CSE 3902: Audio and Heads-Up Displays

Justin Holewinski

The Ohio State University

Audio in MonoGame

Sound Effects

- Add `.wav` to content project
- Use `ContentManager` to load `SoundEffect` instance
- Play sound with `SoundEffect.Play()` (more on this later)
  - Multiple sound effects can be playing simultaneously
  - Don’t replay the attack sound every frame of an attack!

```
SoundEffect effect =
    Content.Load<SoundEffect>("sounds/attack");

effect.Play();
```

Music

- Add `.mp3` to content project
- Use `ContentManager` to load `Song` instance
- Play song with `MediaPlayer.Play()`
  - Only a single song can be playing at a time

```
Song music =
    Content.Load<Song>("songs/background");

MediaPlayer.Play(music);
```

SoundEffectInstance

- The `SoundEffect` class only stores sound effect data
  - Does not have state like “playing” or “volume”
- The `SoundEffectInstance` class stores state about an active sound effect
  - `IsLooped`
  - `Volume`
  - `State` (`Playing`, `Paused`, `Stopped`)
- Use `SoundEffect.CreateInstance()` to get an instance of a sound effect
- Use `SoundEffect.Play()` for a one-shot sound effect without ability to pause/stop/loop
  - Internally pools `SoundEffectInstance` objects

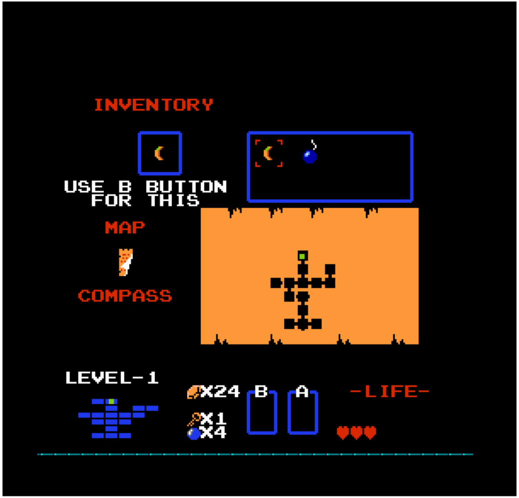
Heads-Up Display

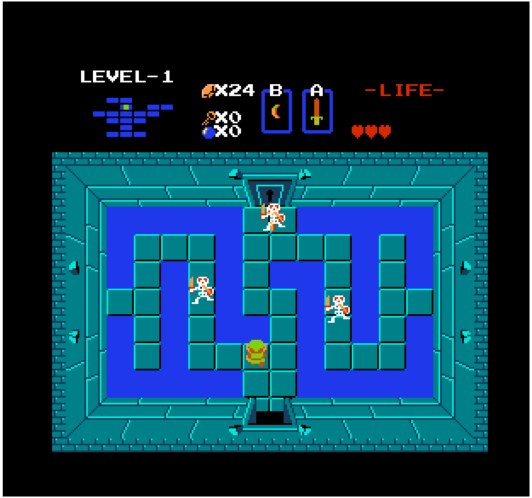
Shows information about the game

- Health, Ammo, Items

Typically not part of the "game world"

- Drawn over game level, or separately
- May have multiple "screens"





Sprint 4 Considerations

Consider what needs to be drawn in your HUD

- Model off original game
- Things to include
  - Player Health
  - Inventory Management
  - Map (with associated pickups)
  - Items (money, bombs, keys)
- Allow game to pause when in inventory management screen
- When in doubt, **model off original game**!
- If using custom game, **implement equivalent functionality**!