GDC | March 21-25, 2022
San Francisco, CA

FROM 'CLASH OF CLANS' TO 'EVERDALE' SCALING FROM SOLO TO SOCIAL

Tristan Williams

#GDC22

Tristan Williams
Senior Programmer

Supercell 2014-
Remedy 2008-2014
Splash Damage 2005-2008
Ratbag 2004-2005

TOPICS

Introduction

Design

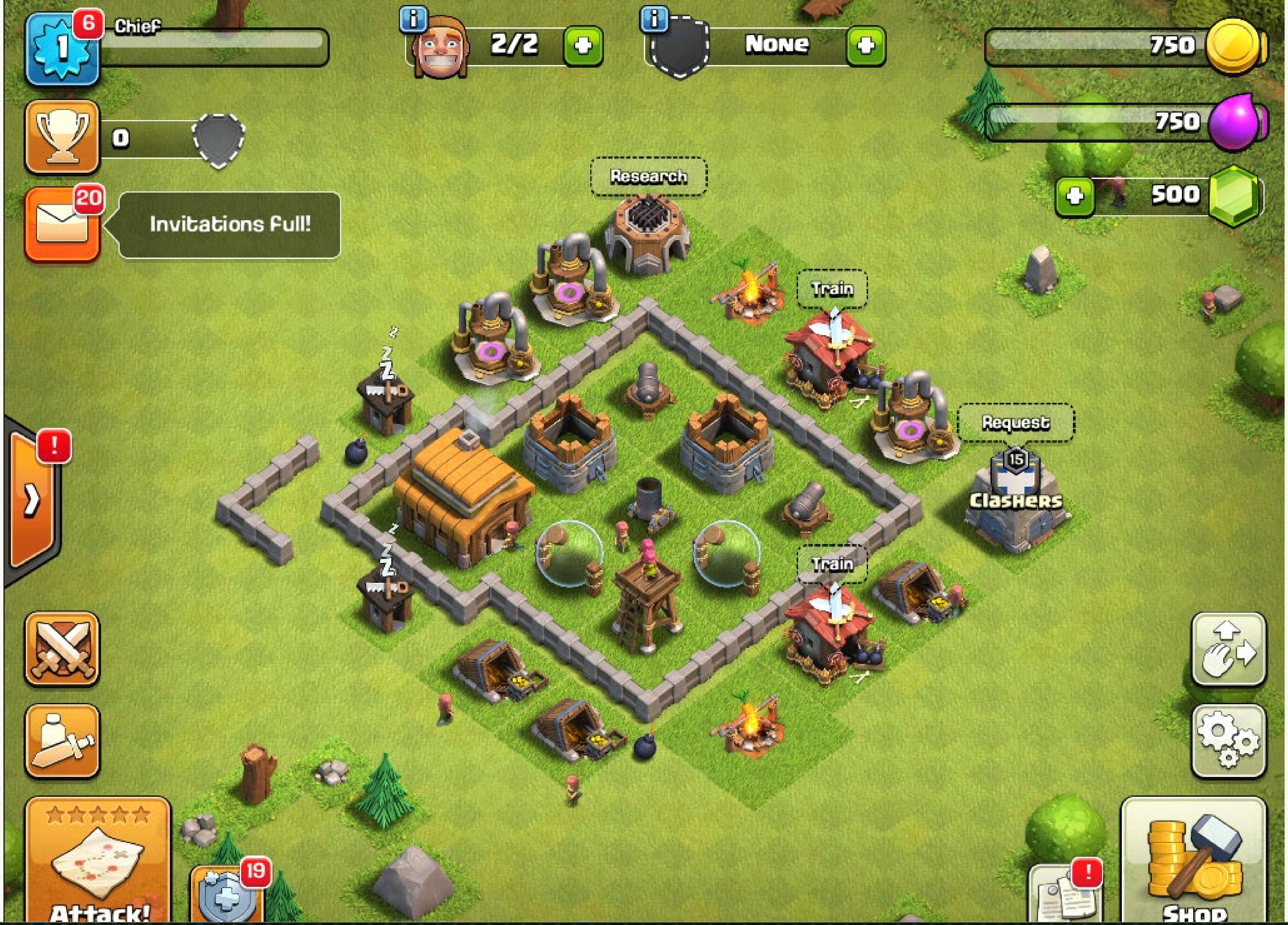
Tech

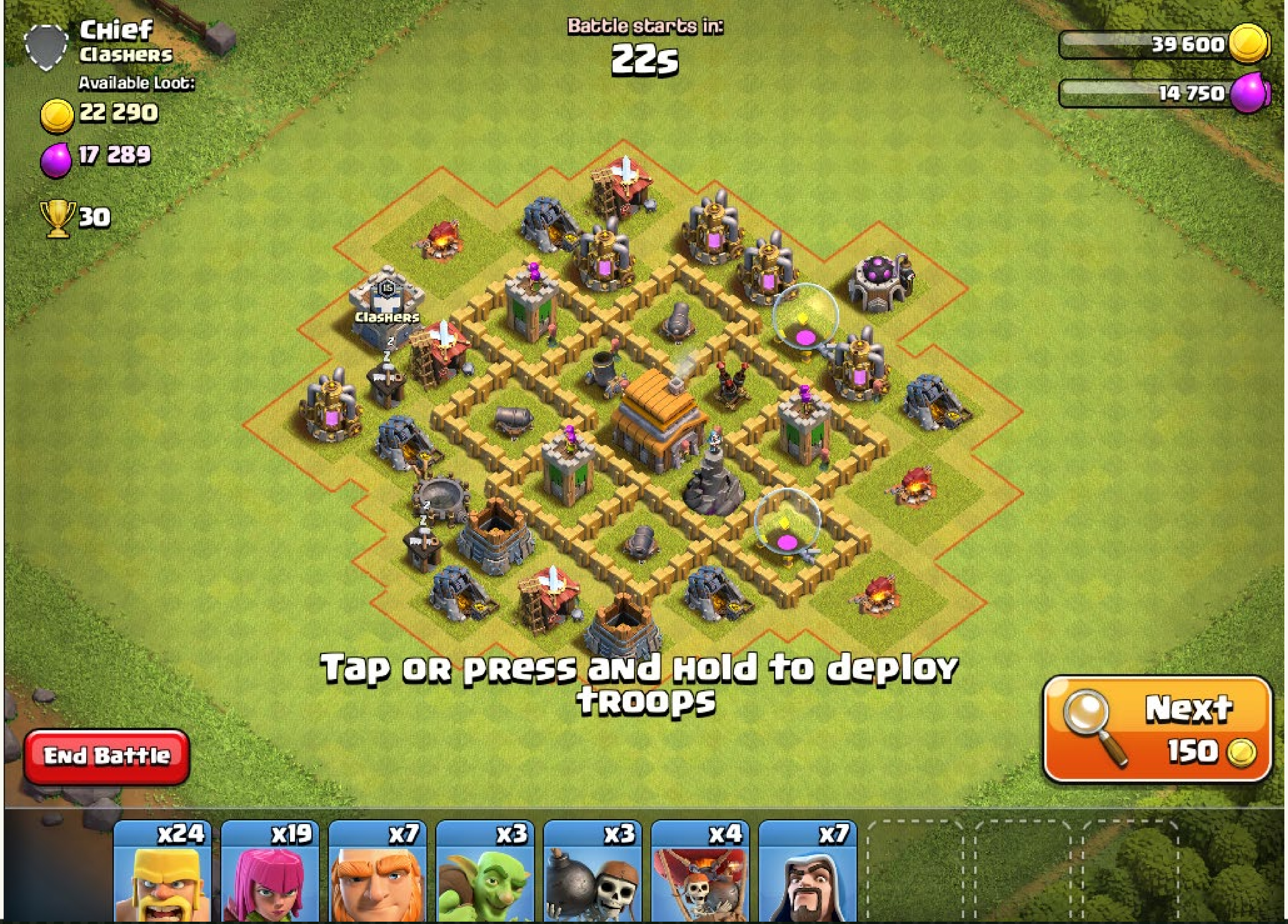
Takeaways

INTRODUCTION

Released 2012

Soft launched 2021


6
1
Chief
2/2
None
750
750
500
0
20
Invitations Full!
Research
Train
Request
15
Clashers
Train
Attack!
19
Shop


Chief
Clashers
Available Loot:
22 290
17 289
30
Battle starts in:
22s
39 600
14 750
Clashers
Tap or press and hold to deploy troops
End Battle
Next
150
x24
x19
x7
x3
x3
x4
x7

CLASH OF CLANS

- One village on screen at a time
- Core gameplay is solo

CLASH OF CLANS

Clans

- Opt-in social layer on top of the solo core game
- Clan Wars – another source of resources
- Your progress is still entirely your own!

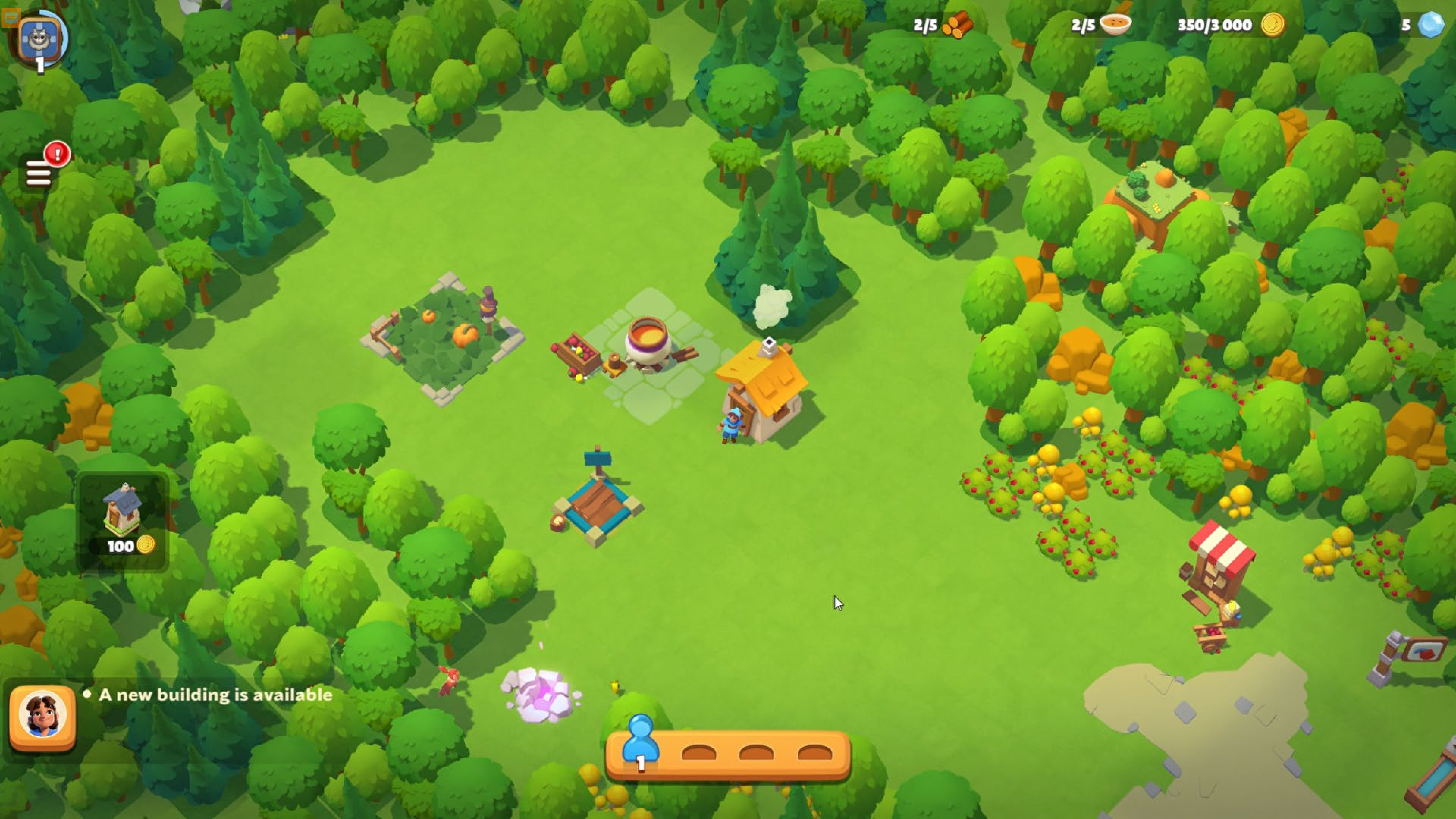
EVERDALE

The Settlers II
VENI, VIDI, VICI

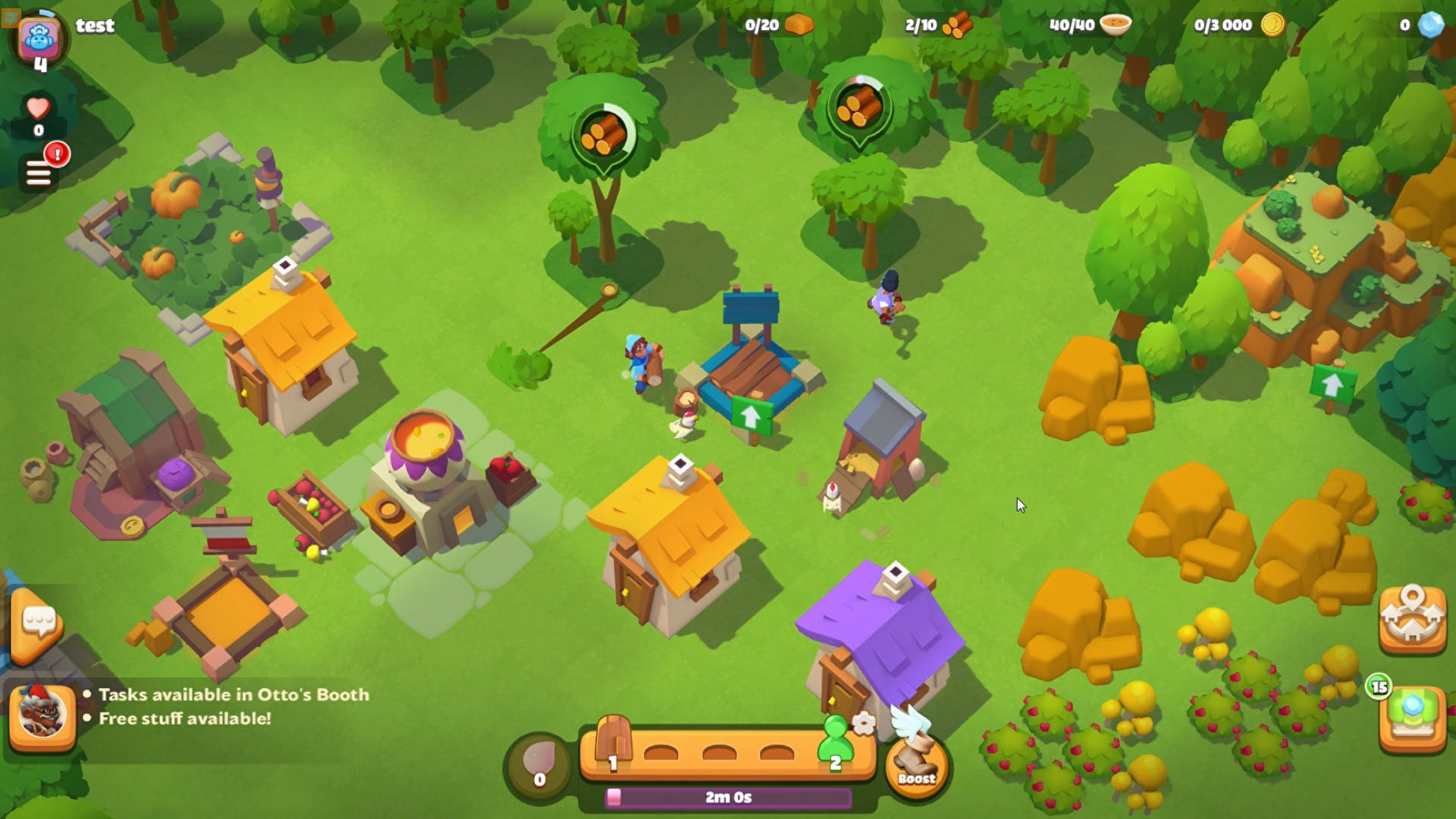
The Sims 4™

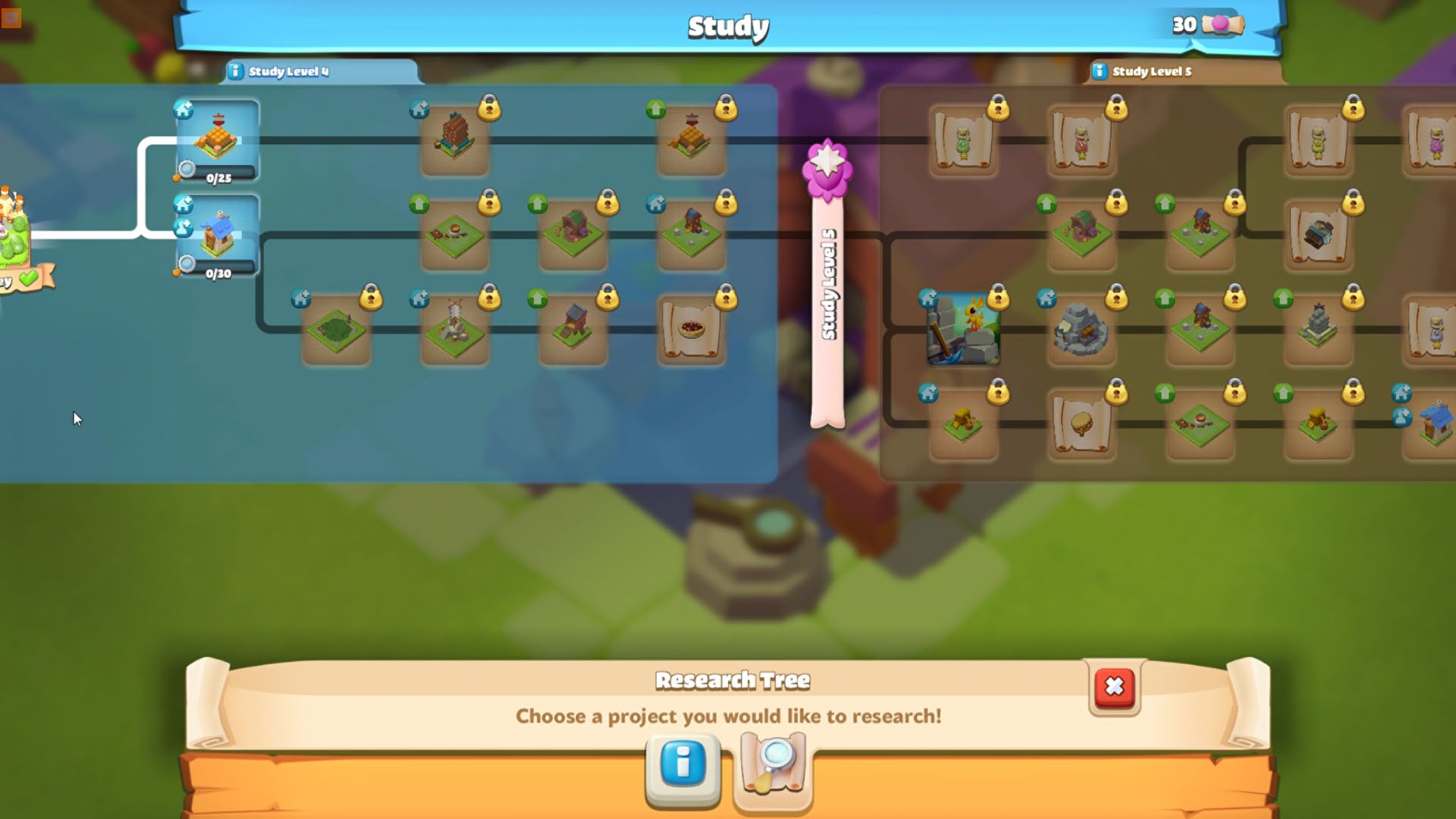
Fallout® Shelter™

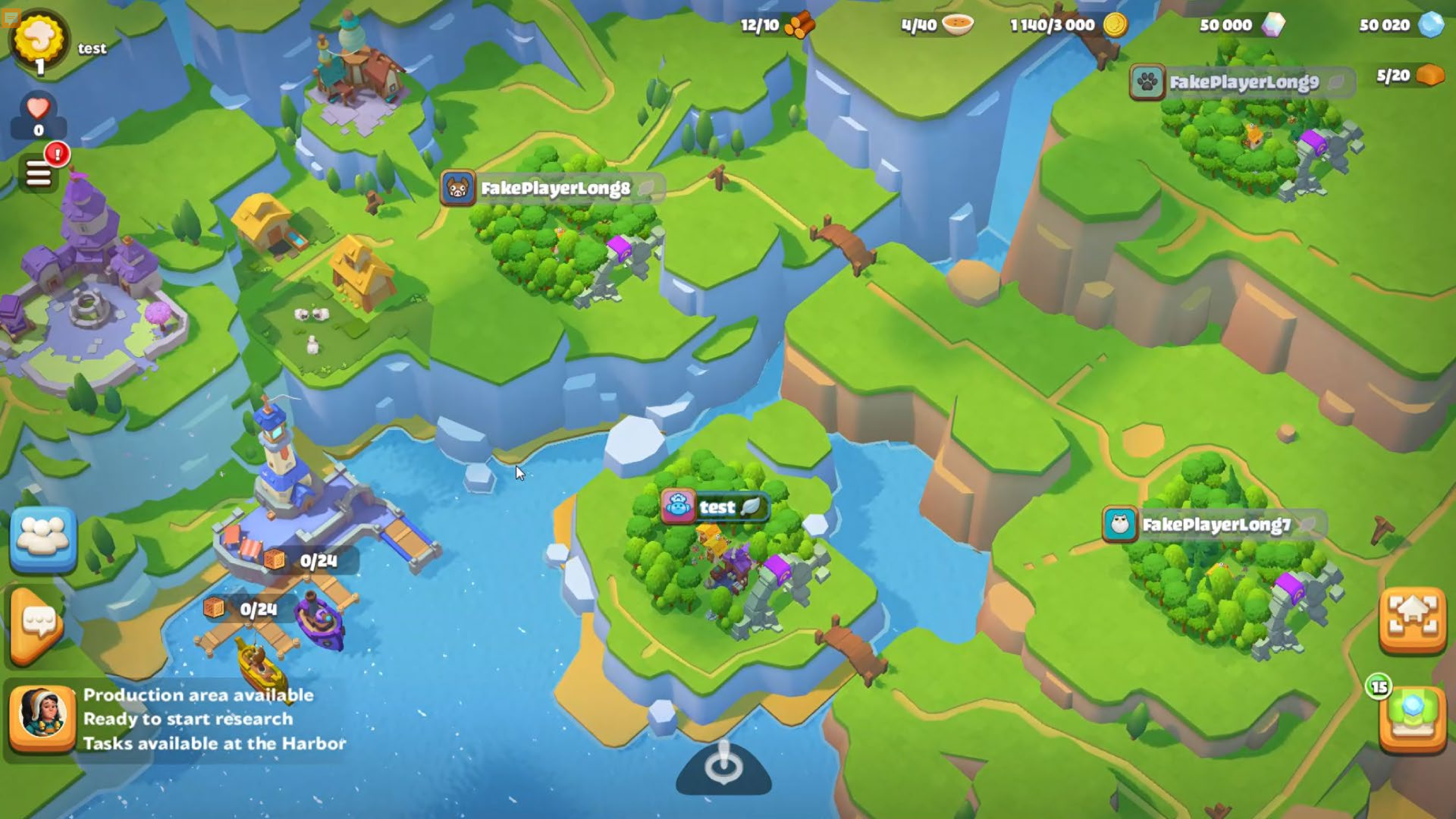
SID MEIER'S
CIVILIZATION VI



A new building is available


test
4
0/20
2/10
40/40
0/3 000
0
0
Tasks available in Otto's Booth
Free stuff available!
0
1
2
Boost
2m 0s
15


Study
30
Study Level 4
Study Level 5
0/25
0/30
Study Level 5
Research Tree
Choose a project you would like to research!



Production area available
Ready to start research
Tasks available at the Harbor

EVERDALE

- Valley is deeply linked to your village gameplay
- Village supports the valley and valley supports the village
- Shared progress
- Deep collaboration

TEAM

- Small, independent teams
- Started ~2016
- <6 people
- Grew to 10-20 for launch
- 4 client programmers & 2 server programmers

THE DREAM

DO GAME DEVS DREAM OF SIMULATED SHEEP?

- Everything starts with some kind of dream
- Dreams come in all shapes and sizes

DREAM “SMALL”

- A cool mechanic
- A unique gameplay idea
- An art style
- Some interesting tech

DREAM “SMALL”

- Often clear reference exists
- But: Differentiation?

DREAM “BIG”

- Lofty goals
- User experience/emotion driven
- “How does the game make me feel?”
- Start from the high fantasy

DREAM "BIG"

DREAM "BIG"

Daunting!

- Mechanics may not be understood yet
- Lack of reference
- The tech may not exist yet

Sounds easy, right?

EVERDALE DREAMS

In the beginning

- Broader audience
- Not about combat
- More collaborative than anything we've made before
- More immersive

EVERDALE DREAMS

"Small" dreams

- Village builder
- Peaceful
- Relaxing

EVERDALE DREAMS

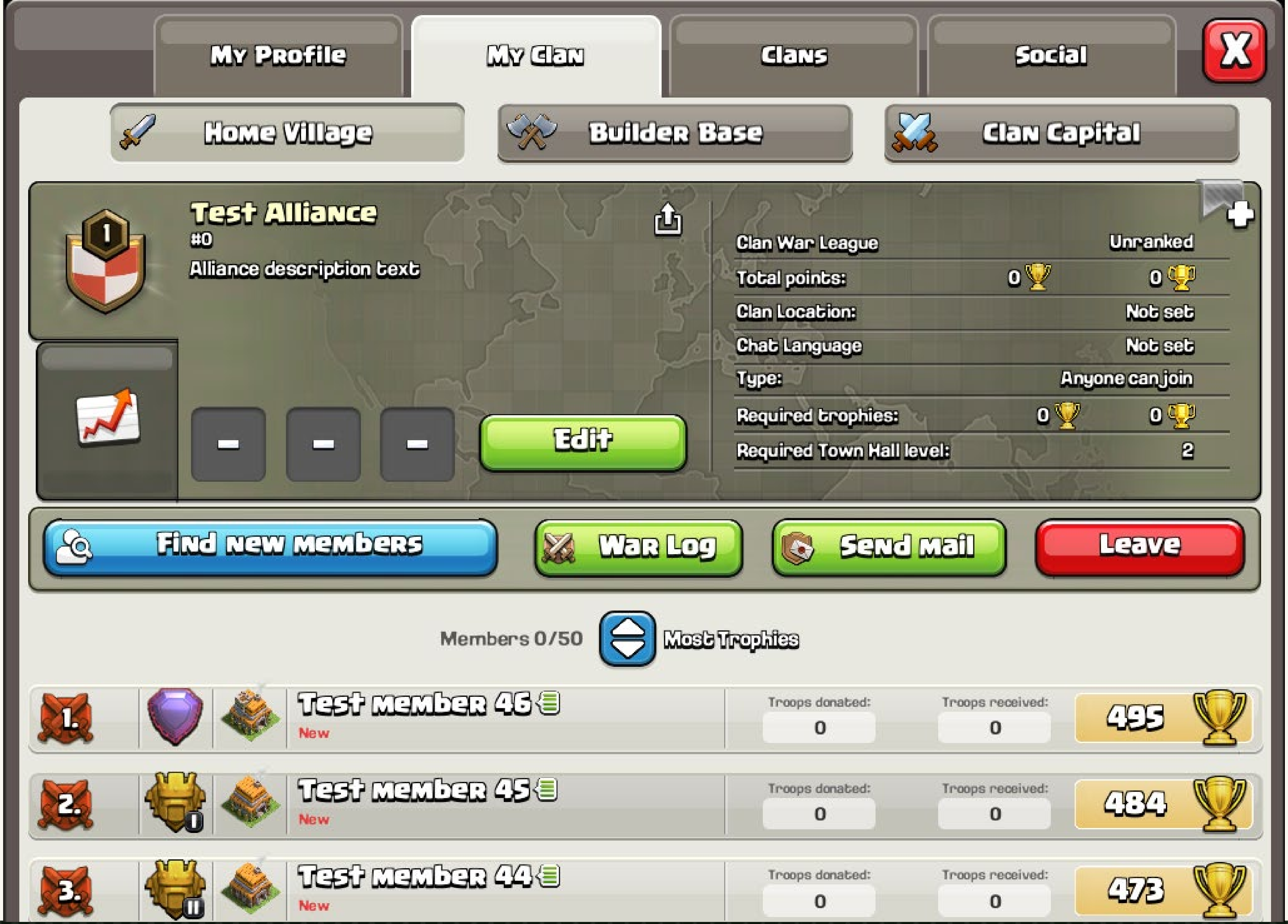
"Big" dreams

- Game with collaboration built in at the core
- Real, meaningful cooperation
- Seamless world
- Multiple villages

THE DREAM

- Every game starts with some sort of dream
- Helps to identify these early
- These will drive tech choices

COLLABORATION

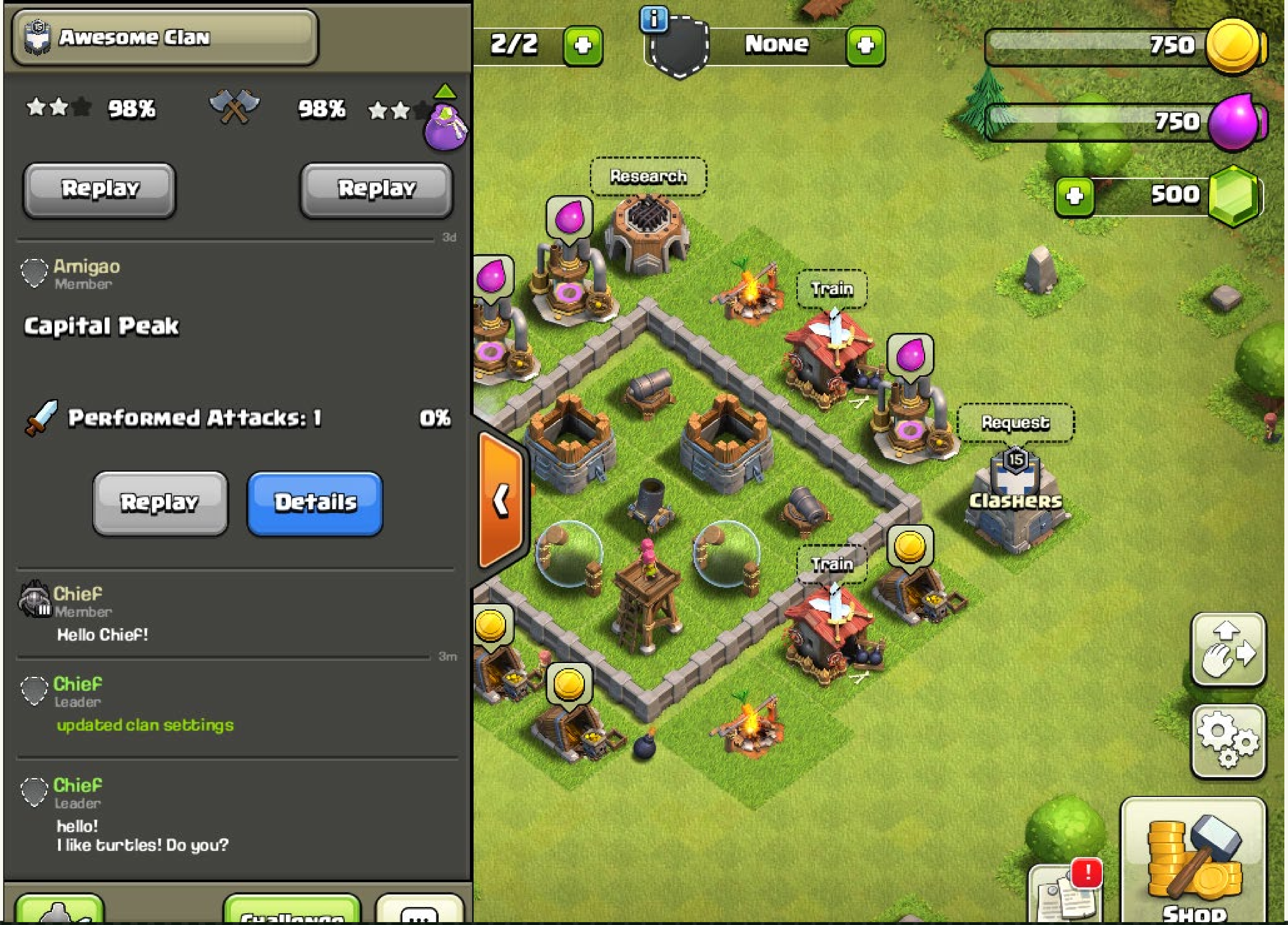


Test Alliance
#0
Alliance description text

Clan War League		Unranked
Total points:	0	0
Clan Location:		Not set
Chat Language		Not set
Type:		Anyone can join
Required trophies:	0	0
Required Town Hall level:		2

Members 0/50

		Troops donated:	Troops received:	
1.	Test member 46 New	0	0	495
2.	Test member 45 New	0	0	484
3.	Test member 44 New	0	0	473



Awesome Clan

98% 98%

Replay Replay

3d

Amigao
Member

Capital Peak

Performed Attacks: 1 0%

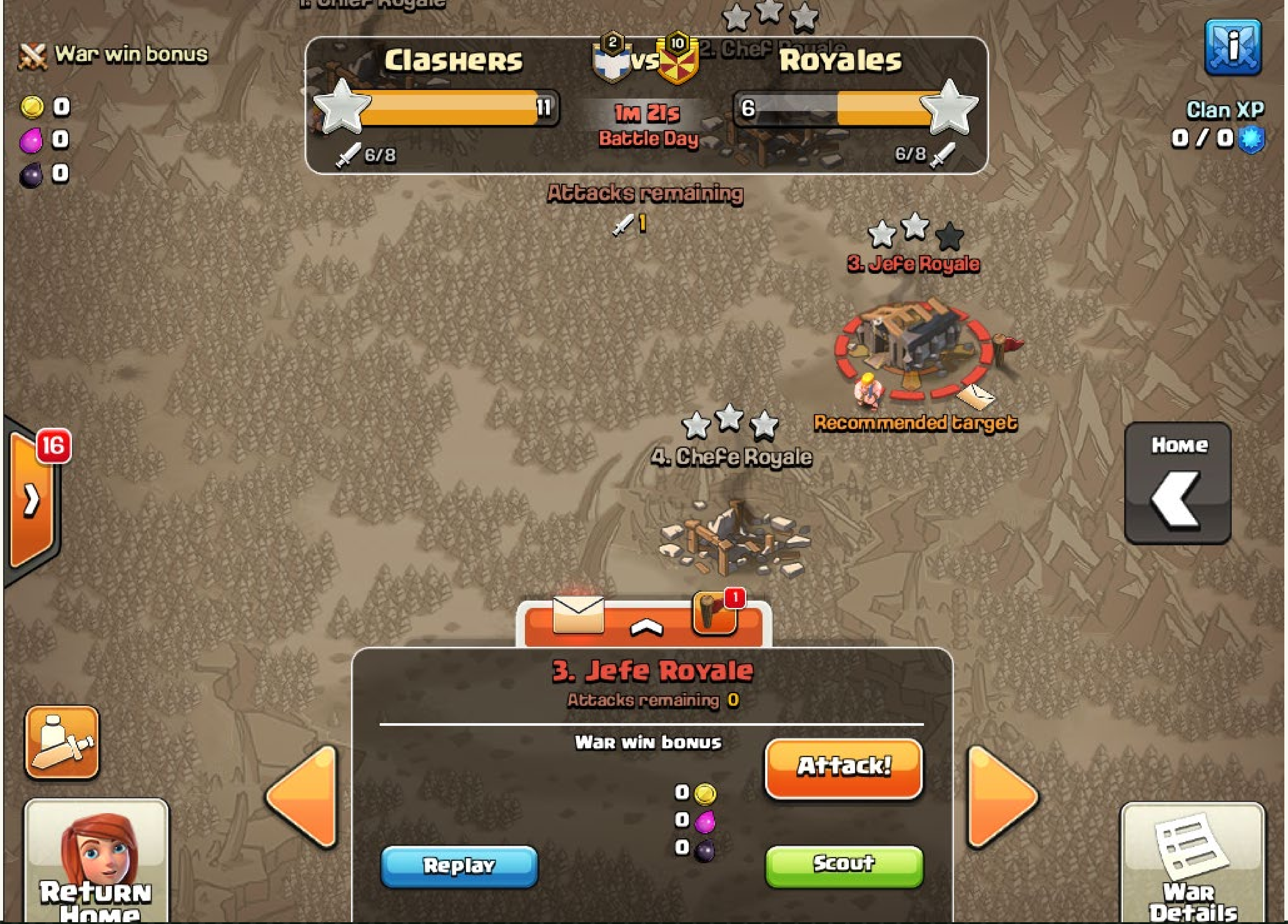
Replay Details

Chief
Member
Hello Chief!

3m

Chief
Leader
updated clan settings

Chief
Leader
hello!
I like turtles! Do you?


War win bonus
0
0
0
Clashers
vs
Royales
11
6
1m 21s
Battle Day
6/8
6/8
Clan XP
0 / 0
Attacks remaining
1
3. Jefe Royale
Recommended target
4. Chefe Royale
2. Chef Royale
16
Home
3. Jefe Royale
Attacks remaining 0
War win bonus
Attack!
0
0
0
Replay
Scout
Return Home
War Details

COLLABORATION++

- Seamless world
- Watch other people play in real time
- Real gameplay interaction in the world – not just in menus
- Teamwork really means something
- Shared goals

Exciting!

COLLABORATION++?

.. Except ..

- Big world, expensive to simulate & render
- Complex to design
- Complex to test

TECH

TECH BACKGROUND

Forked Clash of Clans

- In-house engine
- Single village on screen at a time
- Collaboration only in menus
- 2D, sprite atlas based

TECH BACKGROUND

- Client/server architecture
- Server authoritative, asserts clients in sync
- “Logic” code runs on both client and server
- Logic state persistent

TECH BACKGROUND

- Client & logic code relied heavily on singletons
- Convenient and easy to code
- Could only have one village running at a time

MULTIPLE VILLAGES

- Broke this up into the concept of “contexts”
- Context bundles all the subsystems of village state

MULTIPLE VILLAGES

Benefits

- Run background, headless, copies of the village
- Debug logic verification
- Prediction into the future

Design goal ended up giving us nice technical benefits!

SEAMLESS WORLD

- One “render world” per village
  - Own coordinate system
- One render world for the valley
- View composed by stitching together worlds
  - Render each village with an offset

GOING 3D

“Small dream” - village builder

- Diverse villagers
- Performing lots of different tasks
- Possibility to customize villagers

That’s a lot of permutations of content!

GOING 3D

Made the choice to go 3D

- BUT: no true 3D engine yet!
- Some 3D rendering capabilities eg characters in Brawl Stars

GOING 3D

- Built a simple 3D engine
- Usage as close as possible to our 2D engine
- Camera controls built to replicate Clash of Clans camera

Engine made it's way back to Brawl Stars

COLLABORATION TECH

COLLABORATION TECH

- Multiple players in one seamless world
- Player actions sent to other clients
- In your own village, validation is relatively simple

COLLABORATION TECH

What about influencing shared state?

- Previous games:
  - Manually persisted shared state
  - Manual message handling
  - Manual error resolution for every feature
  - Every feature needed both client & server expertise
- We wanted to have far more elaborate features
  - Tedious, prohibitive development time

FIRST PASS

- Added shared logic state for Valley
  - Shared logic code & context, too
- Well-defined “action” object encapsulating:
  - Player’s action and parameters
  - All validation checks
  - Handling of validation failures
  - Rollbacks

FIRST PASS

- Validate on client, if OK, send to server
- Server validate against authoritative village state
  - Fail: respond to client with fail, client runs action failure code
  - OK: validate against authoritative valley state
    - Fail: back to server village & client, run failure code
    - OK: apply, distribute to other clients

FIRST PASS

Pros

- Neater and easier to see all of the handling in one place
- Can share a lot of logic code with village
- One game programmer can build a complete transaction without help from a server programmer

FIRST PASS

Cons

- Onus is on the programmer to foresee all potential failure cases
  - Implement appropriate rollback code for all cases

- Still tedious, laborious, error prone code!
  - Similar to the “old way” in many respects

- Client/UI code needs to be written so that all edge cases are handled with various “bail out” scenarios
  - Often non-trivial!

IMPROVEMENTS

- Adopt resimulation-based approach
- Server runs all villages in each valley synchronously
- Clients run slightly ahead of server
- Clients able to rewind to last known good state and re-simulate the game if things change
- Server validates & applies actions against the village & the valley at the same time

IMPROVEMENTS

Pros

- No error resolution/rollback required. Only validation & execution needed to define action.
- Very robust
- Server state very well defined at any point in time
- Error resolution (misprediction) can be handled universally in most cases

IMPROVEMENTS

Cons

- CPU & memory requirements on client & server
- Some client/UI scenarios still need careful handling
  - State may change dramatically based on misprediction
- Avoid direct callbacks from logic code to client code
  - eg sounds or effects may be re-triggered many times during resim

IMPROVEMENTS

- Great success!
- Develop complicated collaboration-based game systems much more rapidly
- Weeks instead of months
- Single programmer per feature

ITERATION TIME IS KEY!

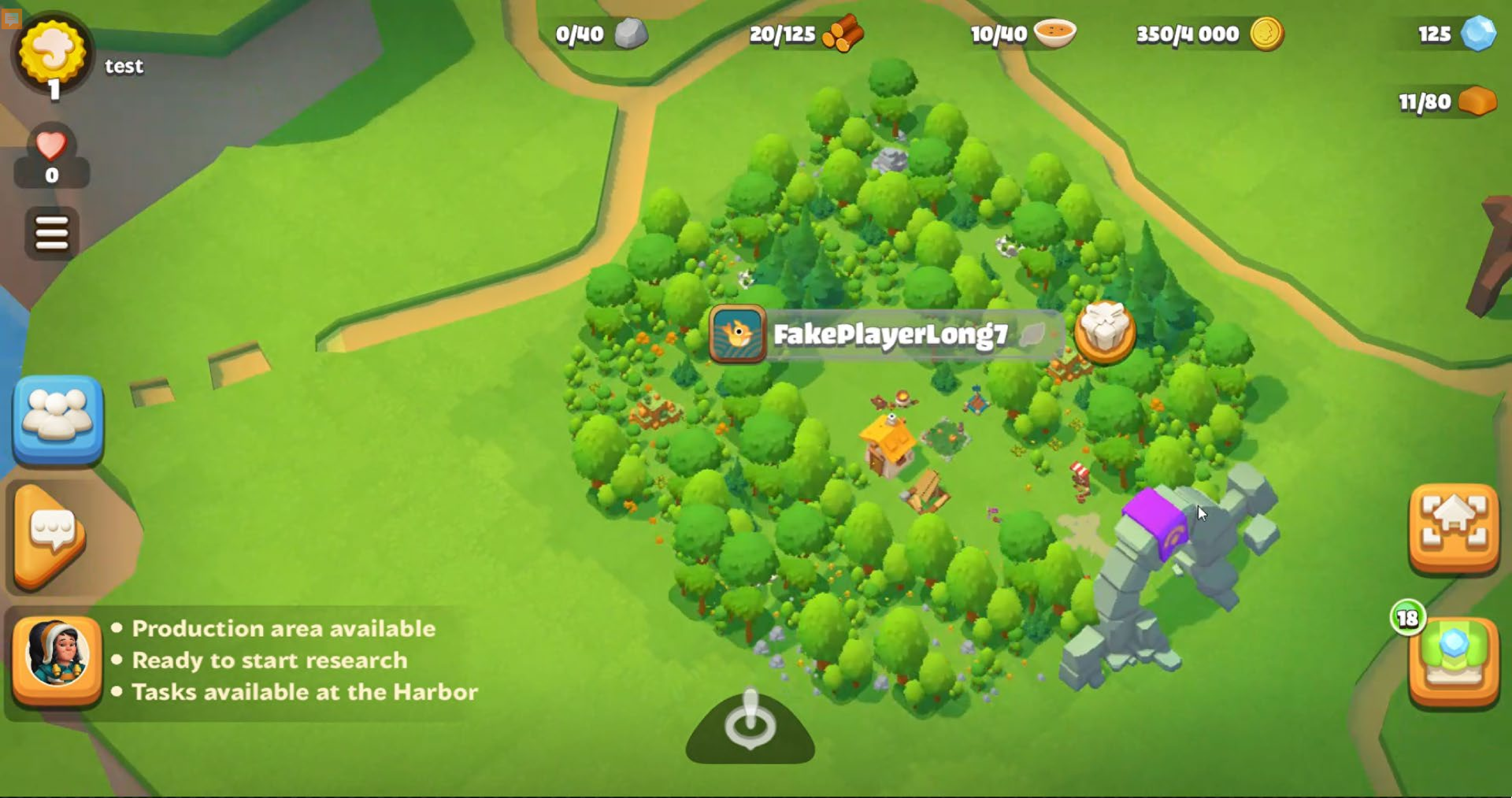
PERFORMANCE

PROBLEM

- ~10 villages
- 100s/1000s of objects per village
- Complex flow-based logic
- Performance definitely an issue!

RENDERING

- Render villages to texture imposters
- Stagger imposter updates
- Near-seamlessly transition to/from impostor



- Production area available
- Ready to start research
- Tasks available at the Harbor

LOGIC

- Relatively slow gameplay
- Reduced logic tick rate to 5Hz
- Client interpolates logical state as necessary

LOGIC

Deterministic variable-length logic update():

- Allow objects/systems to declare how many ticks they can update()
- Many objects can skip updates for multiple seconds at a time
- Client steps tick-by-tick, server can fast forward

LOGIC

- Reduced server load by 80-90%!
- Throttle/stagger logic updates for off-screen villages on client

DETERMINISM

Guaranteeing determinism

- Debug clients run background verification
  - Background thread running fast forward mode to compare
  - Save traces when errors detected
- Server also saving traces when out of sync situations detected

TESTING

With guaranteed determinism & decoupled logic execution:

- Stored gameplay "replay" traces for bug testing
- Regression testing for changes
    - Can simulate all accounts before/after changes and assert same results

TESTING

- Many systems can be validated in background threads while playing

TESTING

- Built AI logic that can play the game
- Connect to load test servers and play
- Run thousands of bots
- Gather balance & stability data
- Isolate rare bugs

RESULT

- Achieved our dreams!
- Game is now in soft launch

TAKEAWAYS

Dream big - but go in with eyes open!

Game design and technology affect each other deeply, and can yield exciting results.

THANKS FOR COMING!

Give feedback!

We're hiring!
https://supercell.com/careers