Interface in Games

Topics in Game Development

UNM

Spring 2008

ECE 495/595; CS 491/591

User Interface (UI) is:

- The connection between game & player
- How player receives information
- How player takes action
- How player is given feedback about effect of his/her actions
- One of the things that can make a game fun… or disappointing and frustrating

Ideal UI:

- Offers maximum control
- Offers information needed to play and enjoy game
- Easy to learn
- Easy to use
- Very clear

Some Important Principles

- Should be consistent
- Should provide useful feedback to player
- Should keep things simple: not require numerous steps to perform an action
- Should spare player’s memory by displaying necessary information or making information easy to find

4 Major Types of Interface

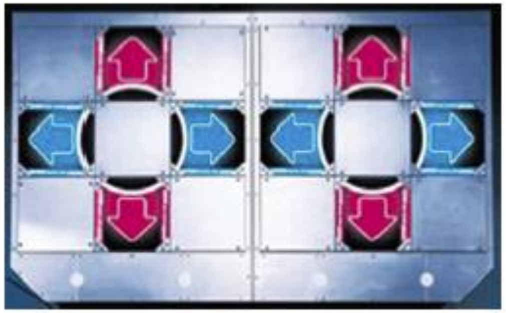
1. Manual
2. Visual
3. Auditory
4. Tactile

Manual Interface (Hardware Based Input Devices)

- Most common:
  - Controllers, joysticks for game consoles
  - Keyboard and mouse for computer games
  - Self-contained hand-held devices, mobile phones
- Each associated with certain game genres – pick platform/device best suited to your game

Other Examples of Manual Interface:

Bass fishing; Samba de Amigo;
Dance Dance Revolution



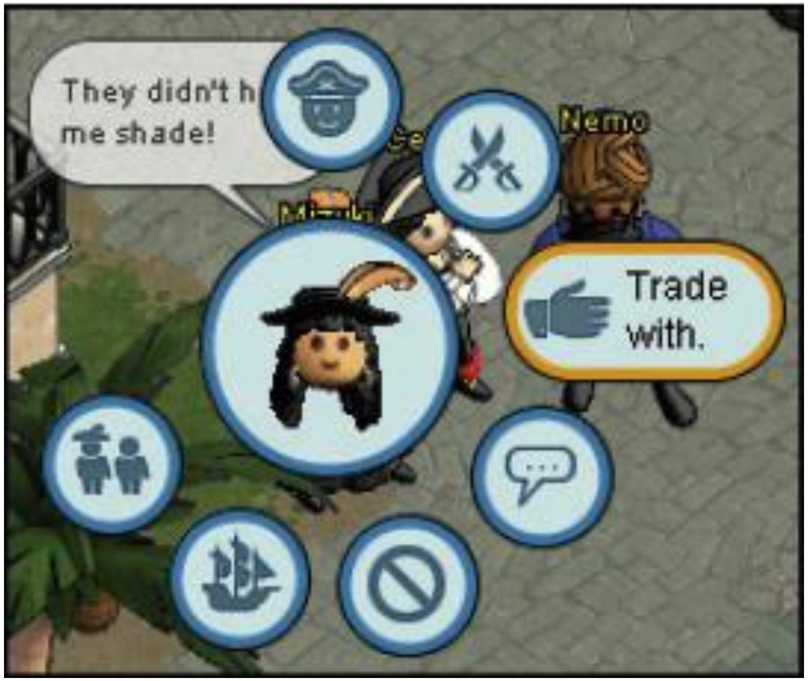
One of the newest: the Wii

And Going Way, Way Upscale: The D-Box GP-100

- Designed for racing games, FPS, flight simulators
- Tactile feedback: acceleration, bumps, turns, stops
- Only $14,999!

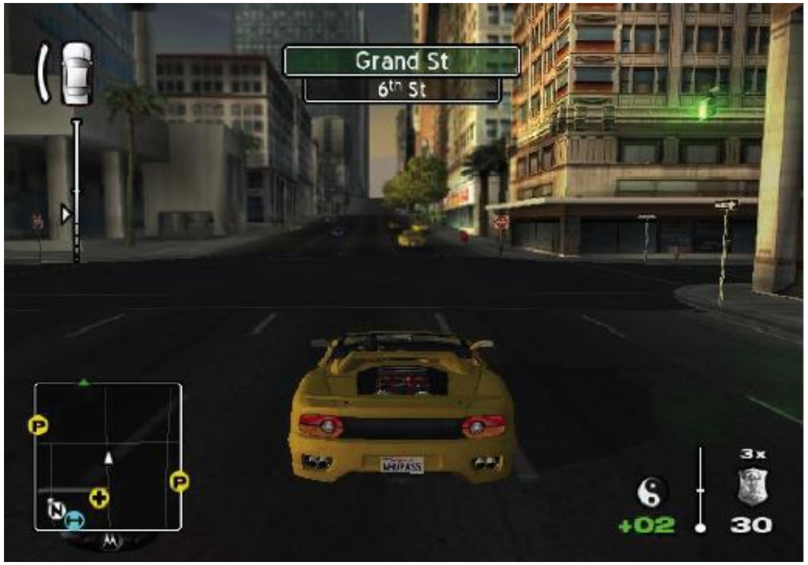
Visual UI: Active type

(Enables interaction – lets player do things)



Visual UI: Passive Type

- Cannot interact
- Provides information like player status, location
- Information provided cannot be changed



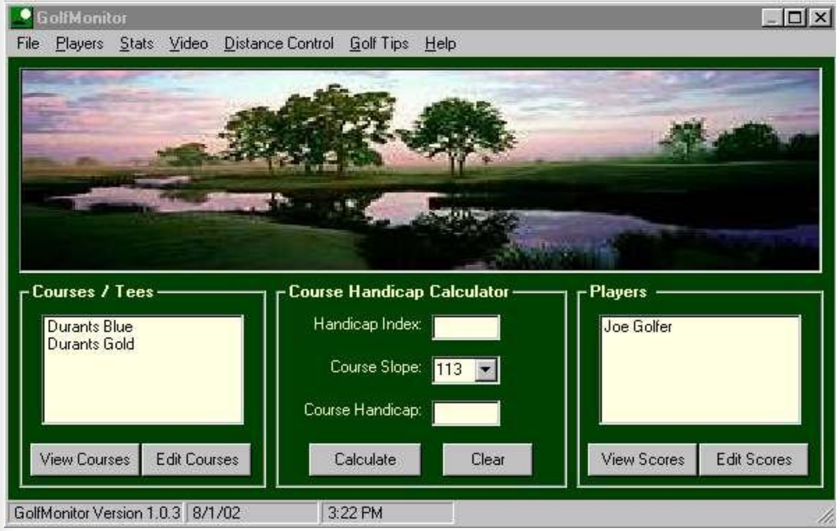
Visual UI Utilizes:

- Numbers
- Text
- Colored lights
- Icons
- Power bars
- Maps
- Screen buttons
- Menus

Auditory UI (Sound)

- May provide feedback when action accomplished: throw grenade, hear *bang*
- May provide warning: rustling leaves = enemy near
- May cue player to do something: dog barks = time to hide
- May be verbal feedback: instructions, praise, danger
- May offer clues about environment: tropics (bird calls), city (traffic sounds)
- May be in form of music: cues, rewards

UI Provides Information About All Important Aspects of Game



Durants Blue
Durants Gold

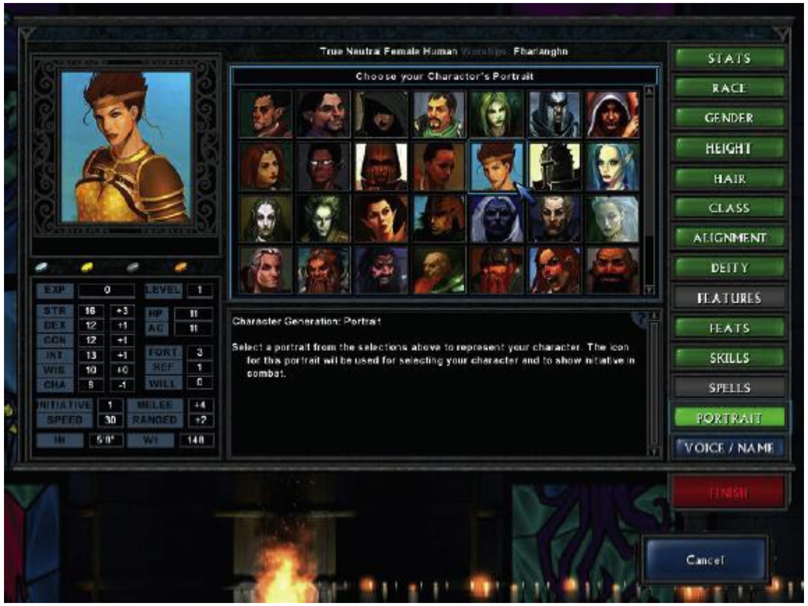
Course Slope: 113

Joe Golfer

GolfMonitor Version 1.0.3 | 8/1/02 | 3:22 PM

- Player’s location, game’s geography (maps, etc)
- Player’s status (score, health, skills)
- Inventory
- What player is doing
- Challenges player is facing
- Whether player is succeeding or failing

UI as a Way for Player to Perform Actions


True Neutral Female Human Worships: Fharlanghn

Choose your Character's Portrait

Character Generation: Portrait

Select a portrait from the selections above to represent your character. The icon for this portrait will be used for selecting your character and to show initiative in combat.

EXP 0 LEVEL 1

STR 16 +3
DEX 12 +1
CON 12 +1
INT 13 +1
WIS 10 +0
CHA 8 -1

HP 11
AC 11

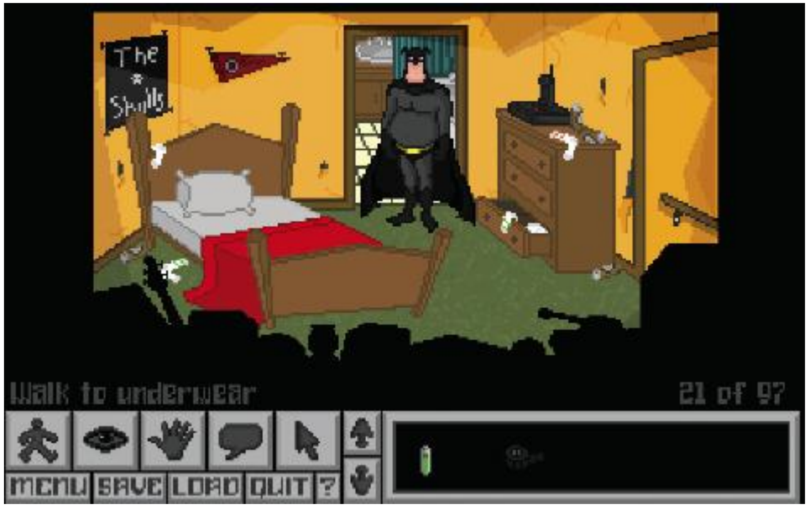
FORT 3
REF 1
WILL 0

INITIATIVE 1 MELEE +4
SPEED 30 RANGED +2

Ht 5'0" Wt 148

- Customize things: avatars, vehicles, real estate, clothing
- Move (run, jump, swim)
- Navigate (travel long distances)
- Pick things up
- Use tools & weapons
- Interact with NPCs, other players
- Collect objects
- Construct & destroy things

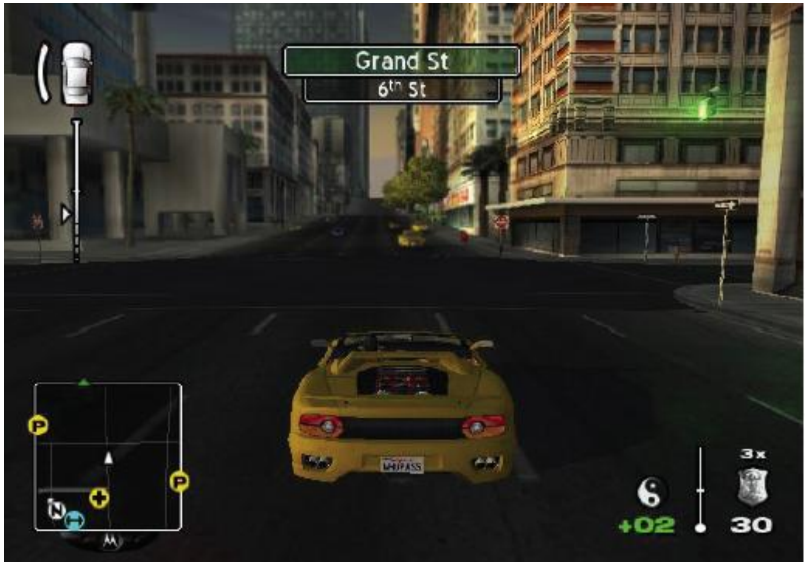
Ways of Offering Visual Interface During Gameplay



One way: Windowed views

- ➢ information on bottom, top, sides of screen (here on bottom)
- ➢ At least 9 layout styles

Second Way: Overlays



- Overlays more immersive; integrated into gameplay
  - Info appears as needed
  - Can be opaque (blocks out background
  - Can be transparent (sometimes hard to read)

Visual Interface and Genre

- Visual interface varies greatly from genre to genre, depending in large part on types of actions players perform
- Best advice: study visual interface in other games in same genre as yours
- Model your game on others in genre
- Innovation NOT a plus in UI!

Saving as a Specialized Action

Saving raises issues of immersiveness, player control

Possibilities include:

- Saving at automatic checkpoints (non-disruptive, but no player control)
- Save to file or save slot (offers player control but at expense of immersiveness
- Quick save (non disruptive, but usually no options)

Taming Interface Complexity

- Simplify, even if some authenticity is sacrificed (example: if player is to fly a jet, don’t try to replicate all the controls in a real cockpit)
- Automate some functions (example: in racing game, car shifts automatically)
- Limit number of steps required to take an action (example: execution of fancy maneuver in a fighting game)

Ideal UI to Strive For:

- Easy to learn, easy to use; intuitive, clear
- Gives player helpful feedback
- Uncluttered, functional, efficient
- In **aesthetic harmony** with game world – in keeping with it, in character

UI for your team projects

- How will the player be given essential information? (location, status, success or failure, etc?)
- Visual interface:
  - What kinds of information will be given visually?
  - What style are you using (windowed or overlays?)
  - What types of visuals will you use? Is it in keeping with your game world?
- Auditory interface:
  - What kinds of information will be given through sound?
  - What kinds of sounds will you be using, and to convey what?