UNIVERSITÀ DI PISA

Punch’em Videogame

Luca Tartaglia

Kamran Mehravar

Farzaneh Moghani

Introduction and Objectives

- *Problem Description*
  - A player fights against two type of enemies, minions and bosses, one at a time. A fight lasts until the opponent is defeated.
  - When the player is fighting a minion and a boss arrives, the fight against the minion is stopped and the minion is queued again recovering some health.

- *Evaluated Scenario*
  - Exponential distribution of the interarrival times and fighting times

- *Objectives*
  - Study the number of minion and bosses fought in the unit of time as a function of the arrival rates
  - Study the queueing time of bosses and minions.

- *KPIs (Key Performance Indexes)*
  - Throughput T
  - Mean Waiting Time E[W]
  - Mean Response Time E[R]

Modeling

Model Description

- **Opponents (Bosses and Minions):**
  - The opponents, sent respectively from the modules Boss and Minion to the module Player, are **considered as messages** (new type defined, called OpponentMessage) that are **placed in a FIFO queue** (different queues are used for each opponent type).
  - The new type of opponent message **has a parameter attached**, the service time, that **is expressed in seconds and represent the opponents life**.
- **The Player:**
  - An opponent message is extracted from the queue (start of the fight) and when the time corresponding to the latter's life (service time) runs out, the battle is declared finished and the opponent defeated.
- **Recovering System:** When a minion is under service and a boss arrives, the player stops the fight and perform these operations:
  1. Takes off the minion under service;
  2. Computes the percentage of the lost life ;
  3. Takes the "x" percentage of the lost life, called recovering percentage (x is set in the configuration and can assume values $0 \leq x \leq 100$);
  4. Applies the recovering percentage to the current life, in order to compute the life to be added;
  5. Sum up the current life and the life to be recovered;
  6. Queue up the recovered minion again.

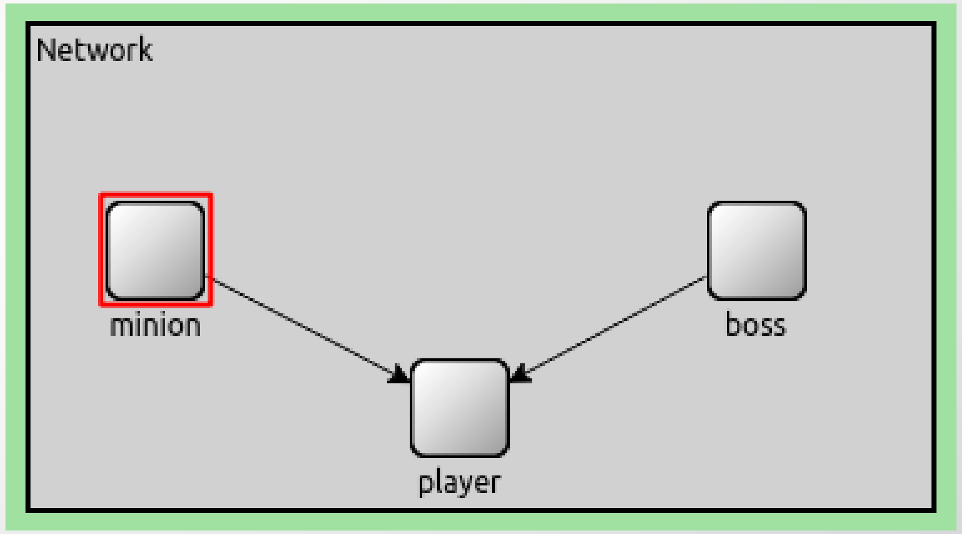
Factors

- $\frac{1}{\lambda}$: Mean of **interarrival time** distributions.
- $\frac{1}{\mu}$ **:** Mean of **service time** distributions.
- **Recover Rate x** : the percentage of the lost life that a minion can recover.

Implementation

- *Modules*
  - **Boss and Minion :**
    1. Generate a random number from interarrival distribution
    2. Generate a random number from service time distribution
    3. Send a message to Player module

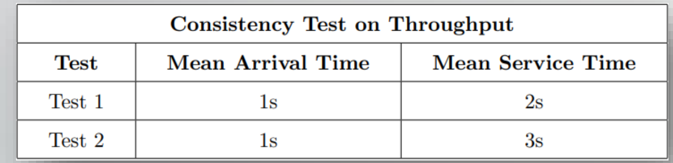
  - ***Player*** **:**
    1. Pushes minion and boss messages in their own FIFO queue.
    2. When needed, it computes the recover life and pushes the minion again in the queue.
    3. Process minion and boss messages.
    4. Record statistics.



- *Messages*
  - **Opponent Message :** New type of messages that hold the service time for each opponent in ***simtime_t*** format.

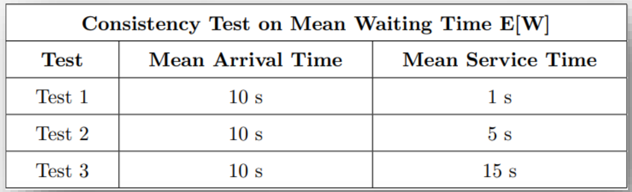
Verification – Degeneracy and Consistency Test

- *Degeneracy Test*
  1. **If $\frac{1}{\lambda}$ and/or $\frac{1}{\mu}$ of minions is set to 0**: the simulation proceeds by only having bosses.
  2. **If $\frac{1}{\lambda}$ and/or $\frac{1}{\mu}$ of bosses is set to 0**: the simulation proceeds by only having minions.
  3. **If $\frac{1}{\lambda}$ and/or $\frac{1}{\mu}$ of both opponents is set to 0**: the simulations ends at the first event.
- *Consistency Test*



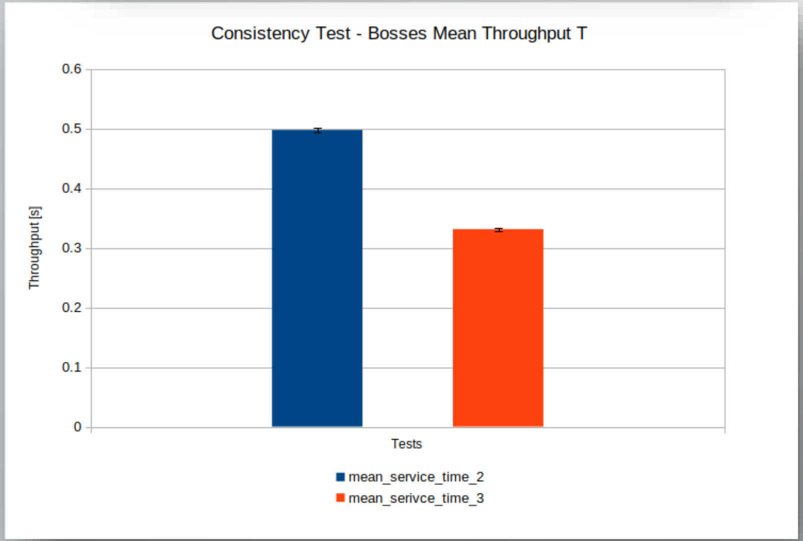
Consistency Test on Throughput

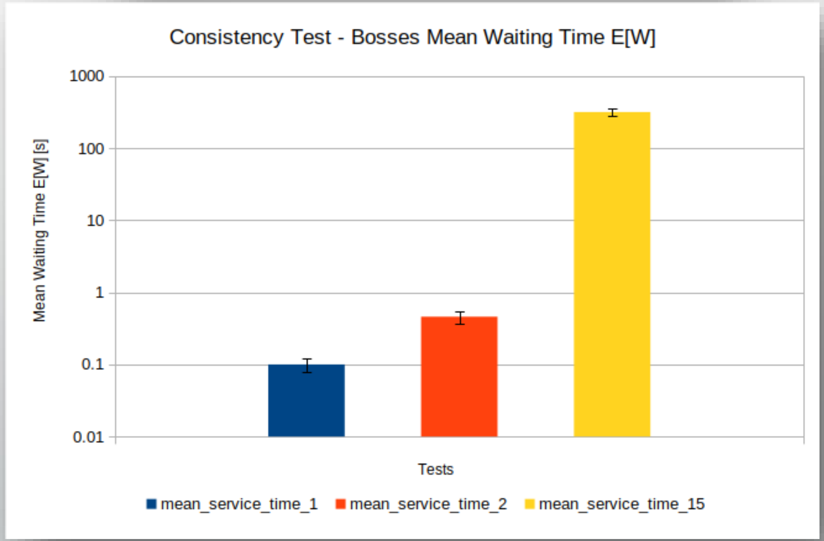
Test	Mean Arrival Time	Mean Service Time
Test 1	1s	2s
Test 2	1s	3s



Consistency Test on Mean Waiting Time E[W]

Test	Mean Arrival Time	Mean Service Time
Test 1	10 s	1 s
Test 2	10 s	5 s
Test 3	10 s	15 s



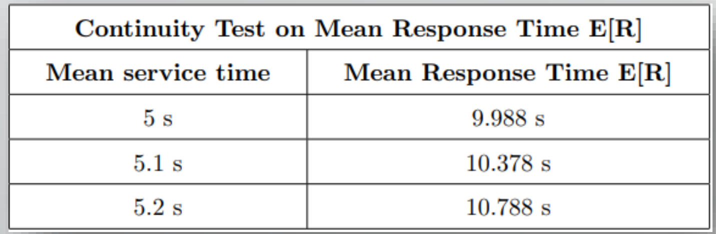


The **Throughput** is higher when the service time decreases and vice-versa.
In terms of the model studied the player defeats more opponents if their life decreases.

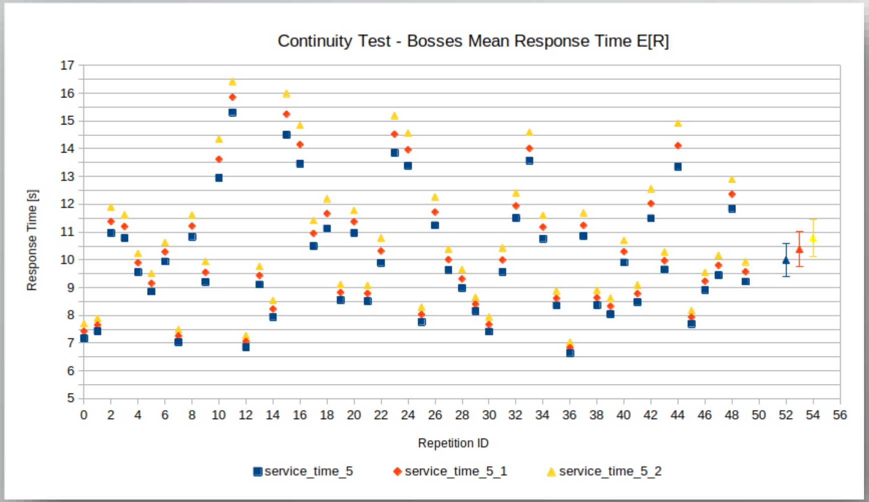
When the service time is lower than the arrival time, the **Mean Waiting Time** takes values around o (the player has enough time to fight and defeat the opponents before a new one arrives)

Verification – Continuity and Monotonicity Test

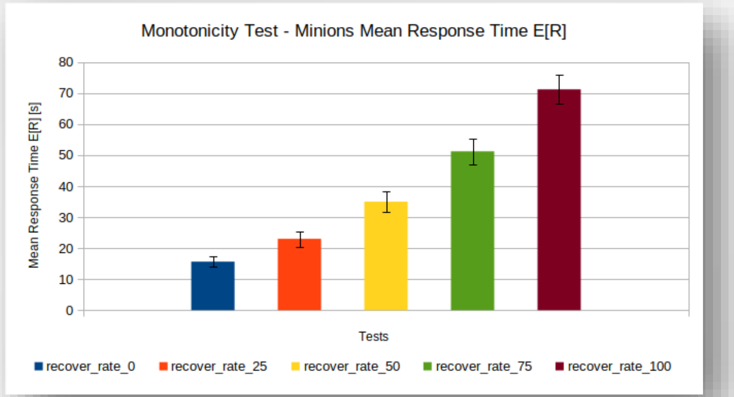
- *Continuity Test*

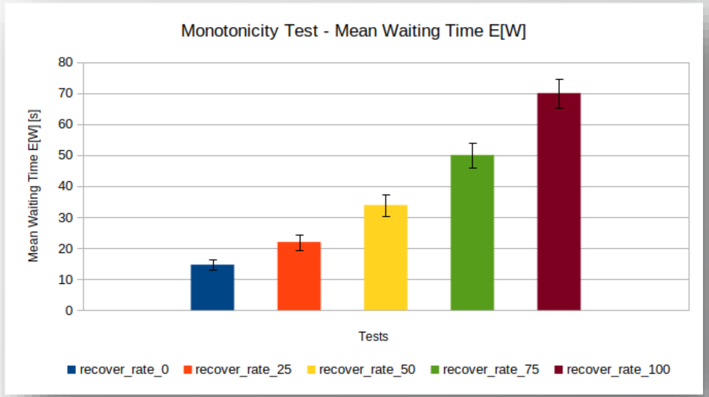


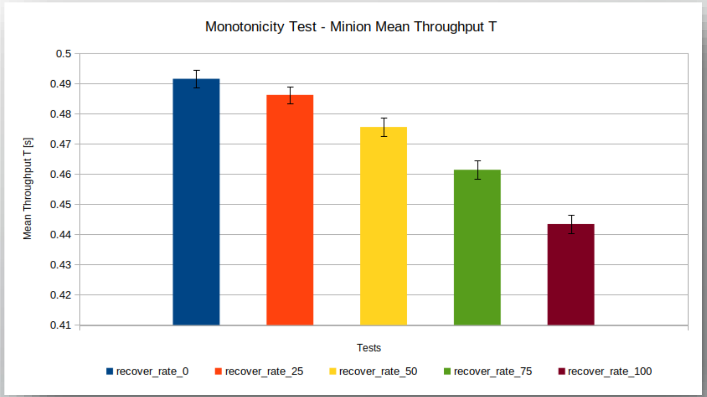
Continuity Test on Mean Response Time E[R]	
Mean service time	**Mean Response Time E[R]**
5 s	9.988 s
5.1 s	10.378 s
5.2 s	10.788 s



- *Monotonicity Test*

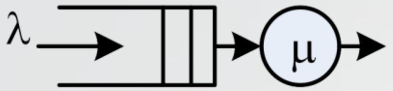






Verification – Against Theoretical Model

- *Assumptions*



1. Only one opponent type (bosses or minions)
2. Exponential interarrival and service time

- *Formulas*
  - ***System utilization :*** $\rho = \frac{\lambda}{\mu}$
  - ***Number of jobs in the system :*** $E[N] = \frac{\rho}{1-\rho}$
  - ***Mean Response Time :*** $E[R] = \frac{E[N]}{\lambda}$
  - ***Mean Waiting Time :*** $E[W] = E[R] - \frac{1}{\mu}$
  - ***Throughput :*** $\gamma = \mu\rho$

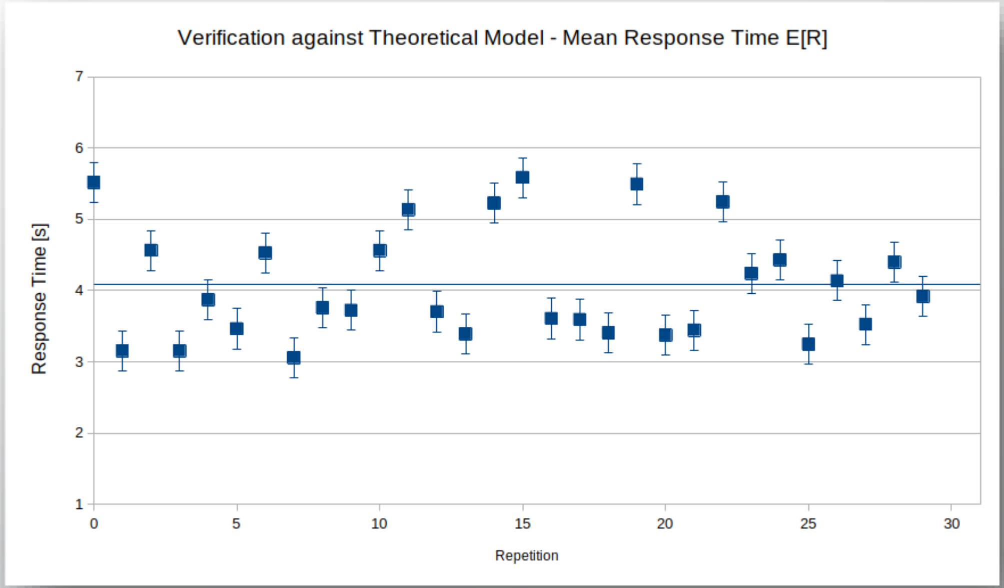
- *Scenario*
  - **Bosses Mean Arrival Time :** $\frac{1}{\lambda} = 1$
  - **Boss Mean Service Time :** $\frac{1}{\mu} = 1$
  - **Simulation duration : 2000 seconds**
  - **Repetitions : 30**

- *Results*

$\rho = 0.8,\ E[N] = 4,\ E[R] = 4,\ E[W] = 3.2,\ T = 1$

Verification Against Theoretical Model results

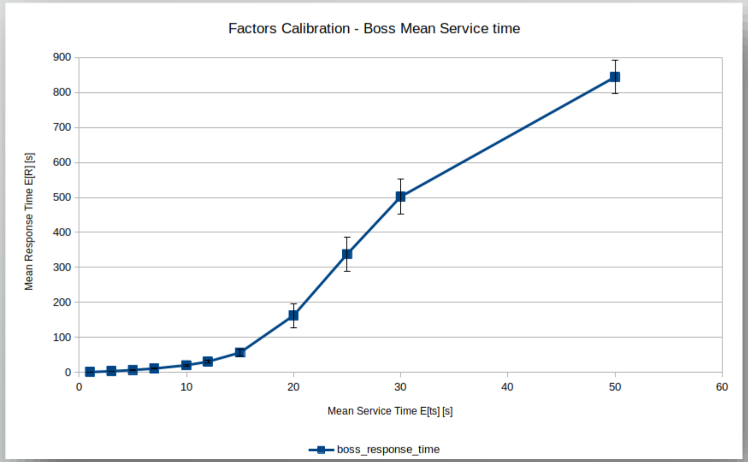
Metric	Value	Confidence Interval
E[R]	4.0825	0.2814
E[W]	3.2818	0.2776
T	0.99845	0.00807



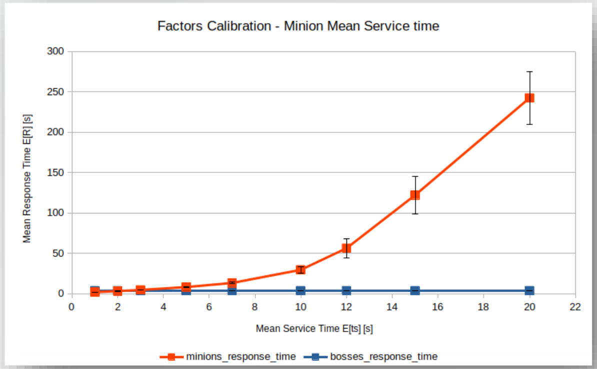
Factors Calibration

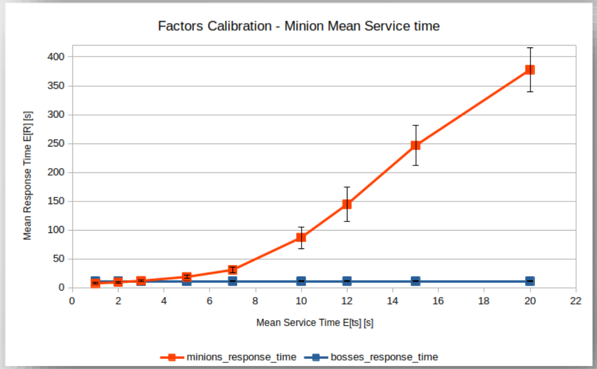
- *Scenario Calibration*

Bosses Factors

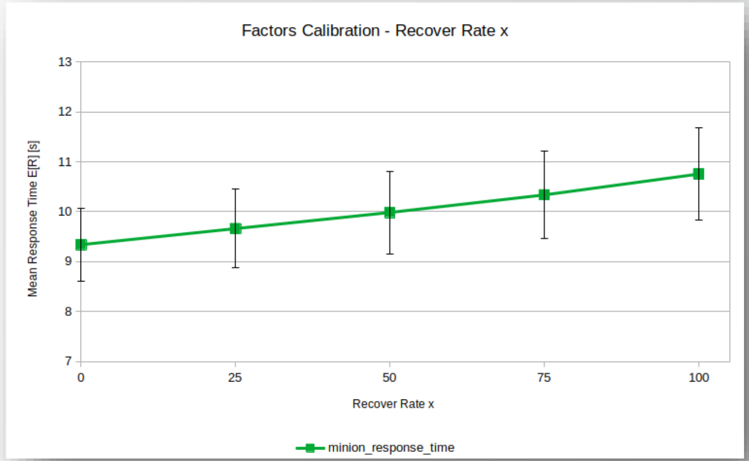


Minions Factors



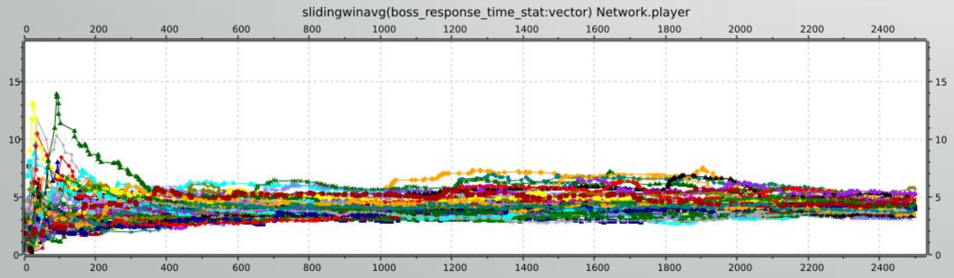


Recover Rate x



- ***Boss Mean Arrival Time*** $\frac{1}{\lambda_b}$: [20s, 25s]
- ***Boss Mean Service Time*** $\frac{1}{\mu_b}$: [5s, 10s]
- ***Minion Mean Arrival Time*** $\frac{1}{\lambda_m}$: [10s, 15s]
- ***Minion Mean Service Time*** $\frac{1}{\mu_m}$: [1s, 7s]
- ***Recover Rate x :*** [0,100]

- *Warm-Up and Simulation Time*

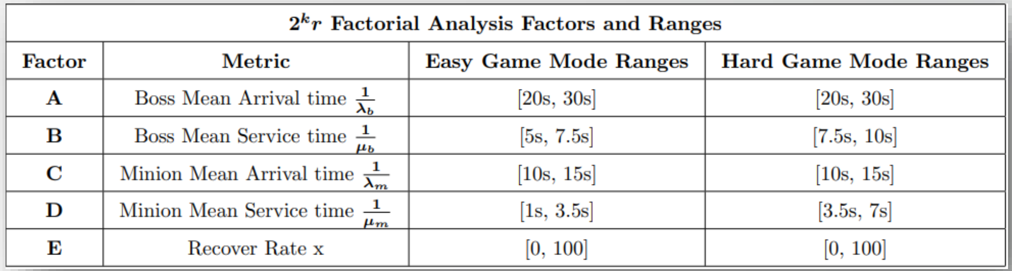


- ***Warm-up Time*** *: 350 seconds*
- ***Simulations Time*** *: 2500 seconds*

2kr Factorial Analysis

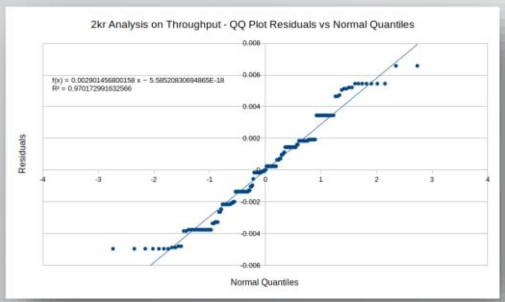
The 2kr analysis is made on both metrics to be analysed, the Throughput T and the Mean Waiting Time E[W], with the aim of understanding the contribution of factors.

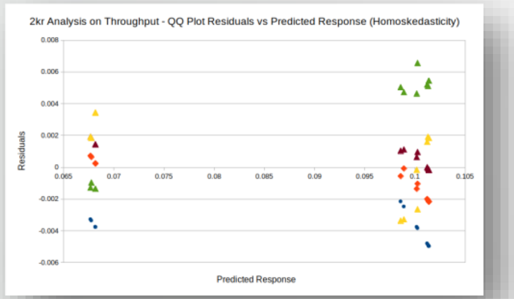
Hypotesis to assess: ***The residuals (errors) are IID's normal RV with a null mean and a costant standard deviation.***



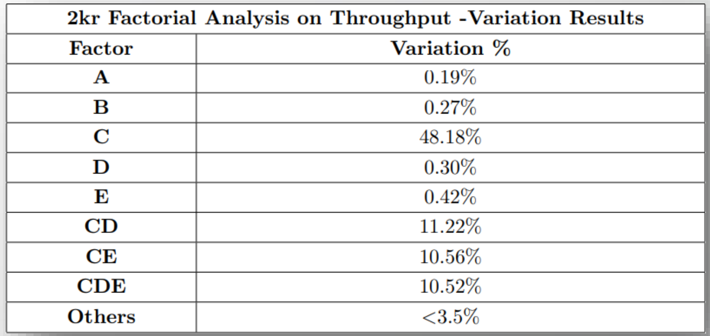
$2^k r$ Factorial Analysis Factors and Ranges			
Factor	**Metric**	**Easy Game Mode Ranges**	**Hard Game Mode Ranges**
A	Boss Mean Arrival time $\frac{1}{\lambda_b}$	[20s, 30s]	[20s, 30s]
B	Boss Mean Service time $\frac{1}{\mu_b}$	[5s, 7.5s]	[7.5s, 10s]
C	Minion Mean Arrival time $\frac{1}{\lambda_m}$	[10s, 15s]	[10s, 15s]
D	Minion Mean Service time $\frac{1}{\mu_m}$	[1s, 3.5s]	[3.5s, 7s]
E	Recover Rate x	[0, 100]	[0, 100]

- ***Throughput***



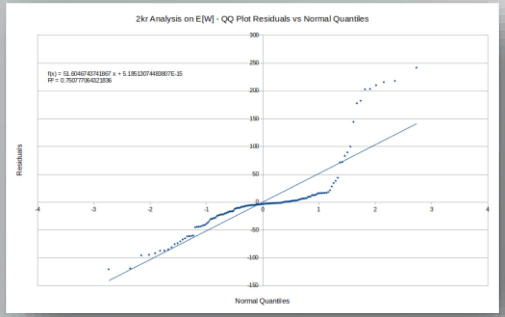


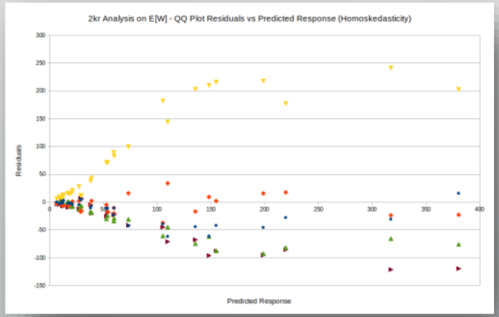
Results



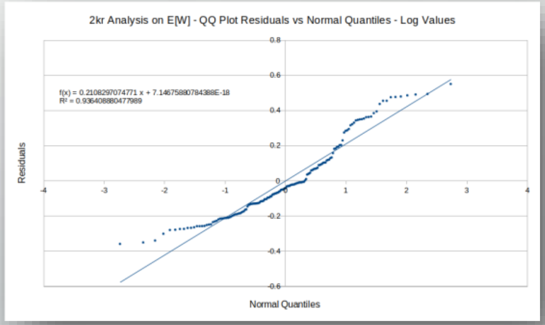
2kr Factorial Analysis on Throughput -Variation Results	
Factor	**Variation %**
A	0.19%
B	0.27%
C	48.18%
D	0.30%
E	0.42%
CD	11.22%
CE	10.56%
CDE	10.52%
Others	<3.5%

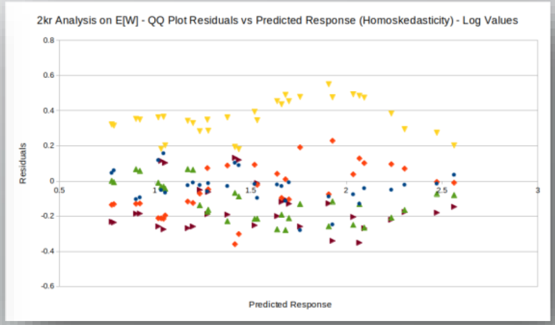
- ***Waiting Time***





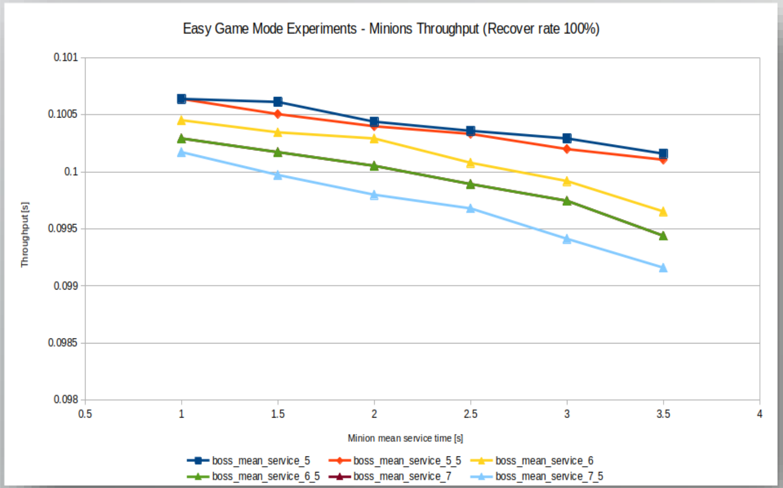
Log Transformation





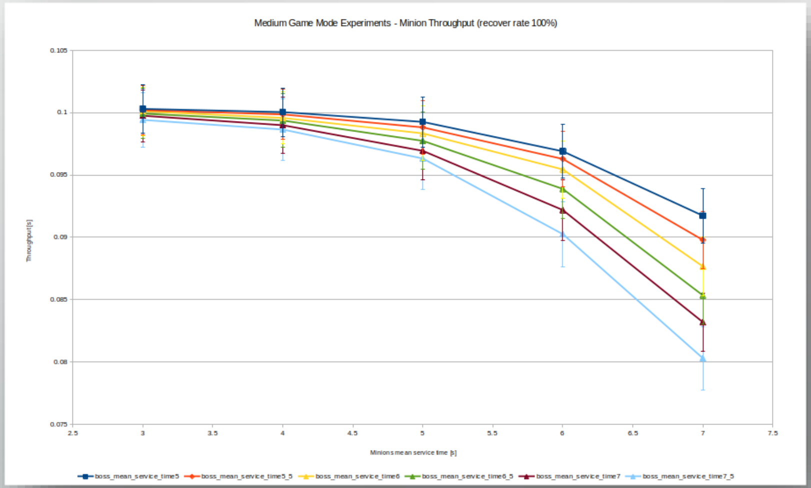
Simulation Experiments - Throughput

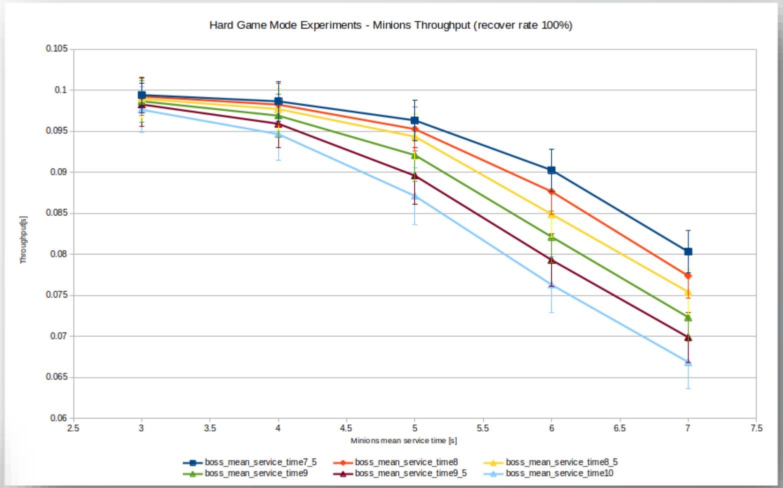
- **Easy Game Mode :** $\frac{1}{\lambda_b} = 20s$ $\frac{1}{\mu_b} = [5s, 7.5s]$ *(steps of 0.5s)* $\frac{1}{\lambda_m} = 10s$ $\frac{1}{\mu_m} = [1s, 3.5s]$ *(steps of 0.5s)* *Recover rate x = [0%, 100%]*
- **Medium Game Mode :** $\frac{1}{\lambda_b} = 20s$ $\frac{1}{\mu_b} = [5s, 7.5s]$ *(steps of 0.5s)* $\frac{1}{\lambda_m} = 10s$ $\frac{1}{\mu_m} = [3s, 7s]$ *(steps of 1s)* *Recover rate x = [0%, 100%]*
- **Hard Game Mode :** $\frac{1}{\lambda_b} = 20s$ $\frac{1}{\mu_b} = [7.5s, 10s]$ *(steps of 0.5s)* $\frac{1}{\lambda_m} = 10s$ $\frac{1}{\mu_m} = [3, 7s]$ *(steps of 1s)* *Recover rate x = [0%, 100%]*



- **Easy Game Mode Throughput Results :** The results are almost identical among the various experiments

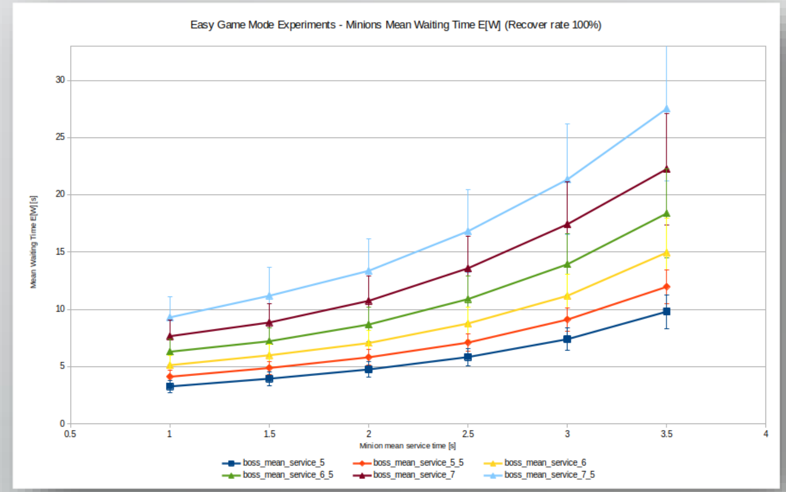
Medium Game Mode Throughput Results : it could take up to 12.5 seconds to defeat a minion (throughput equal to 0.08s), which, for long simulations, could be relevant for the overall performance of the system.

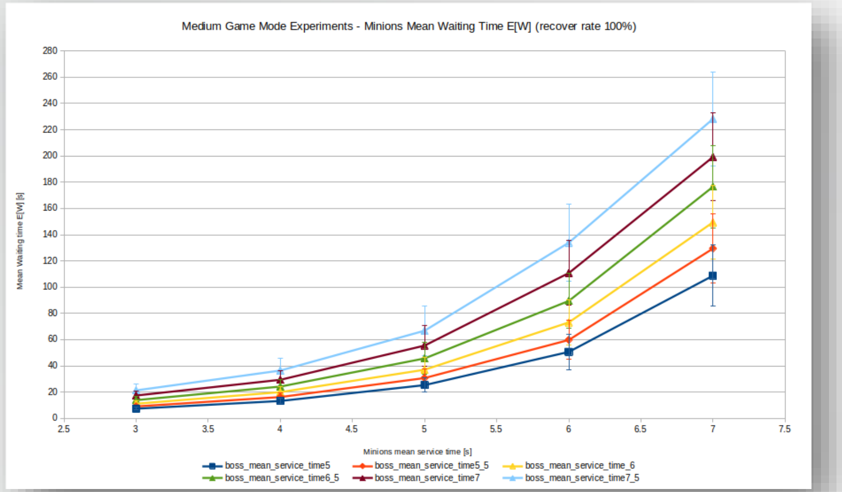


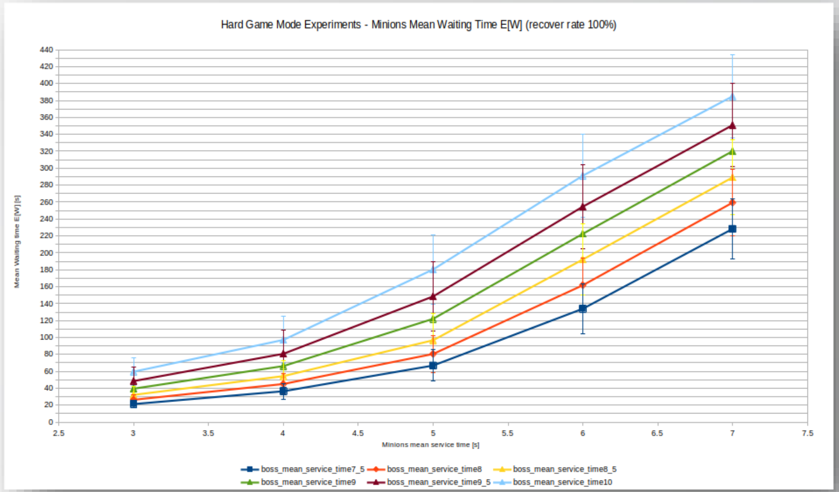


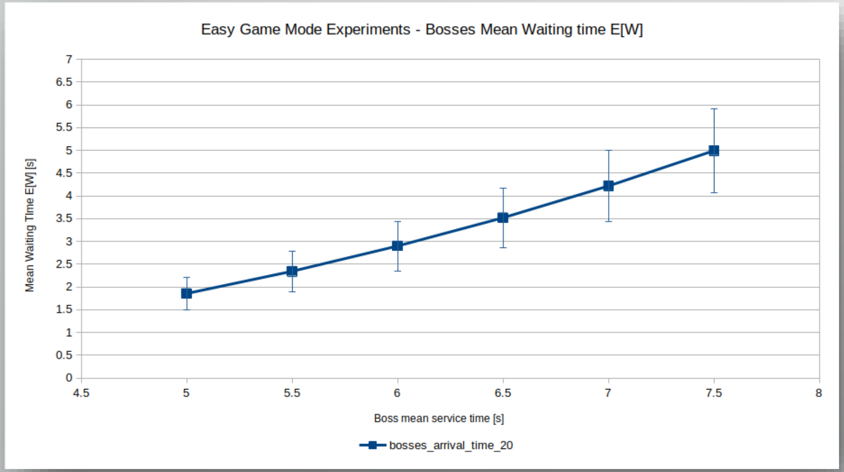
Hard Game Mode Throughput Results: the minion throughput shows a further drop, due to the increase in the bosses mean service time values employed in the hard game mode. the throughput value reached approximately 0.065s, i.e. 1 minion defeated every 15 seconds.

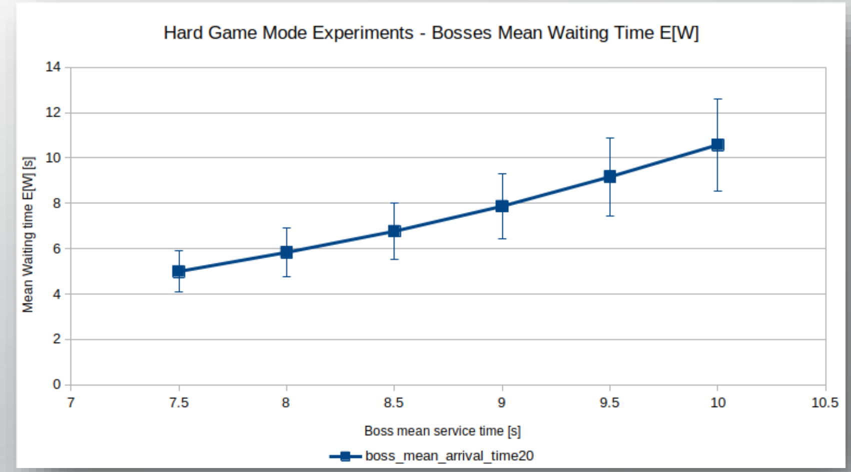
Simulation Experiments – Mean Waiting Time E[W]





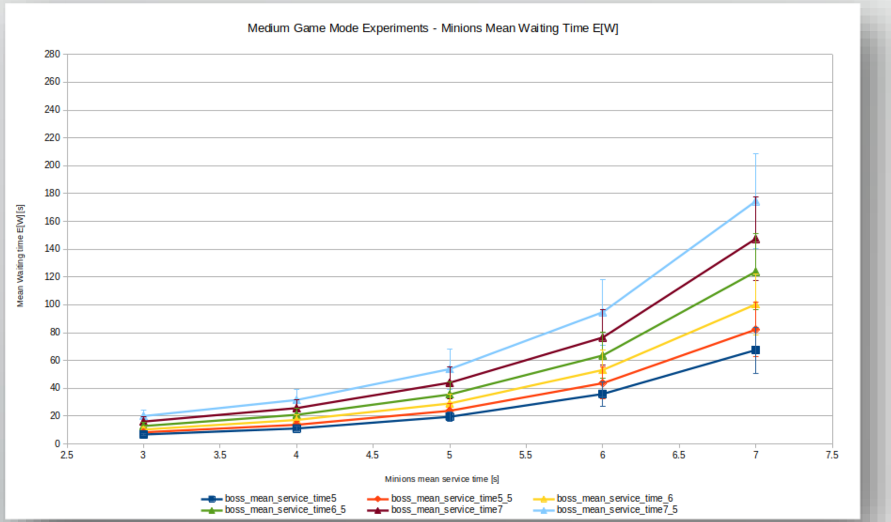


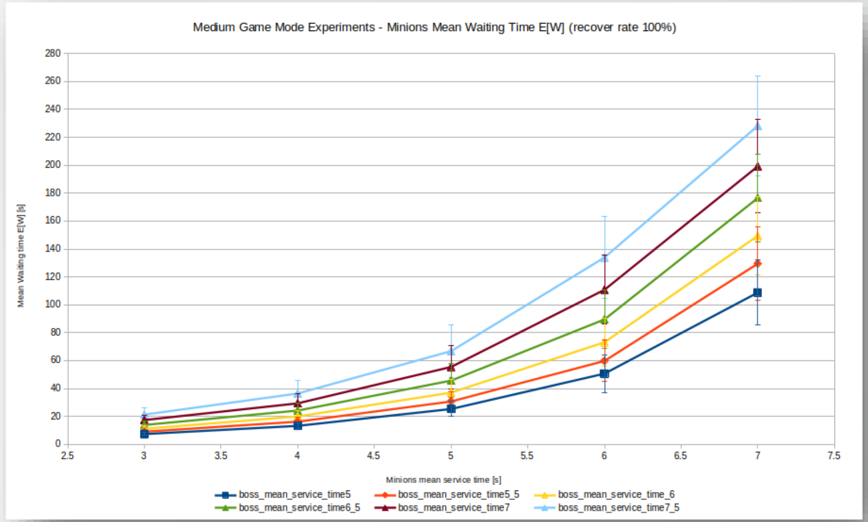




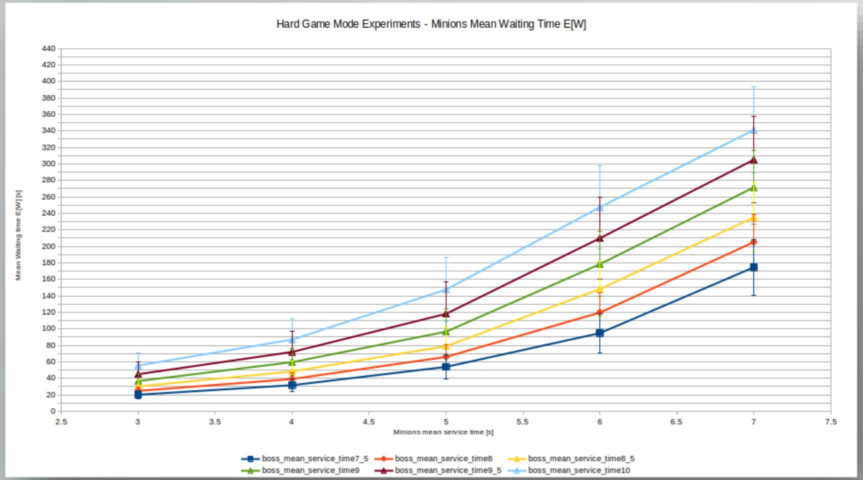
Simulation Experiments – Recover Rate x Effect

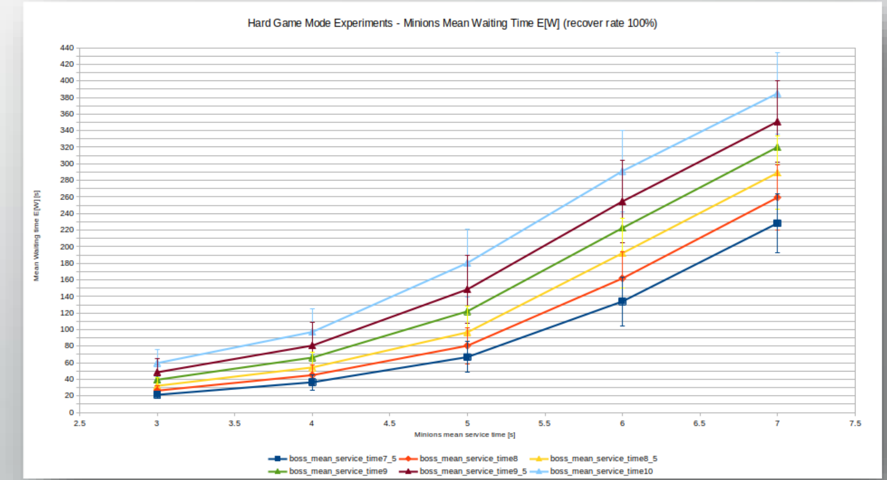
- **Medium Game Mode MEAN WAITING TIME E[W]**





- **Hard game Mode MEAN WAITING TIME E[W]**





Conclusions

- **Why is this study useful and which kind of benefits can it give us?**

  1. Knowing these results, average game experiences can be estimated, predicting how a game might take place and which goals users would achieve.
  2. Being able to study the possible outcomes of a game in advance and set the basic parameters of the game appropriately can improve users performance and, above all, their level of satisfaction and fulfilment, which can be key factors in the success of a videogame.