

Guide to creating games in Builda

Builda is a groundbreaking platform that enables anyone to become a game developer without programming experience.

Builda is a groundbreaking platform that allows anyone to become a game developer without any programming experience. Builda's power lies in its intuitive interface and simple block system. This article will dive into each section of the game creation tool (Create Menu), guiding you step by step to turn your idea into a complete game.

Builda

I. Instructions for creating Builda games

Step 1: Open the Builda App and access the Creation Menu

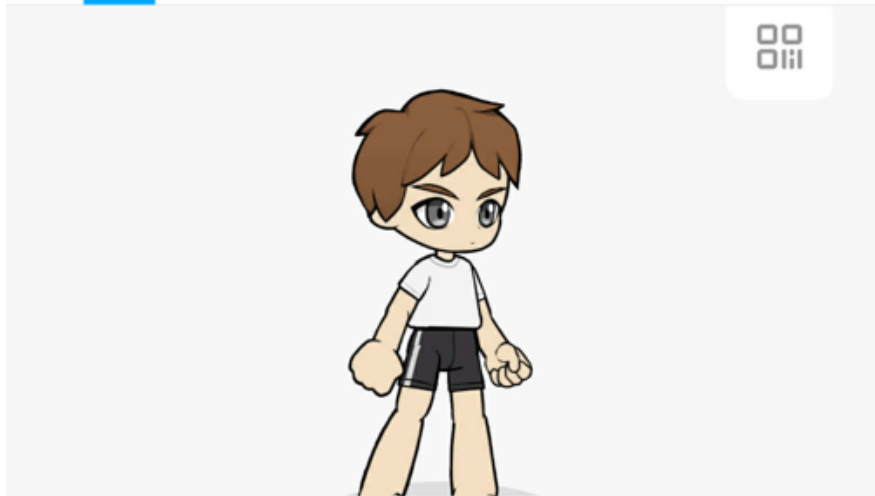
Trang chủ



Đề xuất

Xếp hạng

Đang follow



Gần đây >



câu cá vụn cân
749.1N chơi

Top Games >



13:52



Game



Asset



Bài đăng



1. Open the Builda app on your device (iOS/Android).
2. Find and tap the "Create" icon or the large plus (+) icon in the bottom navigation bar. This is your gateway to all of Builda's creative tools.

Step 2: Start a new project

1. In the Game Creation Menu, select Start New Project.
2. Choose a Template (Available Template) that suits the game genre you want. Including solo games, RPG games, tower defense games, online pvp games, hunter and play genres.

13:55



Game mới

Tạo

Tên

quantrimang.com

Mẫu



Chọn



Solo platform

Độ khó★



Solo RPG

Độ khó★★



Tower defense

Độ khó★★



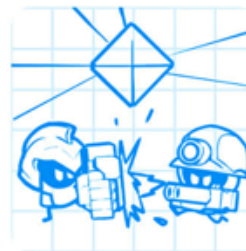
Online PvP

Độ khó★★★



Hunting game

Độ khó★★★★★



Online brawl...

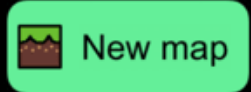
Độ khó★★★★★

Ok

Explore the game creation menu



The screenshot shows a Scratch script with three 'say' blocks. The first block is labeled 'Level select' and has a 'Start map' button below it. The second block is labeled 'Level 1' and the third is labeled 'Level 2'. A watermark for 'centrimang' is visible over the screenshot.



Once you have created your project, you will be taken to the main Studio interface. This interface is divided into sections, each with an essential function.

1. Project

This section manages your game as a whole, from structure to general settings.



Game Settings

13:55

10



8

7

6

5

4

3

2

1

uantrimang



Layer1



Tile Deco Actor Prop Item Weapon



A. Top Bar

1. (Back / Back):

1. **Effect:** Returns to the previous screen, usually the project management interface (Project Menu) or the scene list (Scene List).

2. (M?i t?ên cong)(Undo/Redo / Undo/Redo):

1. **Effect:** Allows the user to undo the last action of placing or deleting an object, or redo the undone action.

3. Map(Map):

1. **Effect:** Default mode for designing and building **the terrain structure** of the game screen by placing *Tiles* , *Actors* , *Props* , etc., on the grid.

4. UI ?(UI Switch / User Interface Switch):

1. **Effect: Switches to User Interface (UI)** elements editing mode . This mode is used to add interactive elements such as buttons, health bars, score bars, or other informational overlays to the game screen.

5. (Kính Lúp)(Search / Search):

1. **What it does:** Open the search bar to quickly find assets in the app's large library or in the project's asset store.

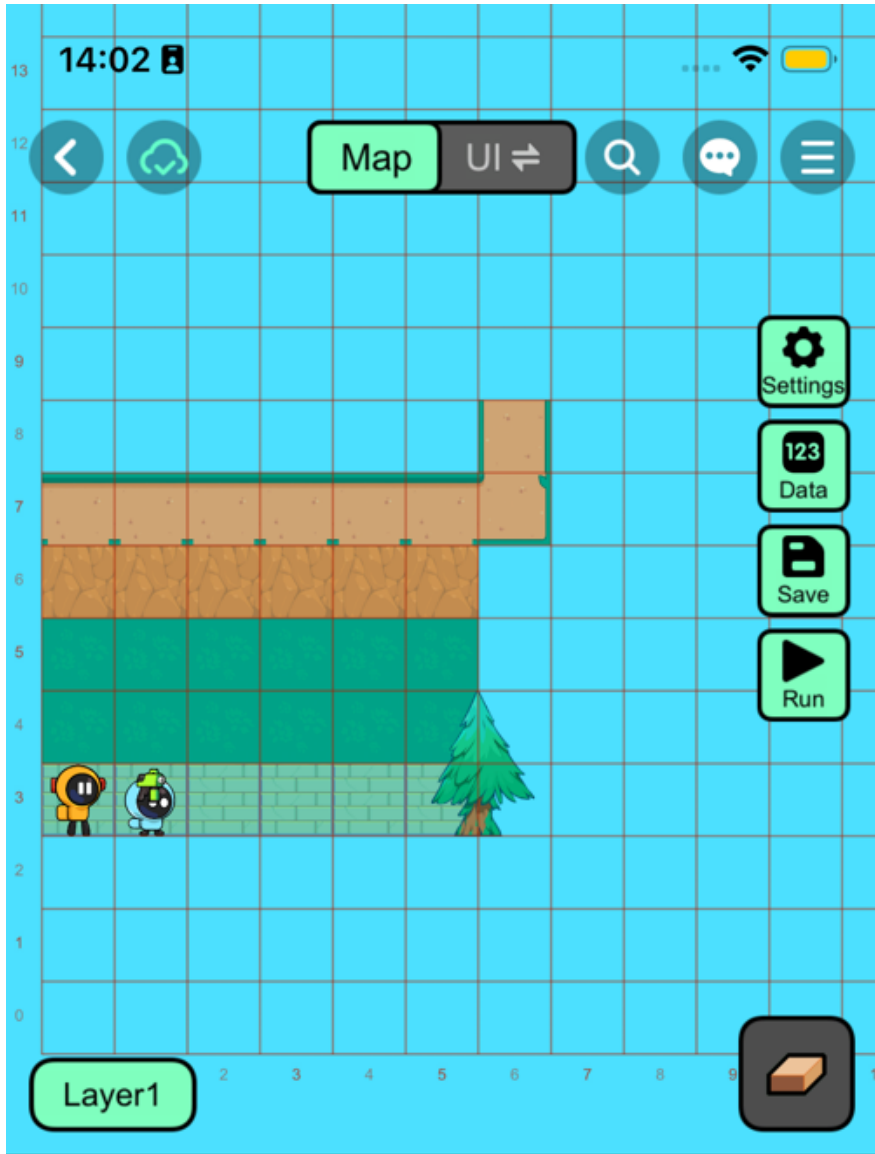
6. (Bi?u t??ng h?i tho?i)(Chat / Conversation):

1. **Purpose:** Could be a button to access community tools or discussion/suggestion features during game creation.

7. (Ba G?ch Ngang)(Menu / Options Menu):

1. **Effect:** Opens the additional options menu for the editor (e.g. mesh display options, scene import/export).

B. Side Panel



1. **Settings(Gears / Settings):**

1. **Effect:** Opens the advanced settings for the Scene. You can adjust physics properties such as **Gravity** , **Scene Size** , **BGM** settings , and Background Color.

2. **Data(123 / Data):**

1. **Function:** Manages and sets up game-related data, including **Variables** , character **Spawn Point** , and game Initial Values.

3. **Save(Save):**

1. **Effect:** Saves all editing progress and changes you have made on the current level to the project.

4. **Run(Test Run):**

1. **Effect:** Start the game immediately in the test environment (Test Environment) for users to quickly check interactions, command block logic and gameplay feel.

II. Main game design area (Grid Area)

1. **Grid System:**

1. **Purpose:** Provides a precise coordinate system (numbered on the X and Y axes) for aligning objects. The grid helps ensure tiles *fit* together perfectly and simplifies level design.

2. **Layer1(Grade 1):**

1. **Effect:** Indicates the current layer being edited. The **Layer** system allows objects to be placed in order of depth (e.g. Layer 1 is the background, Layer 2 is the character, Layer 3 is the foreground item).

III. Object placement toolbar (Asset Palette / Bottom Bar)

This section contains asset categories and specific asset slots to drag and drop into the game.

A. Classification of structures in game creation interface

1. **Tile(Brick/Background):**

1. **Purpose:** Used to build **the static structure** and **terrain** of the level. These are the base blocks (earth, walls, floors) that create the physical space for the character to move around.

2. **Deco(Decoration / Decoration):**

1. **Purpose:** Objects **that are purely aesthetic** , non-collidable, and have no game logic. Used to beautify the environment (e.g., trees, lampposts, paintings).

3. **Actor(Actor/Character):**

1. **Actions:** Objects **that can move** and are **the subject of** complex game logic. Includes the Main Character, Enemies, NPCs (Non-Playable Characters), and other dynamic objects.

4. **Prop(Props):**

1. **Actions:** Physical objects **can be interacted with** but typically do not have artificial intelligence (AI). For example, boxes can be pushed, doors can be opened, traps have simple mechanisms.

5. **Item(Item):**

1. **Effects:** Objects that the player can **collect** to gain benefits or points (e.g. coins, keys, power-up items).

6. **Weapon(Weapon):**

1. **Effect:** Items that are specifically used for **offense** or **defense** (e.g., swords, guns, bullets, shields).

B. Asset Tiles

1. **Plus Sign (+/ Add New) Button:**

1. **Effect:** **Opens the Asset Creator** tool , allowing users to draw (Pixel Art) or design a new object or character to use in the game.

2. Tile Graphics:

1. **Effect:** Represents a specific asset that has been created or imported. The user selects one of these tiles and clicks on the grid to place it on the screen.

3. Small Blue Gear Icon (Logic/Property Setting):

1. **Effect:** Indicates that this object **can be given specialized properties** (e.g. setting a tile to be "ice material" to make it slippery) or have its own **Command Block Logic** attached to it.

2. General Map Settings (Settings - Gear)

13:56

Map

UI

Map settings ?



Map

Map name

Level select

Size

W

32

H

32

BG color

(190,70,100,100%)



Reset map after exiting



Mirror mode



When map is created



When player enters map



When player exits map



When receiving broadcast



Confirm

Texture

Image name

Team

None

Tile

FPS: 59
4.49,18b47

1. **Map(Map):**

1. **Purpose:** The general title for these settings. Usually accompanied by a drop-down arrow, allowing switching between different types of settings (e.g. *Map Settings* and *Physics Settings*).

2. **Map name(Map Name):**

1. **Purpose:** Names the current level. For example, in the image it is "Level select". This name helps to manage easily in the Project Menu and helps players to recognize the level.

3. **Size(Size):**

1. **Values:** W (Width) and H (Height), for example: 32x32.
2. **Effect:** Sets the overall size of the game screen in tiles. This size determines the player's movement range and the space for building terrain.

4. **BG color(Background Color):**

1. **Value:** Color (eg: (190, 70, 100, 100%)) and a blue checkbox.
2. **Effect:** Sets the color displayed in the background behind objects (Tile and Actor). Adjusting the background color helps set the overall atmosphere of the game.

5. **Reset map after exiting(Reset map after exit):**

1. **Type:** Checkbox.
2. **Effect:** If checked (checked), any changes to the object's state (e.g., Prop destroyed, Item collected) will be **reset** to its original state every time the player exits the stage and returns. If unchecked, the state will be preserved (e.g., a solved puzzle will not need to be solved again).

6. **Mirror mode(Mirror/Symmetry Mode):**

1. **Type:** Checkbox.

2. **Effect:** If selected, any object placement on one side of the map will be reflected on the opposite side (often used for levels that require symmetry).

II. Set up logic in the map (Event Logic Settings)

This section defines the Script Blocks that will be triggered when a specific event occurs in the level. Each item has a **plus (+)** button to add logic.

1. **When map is created(When the map is created):**

1. **Purpose:** Defines logic that will run **ONLY ONCE** when the level is first loaded into memory (before the player sees it). Often used to initialize variables, set the initial state of enemies, or run introductory animations.

2. **When player enters map(When the player enters the map):**

1. **Purpose:** Defines logic that will run **EVERY TIME** the player moves from another map to this one. Usually used to display a welcome message, set the starting point, or check conditions (e.g. check if the player has a key). Transfers characters, items, sounds, notifications. to the new map, the resources will be created similarly to when you create a new map.

3. **When player exits map(When the player exits the map):**

1. **Purpose:** Defines logic that will run **EVERY TIME** the player leaves this level to move to another level. Usually used to temporarily save data, set the destination for the next level, or display the results/score screen. Similar to joining a map, except this is exiting the map.

4. **When receiving broadcast(When receiving broadcast signal):**

1. **Purpose:** Defines the logic that will run when a stage receives a **broadcast signal** from an Actor or another stage in the game. This is the core tool for different parts of the game to communicate with each other (e.g. An item broadcasts the "GAME_WIN" signal and the stage receives the signal to display the victory screen).

In addition to the above options, you also have the option to Zoom Map, follow camera or zoom freely below.

III. Operation Buttons

1. **1. Confirm(Confirm):**

1. **Effect:** Apply and save all changes you have made in this *Map Settings* window .

2. **x(Close):**

1. **Effect:** Closes **the** *Map Settings* window without **saving** the changes you just made (unless there is a separate *Save button in this window*).

3. Player Variable Card (123 icon - Data)

13:57

Map

UI ⇌

Data management



Player variable

Config table

+ Create Player variable



Level stars array(云存储)



123

Unlocked Levels(云存储)



123

Current level



Team

None

Tile

FPS: 56
4.49.18b47

This section manages variables whose values change during the game and **are usually saved** for each player.

A. General Functions

1. Create Player variable(Create Player Variable):

1. **Purpose:** Allows the user to create a new variable. These variables are used to store information such as Score, HP, Coins, Item Quantity, etc.

2. Bi?u t??ng Bánh r?ng xanh(Settings / Settings):

1. **Purpose:** Allows editing of variable properties, such as variable name, initial value, variable type (number, text, boolean), and most importantly whether the variable is **stored in the cloud (Cloud Storage)** .

B. Examples of Default/Created Variables

1. Level stars array(Star Level Array - Cloud Storage):

1. **Variable type:** Array – used to store multiple values at the same time.
2. **Purpose:** Usually used to store the number of stars (or completion points) the player has achieved for **each level** in the game (e.g. Level 1 is 3 stars, Level 2 is 2 stars). The symbol (???) indicates that this data is stored on the Builda server (Cloud Save).

2. Unlocked Levels(Unlocked Levels - Cloud Save):

1. **Variable type:** Number or Array.
2. **Purpose:** Stores information about what level the player has **unlocked** . This is important data to maintain game progress.

3. Current level(Current Level):

1. **Variable type:** Number.

2. **Purpose:** Stores the level number the player is currently **playing** . **This variable helps the game know which level to load when the player returns to the game.**

II. Config table tab

1. **Purpose:** This card is used to manage data **that is fixed and does not change** during the game. For example:
 1. **Enemy Attributes:** Strength, Speed, Initial Health.
 2. **Item Price:** Fixed selling/buying price of items in the store. Also set image, quantity, sound, UI.
 3. **Level Settings:** Fixed parameters of each level (like max damage, time limit).
2. **Benefits:** Separating configuration data from game logic allows for quick editing and balancing without changing command blocks.

III. The Importance of Data Management

The Data Management interface is the brain of every game. By defining variables here, Builda users can:

1. **Save Progress:** Ensures players do not lose data when exiting the game.
2. **Creating Complex Logic:** Variables are the foundation for writing conditional (If/Else) blocks and calculating scores.
3. **Customize Your Experience:** Easily adjust level and item values ??in **the Config table** to fine-tune the difficulty and player progression.

4. Edit UI (next to Map)

14:03



15
14
13
12



Select level

5

UI layer1

3

UI

uantrimang

RETURN YOU LOSE CONTINUE YOU WIN 3

FPS: 57
4.4918b47

Interface structure

1. **Grid Area:** Still the main design area, but instead of placing game objects (Tile, Actor), you place UI components here.
2. **UI layer1(UI Class 1):**
 1. **Effect:** Similar to the map Layer system, but specifically for UI. Helps you manage the depth of windows, buttons (eg: Buttons are on top of Popup windows, Popup windows are on top of the game background).

II. UI Asset Palette / Bottom Bar

This section contains drag-and-drop components that make up the user interface.

1. Plus Button (+/- Add New UI Element):

1. **Purpose:** Allows users to create new basic or custom UI elements, such as:

1. **Text:** Add text (e.g. "Select level", "Score").
2. **Image:** Add an image or icon (e.g., currency symbol, background image).
3. **Button:** Add interactive button (can assign command block logic when pressed).
4. **Slider/Bar:** Add slider, HP Bar, or EXP Bar.

2. UI Asset Tiles:

1. **Purpose:** Represents UI templates that have been created or are available in the library. The user selects a template and places it on the screen.
2. **Small Blue Gear Icon:** Indicates that this UI element can have **its properties edited** (color, size, font) or has its own **Command Block Logic** attached to it (e.g. assigning a "**Scene Transition**" command to a button).

III. Typical UI Patterns in the Library

The UI patterns in the image are usually basic notification windows or buttons:

1. **RETURN(Come back):**

1. **Action:** Usually a logical button, assigned to return to the previous screen, main menu, or close the current window.

2. **YOU LOSE(You Lose):**

1. **Effect:** A window or image displaying a failure message, usually appearing when the main character runs out of health or runs out of time.

3. **CONTINUE(Continue):**

1. **Purpose:** A button, often used in the Lose/Pause game screen, assigns a command that allows the player to continue playing (e.g., after watching an advertisement).

4. **Yellow Box (Placeholder):**

1. **What it does:** A blank box that can be used as a **background** for pop-up windows, or as a graphic element that can be recolored and resized to create different shapes.

5. **YOU WIN(You Thang):**

1. **Effect:** A window or image displaying a victory message, usually appearing when the player completes a level objective.

6. **3(Big Count/Clock):**

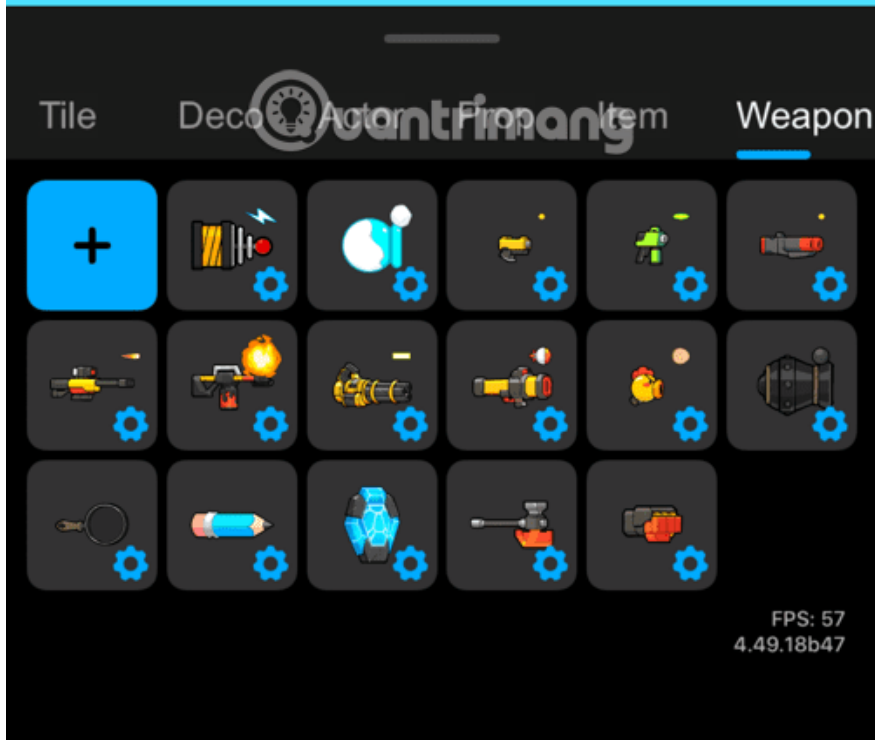
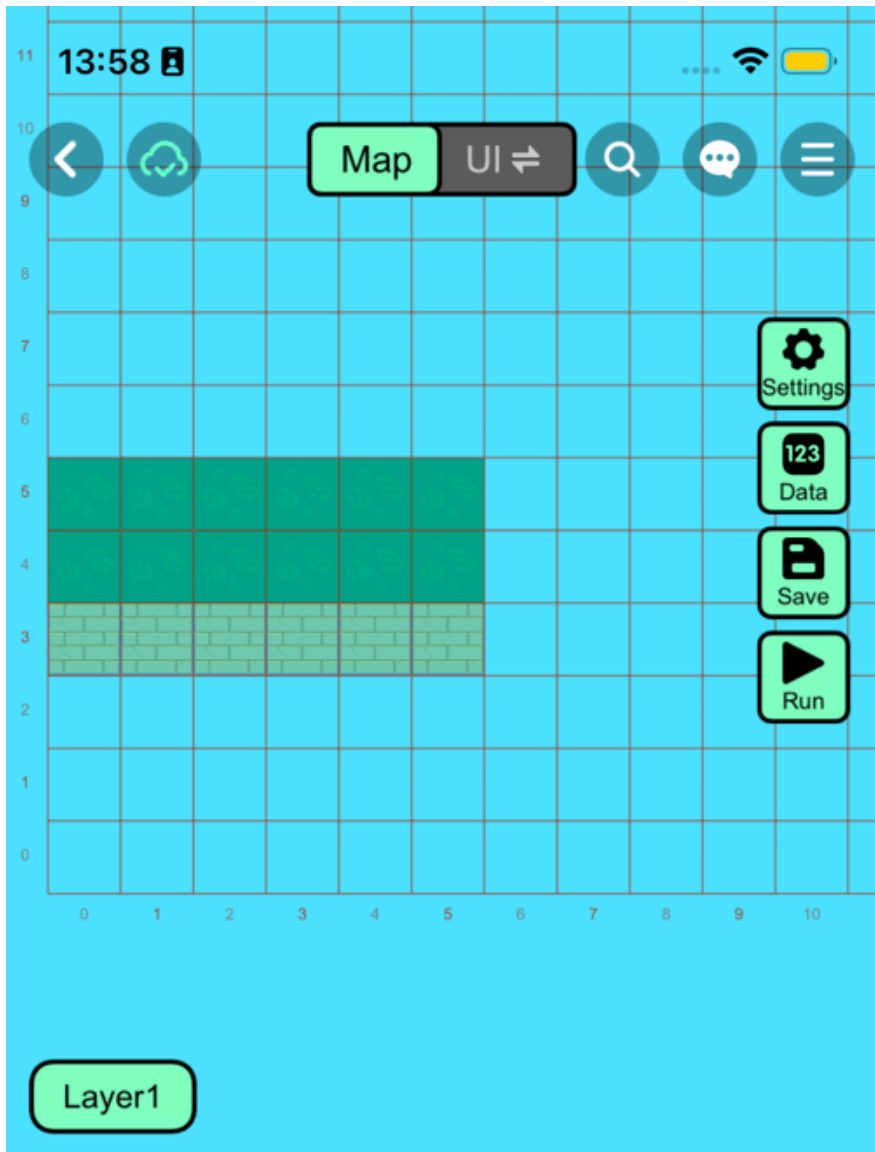
1. **Purpose:** A **large number** display, often used as a countdown at the start of a level or to display an important value (e.g. Number of lives remaining).

7. **(? Khóa)(Lock):**

1. **Function:** Icons that indicate the **locked** state of a level or feature, are used to overlay buttons. Game logic will control the icon to disappear when the level is unlocked.

8. **(Sao và Ô Vương)(Level Rating/Select):**

1. **Effect:** The templates display level completion status (such as number of stars achieved) or boxes for the player to select a level.



5. Evaluate the ability to create games on Builda

Builda is not a traditional game development tool, but a Simplified Game Engine.

Advantage:

1. **No Code Required:** Use drag-and-drop Visual Scripting instead of writing complex code.
2. **Community Integration:** Easily share, playtest, and get feedback from the community right in the app.
3. **Custom Graphics:** Supports simple pixel art and 3D drawing tools, allowing users to design their own game assets.

Limit:

1. Best suited for simple game genres like 2D Platformers, Puzzle Games, Simulations, Story-Based Adventures, and Basic Card/RPGs.
2. Difficulty creating games that require complex physics, AAA graphics, or intensive multiplayer mechanics.

Builda removes the technical barriers of programming, making game creation an accessible creative activity.

Although Builda can seem a bit complicated for those who have never been exposed to game creation, by mastering simple yet powerful tools like Art Assets and Scripting Blocks, anyone can create a unique game without needing too much IT expertise.

You finished reading the article "**Guide to creating games in Builda**" edited by the [TipsMake](#) team. We hope this article has provided you with many useful tech tips and tricks. You can search for similar articles on tips and guides. Thank you for reading and for following us regularly.