

# Scheduling

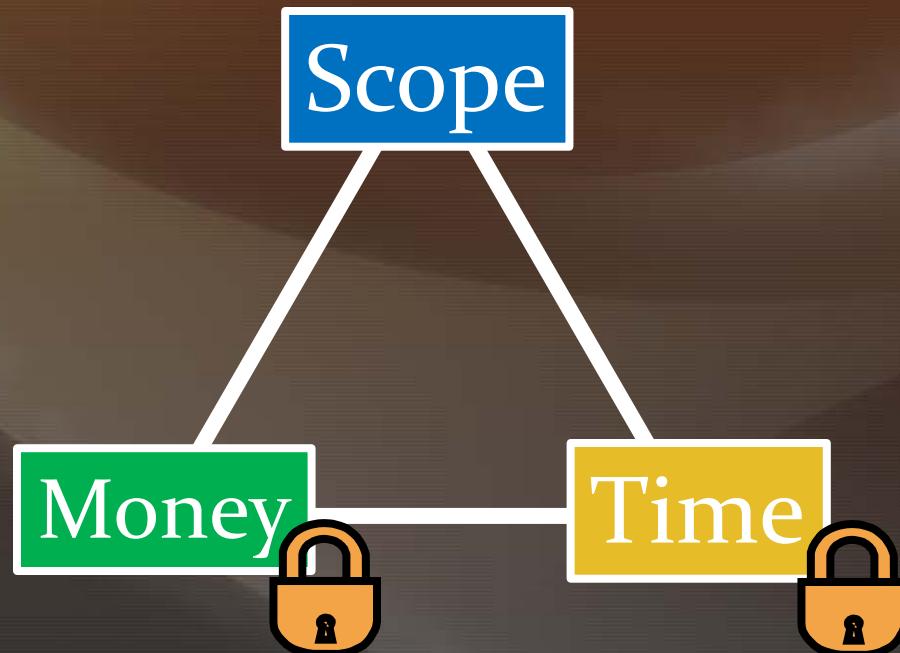


FULL  
STRUCTURE  
of Game Production

# Scheduling

## The Iron Triangle

- ◆ The three things that limit production
  - All fixed can not be fixed (Pick 2)



# Scheduling

## What are the steps to scheduling?

- ◆ First:
  - ◆ Recognize how much time and resources are available

7	8	9	10	11	12	Sprint Planning	Work On Tasks
14	15	16	17	18	19	20	
Work On Tasks							
21	22	23	24	25	26	27	
Work On Tasks							
					Code Freeze	Sprint Review	

- ◆ Second:
  - ◆ Identify what takes priority
  - ◆ Select tasks until that time is filled
  - ◆ Organize tasks in a timeline

## Man-Hours

- ◆ Man-hour: A unit of production equal to what can get done by one person over an hour
- ◆ Consider... travel time to Full Sail
  - How must time is set aside to travel to FS every day?
  - How many days a week?
  - What are the combined man hours of the class?

# Scheduling

## What are the steps to scheduling?

- ◆ First:
  - Recognize how much time is available
  - How many hours will be worked per week?
    - ~40
  - How many developers?
    - ~4 (average)
  - How many weeks?
    - 2 weeks for average sprint
  - How many man hours for a sprint?
    - $40 \text{ hours} * 4 \text{ people} * 2 \text{ weeks} = 320 \text{ man hour sprint}$
- ◆ Second:
  - Identify what takes priority
  - Select tasks until that time is filled
  - Organize tasks in a timeline

# Problems with Man-Hours

- ◆ Man-hours don't take into account time for breaks, interactions, unforeseen complication
  - ♦ It is pure unadulterated work
- ◆ There are other things we do have to do
  - ♦ Stand up meetings
  - ♦ Meetings with management
  - ♦ Peer programming with each other
  - ♦ ...

# Hours

# Problems with Man-Hours

- ◆ Man-hours don't take into account "too many cooks in the kitchen"
  - ◆ Some tasks can't be evenly divided and putting additional workers on it has diminishing returns
  - ◆ Some tasks can't be divided at all
- ◆ How many months does it take 1 woman to make a baby?
  - ◆ ~9 month
- ◆ How many months does it take 9 women to make a baby?
  - ◆ ~9 month
  - ◆ It doesn't take less time by throwing more people at it

# Phases of Production



FULL  
STRUCTURE  
OF GAME PRODUCTION

# Phases of Production

- ◆ Greenlight
- ◆ Pre-Production
- ◆ Open Production
  - First Use/Playable
  - Alpha
  - Beta
  - Gold
- ◆ Post Release

# Green light

- ◆ At this point a concept and funding has been structured
- ◆ Core documentation is written
- ◆ Conceptual artwork is created
- ◆ Paper and electronic prototypes are created
- ◆ Play testing of prototypes completed to prove the idea works

## Green light

- ◆ During the green light phase a studio supports itself.
  - ◆ This is when the studios *burn rate* is most important
- ◆ Burn Rate
  - ◆ The rate at which an the company spends money, in excess of income

# Pre-Production

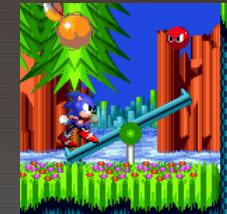
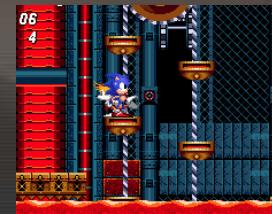
- ◆ Deeper Design Concepts
- ◆ Evaluate Human Resources (Hiring any open positions)
- ◆ Researching
- ◆ Code architecture is created
  
- ◆ Takeaway: A clear definition of the product about to be developed

# Open Production – First Use/Playable

- ◆ Defined: A completed single level that displays most Global aspects of product in a local environment.
  - Note the difference between Global/local features.
    - Global: Things necessary for every portion of the product



- Local: Things only necessary for specific portions of the product.



# Open Production – First Use/Playable

- ◆ The product must be able to sell itself
  - This is when products most often gets canceled
- ◆ We have the 2 sprints during the AHI month to get to first playable

# Open Production - Alpha

- ◆ Defined: Completed the construction of all features.
- ◆ Active new development stops.

# Open Production - Beta

- ◆ Defined: Finalizing content for the product.
- ◆ Could contain the QA process.

# Open Production - Gold!

- ◆ Manufacturing and shipping
- ◆ Passed all TCR Certification

# Post Release

- ◆ Can be the least fun part of the process.
- ◆ Reviews come out
  - ◆ Internet hate machine kicks in overdrive
- ◆ Bug fixes
- ◆ Continuing support for the community, regardless of its size

# Sprint goals



FULL  
S  
tory  
line  
Structure of Game Production

# Sprint 1 (Core)

## ◆ Goals

- The core game feature/mechanic (hook) is fully realized and useable
- All user input methods working
- The game has at least one complete playable experience
- Menu and UI systems created

# Sprint 1 (Core)

## Month 1 (PP2)

Week4	Lecture 7		Lecture 8		
			Sprint 1 (Core) Plan	Sprint 1	↗

## Month 2 (AHI)

Week1	Lecture 1		Lecture 2		
	↗	Sprint 1	↗		
Week2	Lecture 3		Lecture 4		
	↗	Sprint 1	↗		
Week3	Lecture 5		Lecture 6		
	Sprint 1 (Core) Review				
Week4	Lecture 7		Lecture 8		

## Month 3 (PP3)

Week1	Lecture 1		Lecture 2		
Week2	Lecture 3		Lecture 4		
Week3	Lecture 5		Lecture 6		
Week4	Lecture 7		Lecture 8		

- ◆ Starts the last day of PP2
- ◆ Very light hourly commitment
  - ◆ 2 dedicated work days
  - ◆ ~16 hours a person
- ◆ Sprint submission night before last day of week 2
  - ◆ Push to repo
- ◆ Sprint review must be scheduled

# Sprint 2 (First Use)

## ◆ Goals

- The game world is populated with at least one example of each core game object
- Core use / gameplay revisions
- Menu and UI systems finalized
- The game is fun to play
- AHI directives on how to improve the experience have been completed

# Sprint 2 (First Use)

## Month 1 (PP2)

Week4

Lecture 7

Lecture 8

Sprint 1  
(Core)  
Plan

Sprint 1

## Month 2 (AHI)

Week1

Lecture 1

Lecture 2

Week2

Lecture 3

Lecture 4

Week3

Lecture 5

Lecture 6

Week4

Sprint 1  
(Core)  
Review

Sprint 2  
(First)  
Plan

Sprint 2

## Month 3 (PP3)

Week1

Lecture 1

Lecture 2

Week2

Sprint 2  
(First)  
Review

Lecture 3

Lecture 4

Week3

Lecture 5

Lecture 6

Week4

Lecture 7

Lecture 8

- ◆ Starts halfway point of AHI

- ◆ Very light hourly commitment
  - ◆ 2 dedicated work days
  - ◆ ~16 hours a person

- ◆ Sprint submission night before last day of week 4
  - ◆ Push to repo

- ◆ Sprint review day 1 of PP3

# Sprint 3 (Alpha)

- ◆ Goals
  - ♦ The game world is populated with at least one example of each game object
    - ♦ (all game object types)
  - ♦ Overall game progress can be shown
    - ♦ (multiple levels, multiple objectives...)
  - ♦ The game contains its win/loss conditions
  - ♦ Feature complete
    - ♦ Enough asset creation complete to prove the use of features
  - ♦ Any feature not completed by the end of this sprint must be cut from the product

# Sprint 3 (Alpha)

## Month 1 (PP2)

Week4	Lecture 7		Lecture 8		
			Sprint 1 (Core) Plan	Sprint 1	➤

## Month 2 (AHI)

Week1	Lecture 1		Lecture 2		
			Sprint 1		➤
Week2	Lecture 3		Lecture 4		
			Sprint 1		➤
Week3	Lecture 5		Lecture 6		
	Sprint 1 (Core) Review	Sprint 2 (First) Plan	Sprint 2		➤
Week4	Lecture 7		Lecture 8		
			Sprint 2		➤

## Month 3 (PP3)

Week1	Lecture 1		Lecture 2		
	Sprint 2 (First) Review	Sprint 3 (Alpha) Plan	Sprint 3		➤
Week2	Lecture 3		Lecture 4		
			Sprint 3 (Alpha) Review		➤
Week3	Lecture 5		Lecture 6		
Week4	Lecture 7		Lecture 8		

- ◆ Starts day 1 of PP3
- ◆ Average hourly commitment
  - 6 dedicated work days
  - ~48 hours a person
- ◆ Sprint submission night before day 4 PP3
  - Pushed to repo
- ◆ Sprint review day 4 of PP3

# Sprint 4 (Beta)

## ◆ Goals

- ◆ All assets are in their final, non-placeholder form
  - ◆ (sprites, particles, animations, sounds, music...)
- ◆ Final game progress
  - ◆ (all levels, challenges, modes...)
- ◆ All target platforms functioning
  - ◆ (PC, web player, tablet)
- ◆ Various finalizing
  - ◆ Final Credits
  - ◆ Installer created

# Sprint 4 (Beta)

## Month 1 (PP2)

Week4	Lecture 7		Lecture 8		
			Sprint 1 (Core) Plan	Sprint 1	➤

## Month 2 (AHI)

Week1	Lecture 1		Lecture 2		
			Sprint 1		➤
Week2	Lecture 3		Lecture 4		
			Sprint 1		➤
Week3	Lecture 5		Lecture 6		
	Sprint 1 (Core) Review	Sprint 2 (First) Plan	Sprint 2		➤
Week4	Lecture 7		Lecture 8		
			Sprint 2		➤

## Month 3 (PP3)

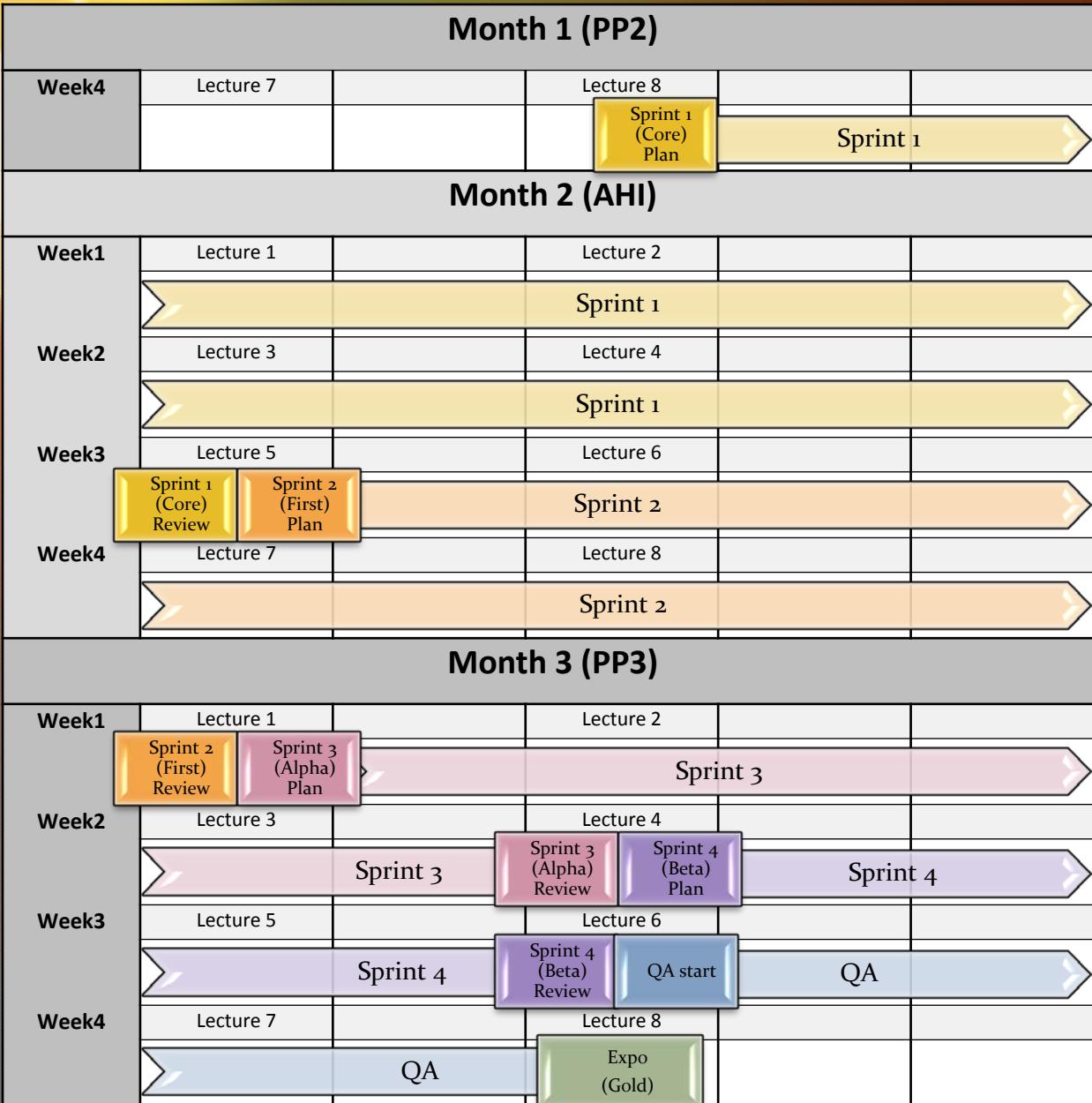
Week1	Lecture 1		Lecture 2		
	Sprint 2 (First) Review	Sprint 3 (Alpha) Plan	Sprint 3		➤
Week2	Lecture 3		Lecture 4		
			Sprint 3 (Alpha) Review	Sprint 4 (Beta) Plan	Sprint 4
Week3	Lecture 5		Lecture 6		
			Sprint 4 (Beta) Review		
Week4	Lecture 7		Lecture 8		

- ◆ Starts day 4 of PP3
- ◆ Average hourly commitment
  - ◆ 4 dedicated work days
  - ◆ ~32 hours a person
- ◆ Sprint submission night before day 6 PP3
  - ◆ Push to repo
- ◆ Sprint review day 6 of PP3

## QA and finalizing

- ◆ Goals
  - Bug testing rounds completed
  - All bugs closed out (fixed preferably)
  - Project post mortems completed
  - Expo presentation

# QA and finalizing



- ◆ Starts day 6 of PP3
- ◆ Average hourly commitment
  - ◆ 4 dedicated work days
  - ◆ 2 testing days
  - ◆ 2 fixing days
- ◆ Expo presentation day 8 of PP3

# Hours

# Man Hours during this project



- Prototyping
  - Pre Production
  - No hours dedicated to production
- Core
    - 16 hrs. per person
  - First Use/Playable
    - 16 hrs. per person
- Alpha
    - 48 hrs. per person
  - Beta
    - 32 hrs. per person
  - Finalizing
    - No hours dedicated to production

~100 hrs. per person

3 person team = ~300 man hours

4 person team = ~400 man hours

5 person team = ~500 man hours

# Sprint planning meetings



FULL  
STRUCTURE  
IN A DAY  
Structure of Game Production

# Sprint planning

## Goal

- ◆ Detailed scheduling in Scrum is handled in a sprint by sprint basis
- ◆ At the beginning of each sprint the entire team will meet to:
  - ◆ Determine an overall sprint goal
  - ◆ Select stories from the product back log to create a sprint backlog
  - ◆ Evaluating difficulty/hours/complexity to the stories
  - ◆ Compared against hours available
  - ◆ Distributing the work load

# PLANNING POKER



FULL  
Stack  
UNIVERSITY  
Structure of Game Production

## Step 1: Bidding

- ◆ Userstory and test cases is read out to the team
  - Answers questions if there are any
  - Modify test cases where needed
    - (Client is involved in this for externally produced projects)
- ◆ Each team member
  - Evaluates how difficult they believe the story is to completing
  - Without bias from other members
  - Pick which of the possible bids best represents how difficult they evaluate the task to be

## Step 1: Bidding

- ◆ Bid Value : Average work time
- ◆ 0 hr.
- ◆  $\frac{1}{2}$  hr.
- ◆ 1 hr.
- ◆ 2 hrs.
- ◆ 3 hrs.
- ◆ 5 hrs.
- ◆ 8 hrs.
- ◆ 13 hr (1 day and a half)
- These number pattern reflect one of the faults in making estimates
  - The larger the estimate the more room for error
- Each value should be thought as a range from the bid below it up
  - Bid of 5 = anything above 3 up to 5
- ◆ 20 hrs. (half a week)
- ◆ 40 hrs. (1 week)
- ◆ 100 hrs. (2 weeks)
- ◆ Unknown
- ◆ Infinite

## Step 1: Bidding

- ◆ Bid Value : Average work time
  - ◆ 0 hr.
  - ◆  $\frac{1}{2}$  hr.
  - ◆ 1 hr.
  - ◆ 2 hrs.
  - ◆ 3 hrs.
  - ◆ 5 hrs.
  - ◆ 8 hrs.
  - ◆ 13 hr (1 day and a half)
  - ◆ 20 hrs. (half a week)
  - ◆ 40 hrs. (1 week)
  - ◆ 100 hrs. (2 weeks)
  - ◆ Unknown
  - ◆ Infinite
- 
- Stories in this project really shouldn't be more than 13 hrs.
    - Because of how personal grades are put together

## Step 1: Bidding

- ◆ Bid Value : Average work time
- ◆ 0 hr.
- ◆  $\frac{1}{2}$  hr.
- ◆ 1 hr.
- ◆ 2 hrs.
- ◆ 3 hrs.
- ◆ 5 hrs.
- ◆ 8 hrs.
- ◆ 13 hr (1 day and a half)
- Special bids:
  - 0: There is no or an inconsequential amount of work to be done to have this completed.

## Step 1: Bidding

- Bid Value : Average work time
  - 0 hr.
  - $\frac{1}{2}$  hr.
  - 1 hr.
  - 2 hrs.
  - 3 hrs.
  - 5 hrs.
  - 8 hrs.
  - 13 hr (1 day and a half)
- Special bids:
  - **Unknown:** When there is not enough information to make a bid.
  - If any team member puts down an "?" bidding stops until receiving further explanation.

## Step 1: Bidding

- ◆ Bid Value : Average work time
  - ◆ 0 hr.
  - ◆  $\frac{1}{2}$  hr.
  - ◆ 1 hr.
  - ◆ 2 hrs.
  - ◆ 3 hrs.
  - ◆ 5 hrs.
  - ◆ 8 hrs.
  - ◆ 13 hr (1 day and a half)
- Special bids:
  - **Infinite**: The user story is completely understood, but will never be able to be completed during a sprint.

## Step 1: Bidding

- ◆ Bid Value : Complexity points
  - ◆ 0
  - ◆  $\frac{1}{2}$
  - ◆ 1
  - ◆ 2
  - ◆ 3
  - ◆ 5
  - ◆ 8
  - ◆ 13
  - ◆ 20
  - ◆ 40
  - ◆ 100
  - ◆ Unknown
  - ◆ Infinite
- 
- With experience it simplifies
    - Complexity values compare it to previous work.

# Faulty estimates

## Step 1: Bidding

- ◆ This first step (bidding) must be done in a vacuum
  - ◆ Without influence of other team members
  - ◆ Allows everyone to think about the story
  - ◆ Gives people a place to defend and forces them to make their estimate for a reason
  - ◆ Allows everyone to have a voice in the estimation

## Step 2: Negotiation

- ◆ Each team member reveals what bid they decided upon on the previous step at the same time
- ◆ If bids differ the team must discuss why and come to an agreement on the task's value

## Step 3: Allocation

- ◆ After every user story has agreed upon values, user stories must have owners committed to them.
  
- ◆ The story's owner will be the person
  - ◆ Best equipped to tackle the story
  - ◆ Responsible for completing all task related to the story before the end of the sprint
  - ◆ Owner can not change without management approval
    - ◆ Owners really shouldn't change after they are set.

## Step 3: Allocation

- ◆ Once all of the stories have owners tally up the total value of each person
- ◆ If the distribution of tasks is not even, rebalance the tasks' owners until it is.
  - You may have to give up what you would like to work on in favor for what the team needs
- ◆ Make sure the work load matches up with the sprint length
  - If there aren't enough hours to fill the schedule we take more stories from the product backlog
  - If we are over hours we return things to the product backlog that are unnecessary for the sprint goal

# Estimating Mistakes



FULL  
STRUCTURE  
of Game Production  
Structure of Game Production

# Faulty estimates

## Estimating is hard

- ◆ One of the toughest components of production in general
  - ◆ You get better with practice, but you will never be perfect

“I don’t know how long it will take until it is already done”

versus

“If it is exactly like some previous work, there would be absolutely no point doing it”

# Faulty estimates

## D.I.U.I.S

- ◆ Tunnel vision
  - ◆ Not thinking about all the work involved is the biggest cause of bad estimates
- ◆ We are doing more than just coding
  - ◆ Design
  - ◆ Implement
  - ◆ Use test
  - ◆ Integration
  - ◆ System test

## Underestimating

- ◆ Resist the urge to underestimate (underbidding)
- ◆ How it is a problem:
  - Not having enough time to complete the work
    - Having to cut features/stories that have already been committed to
    - Crunch/overwork (getting into this a little later)
  - Missing deadlines

# Overestimating

- ◆ Resist the urge to pad estimates  
(sandbagging, overbidding)
- ◆ Student Syndrome (Planned procrastination)
  - ◆ Only starting to fully apply oneself to a task at the last possible moment needed to complete the work before a deadline.
- ◆ Parkinson's law
  - ◆ “Work expands so as to fill the time available for its completion”
- ◆ How is it a problem:
  - ◆ Lower quality, less feature rich products, and less sales
  - ◆ If you know you have more time than you need, your workflow suffers.

# Planning poker

## Class Activity

- ◆ Planning poker examples

# Avoid burn out



FULL  
STRUCTURE  
of Game production  
Structure of Game production

# Avoid burnout

# Quality of life

- Quality of life discussions are a thing in the industry for a reason
- There have been quite a few terrible examples in the industry's history
  - EA: The Human Story (ea\_spouse) (2004)
    - "The current mandatory hours are 9am to 10pm -- seven days a week -- with the occasional Saturday evening off for good behavior (at 6:30pm). This averages out to an eighty-five hour work week. Complaints that these once more extended hours combined with the team's existing fatigue would result in a greater number of mistakes made and an even greater amount of wasted energy were ignored."

EA: The Human Story - ea\_spouse [entries|archive|friends|userinfo]  
ea\_spouse  
[ userinfo | livejournal userinfo ]  
[ archive | journal archive ]

EA: The Human Story [Nov. 10th, 2004 | 12:01 am]  
  
My significant other works for Electronic Arts, and I'm what you might call a disgruntled spouse.  
  
EA's bright and shiny new corporate trademark is "Challenge Everything." Where this applies is not exactly clear. Churning out one licensed football game after another doesn't sound like challenging much of anything to me; it sounds like a money farm. To any EA executive that happens to read this, I have a good challenge for you: how about safe and sane labor practices for the people on whose backs you walk for your millions?  
  
I am retaining some anonymity here because I have no illusions about what the consequences would be for my family if I was explicit. However, I also feel no impetus to shy away from sharing our story, because I know that it is too common to stick out among those of the thousands of engineers, artists, and designers that EA employs.

# Avoid burnout

# Quality of life

- Quality of life discussions are a thing in the industry for a reason
- There have been quite a few terrible examples in the industry's history
  - Wives of Rockstar San Diego employees have collected themselves (RockstarSpouse) (2010)
    - "The reigns whip again, and it becomes mandatory to work close to twelve hours a day including Saturdays, regardless if an employee has finished all his duties prior."

The screenshot shows a Gamasutra blog page. At the top, there is a navigation bar with categories: ALL, CONSOLE/PC, SMARTPHONE/TABLET, INDEPENDENT, VR/AR, SOCIAL/ONLINE, and a logo for GAME DEVELOPER ON GAMASUTRA. On the left side, there is a sidebar with a Member Login form (Email and Password fields, Login button, Forgot Password? and Sign Up links), social sharing icons (Facebook, Twitter, RSS, Email), and category links for PROGRAMMING, ART, AUDIO, DESIGN, PRODUCTION, and BIZ/MARKETING. The main content area features a blog post titled "Wives of Rockstar San Diego employees have collected themselves" by Rockstar Spouse on 01/07/10 at 08:33:00 am. The post has 152 comments and sharing options for Twitter, Facebook, Google+, and RSS. A note at the bottom states: "The following blog post, unless otherwise noted, was written by a member of Gamasutra's community. The thoughts and opinions expressed are those of the writer and not Gamasutra or its parent company." Below the note, there is a quote from the post: "To whomever it may concern," and a summary: "In response to the unfortunate circumstances, some wives of Rockstar San Diego employees have collected themselves to assert their concerns and announce a necessary rejoinder, in the form of

# Avoid burnout

# Quality of life

- Quality of life discussions are a thing in the industry for a reason
- There have been quite a few terrible examples in the industry's history
  - Why Did L.A. Noire Take Seven Years to Make? (2011)
    - "It's abhorrent that these young kids are being thrown into a 24/7 corpse grinder with perpetual crunch and weekend overtime,"



L.A. NOIRE / 24 JUN 2011

## WHY DID L.A. NOIRE TAKE SEVEN YEARS TO MAKE?

 **Share.** Examining the troubled development of Team Bondi's opus.

BY ANDREW MCMILLEN → Team Bondi's film noir-inspired detective thriller [L.A. Noire](#) was released last month to critical and commercial success. Set in a lavish recreation of 1947 Los Angeles, the game eschewed a familiar open-world design for case-by-case detective gameplay that revolved around examining crime scenes and interrogating suspects. Featuring a vast city, cases that adjusted depending on the player's actions and choices, and sophisticated motion capture technology that had never been used in a video game before, it was a mammoth project.

So mammoth, in fact, that it took over seven years to complete, with a publisher switch from Sony to Rockstar midway through. That's a lot of development time.

# Quality of life

- Quality of life discussions are a thing in the industry for a reason
- There have been quite a few terrible examples in the industry's history
  - Inside Konami: public shaming, tyrannical management and punitive reassignment (2015)
    - "employees who disappoint or are deemed to have underperformed are reassigned to what can only be described as punishment duties. Talented composers, programmers and other well trained specialists are plucked from their hard-earned positions and redeployed as workers in Konami's pachinko factories, as office security guards or as janitors on garbage duty at the firm's fitness clubs."

## Inside Konami: public shaming, tyrannical management and punitive reassignment

By Dan Pearson

[Facebook](#) Recommend 230

[Twitter](#) Tweet

[LinkedIn](#) Share 27

MON 03 AUG 2015 12:09PM GMT / 8:09AM EDT / 5:09AM PDT

PUBLISHING

**Nikkei paints a picture of publisher not just in disarray, but dissolution**

Konami, already a publisher mired in bad press, has today found itself at the centre of a maelstrom of ill-will thanks to a series of revelations about HR practices at the firm that wouldn't seem out of place in the military or a prison - practices that denigrate, demoralise and dehumanise the people working there.



Games Jobs

Search 1605 live games in

Job search

UK & Europe North America

# Avoid burnout

# Quality of life

- Quality of life discussions are a thing in the industry for a reason
- There have been quite a few terrible examples in the industry's history
  - Why 'crunch time' is still a problem in the video game industry
    - "Between 2014 and 2015 we saw 38 and 37 percent of developers, respectively, stating they don't get compensation for crunch time. That's ringing an alarm."
    - Response: Game developers must avoid the 'wage-slave' attitude: Alex St. John (2016)
      - "You can't "make fun" on a schedule, under budget, on time with a bunch of people who are all grumbling about what a miserable time they are having finishing a game together."

## Why 'crunch time' is still a problem in the video game industry

Game developers must avoid the 'wage-slave' attitude

Indie developer Rami Ismail responds to critic of work-life balance in game industry



Amilia St. John [Follow](#)

Front End Developer, fighting for women in tech. Twitter: @milistjohn Email: stjohnmili@gmail.com

Apr 21 · 10 min read

I am Alex St. John's Daughter, and He is Wrong About Women in Tech



# Quality of life

- Quality of life discussions are a thing in the industry for a reason
- There have been quite a few terrible examples in the industry's history
  - "We're definitely at the point where something's gotta give" (2016)
    - "When I was making sacrifices, did it affect my family? Yes, but it was primarily affecting me and I could make that choice. But when I look at other people... I mean, my health really declined, and I had to take care of myself, because it was, like, bad. And there were people who, y'know, collapsed, or had to go and check themselves in somewhere when one of these games were done. Or they got divorced. That's not okay, any of that. None of this is worth that."

## "We're definitely at the point where something's gotta give"

By Matthew Handrahan

 Recommend 3.2k

 Tweet

 Share

224

THU 06 OCT 2016 9:43AM GMT / 5:43AM EDT / 2:43AM PDT

 DEVELOPMENT

**Amy Hennig calls AAA development "an arms race that is unwinnable" where crunch is still a big problem**

With a CV that includes Crystal Dynamics, Naughty Dog and now Electronic Arts, Amy Hennig has worked in AAA console development for more than 20 years. But reaching that cutting-edge and staying there demands significant personal sacrifice, and Hennig believes the AAA industry has reached the point where,



Games Jobs

Search 1557 live games in

Job search

UK & Europe

North America

# Be in a state to work

- ◆ Focus
  - Stay in a state where you are able to work
  - Get sleep
  - Eat
  - Take breaks when you need them
  - Maintain humanity outside of development

# Avoid burn Out

# Avoid burn out

- ◆ Don't burn yourself out unnecessarily
- ◆ You are human, you need down time.

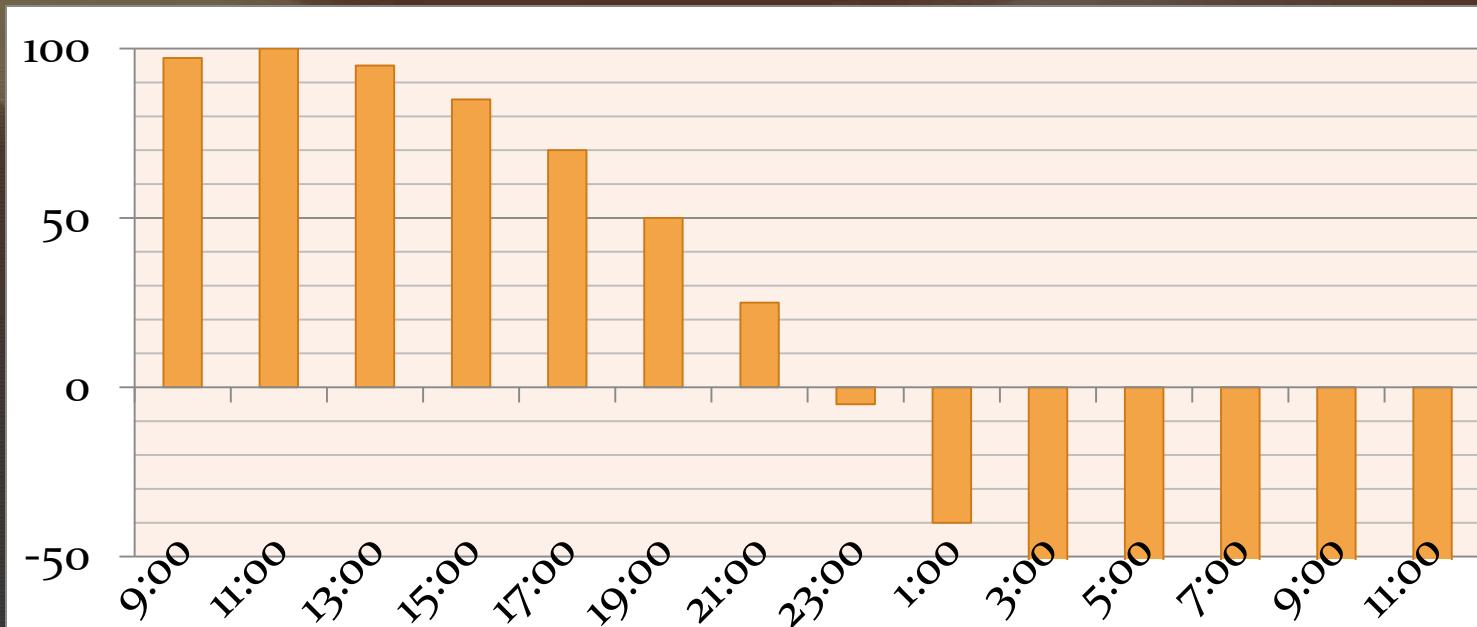


In order to achieve anything up here

Don't ignore the things down here

# Daily productivity

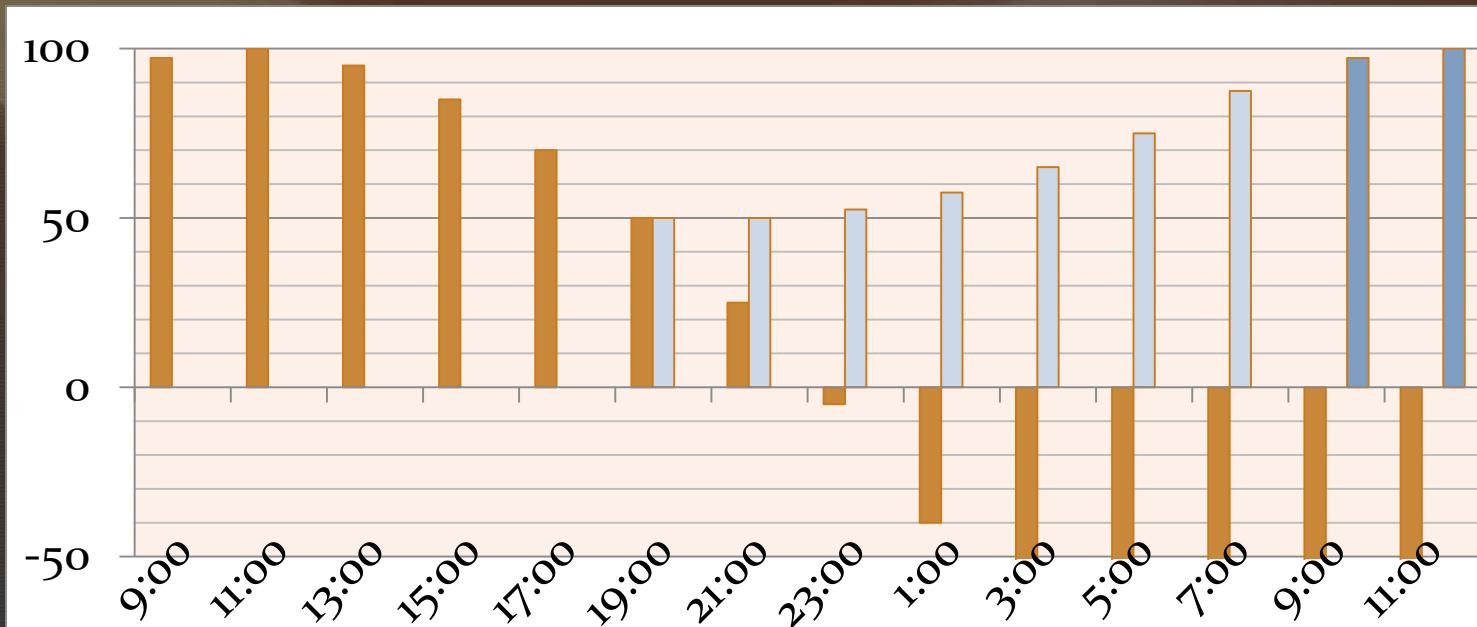
- ◆ Productivity varies over the course of the day
  - ◆ The greatest productivity occurs in the first 4 - 6 hours
  - ◆ After enough time working, productivity approaches zero
  - ◆ Eventually productivity becomes negative



# Avoid burnout

# Daily productivity

- ◆ You will be more productive if you allow yourself to take breaks
  - Let your mind rest periodically though the day
  - Stop working at the end of the day
  - Sleep!



# Long term productivity

- ◆ Crunch can lead to short term gains
- ◆ Continued crunch produces less than a standard 40 week in a couple months



# Being “on the clock”

- ◆ We will be pushing you to an aggressive schedule.
- ◆ This can be done without the need to crunch
- ◆ But that requires solid focused work when “on the clock”
  - You can get a ton done during lecture and lab periods if you focus

# Additional

# Additional Resources

- ◆ Evan Robinson: Why Crunch Modes Doesn't Work—Six Lessons
  - ◆ <http://www.igda.org/?page=crunchsixlessons>

## Why Crunch Modes Doesn't Work: Six Lessons

**There's a bottom-line reason most industries gave up crunch mode over 75 years ago:  
It's the single most expensive way there is to get the work done.**

by Evan Robinson

### Executive Summary

When used long-term, Crunch Mode slows development and creates more bugs when compared with 40-hour weeks.

More than a century of studies show that long-term useful worker output is maximized near a five-day, 40-hour workweek. Productivity drops immediately upon starting overtime and continues to drop until, at approximately eight 60-hour weeks, the total work done is the same as what would have been done in eight 40-hour weeks.

In the short term, working over 21 hours continuously is equivalent to being legally drunk. Longer periods of continuous work drastically reduce cognitive function and increase the chance of catastrophic error. In both the short- and long-term, reducing sleep hours as little as one hour nightly can result in a severe decrease in cognitive ability, sometimes without workers perceiving the decrease.

# Avoiding the burn out

- ◆ This is where the sprint structure and scrum should come into play
  - ◆ Be honest with estimates
  - ◆ Don't schedule more work than you have time for

# PP3 Grading



FULL  
STRUCTURE  
OF GAME PRODUCTION

Structure of Game Production

# Grading Scale

# Overall Grade

- ◆ Your overall grade is a combination of a few sections
  - ◆ Team Evaluation
  - ◆ Personal Evaluation
  - ◆ GPS

# Team Evaluation

- ◆ The team portion of your grade represents the total team accomplishments during the project as a whole.
  - Graded on a milestone basis
- ◆ Each milestone is graded on its own based on the milestone's goals
  - During first use/playable is the game playable?
  - During Alpha are all the features completed?
- ◆ Each milestone has its own rubric that can be found in the PP3 syllabus or on \\studentvfile\\SGP\\Rubrics

# Personal Evaluation

- ◆ Based on completion of the work that you commit to for the duration of the project.
  - ◆ Graded on each major milestone
- ◆ **Personal Tasks Completed**
  - ◆ Completed all assigned userstories with a high level of quality.
- ◆ **Team Practices**
  - ◆ Updated task board tasks and completed peer reviews.
- ◆ **Maintained Communication**
  - ◆ Informed production staff of any work issues in a timely manner.

# Grading Scale

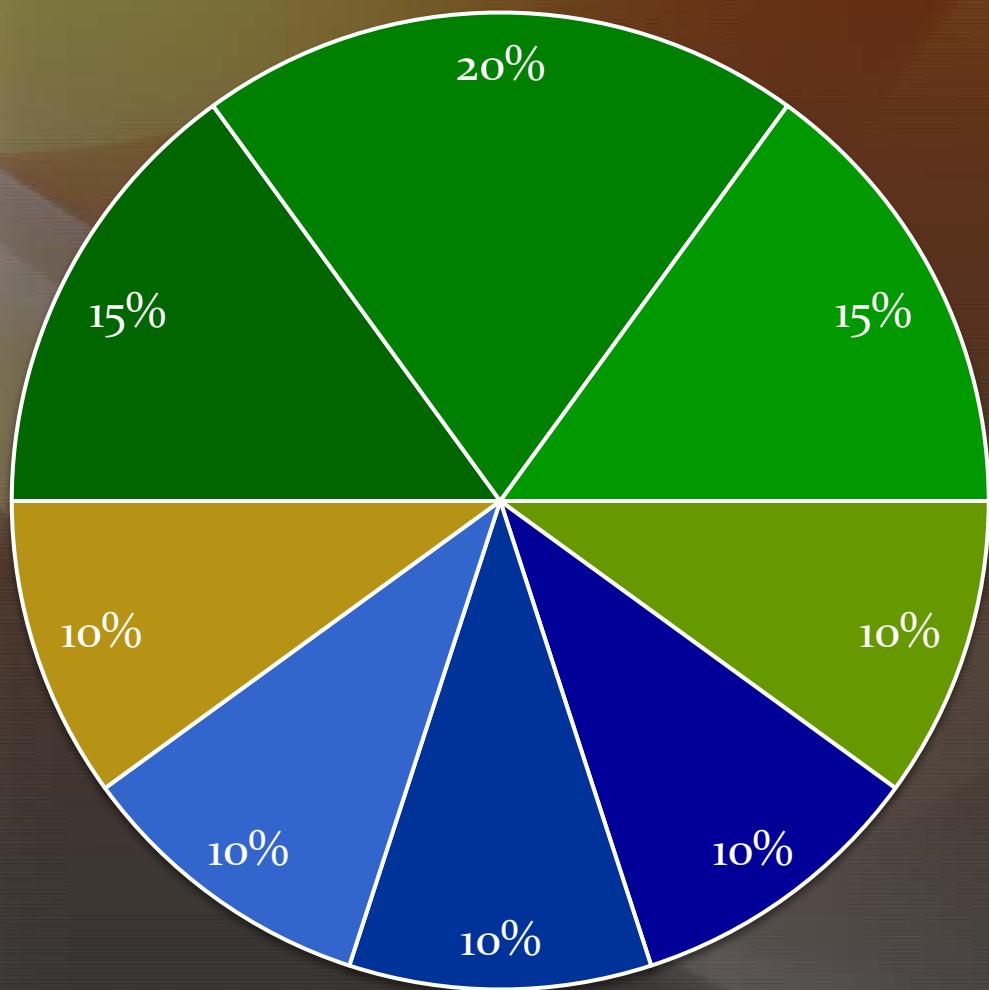
## GPS

- ◆ The faculty reserves the right to identify what is considered unprofessional and what constitutes a breach of the assignment.
- ◆ Common GPS problems include
  - Not complying with imperatives from assigned by EP or APs
  - If you choose to come in and work during the other lecture or lab period and are loud or disruptive during the other class' lecture
  - Being off task during free work hours
- ◆ If you have any questions don't hesitate to ask.

# Grading Scale

## PP3 Weighting

Project Milestones (Team): 45%



- First Use Milestone
- Alpha Milestone
- Beta Milestone
- QA and Finalizing
- First Use Personal
- Alpha Personal
- Beta Personal
- GPS

# Today's Tasks



FULL  
STRUCTURE  
of Game Production

Structure of Game Production

# Things that must be done today

- ◆ Sprint planning
  - ◆ Set Up Sprint backlog
  - ◆ Sprint goal
    - ◆ User stories moved from product backlog to sprint backlog that support the sprint goal
    - ◆ Test cases and task lists reviewed for all sprint user stories
  - ◆ Story allocation
    - ◆ Planning poker completed for all stories
    - ◆ User story owners assigned for all stories
      - ◆ (16 hours per person)
- ◆ Pre Production Review
  - ◆ Grade Pre-pro tasks with Course director

# Sprints

## Getting to 1<sup>st</sup> playable Sprint 1 (Core)

- ◆ Goals
  - The core game feature/mechanic (hook) is fully realized and useable
  - All user input methods working
  - The game has at least one complete playable experience
  - Menu and UI systems created
- ◆ Very light hourly commitment
  - 2 dedicated work days
  - ~16 hours a person

# Still needs to be completed

- ◆ Class Critiques filled out
  - ◆ We do read the critiques
    - ◆ Our boss reads them
    - ◆ We discuss them amongst ourselves
    - ◆ Comments > filling out bubbles
  - ◆ Make sure the comments are unambiguous
    - ◆ Name the course director / lab specialist you are talking about when you do
    - ◆ The more specific the feedback is the easier it is for us to respond to it.
- ◆ Career module
  - ◆ Make sure all tasks have been completed