Gamedev Glossary: What Is the “Game Loop”?

by PotHix 30 Nov 2012

Length: Short Languages: English

Gamedev Glossary

In this post, I'll explain the heart of every game: the *game loop*! All the code that makes the game interactive and dynamic goes in the game loop, but is separated into different pieces. The game loop itself is a controlled infinite loop that makes your game keep running; it's the place where all your little pieces will be updated and drawn on the screen.

Initialize, Update and Draw

The game loop is the central code of your game, split into different parts. Generally, these are: *initialize*, *update* and *draw*.

The **initialize** phase is used to do any necessary game setup and prepare the environment for the update and draw phases. Here you should create your main entities, prepare the menu, detect default hardware capabilities, and so on.

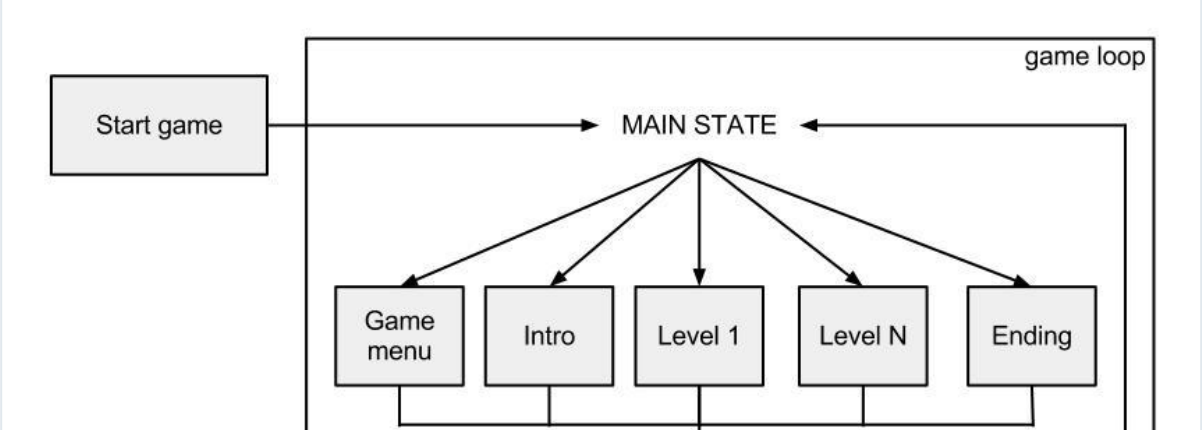
The main purpose of the **update** phase is to prepare all objects to be drawn, so this is

where all the physics code, coordinate updates, health points changes, char upgrades, damage dealt and other similar operations belong. This is also where the input will be captured and processed.

When everything is properly updated and ready, we enter the **draw** phase where all this information is put on the screen. This function should contain all the code to manage and draw the levels, layers, chars, HUD and so on.

Managing States

The loop will keep running over and over again throughout the game, so you must make sure it runs the correct parts of your code for each part of your game. A *state machine* is generally used for this task; this manages a global state for your game loop and redirects the flow to the correct part of the code based on it.



Typical state machine

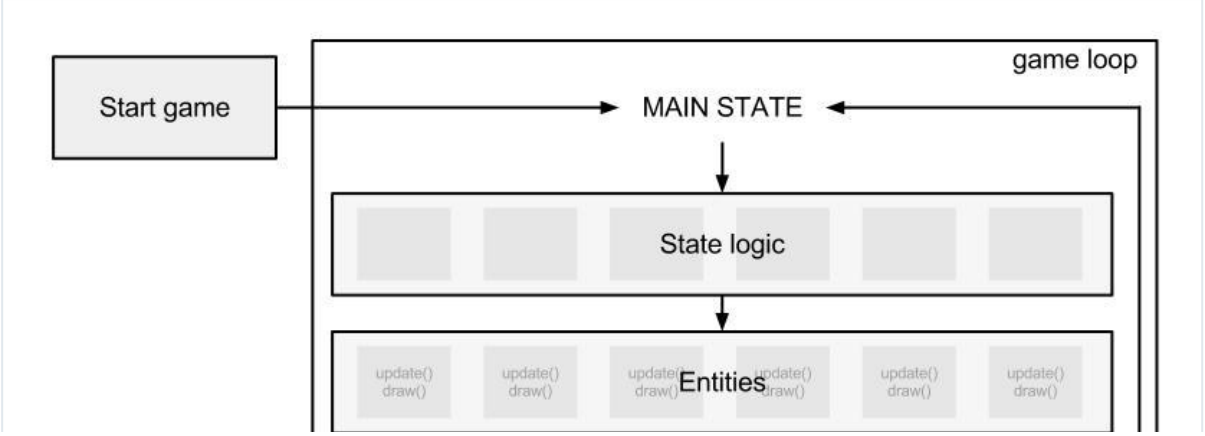
A simple example following the picture above: a new game starts and proceeds to the Main State. This will load the game menu and then keep running and executing the menu logic until it receives an input event (a mouse click, keyboard press, or similar).

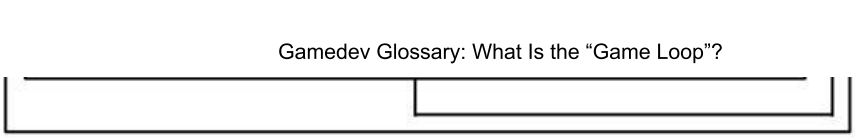
This event then triggers a change in the main game state, moving the execution flow to the intro scene, so the game menu code is not executed any more. The game loop keeps running, but it executes a different part of the code - the part corresponding to the intro scene.

Entities Within the Game

Now that you have all your states under control you'll need to interact with all your entities (chars, objects, NPCs, and so on) and update their properties (status, health, position, ...) to actually move objects and players on your game.

So as well as executing the specific part of the code based on the current state, the game loop also has an inner loop that executes the specific part of the code for each *entity* in the current state.





Interacting with entities

The same behavior happens in the *draw* phase, where each entity gets drawn.

Conclusion

The game loop is the heart of the game, and it is not too difficult to understand its concept if you're coming from any different software development branch. There's a lot more that could be said about it, but I hope this post helps you to start understanding the core of what it's about!

PotHix

PotHix is a hobbyist game developer that works mainly with HTML5 and Javascript for games on his spare time. His latest game was Skeleton Jigsaw (http://plaev.me/skeleton-jigsaw) that is available open source on his organization

on Github: http://github.com/plaev.

Name

JawBfl • 6 years ago

I recommend this tutorial on the game loop and achieving constant FPS
http://obviam.net/index.php...

4 • Reply • Share ›

barrkel • 6 months ago

Initialize isn't part of a game loop. That's something that happens before you enter a game loop. A game loop is a simulation that takes in input, computes a new world state (often based on a time delta for a realtime simulation, like most games are), renders the state and then goes around again. So the parts of the loop are Input, Update and Output (or Render).

The first diagram is misleading also. It seems to give the impression that the game loop relates to navigation around a state diagram. The game loop is simply a programming loop (could be as simple as `while (true) { ... }`), with polling checks for input and network traffic, world update, possibly network traffic output, and a bunch of rendering (video and audio).

• Reply • Share ›

Mike • 4 years ago

Very helpful :)

• Reply • Share ›