Foundations of Interactive Game Design (80K)

week one, lecture two
Today

- First quiz!
- Demo of *Thrive* from 80K last year
- Making meaning with simple actions
- Three game demos
- Event-based programming
- Operational logics
- Reminder of what’s coming up
- We may not finish...
Quiz
Did you register your clicker?

A: Yes
B: No
C: Not sure
Which is more important to *Journey* than *Combat*?

A: Multiplayer engagement
B: Narrative progression
C: Technology showcase
D: World design
Notes on quizzes

• Operating someone else’s clicker is academic dishonesty

• Your lowest two quiz grades will be dropped (covers illness, transport failure, clicker not registered yet, forgetting clicker)

• Email your section TA before any lecture or section you will have to miss
Demo: *Thrive*

by James McKenna
You are standing at the end of a road before a small brick building. Around you is a forest. A small stream flows out of the building and down a gully.

> enter

You are inside a building, a well house for a large spring.

There are some keys on the ground here.
There is a shiny brass lamp nearby.
There is food here.
There is a bottle of water here.

> get lamp

Ok.
You are standing at the end of a road before a small brick building. Around you is a forest. A small stream flows out of the building and down a gully.

> enter

You are inside a building, a well house for a large spring.
Adventure, Warren Robinett, 1978
USE YOUR JOYSTICK CONTROLLER WITH THIS ATARI® GAME PROGRAM. BE SURE THE CONTROLLER IS FIRMLY PLUGGED INTO THE LEFT CONTROLLER JACK AT THE REAR OF YOUR ATARI VIDEO COMPUTER SYSTEM. HOLD THE CONTROLLER WITH THE RED BUTTON TO YOUR UPPER LEFT TOWARD THE TELEVISION SCREEN. SEE SECTION 3 OF YOUR OWNER'S MANUAL FOR FURTHER DETAILS.

NOTE: ALWAYS TURN THE CONSOLE POWER SWITCH OFF WHEN INSERTING OR REMOVING AN ATARI GAME PROGRAM. THIS WILL PROTECT THE ELECTRONIC COMPONENTS AND PROLONG THE LIFE OF YOUR ATARI VIDEO COMPUTER SYSTEM.

NOTE: FOR MORE ENJOYMENT OF THE ADVENTURE GAME PROGRAM, PLEASE READ ALL INSTRUCTIONS CAREFULLY BEFORE BEGINNING PLAY. KEEP THIS INSTRUCTION BOOKLET HANDED FOR QUICK REFERENCE.

HOW TO PLAY

An evil magician has stolen the Enchanted Chalice and has hidden it somewhere in the Kingdom. The object of the game is to rescue the Enchanted Chalice and place it inside the Golden Castle where it belongs.

This is no easy task, as the evil magician has created three Dragons to hinder you in your quest for the Golden Chalice. There is Yongle, the Yellow Dragon, who is just plain mean, there is Grundle, the Green Dragon, who is mean and ferocious; and there is Rhindle, the Red Dragon, who is the most ferocious of all. Rhindle is also the fastest Dragon and is the most difficult to outmaneuver.

There are three castles in the Kingdom: the White Castle, the Black Castle, and the Golden Castle. Each castle has a Gate over the entrance. The Gate can be opened with the corresponding colored Key. Inside each Castle are rooms (or dungeons), depending on which Skill Level you are playing.

The Castles are separated by rooms, pathways, and labyrinths. Common to all Skill Levels is the Blue Labyrinth through which you must find your way to the Black Castle. Skill Levels 2 and 3 have a more complicated Kingdom (see SKILL LEVELS section).

USING THE CONTROLLER

You can move in any of the eight directions with the joystick in that direction (see diagram). Each area shown on your television screen will have one or more barriers or walls, through which you CANNOT pass. There are one or more openings. To move from one area to an adjacent area, move "UP" the television screen through one of the openings, the adjacent area will be shown on your television screen.

CONSOLE CONTROLS

Choose the Skill Level you wish to play by depressing the game select switch. To begin play depress the game reset switch.

If you get "eaten" by one of the Dragons, do not despair! Just depress the game reset switch and you will be "reincarnated" and placed back in front of the Golden Castle. Unfortunately, any Dragons you may have slain (see GOOD MAGIC will also be reincarnated. If you were carrying any object with you, it will remain where it was.

If you have finished one game and wish to begin another depress either the game reset or the game select switch. The number of the skill level at which you were playing will appear on the television screen. Depress the game reset switch and begin play.

DIFFICULTY SWITCHES

By using the difficulty switches you can increase or decrease the difficulty of the game at each Skill Level. When the left difficulty switch is in the B position, the Dragons will hesitate before they bite you. Therefore, when the left difficulty switch is in the A position, it is more difficult to escape them.
What did you notice about playing Adventure?
Playing Adventure

- Moving, running into walls
- Running into objects, picking them up
- Running into enemies with objects (combat)
- Running into objects with objects (keys)
- Bridging walls with special object
Does this sound familiar from somewhere?
Event-based programming
This looks simple

But it introduces much that is important to games
Is computation Q&A?
Beyond Turing Machines

Turing Machines are a common way of thinking abstractly about processes... but they don’t account for interaction:

- **Claim:** Interaction-machine behavior is not reducible to Turing-machine behavior.

- **Informal evidence of richer behavior:** Turing machines cannot handle the passage of time or interactive events that occur during the process of computation.

- **Formal evidence of irreducibility:** Input streams of interaction machines are not expressible by finite tapes, since any finite representation can be dynamically extended by uncontrollable adversaries. (Peter Wegner)
What does this have to do with Game Maker?
When fruit collides with a wall, this is an “event”

When a player clicks on fruit, this is an “event”
Event-based programming

• A simple idea:
  Wait for a specific event, then do something

• Learning to decompose problems this way will improve your Game Maker ability

• Games update everything as fast as they can (calculating physics, rendering reflections, updating AI, etc) and monitor for events based on ongoing action and player input
Basic example

- Event Zoe is created, Zoe grabs closest toy
- Event Zoe grabs ball, Zoe throws ball random direction
- Event ball collides with Noah, he catches
- Event ball collides with...
Back to our story
The Legend of Zelda, Shigeru Miyamoto, Takashi Tezuka, et al, 1986–7
What did you notice about playing *The Legend of Zelda* compared with *Adventure*?
Zelda versus Adventure

- More complex world
- More complex enemies and attacks
- More objects, inventory
- Resource management, vendors
- Running into things still fundamental: moving through world (boundaries), picking up rewards, fighting enemies
Let’s take a step back
How do games mean?
What does this mean?
And *how does this mean?*
Not this
or this
It’s this

Game State Presentation

Computational Process

Player Experience
How does collision detection mean?

- Visually: A ball runs into a paddle, then something happens (it bounces off). Like how meaning happens in a movie.

- Experientially: As we play, an experience unfolds. Balls keep running into paddles, from different angles, and keep bouncing off — we get a feel for how the world works.
The foundation of game meaning

- The *experience* of collision detection is only made possible by an underlying process.
- There are many specific collision detection algorithms, but all support an abstract process wedded to a communicative goal: when two virtual objects touch, something happens.
These are operational logics
Operational logics

- A communicative goal — “virtual objects can touch”
- combined with an abstract process — “when two coordinate spaces overlap, do something”
- supporting ongoing media (re)presentation and audience experience
Implemented in many ways

• When you play Pong on an Atari VCS, the 2D collision detection is implemented in hardware.

• When you build a game using XNA, the 3D collision detection is implemented in software.

• Obviously, implementations differ fundamentally, but the logic — that virtual objects can “touch” — is the same.
When fruit collides with a wall, this is an example of an operational logic.
Further meanings

• Basic operational logics are articulated with game systems and themes

• Meaning approaches become conventional: running into walls, picking up objects

• Games build on this conventional knowledge like film builds on our familiarity with cross-cutting, etc
What are creative ways to tap into this experience of meaning?
Passage, Jason Rohrer, 2007
What did you notice about playing Passage compared with Adventure and Zelda?
Collision detection — and other elements of spatial navigation — re-mapped to options and constraints in life choices

Splits apart conventional, almost-invisible logic — same abstract process, new communicative goal

Could easily be made in Game Maker...
What’s next in 80K

• Read first chapter of Fullerton for Friday
• Sections this week introducing Game Maker and sections
• Customized version of first tutorial due in section next week
• Look over the syllabus, start experimenting with Game Maker — play some GM games?