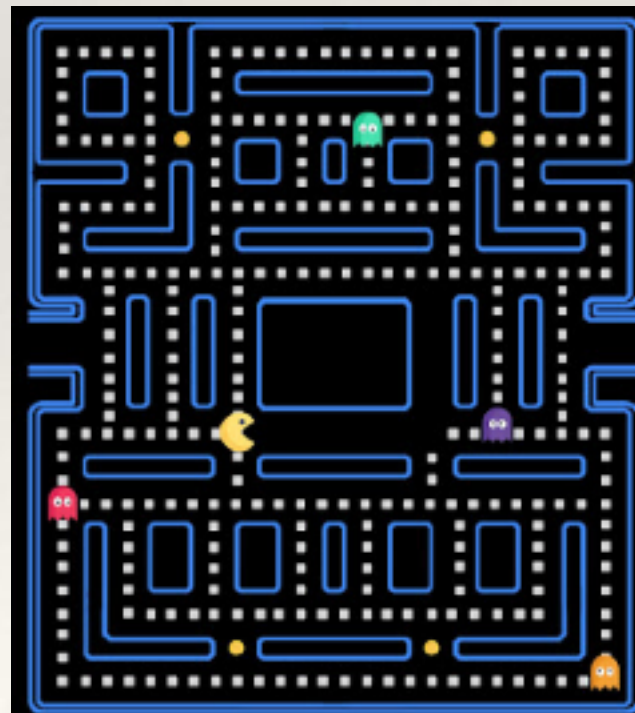


Klasa kao realizacija softverske mašine

- ❖ Pretpostavimo da pravimo softver za neku arkadnu igricu, u kojoj se različiti objekti, tzv. likovi, kreću uporedo po igračkom polju. Kako da implementiramo to nezavisno i uporedo kretanje koje se (makar prividno) dešava istovremeno?
- ❖ Ideja:
 - svaki lik (*character*) predstavimo objektom, po potrebi određene izvedene klase
 - svaki lik implementira operaciju koja izvodi jedan korak (*step*) svog kretanja; ova operacija treba da bude kratka i da izvrši jedan elementaran pomeraj, po što je moguće kraći
 - jedna glavna petlja “proziva” sve likove na igračkom polju i svakome daje da se pomeri za po jedan korak
- ❖ Na ovaj način možemo da stvorimo privid uporednog kretanja objekata kao *aktivnih* entiteta



Klasa kao realizacija softverske mašine

```
class Character {
public:
    ...
    virtual bool step ();
    ...
protected:
    virtual void calcMove (int& dx, int& dy);
    int myX, myY;
};

class Pacman : public Character {
public:
    ...
    virtual bool step ();
    ...
};

class Ghost : public Character {
public:
    ...
    virtual bool step ();
    ...
};

class Engine {
public:
    ...
    void run ();
    ...
private:
    list<Character*> myChars;
    ...
};
```

```
void EGINE::run () {
    while (!gameOver) {
        for (Character* c : this->myChars) {
            c->step();
            ...
        }
    }
}

bool Ghost::step () {
    int dx, dy;
    this->calcMove(dx,dy);
    if (dx!=0 || dy!=0) {
        theField->clearGhostSprite(myX,myY);
        this->myX += dx;
        this->myY += dy;
        theField->drawGhostSprite(myX,myY);
    }
    return true;
}

bool Pacman::step () {
    ...
}
```