each object behaves and appears to the player.



Instance a copy of an object which shares the same properties as the original object but behaves as a new object in its own right.

Object and environment properties.

Design – students can include details of the object and environment properties by making details notes alongside their level designs.

Implementation - examples include size, colour, visibility and whether solid, open, closed. The initial state is defined in the game but events and triggers can modify one or more of these properties.

Progression – at the most basic students can use different animations for each direction, for a more advanced game students should be using different parameters across multiple levels.



**DIGITAL
SCHOOLHOUSE**
together with



Interactions, events and actions.



ukie



PlayStation.

SEGA[®]



Warwickshire
County Council



UBISOFT



Interactions, events and actions interactions are the results of how the player relates to the game objects during gameplay. An event is something (or a sequence) of things that happen in the game. Actions are events that happen as a result of an interaction.



Interactions, events and actions.

Design – students can include event, action sheets and can also design storyboards which explain interactions in their game.

Implementation - for example, an interaction might be a player approaching a door in a game, which is initially closed. The trigger may be proximity to the door the action is to open the door. Examples of events would be game start, game end and a key press.

Progression – at the more basic level students will use simple linear events with a single action, more advanced games could include events with multiple actions and events which use pipelining (the action of one event is used as the event trigger for another).



**DIGITAL
SCHOOLHOUSE**
together with



Collisions.



ukie



PlayStation.

SEGA®



**Warwickshire
County Council**



UBISOFT



Collisions this another form of event which occurs when two sprites or game elements collide during gameplay.



Collisions.

Design – students can include flowcharts, storyboards and event, action tables detailing what will happen as the result of different collisions in the game.

Implementation - an example would be the player character hitting a brick wall or other game hazard.

Progression – at the most basic students will create collisions using the pre-set hit box and collision event, more advanced games could include more precise hit boxes or use different actions based on various collision conditions.



**DIGITAL
SCHOOLHOUSE**
together with



Using algorithms.



Cukie



PlayStation.

SEGA



Warwickshire
County Council



UBISOFT



Algorithms a series of steps in a sequence with a defined end point.



Algorithms.

Design – students could create flowcharts or storyboards to explain the algorithms they will use in their games.

Implementation - examples of where these could be used include scoring systems or speed of movement in the game.

Progression – progression can be demonstrated through the complexity of the algorithms used e.g. a simple scoring system at the more basic end and a linked list at the more advanced end of the spectrum.



**DIGITAL
SCHOOLHOUSE**
together with



Triggers.



ukie



PlayStation.

SEGA®



**Warwickshire
County Council**



UBISOFT



Triggers an event that takes place in order to make another process happen.

Triggers.

Design – students can demonstrate this through flowcharts, storyboards and event, action tables.

Implementation – students can implement this as events in their game that trigger different actions.

Progression – progression for triggers can be demonstrated in the complexity of the trigger, for example, at it's most basic a simple collision as a trigger and at the more advanced end of the spectrum a trigger which is pipelined from a previous action or an event that has multiple conditions.



**DIGITAL
SCHOOLHOUSE**
together with



Saving and exporting a digital game.



cukie



PlayStation.

SEGA[®]



Warwickshire
County Council



UBISOFT



Saving and exporting a digital game saving refers to saving the game in its native format to allow for further modification, exporting refers to exporting the game in a format which would allow it to be played on different computer systems or platforms.



Saving and exporting a digital game.

Design – students can include details of the file names they will use for their files and explain why they will implement versioning when creating their game.

Implementation – students can save in a format that is editable in Construct 3 or can export into a variety of formats.

Progression – at the more basic end students can use the HTML5 export which creates a zip folder, to demonstrate more advanced progression students might look at the different settings needed to export directly to a gaming platform such as the app store or export and then convert their file to an exe.