Introduction:

Game development does not use functional. Carmack and Sweeney suggests switching to functional.

Related work:

- Usability of F#
- ECS in other engines
- F# in NU
- Other people investigating F# in Unity?

Research:

- Evaluation strategies
- Functional Reactive Programming
- Usability Evaluation Methods
- Garbage Collection
- Concurrency in .NET
    - C# job system + ECS
    - Async in F# and C# (kun hvis vi inkluderer vores concurrency benchmark)

Den næste del af rapporten er vi lidt mere usikker på. Vi ser to løsninger:

Experiments (v1):

- Lenient and concurrency + task sizes in .NET Core Tasks

Herefter skriver vi at lenient evaluation ser lovende ud og konkluderer at det kunne være spændende at undersøge i Unity. Derefter konkluderer vi at standard .NET concurrency med Tasks og Async Workflows ikke virker i Unity og siger at vi lavede en sekventiel løsning i stedet.

- Benchmarking FRP in Unity
    - Garbage Collection
    - Implicit concurrency?
    - Entity Component System
- Cognitive Dimensions comparison

Experiments (v2):

Vi starter afsnittet med en kort beskrivelse af lenient og nævner hvordan man kan teste det i .NET. Herefter skriver vi at vi lavede en serie af benchmarks som så lovende ud, men at de egentlig er ubrugelige i Unity. Vi flytter altså hele afsnittet omkring concurrency og critical workload ned i appendix. Derefter

- Benchmarking FRP in Unity
    - Garbage Collection
    - Implicit concurrency?
    - Entity Component System
- Cognitive Dimensions Comparision

User Tests:

- Setup
- Champagne Analysis
- Cognitive Dimensions Side-by-Side Analysis

Discussion:

Conclusion: