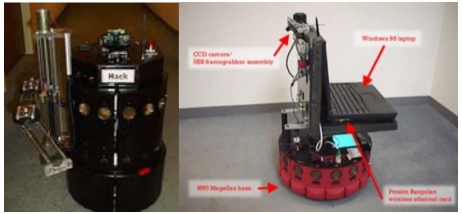
Hello World

Dr. Robert Zubek, SomaSim LLC

EECS-397/497: Game Development Studio
Winter Quarter 2018
Northwestern University

whoami

(x3)


Hack
CCD camera/ USB framegrabber assembly
Windows 98 laptop
RWI Magellan base
Proxim Rangelan wireless ethernet card

Robert Zubek
@rzubek

EA

Three Rings

SEGA

zynga

somasim

Whirled

What is this course?

We’ll be making games!

It’s not about

- Graphics programming
- Engine programming
- Networking programming
- Animation or 3D modeling
- Learning to use specific tools

What is this course?

It *is* about

- … creating aesthetic experiences
- … working in teams
- … planning out your designs
- … building what you planned
- … iterating
- … failing
- … experimenting
- … and making a game

How will this work?

Studio class

- Have you taken an art studio class? That’s the model ☺

The goal: make a game you’re proud of, by the end of the quarter

Subgoals:

- develop skills in designing and building entertainment products
- develop critical vocabulary for analyzing games, your own and by others
- develop skills in giving and evaluating design feedback

How will this work?

The process:

- Intro to game design theory
- Come up with designs you like
- Split into groups
- BUILD IT!
- Iterate on a weekly basis
- Get feedback from your peers - in class, every week
- Give feedback to your peers - in class, every week
- Lectures reduced to short mini-talks

- Demo day + pizza party to show off everyone’s games

Project work

January: PREP

- Everyone comes up with design ideas
- Self-organize into groups of 3 people around ideas you like
- Spend first three weeks of class on planning things out

January 29: GO TIME!

- Weekly sprints and playtests

March 12: DEMO DAY

- (Monday of reading week)

Playtests and Iteration

Focus on practice and feedback

- Each week we bring prototypes of our games
- Everyone either shows game or gives critique
- At the end of the session, tally up feedback, use for planning next sprint

This will start out very simple at first, when prototypes are simple

Gets more interesting once the games get more complete

The point is to develop your skills in development as well as in evaluating

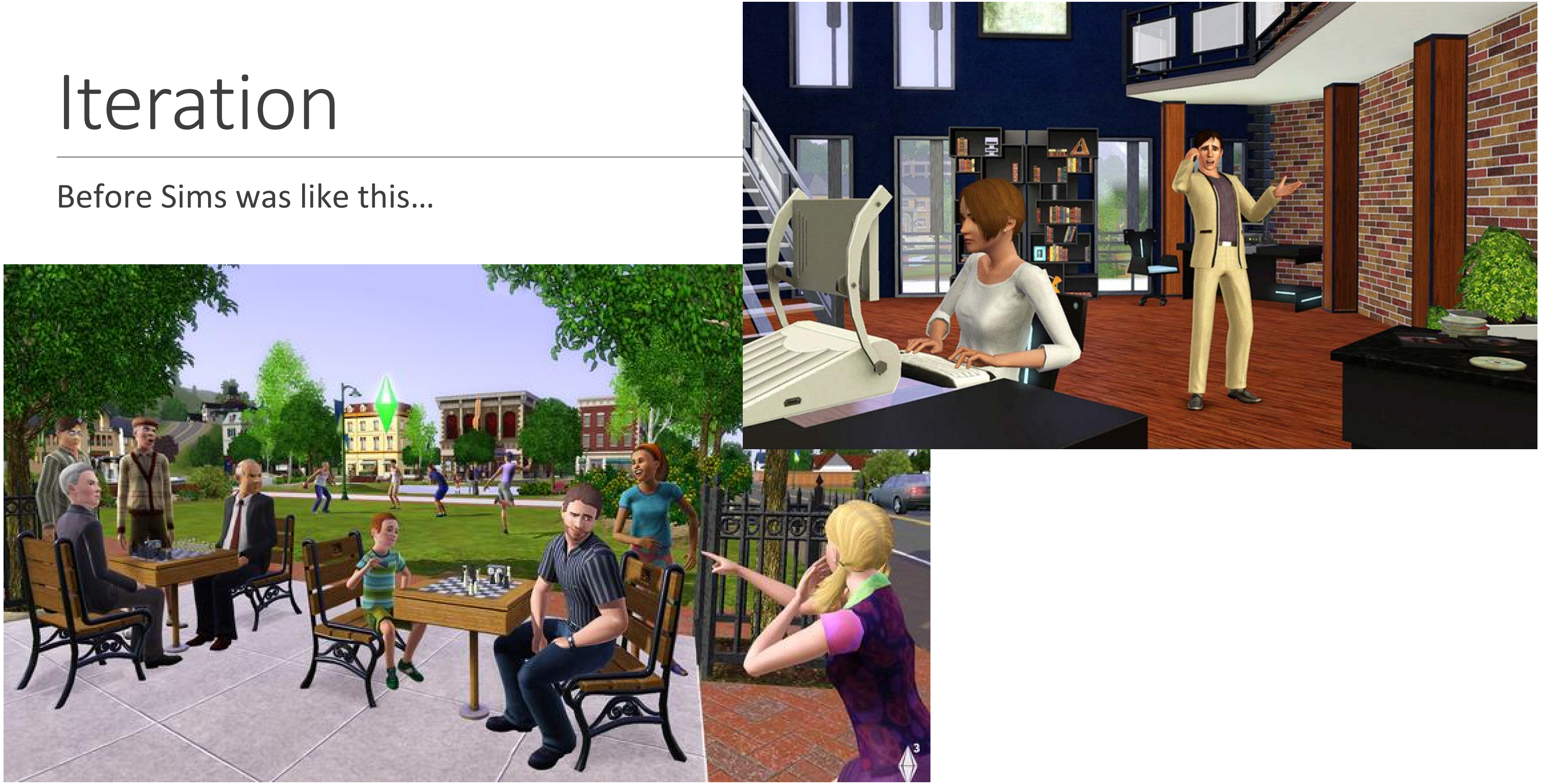
Why so much focus on iteration?

GAMES !=

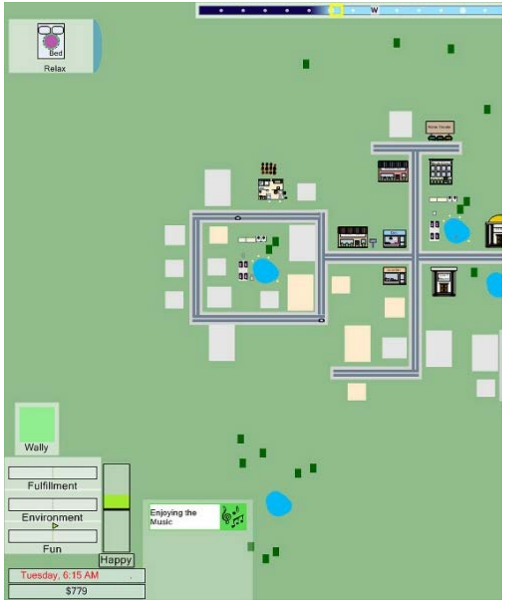
(Athena, sprang out fully formed from the head of Zeus)

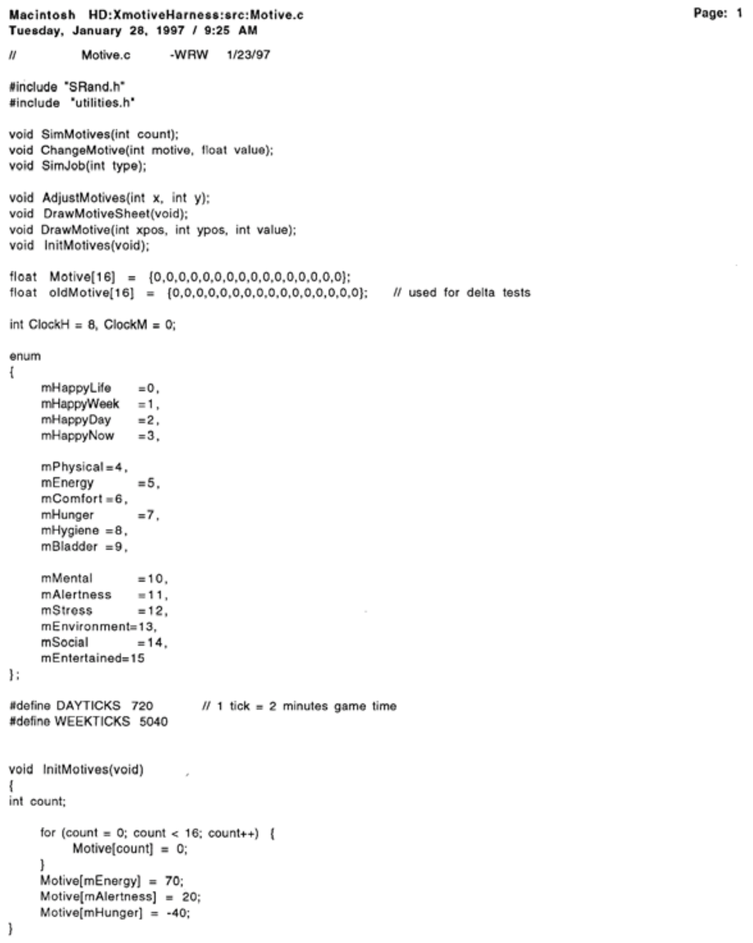
Iteration

Before Sims was like this…

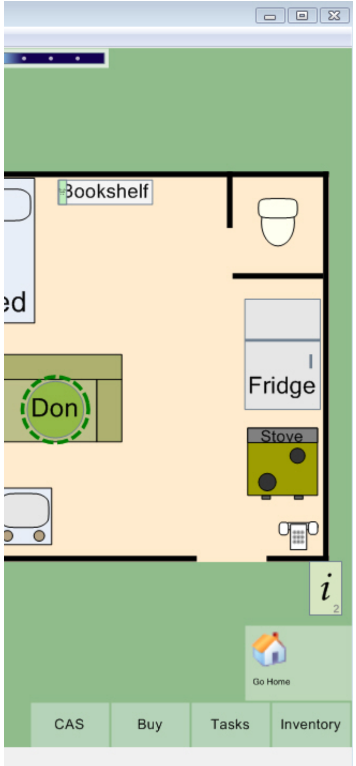


Iteration





```
Macintosh  HD:XmotiveHarness:src:Motive.c                                    Page:  1
Tuesday,  January  28,  1997  /  9:25  AM

//              Motive.c         -WRW     1/23/97

#include "SRand.h"
#include  "utilities.h"

void  SimMotives(int  count);
void  ChangeMotive(int  motive,  float  value);
void  SimJob(int  type);

void  AdjustMotives(int  x,  int  y);
void  DrawMotiveSheet(void);
void  DrawMotive(int  xpos,  int  ypos,  int  value);
void  InitMotives(void);

float   Motive[16]   =   {0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0};
float   oldMotive[16]   =   {0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0};     //  used  for  delta  tests

int ClockH = 8, ClockM = 0;

enum
{
      mHappyLife      =0,
      mHappyWeek      =1,
      mHappyDay       =2,
      mHappyNow       =3,

      mPhysical=4,
      mEnergy         =5,
      mComfort =6,
      mHunger         =7,
      mHygiene  =8,
      mBladder  =9,

      mMental         =10,
      mAlertness      =11,
      mStress         =12,
      mEnvironment=13,
      mSocial         =14,
      mEntertained=15
};

#define DAYTICKS   720          // 1  tick  =  2  minutes  game  time
#define WEEKTICKS   5040


void  InitMotives(void)
{
int  count;

      for  (count  =  0;  count  <  16;  count++)  {
            Motive[count]  =  0;
      }
      Motive[mEnergy]  =  70;
      Motive[mAlertness]  =  20;
      Motive[mHunger]  =  -40;
}
```



How do we do that?

1. Build something
2. Get feedback
3. Evaluate feedback, decide next steps
4. GOTO 1

You'll learn how to **give** design feedback … and how to **receive** design feedback

Final Project

One game, due and demoed on last day of class

Evaluated on

1. the quality of the game

2. your ability to deliver what you planned

Playing to your strengths

We’re programmers, not visual artists

Don’t make games that require a lot of art work

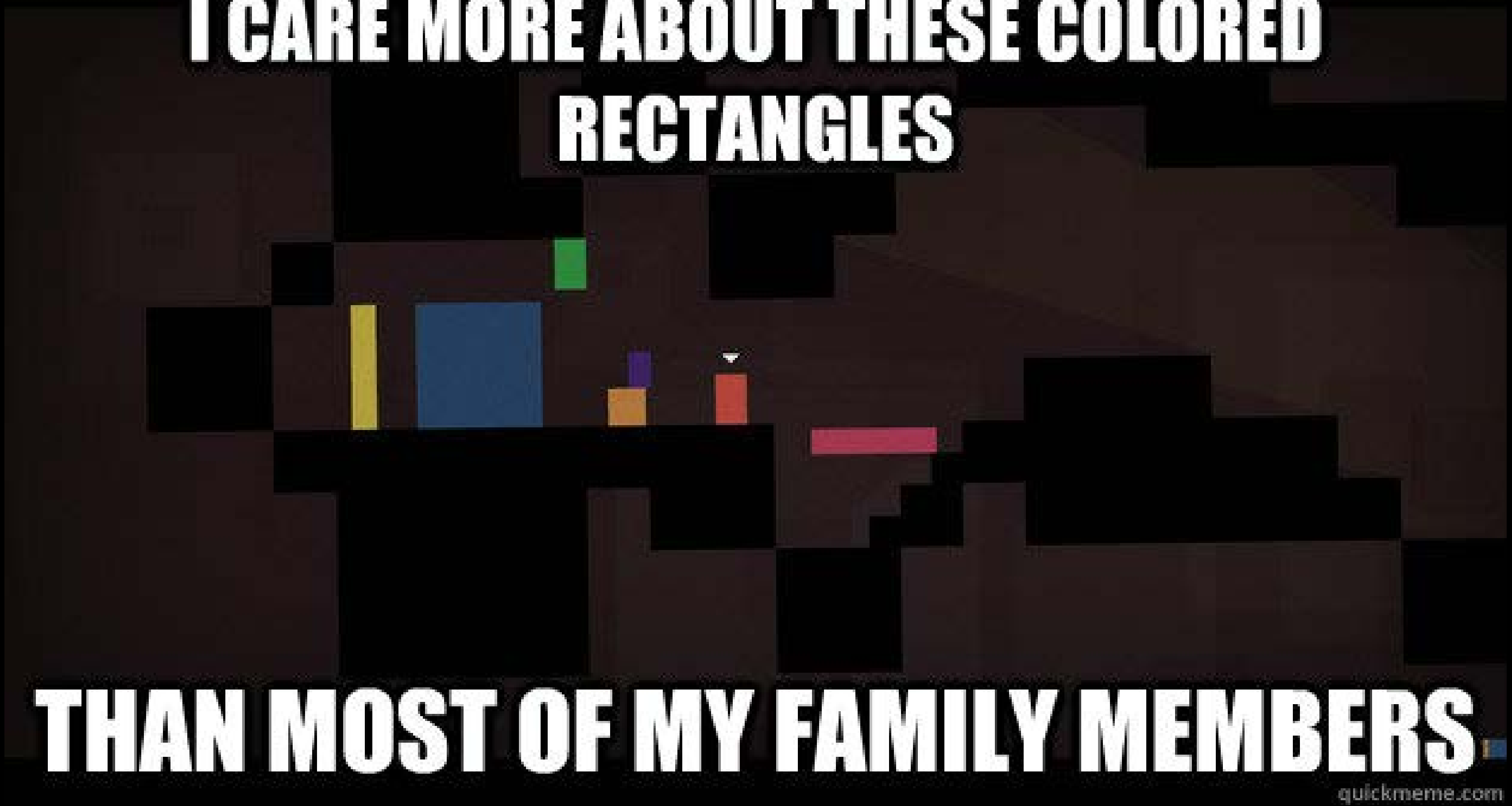
- But reusing existing assets is okay
- Using stuff from the Unity Asset Store is okay (just let me know)

Optimize for fun gameplay

Gameplay, not Graphics

Gameplay, not Graphics

Maybe that's what the dots were for. They were there to extend
John's reach, to make his performance even more impressive.
John liked the thought. He decided to keep them.


I CARE MORE ABOUT THESE COLORED RECTANGLES
THAN MOST OF MY FAMILY MEMBERS
quickmeme.com

High-level schedule

January

Week 1: Intro to class, game design theory

Week 2: Game design theory continued

Week 3: Preparing your designs and plans

Week 4: Groups formed, group presentations
→ Start project work!

February

Weeks 5-10:

- Playtests – bring your game to class
- Everyone shows their game once a week
- Also occasional mini-talks on specific topics

March 12 (Monday of reading week):

- Game Demo Party

Tools

Engine: Unity 2017.x

- Free, cross-platform Win + OSX
- You should already know it from the prereq
- Great self-guided tutorials are available

Language: C#

- Sorry, no JavaScript, proper language please :)
- If you don’t know it yet, need to learn it fast!

Teamwork

- Google docs + spreadsheets
- Source control (git)
- Scheduling: Google spreadsheets or Trello

Note on workload

This class will be a significant time commitment

- Programming time
- Teamwork time (planning, meetings, emails)
- You will have to coordinate schedules, have regular meetings
- **You should budget min 10h/wk for out-of-class work**

You must be a self-sufficient programmer

- Be able to program and debug in C#, find online resources, etc

Note on workload (cont’d)

This class will be a significant time commitment

You must already be a self-sufficient programmer

Use your time wisely

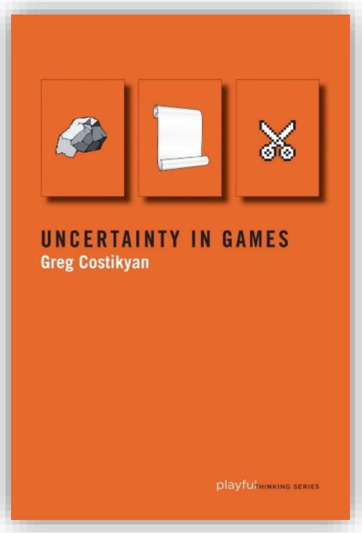
- From now till mid March = not a lot of time to make a game!

Class attendance required

- We’ll be playtesting each other’s games – be here to demo or give feedback

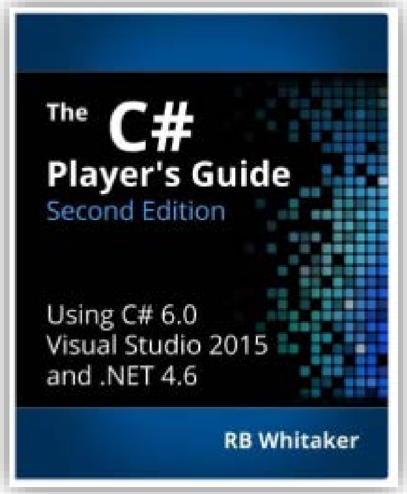
Required Reading

1. My lecture notes
2. Costikyan, *Uncertainty in Games* →



Optional

Reference book from your prereq class



Grading

Group project outcomes: **40%**

- Final project grade (from me): 25%
- Playtest scores (from your classmates): 15%

Participation: **40%**

- Homeworks, project proposals: 15%
- Sending in weekly reports and assessments: 20%
- Doing playtests and sending in your scores: 5%

Teammate assessment of your work:**20%**

Late policy

Group project:

- There is no late policy
- There are no extensions
- Your team either makes its milestones and ship date, or it doesn't

- (Fortunately since you're iterating and always showing something, this should not be a problem)

Homeworks

- Lose 25% when late + 25% for each additional 24h

Course materials

Syllabus, slides, and everything else will live here (also linked from Canvas):

http://robert.zubek.net/docs/games-studio-2018

Homework

This week

- Read lecture notes chapters 1 & 2
- Homework due Sunday, more details after next lecture

On Wednesday we start on game design

Q&A