

Game Objects

Game State

Game Mode

Game Objects

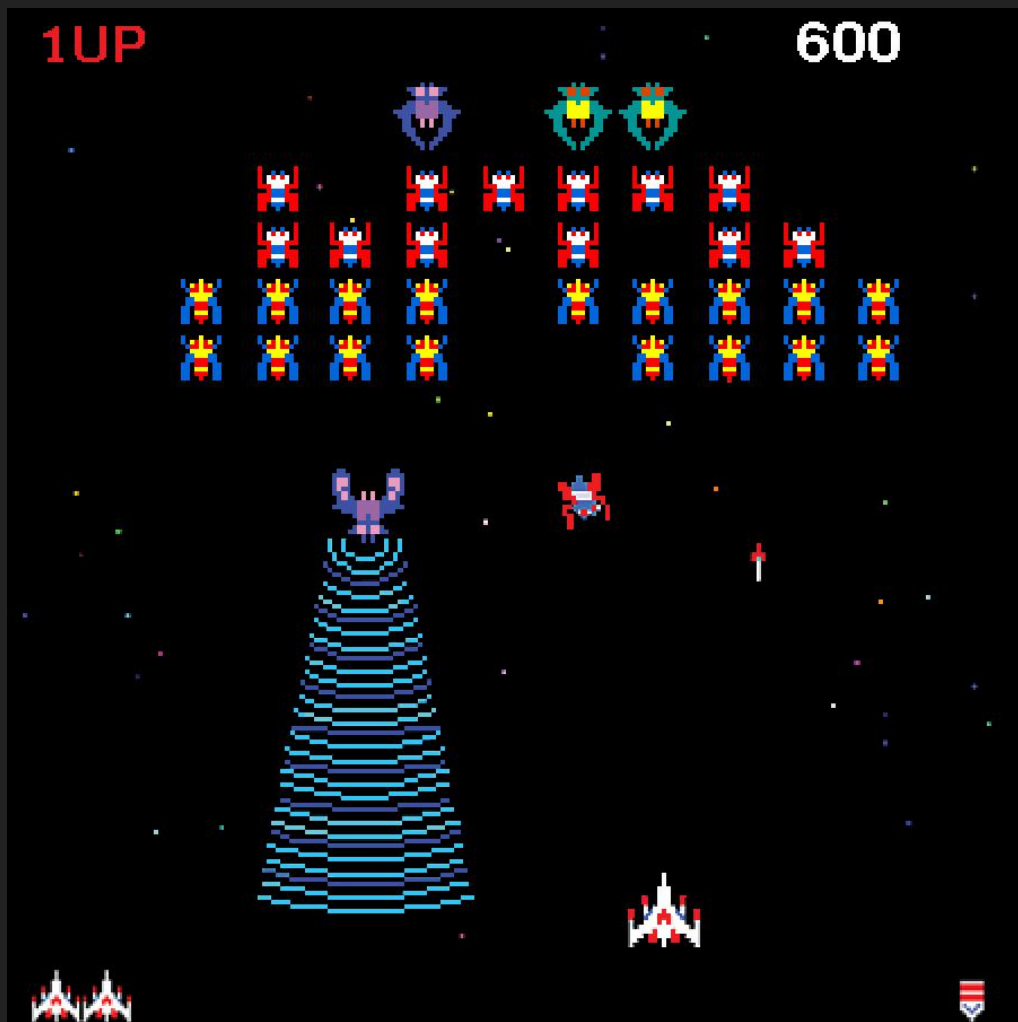
(entities)

Entities help us to organize and manage game objects.

```
class Entity {  
public:  
    glm::vec3 position;  
    glm::vec3 movement;  
    float speed;  
  
    GLuint textureID;  
  
    Entity();  
  
    void Update(float deltaTime);  
    void Render(ShaderProgram *program);  
};
```

1UP

600





250

100

20/20



Norad II
General Duke
Kills: 0

700/700
250/250



MENU



Object Pool

(optimizing for tons of objects)



Object Pool

Create/Allocate objects ahead of time.

Use a bool for active or not.

Maximum number of objects.

You can test with max amount of objects.

Less prone to memory leaks.

Object Pool

```
#define MAX_BULLETS 100;

int nextBullet = 0;
Entity bullets[MAX_BULLETS];

void initialize() {
    for (int i = 0; i < MAX_BULLETS; i++) {
        bullets[i].active = false;
    }
}

void fire() {
    bullet[nextBullet].position = // somewhere
    bullet[nextBullet].active = true;

    nextBullet++;
    if (nextBullet == MAX_BULLETS) nextBullet = 0;
}
```


Game State

entities, score, lives left, time left, etc.

Game State

```
struct GameState {  
    Entity player;  
    Entity enemies[10];  
    Entity items[5];  
    int score;  
};
```

```
GameState state;
```

Game Mode

Game Mode

Arcade games typically feature an “attract” mode.
Also called “demo” mode. Sometimes seen in NES games.

Game plays itself, shows high scores, cut scenes, etc.

Rygar:

<https://www.youtube.com/watch?v=jV2iT9LnCD4>

Street Fighter II:

<https://www.youtube.com/watch?v=TU1C1ihW2mq>

Game Mode

Modern Console/PC Games typically do not have Attract/Demo modes.





Main Menu



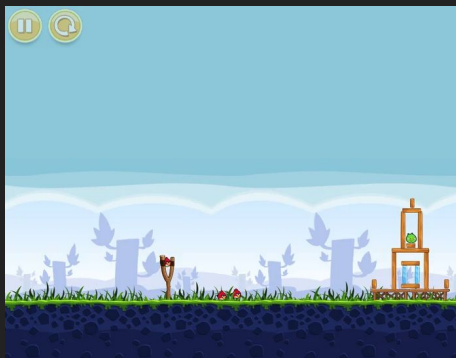
Chapter Select



Level Select



Cut Scene



Game Level



Win Screen

Game Mode

```
enum GameMode { MAIN_MENU, GAME_LEVEL, GAME_OVER };  
GameMode mode = MAIN_MENU;
```

Game Mode

```
void ProcessInput() {  
    switch (mode) {  
        case MAIN_MENU:  
            ProcessInputMainMenu();  
            break;  
  
        case GAME_LEVEL:  
            ProcessInputGameLevel();  
            break;  
  
        case GAME_OVER:  
            ProcessInputGameOver();  
            break;  
    }  
}
```


Game Mode

```
void Update() {  
    float ticks = (float)SDL_GetTicks() / 1000.0f;  
    float deltaTime = ticks - lastTicks;  
    lastTicks = ticks;  
  
    switch (mode) {  
        case MAIN_MENU:  
            UpdateMainMenu(deltaTime);  
            break;  
  
        case GAME_LEVEL:  
            UpdateGameLevel(deltaTime);  
            break;  
  
        case GAME_OVER:  
            UpdateGameOver(deltaTime);  
            break;  
    }  
}
```

Game Mode

```
void Render() {  
    glClear(GL_COLOR_BUFFER_BIT);  
  
    switch (mode) {  
        case MAIN_MENU:  
            RenderMainMenu();  
            break;  
  
        case GAME_LEVEL:  
            RenderGameLevel();  
            break;  
  
        case GAME_OVER:  
            RenderGameOver();  
            break;  
    }  
  
    SDL_GL_SwapWindow(displayWindow);  
}
```

Game Mode

(or make a class for each mode)

```
MainMenu mainMenu;
GameLevel gameLevel;
GameOver gameOver;

void Render() {
    switch (mode) {
        case GAME_LEVEL:
            gameLevel.Render();
            break;

        // .. other modes
    }
}
```

```
class GameLevel {
    GameState state;

    public void Render() {
        state.player.Render();
        for (int i = 0; i < enemies.length; i++) {
            state.enemies[i].Render();
        }
    }

    // Other stuff
}
```

Characteristics of Games

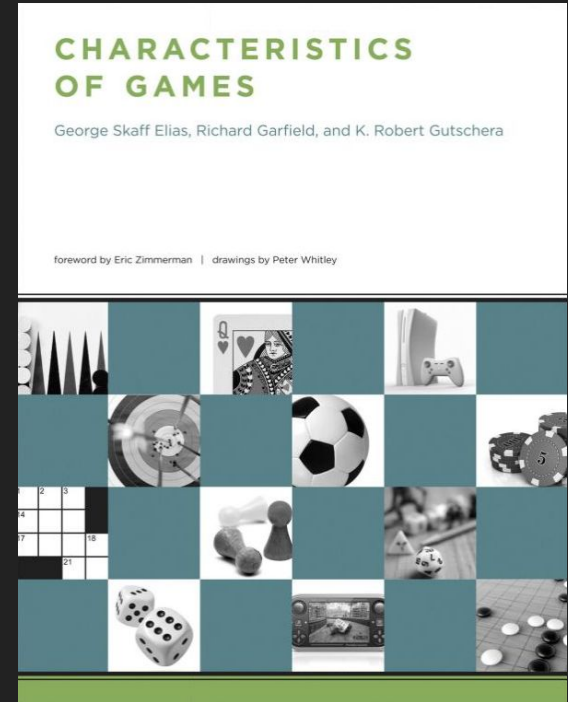
Characteristics of Games

Game Mechanics

Length of Playtime

Number of Players

Heuristics



Let's Talk About
Game Mechanics

Length of Playtime

Length of Playtime

Atom

Game

Match

Session

Campaign



Increasing
Duration

Atom



Game



Match



Session and Campaign



Session and Campaign



Number of Players

1 Player Games

Pure one-player games

Player against a system

(puzzle games, narrative games, tetris, asteroids)

One human player vs simulated opponents

Shares properties with 2-player and multiplayer games.

(StarCraft, Civilization, Street Fighter)

1 Player Games

Relatively uncommon before computers.

Uncertainty of outcome is at the heart of games!

Human opponents provide uncertainty.

Without them, uncertainty needs to come from somewhere else.

2 Player Games

Tennis, Chess, Battleship, Hearthstone, etc.

Some multiplayer games can reduce to be 2-player games.
(Scrabble, StarCraft)

Harder to upscale a 2 player game to a multiplayer game.
(Issues with politics and kingmaking)

Two player games are the iconic examples of games. Pure competitions.

Multiplayer Games

1-sided (players vs. environment)

2-sided (sports games, capture the flag, etc.)

Large multiplayer games (sort of multi-sided with 1 person on each side)

Scrabble, Monopoly, Risk

Deathmatch / Battle Royale

Can be difficult to design due to player interactivity and politics.

What's up with Battle Royale being a genre now?

Characteristics of Multiplayer Games

Races

Competition of essentially one-player games. Low interaction between players. (Clue, Trivia Pursuit, many other board games).

Brawls

Essentially 2-player games with more players. High interaction between players. (Risk, Battle Royale)

Characteristics of Multiplayer Games

Strict Elimination

Player is “out of the game” and no longer playing.

Perceived Elimination

Player has effectively lost the ability to compete or win.



Characteristics of Multiplayer Games

Interactivity Between Players

Ability to influence other player's progress (for better or worse).

Politics

Often a meta activity outside the rules of the game.

Kingmaking

Typically when player perceives they are eliminated, they do something to choose a player who wins.

Heuristics

Rules of Thumb
Strategies from Similar Situations
Educated Guess

Note: Not just humans, but AI also use Heuristics for searches/planning.

Heuristics

As players (especially new players) of a game, we need to know who is winning or losing and what to do next.

How do I get better at this game?

Heuristics

Positional Heuristics

What is the current state of the game - Who is winning and by how much?

Directional Heuristics

What strategy should I follow - Which move should I make next?

Heuristics

Players get better at games by learning more sophisticated heuristics for a game.

For some games and some players, the process of learning is the main appeal of the game.

For reflex games, getting better might mean improving your reflexes, but for games that do not require reflexes, getting better is based on improving heuristics.

Game Designers!

If your players find a heuristic which works and solves all situations, they will use it all the time!

(and not use other strategies you spent so much time designing for)

Heuristics

Should exist on all levels from beginner to advanced.

Some should be easy for players to discover on their own. Some should be more difficult.

Should be rich enough to cover most situations, so the player is never without guidance, but not so powerful as to completely solve all situations.

Should be satisfying, so the player feels like they are exercising judgement, not executing a computer program.

In-Class Exercise

Uno

(With New Mechanics)