welcome to

GAMELET DESIGN

for education

Office hours

- Alexander Repenning: ecot 826, 10am, Tuesdays
- Clayton Lewis: tbd

Announcements

- Final call: If you like to have access to lab see me after class: Buff oneCard#:
- Home work submission: **this week only**: you have until Jan 24, 11:59 pm to fix submission updates
  - Make sure there is a source: click a source link, upload again if needed
  - Make sure your thumbnail is 256 x 256

Objectives

- Gamelet design
- Quick peek at some Froggers
- Hands on: start game #2: Sokoban

Gamelet design

In 6 easy steps

Problem Analysis

Game Analysis Outline

1) *List **agents***: project description nouns
   a) categorize agents: {user controlled, AI controlled, passive/props}
   b) define user control and AI
2) *Determine **methods**:* project description verbs
3) *Determine **states**:* project description adjectives
   a) ***visible states*** -> agent depictions
   b) ***hidden states*** -> agent attributes
4) *Draw **storyboards***: game levels and situations
5) *Define **interaction***: sequence diagrams
6) *Define **change***: state diagrams

FROGGER

You are a frog. Your task is simple: hop across a busy highway, dodging cars and trucks, until you get the to the edge of a river, where you must keep yourself from drowning by crossing safely to your grotto at the top of the screen by leaping across the backs of turtles and logs. But watch out for snakes and alligators! (Sega, 1980)

1) *List agents*: project description nouns

- a) Categorize agents: {user controlled, AI controlled, passive/props}
  - User controlled:
    - Cursor keys
    - Sound, speech recognition (Mac only)
- b) Define user control and AI
  - AI from simple to complex
    - Random movement
    - Tracking with path finding
    - Collaborative Diffusion

What are the Objects/Agents?

You are a frog. Your task is simple: hop across a busy highway, dodging cars and trucks, until you get the to the edge of a river, where you must keep yourself from drowning by crossing safely to your grotto at the top of the screen by leaping across the backs of turtles and logs. But watch out for snakes and alligators!

What are the Objects/Agents?

- Mark nouns

You are a frog. Your task is simple: hop across a busy highway, dodging cars and trucks, until you get the to the edge of a river, where you must keep yourself from drowning by crossing safely to your grotto at the top of the screen by leaping across the backs of turtles and logs. But watch out for snakes and alligators!

2) Determine methods: project description verbs

What are the Operations/Methods?

You are a frog. Your task is simple: hop across a busy highway, dodging cars and trucks, until you get the to the edge of a river, where you must keep yourself from drowning by crossing safely to your grotto at the top of the screen by leaping across the backs of turtles and logs. But watch out for snakes and alligators!

What are the Operations/Methods?

- Step #2: Mark **verbs**
- You are a frog. Your task is simple: hop across a busy highway, dodging cars and trucks, until you get the to the edge of a river, where you must keep yourself from drowning by crossing safely to your grotto at the top of the screen by leaping across the backs of turtles and logs. But watch out for snakes and alligators!

3) Determine states: project description adjectives and quantitative nouns

a) ***visible states*** -> agent depictions

e.g. human emotions: happy vs.sad face

e.g live frog vs. crushed by truck frog

b) ***hidden states*** -> agent attributes

a) Hunger level, age,

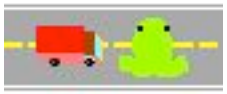
4) *Draw storyboards*: game levels and situations

5) Define interaction: sequence diagrams

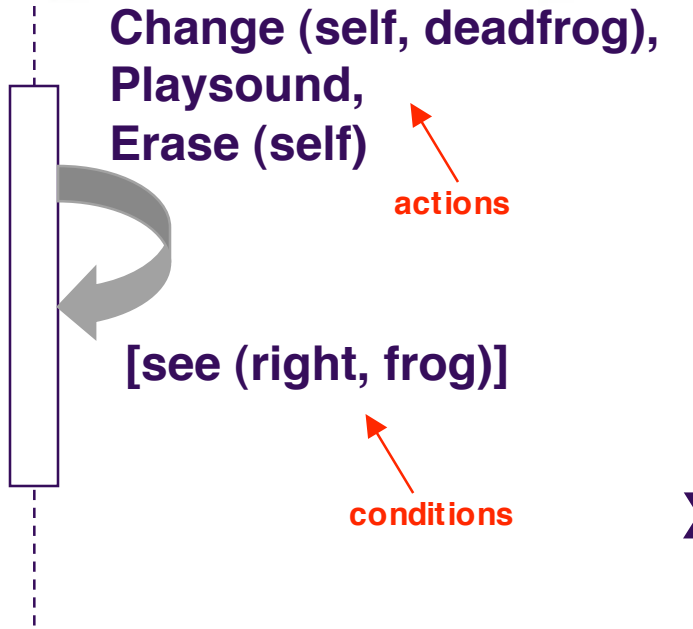
- Which agents are interacting with what other agents?
  - e.g., collision: Truck impacts Frog
- Who deals with interaction?
  - Who need to have the main set of rules?
  - e.g Frog OR truck
- How to deal with interaction
  - Polite: observe environment, change yourself
  - Sneaky: Observer environment, change others
  - Flexible: send messages

Frog meets truck1: user agent centric

Situation: Truck hits frog

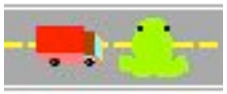


Frog sees truck to left. It changes its look to dead frog, plays sound, and erases itself



Frog meets truck2: AI agent centric

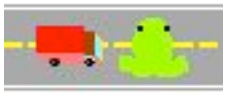
Situation: Truck hits frog

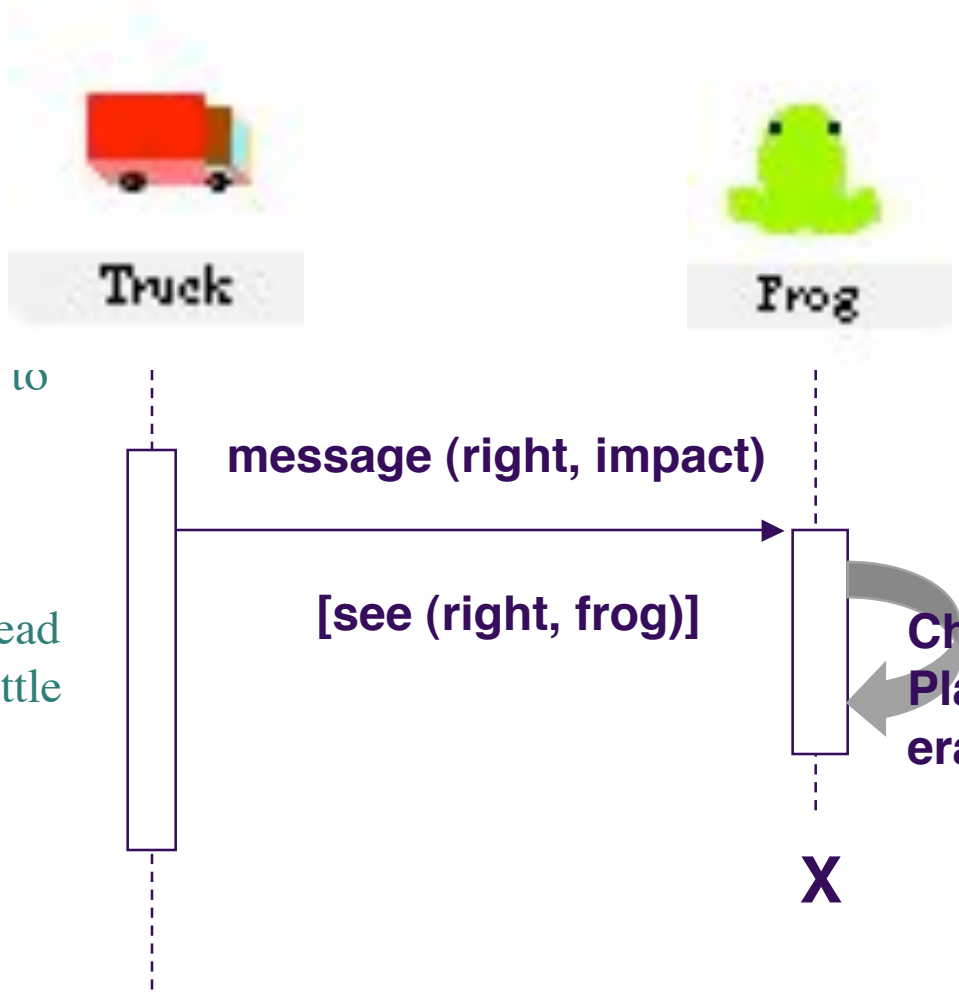


Truck sees a frog to the right: the truck changes the look of the frog to look dead, plays a sound, and erases the frog.

Frog meets truck3: messages

Situation: Truck hits frog





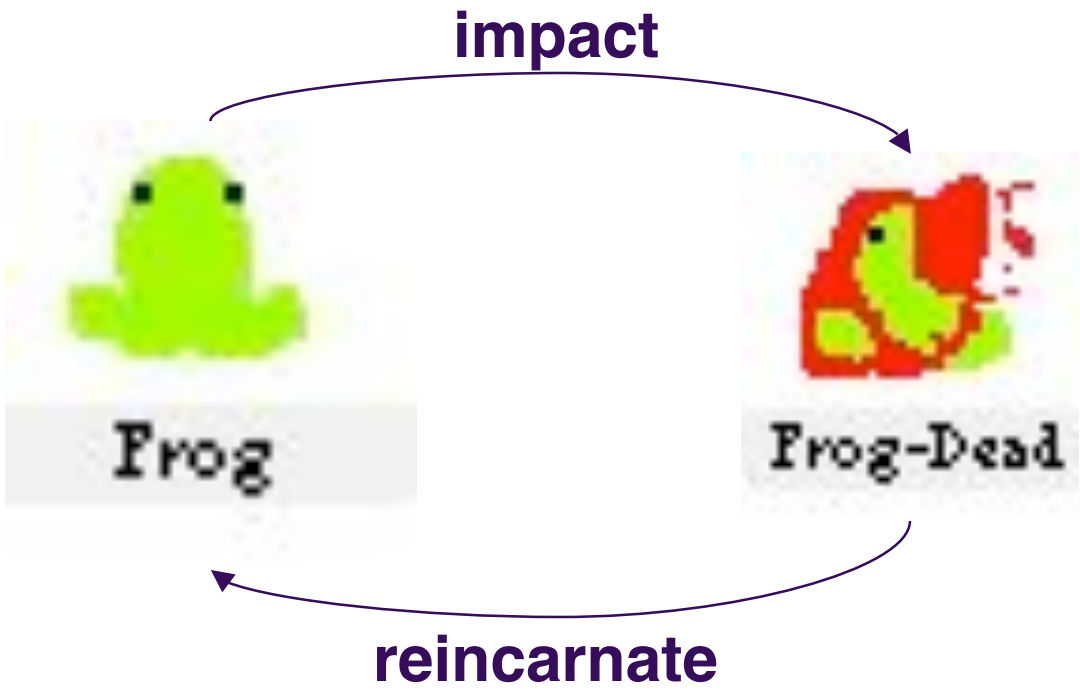
Truck sees a frog to the right

Sends impact message to frog

Frog looks like dead frog and after a little while disappears

6) Define state transitions: state diagrams

- Change visible or hidden state?
- What causes the change to happen

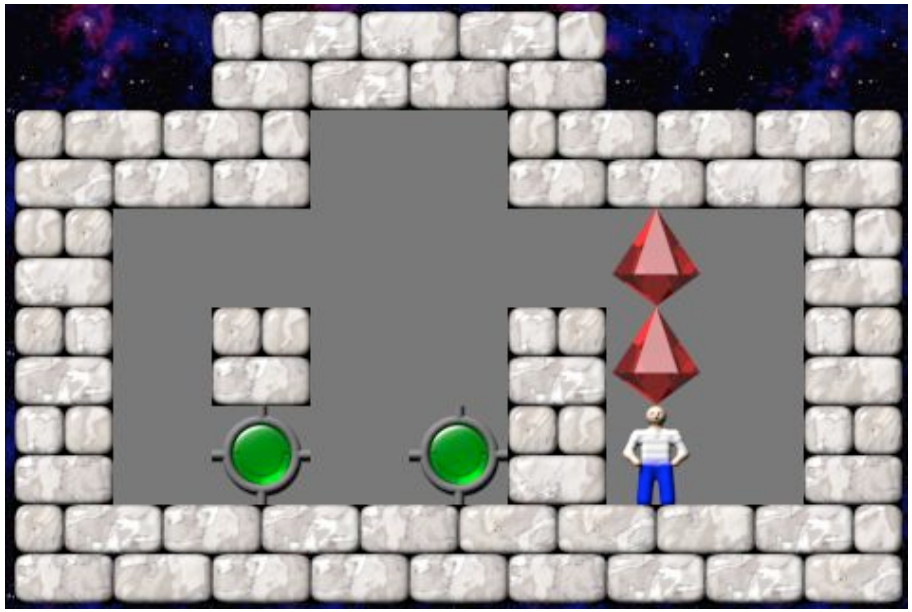


Programming homework #2

Sokoban

Sokoban

Sokoban (Japanese for "warehouse keeper") is a transport puzzle in which the player pushes boxes around a maze, viewed from above, and tries to put them in designated locations. Only one box may be pushed at a time, not two, and boxes cannot be pulled. As the puzzle would be extremely difficult to create physically, it is usually implemented as a video game.



http://en.wikipedia.org/wiki/Sokoban

Why Sokoban?

- More complex agent interaction
  - Small number of agent classes: 4
  - Hint: you will need messages
- Manages Flow entirely through game levels
  - No need to timing, or large cast of icons
  - Dedicated communities of level building:
    - http://www.sourcecode.se/sokoban/levels.php

Homework

- Submit via GORP/sokoban
  1. Applet
  2. Source
  3. Thumbnail 256 x 256
  4. UML diagram (submit as attachment)
- Due: **Monday**, Jan 30, 11:59 pm

score

- Minimal: 100
  - Cursor controlled agent pushing boxes
    - Pusher may never step onto box
    - Box should not jump away from pusher
  - Working game over detection
    - No hardwiring, no counter
  - 3 levels: automatic switching
    - Simple -> medium -> hard
- Added points: 20
  - Additional levels: mention in description when levels are original
  - Extended game concept: describe if present
  - Original game art

How to make UML diagrams

- Make it look like the ones in these slides:
  - Show situation
  - Explain what is going on
  - Show actions and conditions
- To make diagram
  - Paper and pencil -> scan in
  - PowerPoint
  - …