Foundations of Interactive Game Design (80K)

week one, lecture two
Today

- More demos of Game Maker games and questions for their creators
- Event-based programming
- Operational logics
- Game structure: players, objectives, procedures, rules, resources, etc
- Platform and hidden object games
- Preview and learning objectives for next week
Demo: *Salvage*

by Jon Holtan & Sam Wolpert
Demo:
Zool: Ninja of the Nth Dimension: ReGremliffied
by Alexander McCaleb
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 the clown collides with a wall, this is an “event”

When a player clicks the clown, 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...
Operational logics
How do games mean?
What does this mean?

Something ran into something
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.
- Such combinations are operational logics.
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 the clown collides with a wall, this is an example of an operational logic.
Logics matter to players

- Logics aren’t just dry “rules” of games — they are how the world is alive
- We see this in online videos of collision detection
- They don’t focus on showstopping bugs or everyday occurrences, but on humor and surprise at the world’s fundamental operations being violated
For example...
Graphical logics are the abstract operations and communicative goals associated with movement, collision detection, and physics.

- **Movement** – objects move in space
- **Collision** – object overlap triggers events
- **Physics** – movement governed by laws
Isn’t this just graphics?
Fiction logics

• *Quest flag* logic for quests / missions (milestone-based progression).

• *Dialogue tree* logic for NPC interactions — discussion, provocation, quest acceptance/completion, etc (directed graph).

• The interfaces change, the underlying logics remain.
Resource management

• As Michael Mateas observes, resource management logics are the abstract operations associated with acquiring, using, and transforming resources (e.g. food, money).

• Allocation – selecting sources, sinks, and transformations to apply to a resource.

• Random events – events within the fictional world that modify quantities or rate constants.
Resource acquisition, allocation, transformation

Try your hand at governing ancient Sumeria for a ten-year term of office.

Hammurabi: I beg to report to you.
In year 1, 8 people starved. 5 came to the city.
Population is now 100.
The city now owns 1000 acres.
You harvested 3 bushels per acre.
Rats ate 200 bushels.
You now have 2800 bushels in store.
Land is trading at 26 bushels per acre.
How many acres do you wish to buy?
Resource allocation and random events
Time as resource, random events (burglar)
Formal model of resource logics

Machinations:
Joris Dormans
Operational logics support game mechanics
Game mechanics

• Some game genres are named in ways that hint at their core mechanics

• Can people here think of examples?

• First-person and third-person *shooters*

• More obscurely: jumping for *platformers*, finding/selecting for *hidden object* games

• Both jumping and shooting are supported by collision detection
Describing mechanics

• The core mechanics of a game are what players do over and over when playing — jumping, shooting, extracting resources, moving pieces.

• There are other mechanics for other things that can be done — swimming, castling.

• Some designers and theorists use the term “mechanics” for everything that governs the behavior of the game system — every rule.
Like Fullerton’s procedures (and rules)
Fullerton’s formal elements

- Players: what other entertainment demands consumer participation?
- Objectives: specific goal(s) for players
- Procedures: actions or methods of play allowed by the rules
- Rules: object definitions, concepts, what is allowed/enabled
- Conflict: objectives can’t be achieved directly
- Outcome: uncertain and unequal
Specifics of mechanics

• A core mechanic of Chess is piece movement
• The fact that movement is turn-based is very important
• So is the fact that pieces move in defined ways that are different from each other
• This interacts with the capture mechanic — ending a movement on the same space as an enemy piece
• This produces dynamic of projecting force
Next week we’ll go further
Choice of platformer or hidden object tutorial
Next week

• First tutorial due (in section)
• One Fullerton chapter for Tuesday & Thursday
• Write your one-page mechanics analysis
• Second tutorial and team selection assigned
Learning objectives, week 2

• Game Maker: Variables, properties, collision detection, and scripting (section)

• Game design concepts and innovation (section, text, & lecture)

• Basics of computer game genres and computer game history (lecture)

• Deeper on Fullerton’s formal elements (text & lecture)

• Types of play (including Huizinga’s) and types of fun (including Koster’s) (lecture)