

Castle Story

- Tayla Sommer

—



—

Genre	Sandbox, Real Time Strategy
Kickstarter	27th July 2012
Early Access	23rd September 2013
Release	17th August 2017
Publisher	Sauropod Studio
Studio	Sauropod Studio
Game Modes	Single Player, Multiplayer, Coop
Game Engine	Unity
Plattform	Linux, Mac OS, Windows

—

Game Description

Castle Story is a sandbox real-time strategy voxel based building game developed and published by Sauropod Studio. The game was funded via Kickstarter in 2012, had its full release in 2017 and is since 2019 no longer in development.

There are multiple game modes in the game: Sandbox is for relaxed building of structures, and castles, you do this by using your „Bricktrons“ which are small yellow people which you can control and say what to harvest, build or attack. Conquest is either a PvP, Coop PvE or singleplayer PvE game mode, in which you need to take control of crystals around the map and defeat the rockmonsters called „Corruptrons“. And at last there is Invasion in which you need to defend your home crystal against hordes of these Corruptrons.

The game is quite simple and gameplay only lasts about 2-4 hours. Since the game is quite unstable it becomes more laggy the more time passes. With all its faults and errors, I still love the game and have spend thousands of hours in it.

Trailer

<https://www.youtube.com/watch?v=wiyd3o0CXVY&pp=ygUUQ2FzdGxIIIFN0b3J5IFRyYWlsZXI%3D>

My Fan Trailer

<https://www.youtube.com/watch?v=aKwsXxgomFI>

—

Objective Sound Analysis

Music

The music provides a good indication for the style of the entire sounddesign. It is loud, sometimes hectic, it has a steady rhythm and a lot of digital instruments.

[castle_story_soundtrack_example.mp3](#)

Soundscape

As the music already suggests, the general soundscape of the game is quite filled with all kinds of sounds. You might think that this could overwhelm one, but they are easily recognizable and if a Bricktron is in danger or certain materials are being harvested are easy to differentiate from one another.

[Soundscape](#)

Ambience

A strong comparison to the hectic music and soundscape does the ambience not need to be extremely loud nor prevelent. It only provides a bit of wind and foliage sounds and every now and then you can hear a bird chirping. It is quite rare that you only hear the ambience as there is always something happening and you mostly spend your time around your Bricktrons.

[Ambience](#)

UI Sounds

The UI sounds are loud and mostly high pitched. This is important, for all the other sounds might drown out any actions taken using the UI. They are distinct enough to be easily recognizable and its easy to hear the intend of the sound. Was it a successful input? Or did something not work out? Is something in the way? Have I placed my object or removed it? etc.

[Main Menu UI](#) [Ingame UI](#)

Bricktrons

The Bricktrons are the characters you control. Every now and then they make a noise, you can either control them directly or use the auto assign function that lets your Bricktrons automatically assign to

certain tasks. Everytime you give them an order or they assign themselves a task, they will make a noise. This allows you to know when your bricktrons have assigned themselves a new task and if your order has gone through.

Since you might be getting attacked during your conquest or invasion game you need to arm your Bricktrons. Once they wear the equipment their walking sound will differ from not wearing any gear. This also allows you to hear and identify an enemy player appraching you with armoured Bricktrons, allowing you to react and defend yourself.

[Walking](#) [Attacking](#)

Corruptrons

The Corruptrons are your rocky adversary. They come to try to corrupt your home crystal and destroy your structures and Bricktrons. There are 4 different Corruptrons and each of them makes different sounds, allowing you to identify what might be coming out of the fog of war.

[Enemies](#)

Other sounds

Since you spend your time so close to the construction of castles and crafting of items, it is important to gain audible feedback to when something is placed. Your attention might have to shift a lot, so knowing that your reinforced door has been build or a set of armour is important.

[Building](#)

[Digging](#)

[Mining](#)

[Chopping](#)

Subjective Sound Analysis

I really like the entire sound design of Castle Story. While some sounds might sound annoying, you do not encounter them all too often during your playtime. The sounds are easily recognizable and have a cartoonish flare to them, which fits perfectly to the visuals and the great soundtrack. If you love games where you just build structures of your design it is easy to fall in love with this one. It is cute but can also be epic at the same time.

Comparison with Sapiens

Genre

Both games follow the same genre and provide a similar gameplay loop. Gather supplies, build structures, protect your people, rinse and repeat. It is quite simple but in my opinion both games do it in a great way and provide different ways for construction.

Music

While both have very well made music, they could not differ more in style. Castle Story sometimes being extremely hectic and fast while Sapiens takes its time, sit back and relax. The music differs but is extremely fitting for each game.

Sound

The sound, while similar in some aspects, give of quite different vibes from one another. Castle Story keeps this cartoonish, cutesie style within their sounddesign, while Sapiens tries to be more realistic in their approach. Castle Story feels much louder and quicker with its sound design, while Sapiens is extremely laid back, sometimes having no sound at all for certain actions, just because it is not needed.

All in all I do enjoy both games and while they can be quite different in their vibe, they tickle the creative senses in me in just the right way. Back to building I go!

From:

<https://wiki.zhdk.ch/gamesoundopedia/> - **game sound dokumentation**

Permanent link:

https://wiki.zhdk.ch/gamesoundopedia/doku.php?id=castle_story&rev=1686233931

Last update: **2023/06/08 16:18**

