Synchronization of Music and Gameplay in 140

Jakob Schmid, audio, 140

me

Jakob Schmid

Audio programmer by day (Playdead)

Composer by night

140

Design and programming: Jeppe Carlsen

Visual design: Niels Fyrst, Andreas Peitersen

Audio: Jakob Schmid (me)

Hobby project, 3 years

Unity 3 Free

IGF

IGF award 2013

Excellence in Audio

- honorable mention, Technical Excellence

Spilprisen 2014

Sound of the Year

Nordic Game Award 2014

Artistic Achievement

release

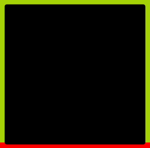
140 is out:

- Steam
- Humble Store

talk

- 140 music and game interaction
- 140 audio production
- Unity 4 music programming tips

140 demo



music and game interaction

music and game interaction

Goals:

- Control game elements from music
- Control music from game progression

music and game interaction

Goals:

- Control game elements from music
- Control music from game progression

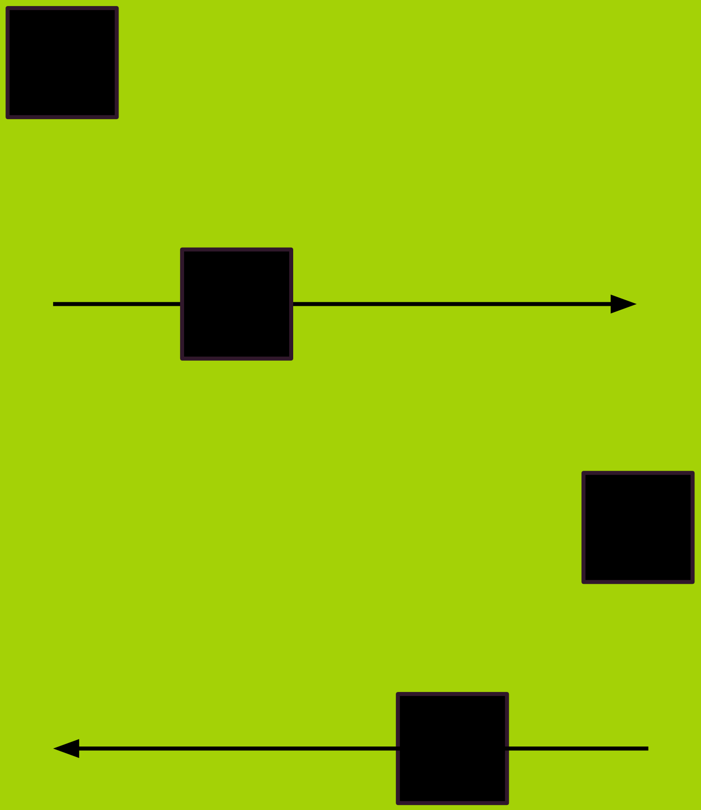
game element control

- Play music loop
- Use audio time from loop to control game elements (instead of game time)

game element control

- Get audio time from playing loop
- When next musical beat reached, raise event
- Game elements listen for events and trigger animation on beats

game element control



1. wait for 16th note #1
2. start moving
3. wait for 16th note #8
4. start moving

tempo

How to get 16th notes from audio time?

tempo

How to get 16th notes from audio time?

```
16th notes: 140 beat/m * 4 note/beat
          = 560 note/m
          = 560/60 note/s
            60/560 s/note
```

tempo

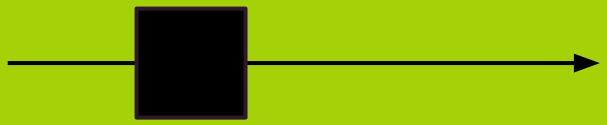
How to get 16th notes from audio time?

```
16th notes: 140 beat/m * 4 note/beat
          = 560 note/m
          = 560/60 note/s
            60/560 s/note
```

~ A new 16th note every 0.107 s

game element control

Using loops and audio time,
we can control game elements from music.



music and game interaction

Goals:

- Control game elements from music
- Control music from game progression

music and game interaction

Goals:

- Control game elements from music
- Control music from game progression

unity 3 audio timing

Using relative timing,
play sound at audio time **`T`**:

```
  AudioSource.Play(delay)
```

unity 3 audio timing

Using relative timing,

play sound at audio time **`T`**:

```
now = AudioSettings.dspTime
delay = T - now
AudioSource.Play(delay)
```

unity 3 audio timing

Using relative timing,

play sound at audio time `T`:

```
   now = AudioSettings.dspTime
   delay = T - now
   AudioSource.Play(delay)
```

- Audio is running in a different thread!

Timing can be inaccurate: e.g. audio update between getting time and playing sound

what we needed

140 is a precise rhythm game.

We needed sample-accurate timing for music!

what we needed

140 is a precise rhythm game.

We needed sample-accurate timing for music!

- That means a precision of 0.00002 s

how we did it

Simple solution with sample-accurate timing:

- Loops of fixed length (or a multiple)
- **Start all loops in same frame**, possibly muted
- Stopping a loop is OK, never restart

how we did it

Simple solution with sample-accurate timing:

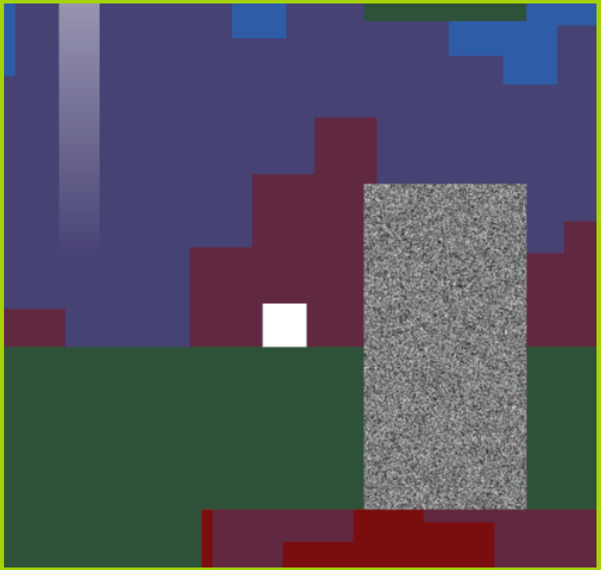
- Loops of fixed length (or a multiple)
- Start all loops in same frame, possibly muted
- Stopping a loop is OK, never restart

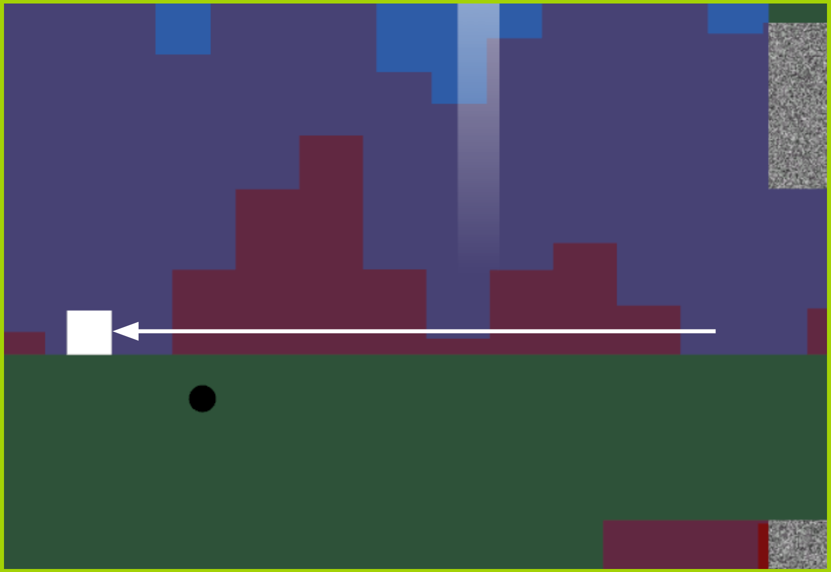
During game progression:

- Control volume/muting and pan
- Never change pitch

how we did it

Simple **attenuation** and **panning** for music loops:





□ ○ audio source

audio receiver

□ ←——→ ○

calculate volume and pan from difference in position

music control

By using loops and modifying volume and pan, we can control music from game progress.

audio production

development limitations

- Looped tracks
- Fixed tempo, 140 BPM
- Fixed key, Cm

inspired by ancient hardware

FM synthesis

samples

toolset limitations

- Ableton Live, no plugins
- FM synthesis: Operator
- Samples: Simpler

ableton demo

unity music programming tips

unity 4 playback timing

Unity 4 has absolute playback timing! Yay

- enables sample accuracy without loops

unity 4 playback timing

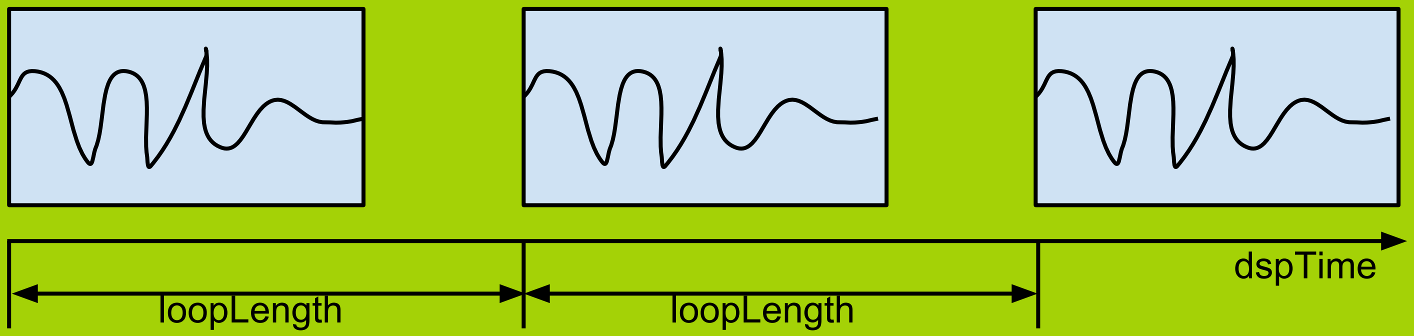
Unity 4 has absolute playback timing!

- enables sample accuracy without loops

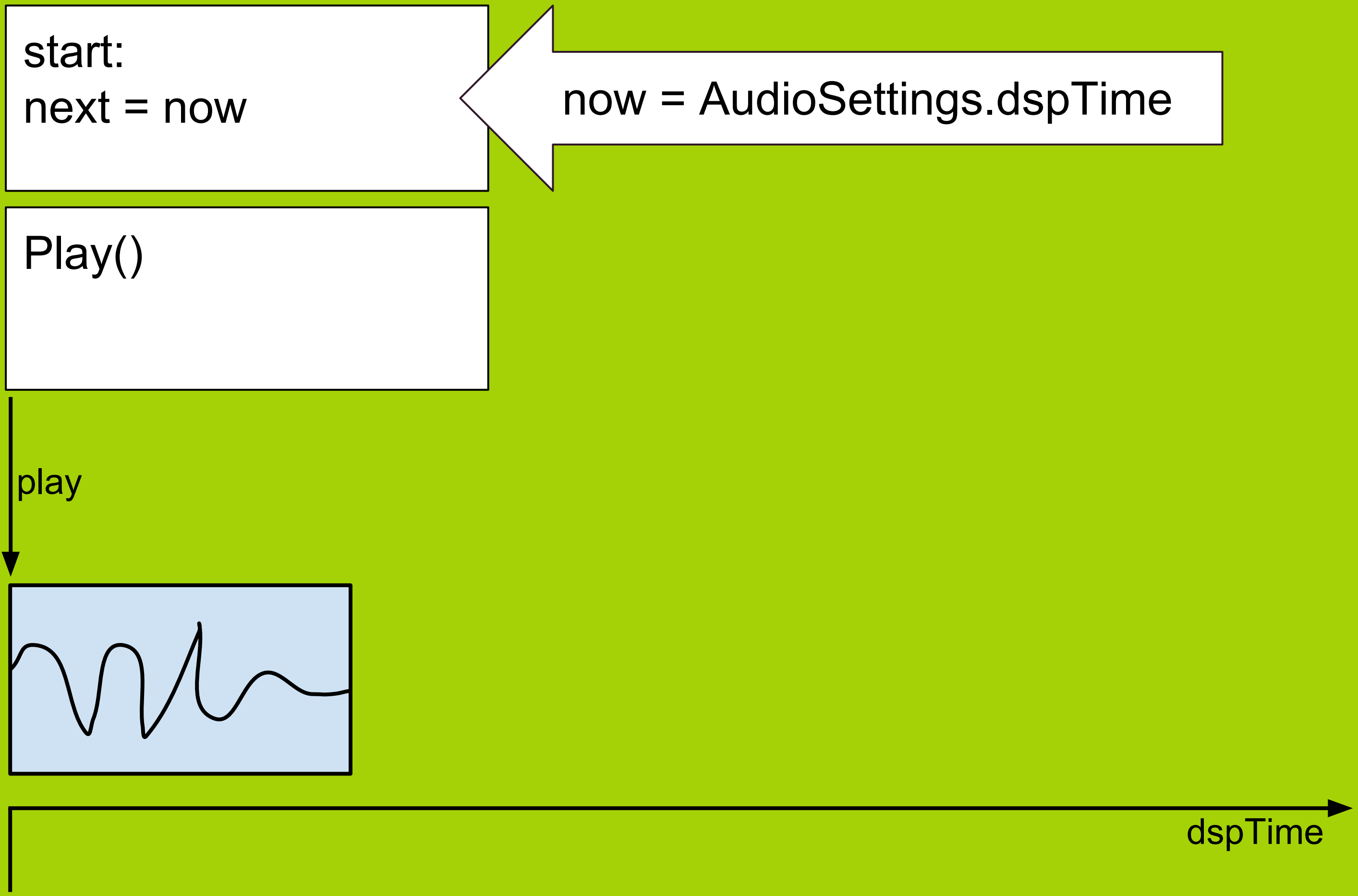
Toolset:

```
    AudioSettings.dspTime
    AudioSource.PlayScheduled(time)
```

audio timing goal



naive solution



start:
next = now

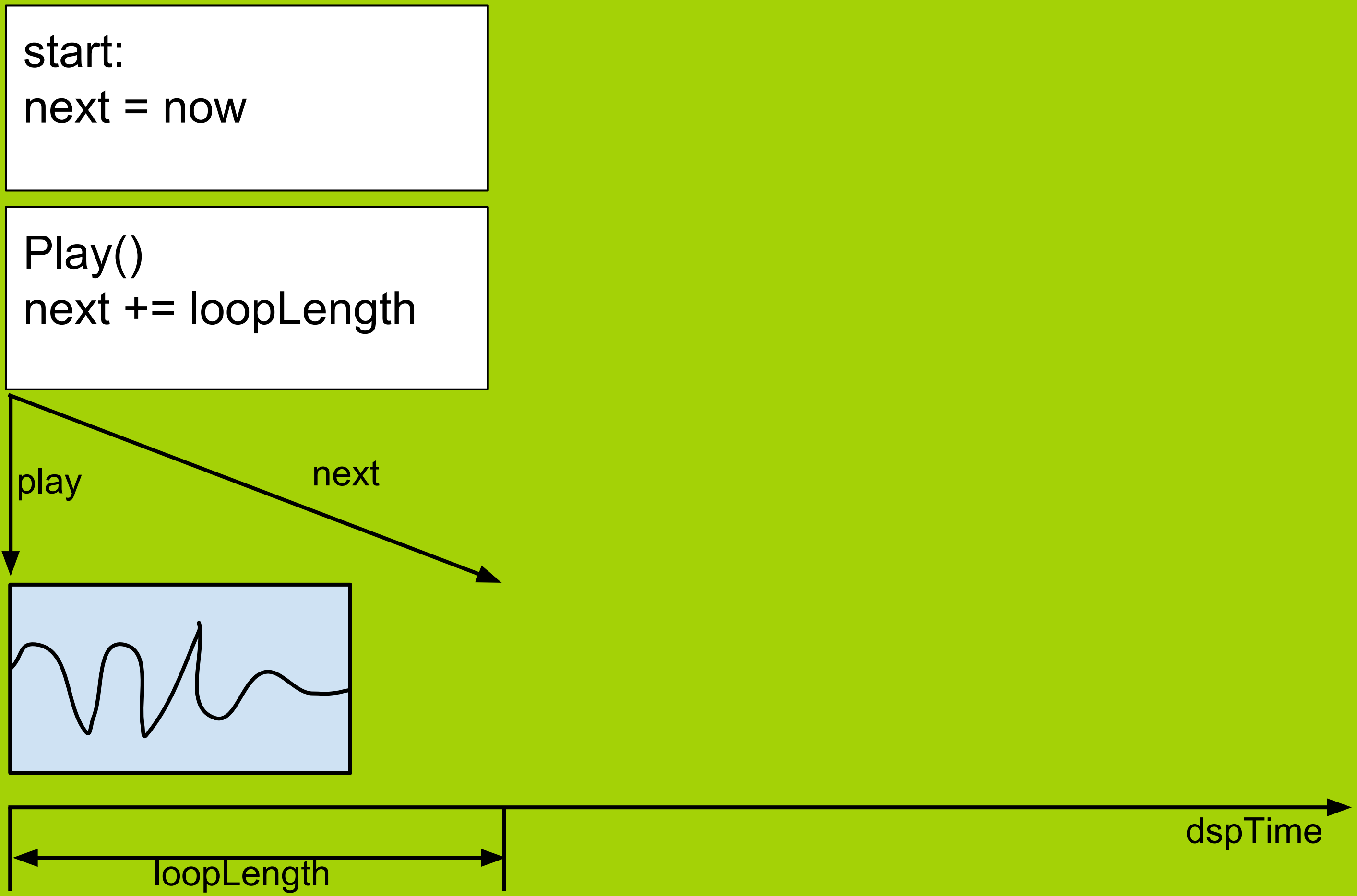
now = AudioSettings.dspTime

Play()

play

dspTime

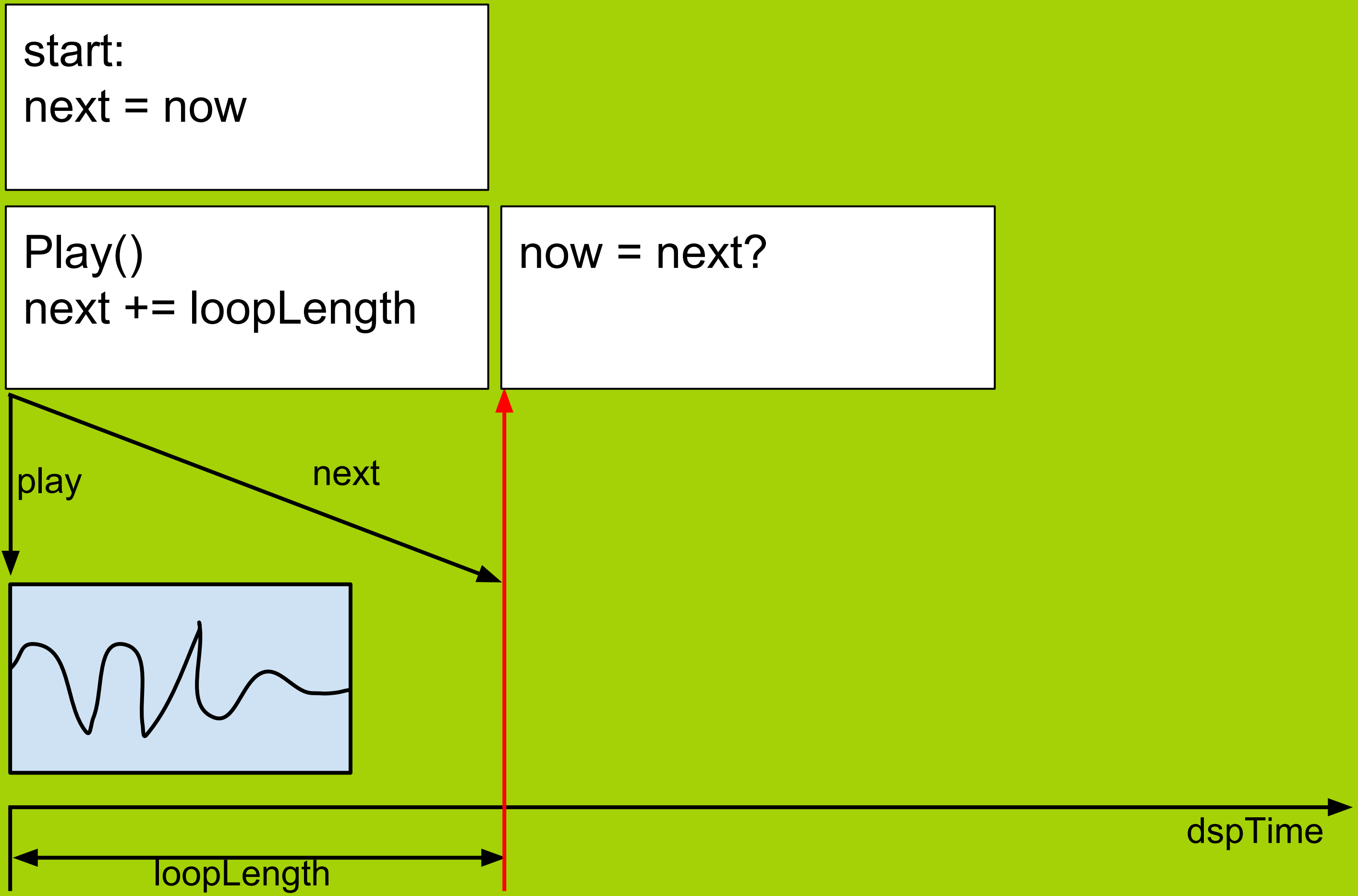
naive solution



start:
next = now

Play()
next += loopLength

naive solution

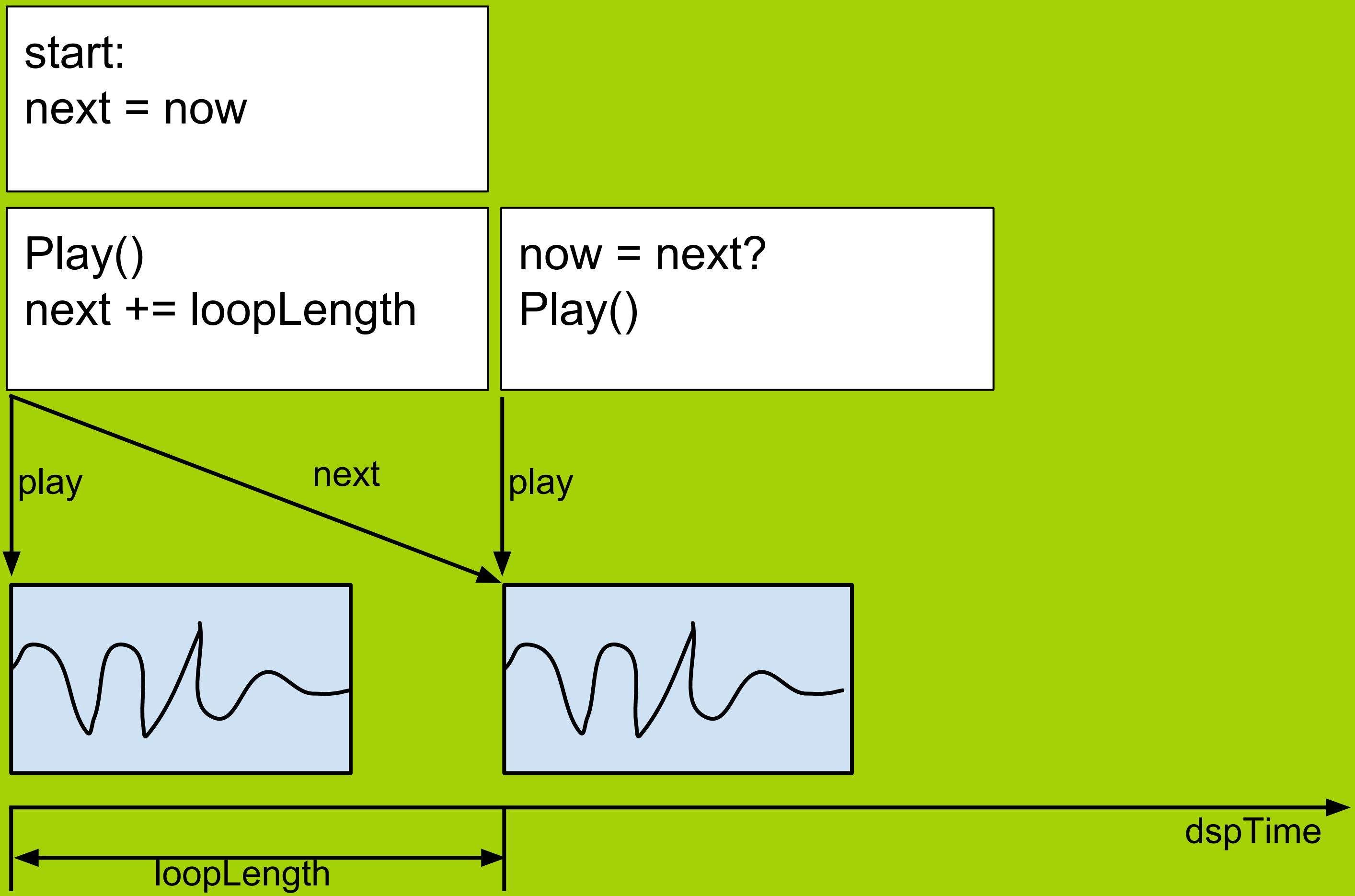


start:
next = now

Play()
next += loopLength

now = next?

naive solution

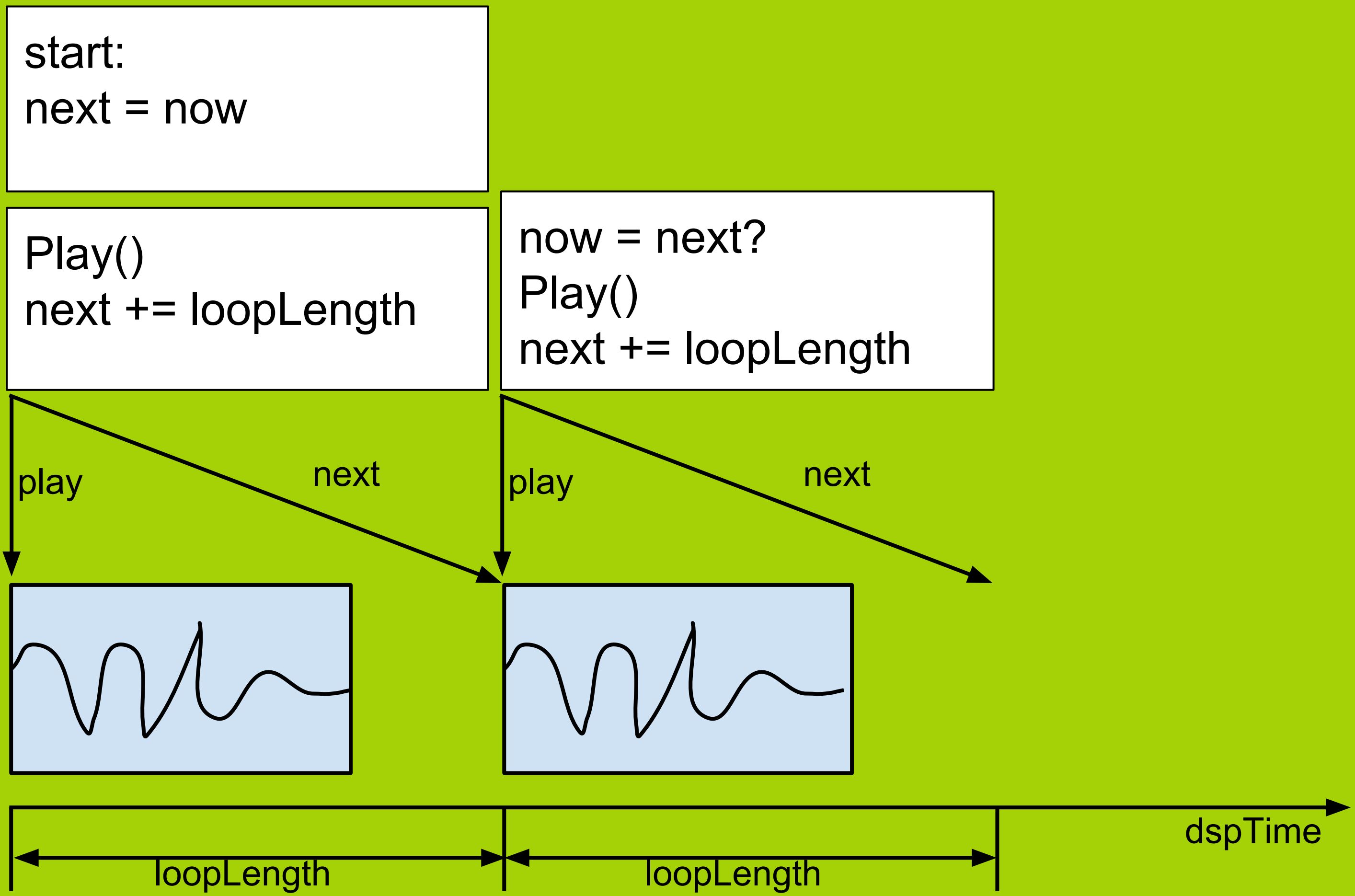


start:
next = now

Play()
next += loopLength

now = next?
Play()

naive solution



start:
next = now

Play()
next += loopLength

now = next?
Play()
next += loopLength

problem

Scheduling a sound for ‘now’ is already too late!

Audio is running in a different thread.

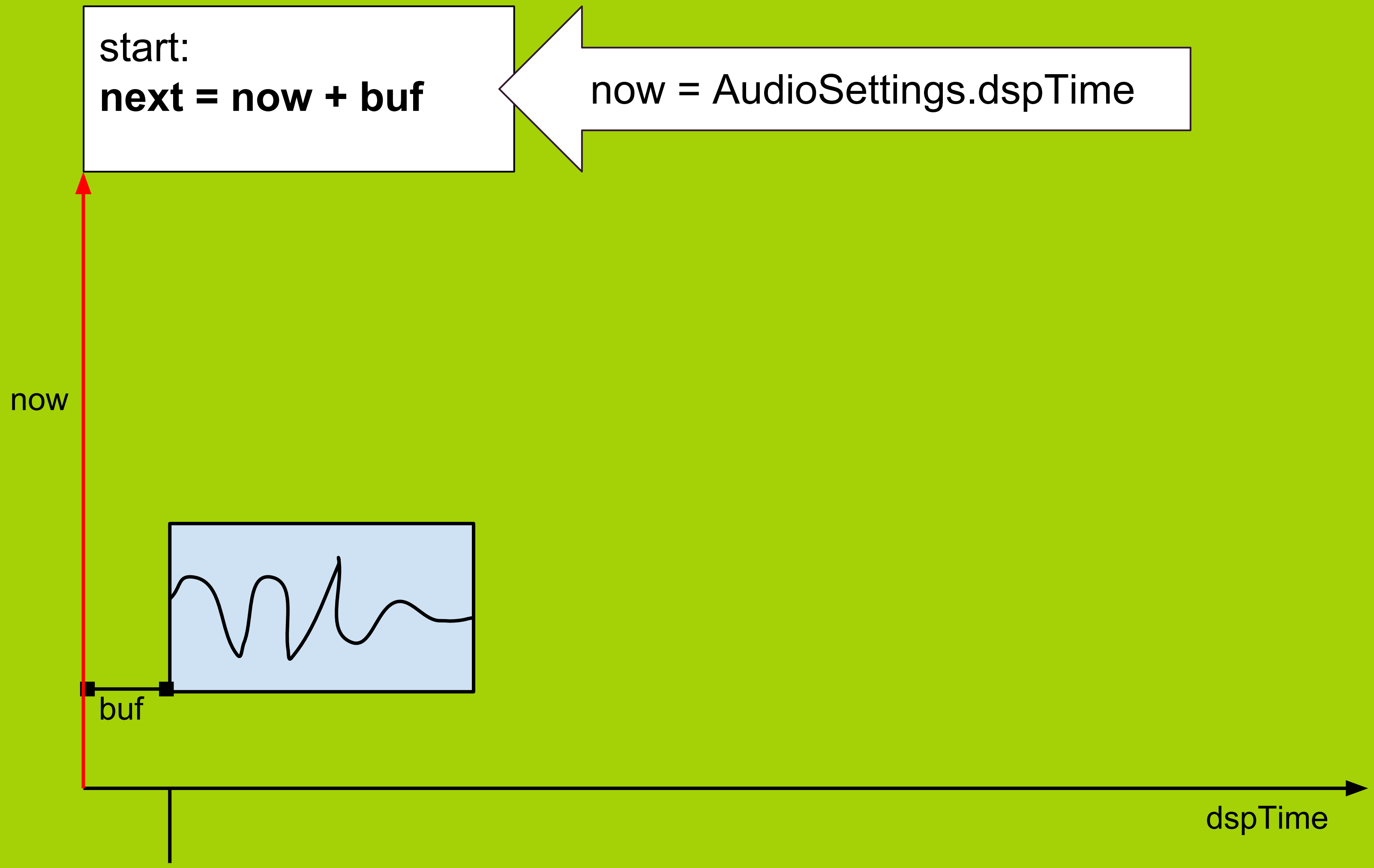
solution

Scheduling a sound for ‘now’ is already too late!

Audio is running in a different thread.

- Add buffer time!

unity 4 audio timing



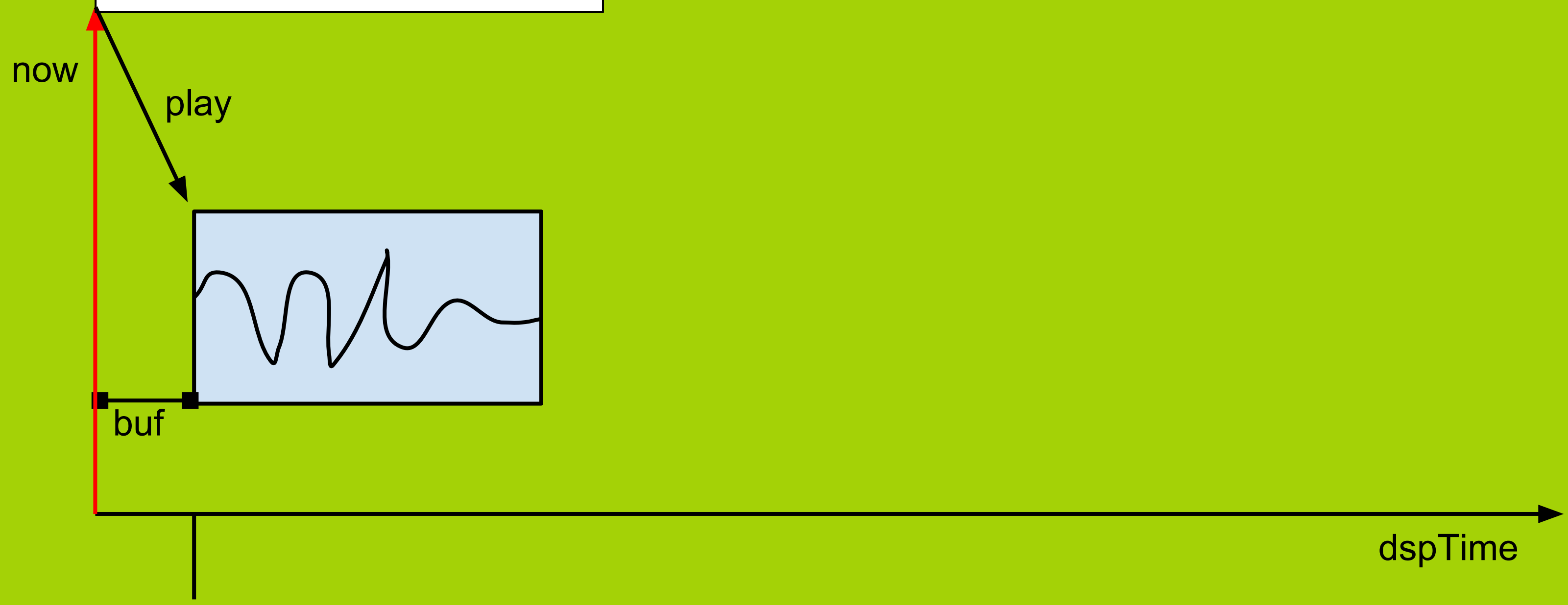
start:
next = now + buf

now = AudioSettings.dspTime

unity 4 audio timing

start:
next = now + buf

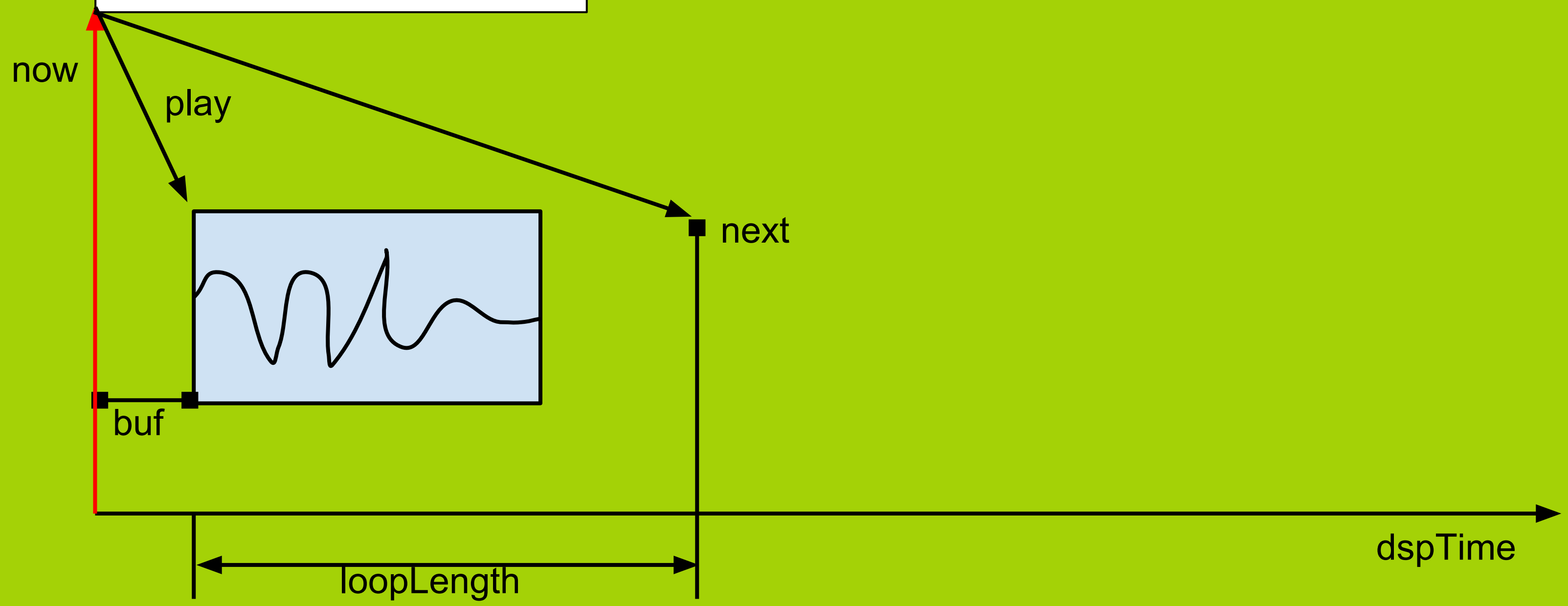
PlayScheduled(next)



unity 4 audio timing

start:
next = now + buf

PlayScheduled(next)
next += loopLength

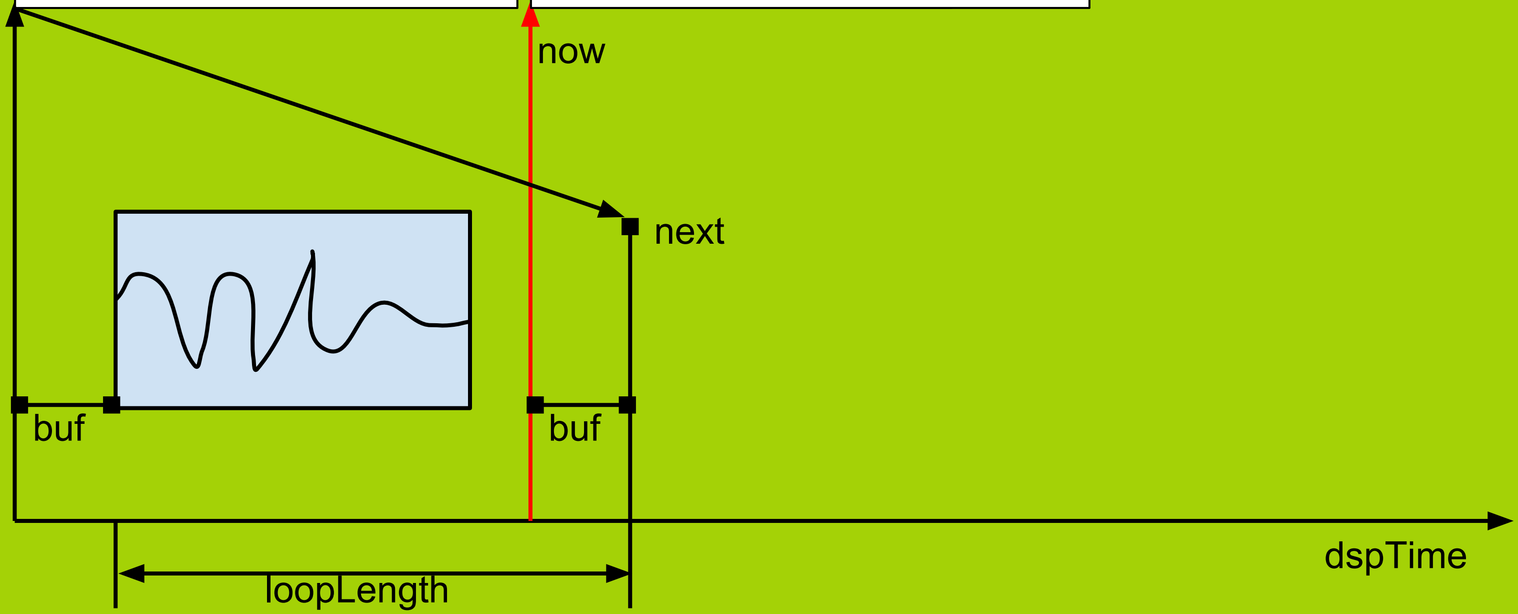


unity 4 audio timing

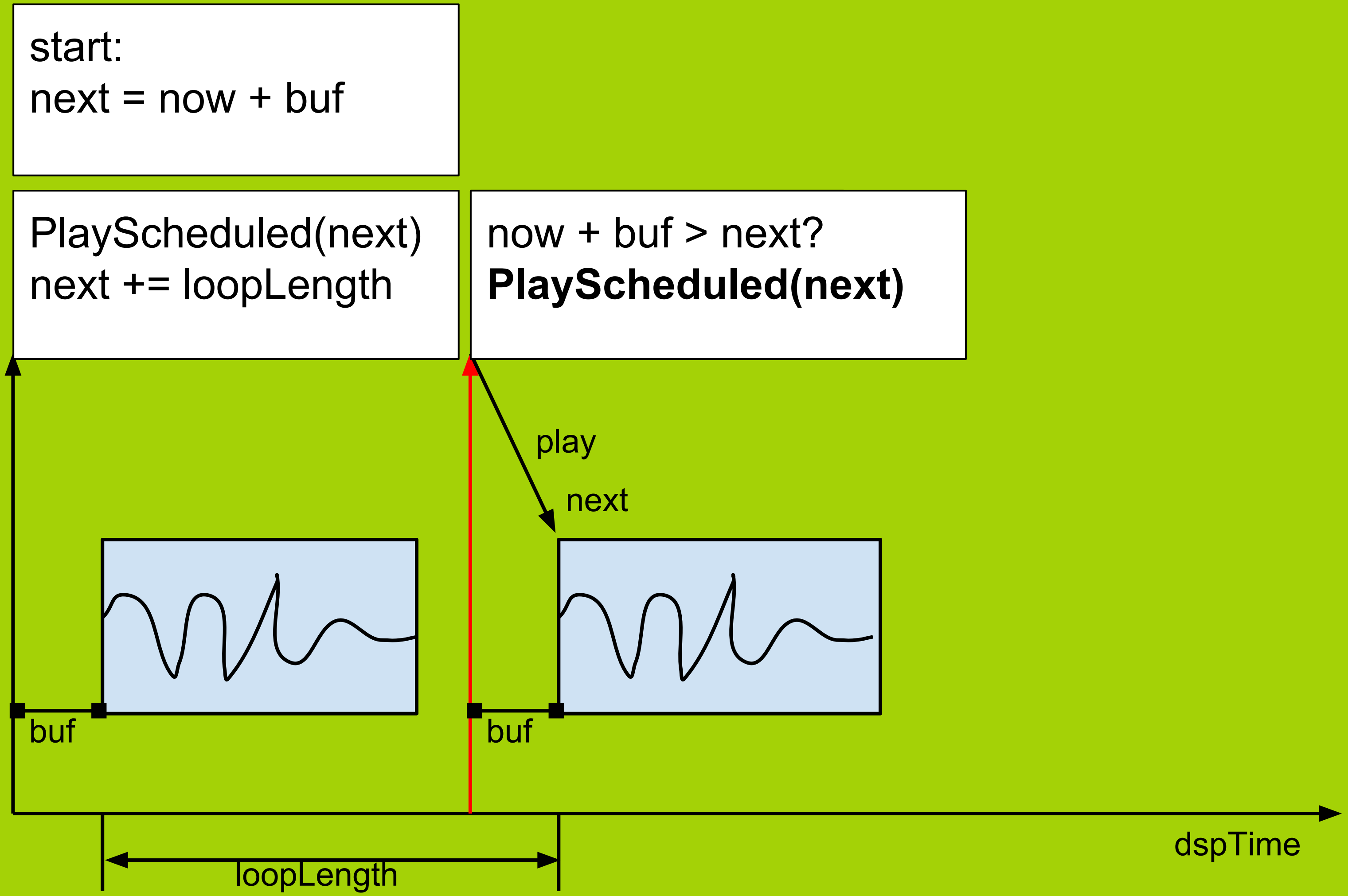
start:
next = now + buf

PlayScheduled(next)
next += loopLength

now + buf > next?



unity 4 audio timing

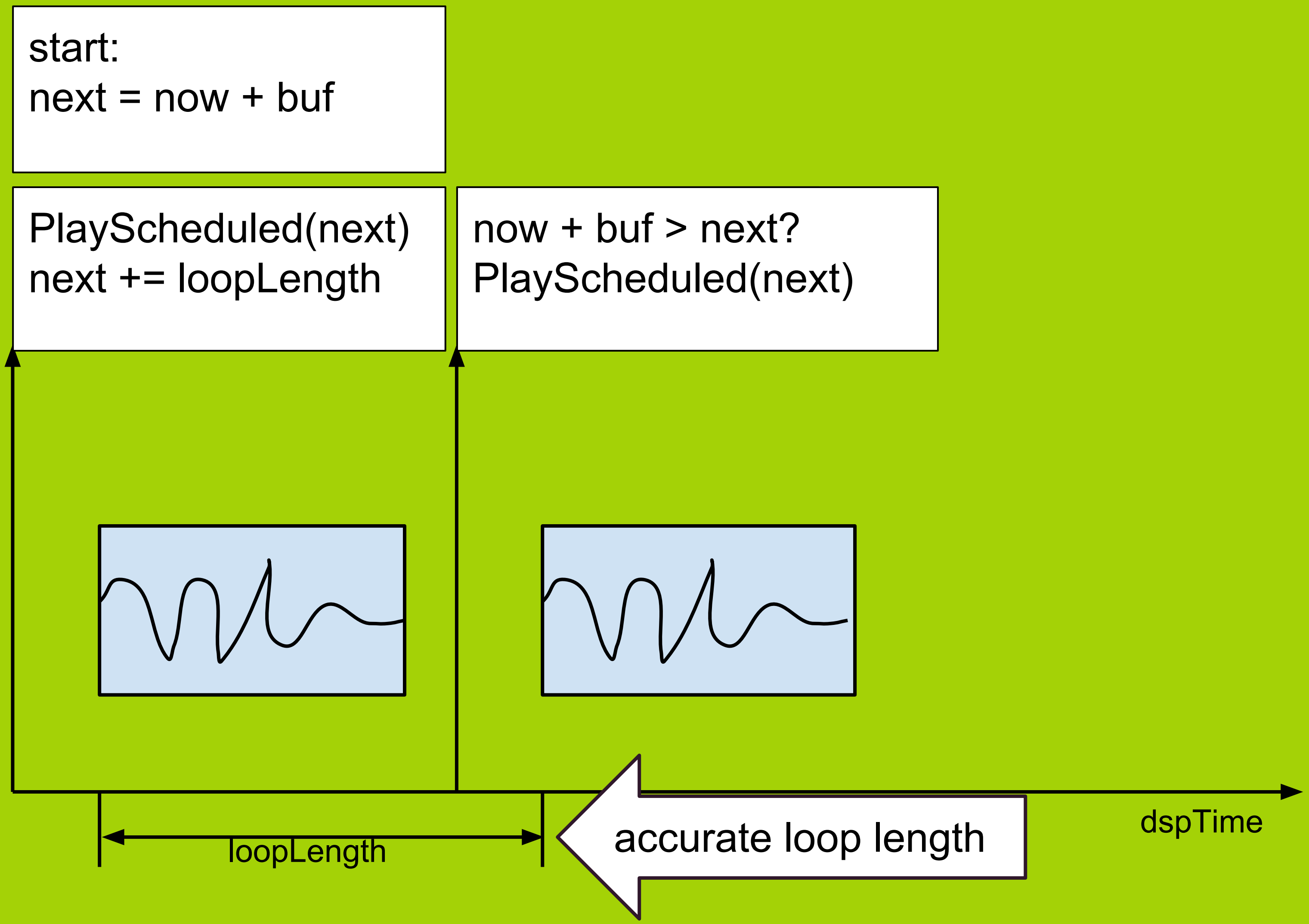


start:
next = now + buf

PlayScheduled(next)
next += loopLength

now + buf > next?
PlayScheduled(next)

goal



start:
next = now + buf

PlayScheduled(next)
next += loopLength

now + buf > next?
PlayScheduled(next)

pseudocode

```
Start:
    buf = 0.1 // as low as possible
    next = AudioSettings.dspTime + buf
Update:
    now = AudioSettings.dspTime
    if(now + buf > next)
        audio.PlayScheduled( next )
        next += loopLength
```

audio timing demo

music control demo

`www.schmid.dk/gallery/play_scheduled/`

- with example C# code

Synchronization of Music and Gameplay in 140

jakob @ schmid.dk