

# The Prototyping Process

Prove your ideas before committing to it





**Adi Zhavo**

Game Developer at Gram Games

# Have the right mindset

- Don't think of your prototype as a game but as a product
- Don't think as a gamer, it will help you stick to the plans
- You might not investing money, but you're investing your time and efforts
- Having a gamer's mentality will make you lose time in details that are not needed
- Aim for a deliverable that will be published on the store

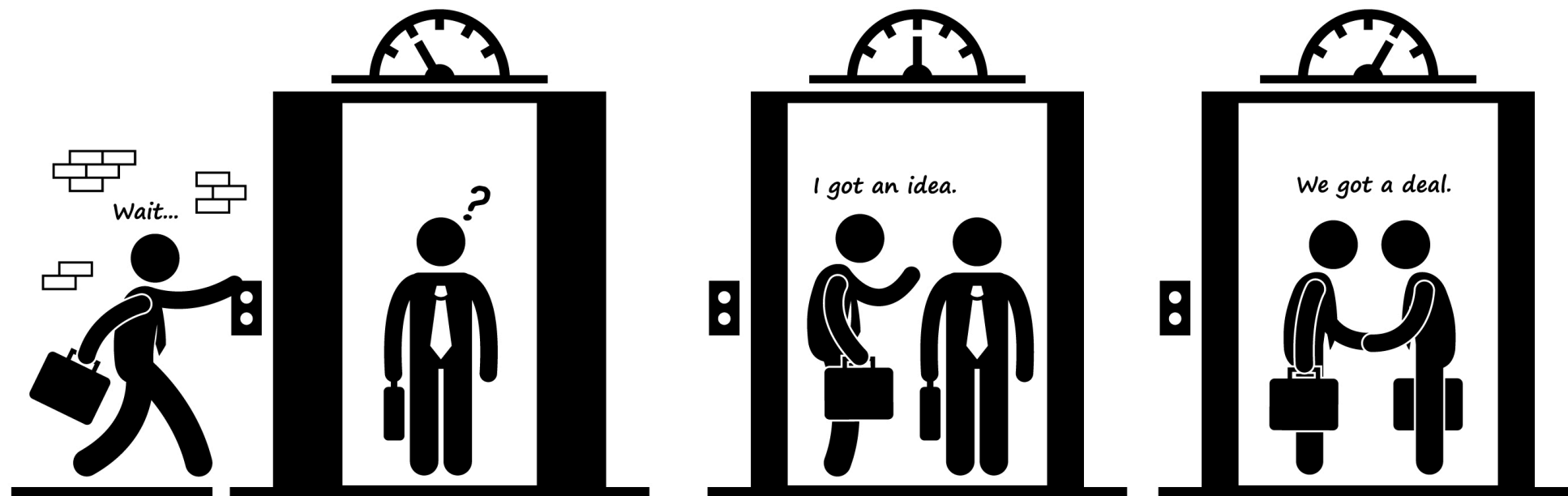
# Have a clear vision

- Core mechanic
- Core interactions
- Theme
- Audience
- Genre - study what other similar games do
- NEVER CLONE
- How will the game scale

# Define the scope

- 1 week time
- At most one meta/economy loop
- Don't focus on the content, you're trying to prove the core
- Think of a clear on-boarding (tutorial)
- Target early retention
  - $D1 > 40\%$ ... Good to have a  $D7 > 10\%$  but its not required
- Always plan for the unknown unknowns

# Work on your pitch



- Describe the core mechanic and the main loop in a sentence
- Don't describe the content
- Don't describe how the game scales

# Example

- It's a game about heroes that have different skins and different abilities with different cool-downs and you can evolve them by crafting weapons.



- It's a game about matching the same items to evolve them so it clears the room for gold. Gold is used to buy more items.



# Plan with your team

- 3-4 people in a team
- Plan the code and art progress for the week and have deliverables each day
- Be ready to wear different hats
- Be flexible in changing things that are not major, never re-pivot until the end of the week
- Communication is key!!!
- Playtest as a team at the end of each day



# Coding?

- Don't build a framework unless you really need one
- Use GIT or any VCS
- Turn your mind off and hack, you need speed not quality
- Get the core done in half a day
- Use what you know, don't explore
- Use a simple engine and a simple programming language
- Be a quick coder, not an engineer

# Art and visual effects (VFX)

- Work in 'phases'
- Do something quick and understandable for the user
- Re-use as much as you can
- Use only 1 visual effect
- Better integrate the assets later but not too late

**Audio... do you need it ?**

**No, thank you...**

# Publishing and Analytics

- Focus on Android, publishing and updates are much faster
- Use 3rd party tracking SDKs
- Implement tracking events
  - First Login of the day
  - Start and end of sessions
  - Sessions length
- What to extract ?
  - Daily retention %
  - Average session length
  - Average amount of sessions per day

# Marketing and User Acquisition (UA)

- Have good store assets and a gameplay video
- Use 1-2 Ad networks for the acquisition
- Get around 1000-2000 users
- Acquire through the whole week
- Acquire from tier 1-2 countries

# Community and Support

- Ignore all emails from users
- Ignore store ratings
- Ignore store reviews unless its about a game breaking bug

# Review Key Performance Indicators (KPIs)

- Remove the app from the store
  - Risk of cloning
  - Risk of app being published on other websites
  - You don't want people to complain too much about the game
- Check retention
  - < 40% ?
  - > 40 % ?
  - Check your sessions
  - Delete the codebase and start fresh
  - Maybe try one more iteration
  - Use the numbers for investment or publisher support
  - Kill it
  - Use the numbers to grow your team

**Does this process  
work for indie companies?**



**Does this work for bigger  
games ?**

# Thank you!

# Questions ?



[gram.gs](https://gram.gs)



[linkedin.com/in/adizhavo](https://linkedin.com/in/adizhavo)



[github.com/adizhavo/gg\\_prototyping\\_process.pdf](https://github.com/adizhavo/gg_prototyping_process.pdf)



GRAM GAMES