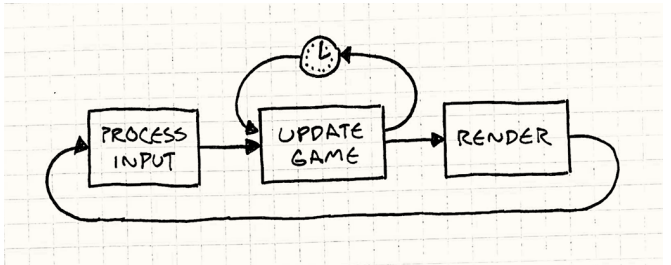
USC Viterbi
School of Engineering
Department of Computer Science
GamePipe Laboratory

Game Loops vs. Core Game Loops

(Say what now?)

Game Loop (Engineering):



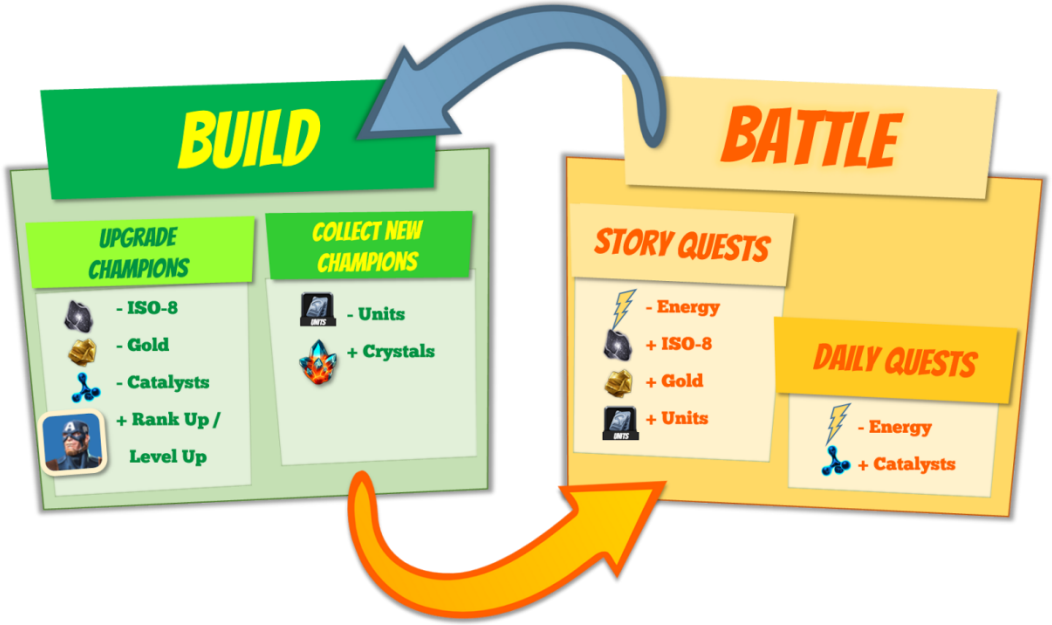
- All the code that makes the game interactive and dynamic goes in the game loop, but is separated into different pieces**:**
  - **Initialize**
    - Any necessary game setup and prepare the environment for the update and draw phases.
      - Main Entities (assets), Prepare the Menu, Detect default hardware capabilities
  - **Update**
    - All objects to be drawn are made ready:
      - Physics code, coordinate updates, health points changes, char upgrades, damage dealt
  - **Draw**
    - The ready levels, layers, chars, HUD are drawn

Core Loop (Design):

- A core series of actions that need to be learned and executed to become skilled**:**
  - **Basic mechanics sewn together**
    - But instead of merely defining an action in game, it needs to contribute to other aspects of the game. It affects later choices or strategies.
  - **Existing mechanics – don't invent new ones**
    - Maybe you buy new things? <- No.
    - Maybe you get money for some store? <- No.
    - Maybe you get could get better and bigger? <- No.
  - Not because these are bad, but the word 'Maybe' can give the impression you will stack mechanics.

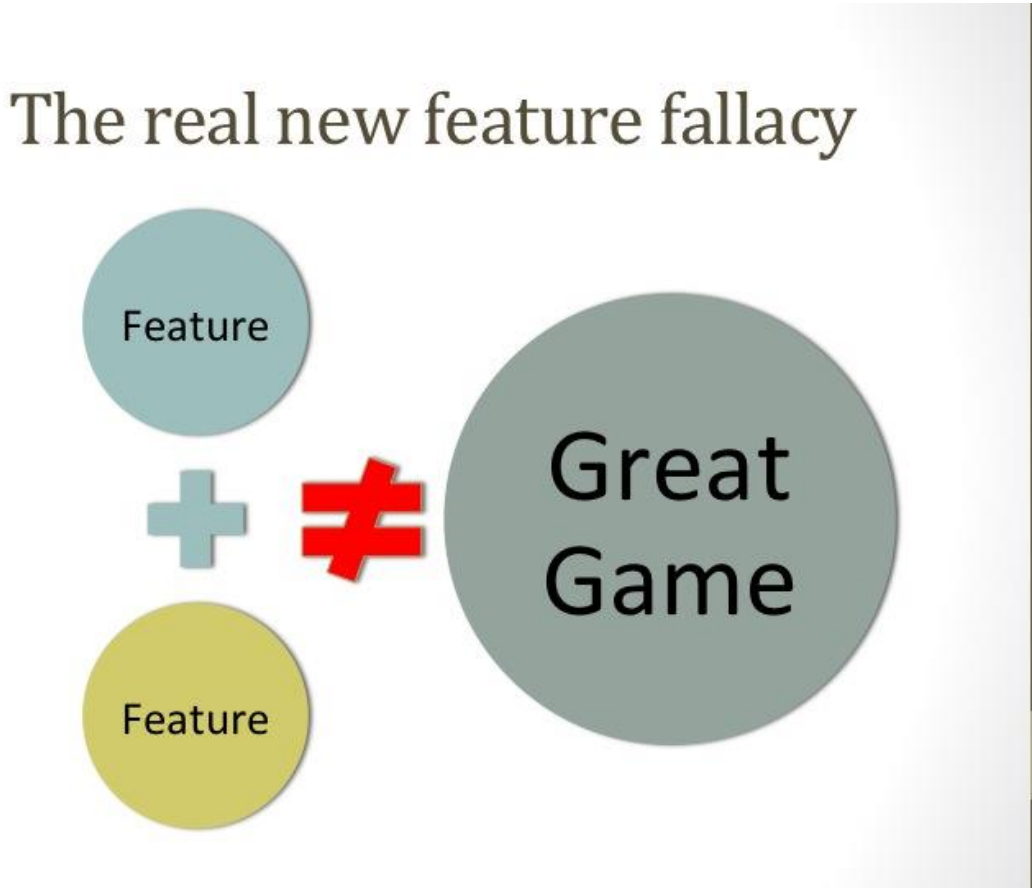
Core Loop (Why):

- A core series of actions that ties back into the game engages the player to strategize better:
  - **Not the same gameplay per player**
    - If you have a loop that allows you to change the gameplay later you have more options of gameplay.
    - Longer core game loops, longer the player plays.
  - **Circular**
    - **How you play determines not just rank or other reward but future Gameplay.**
  - **No blocking points**
    - So ‘Top Score’ as end of game means nothing
      - it must go somewhere that has relevance in game
      - **‘Store’ is ok, but use all rewards in-game even better**



Good Core Loop – use what you have

- Character runs on the board and jumps, gets stuff – what is my loop?
  - **Character get what throughout game?**
    - Gets coins.
      - What can they do with them?
    - Maybe buy stuff later?
      - No. Can you use them in game?
    - Not yet.
      - Use them in the game – change to mushrooms
    - To change what?
      - Well, what does he do?
    - Runs.
      - On what?
    - Platforms.
      - Then affect both – him and platforms



Find your Core Loop – Challenge the static

- A good way to determine what is rewarding in game is choose something static in your game
  - **What is a ‘constant’ in your game?**
    - Levels – then make them move
    - Character size/speed – change it according to powerup
    - Always up/down, or left to right – change gravity, or character direction
    - Always runs from enemy – change the hunter to hunted
    - Are you using your environment?
      - Player can change it if they choose
    - Are you using your rewards?
      - Player can use them immediately in game

Game Loop allows evolution of Core Loop

- Priority is to make engineering environment (game loop) where design (core loop) can continue evolving and being tweaked.
- Try to give player's acquisitions come back directly into gameplay
- Try to make any advantage also have responsibility (downside)
  - If I am bigger, I am slower
  - If I am faster, I am more vulnerable
  - If I use big weapon, it has cooldown and I have no weapon
  - If I choose one path, then I cannot do another

So to test this Game/Core Loop environment:

- Everyone needs to see if the game works on small screen.
- You need to create commented code where your team can see the core loop.
- Write down ideas and study other games how they allow decisions for player that affects the gameplay.
  - Not just bigger/better/more coins, but changes gameplay itself a bit.

Next week: Port your game!

- Everyone needs to see if the game works on small screen.
- This will force your team to have a setup where others can read the commented code, so others know what to change and sketch with.

  - **Technical: Get it ported (Game Loop)**
  - **Design: Design looping gameplay (Core Loop)**