Programiranje 2

Pokazivaci i funkcije

dr Đorđe Obradović

Singidunum - Centar Novi Sad

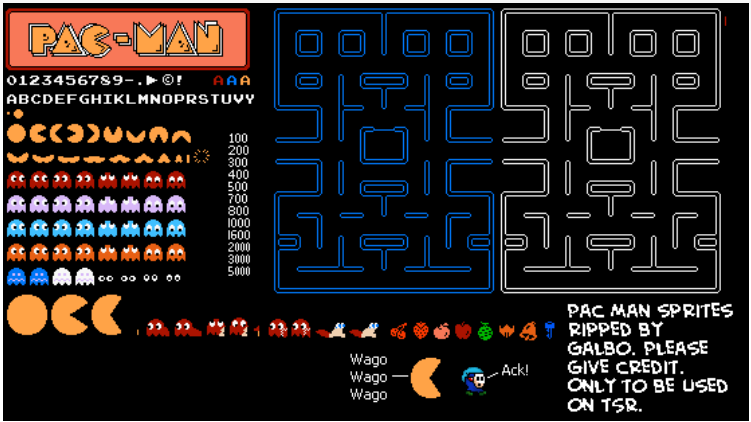
Igra 03 - Pacman

Uvod

U ovom primeru biće prikazana implementacija kretanja dva igrača.

Novost u odnosu na prethodne primere je poboljšan mehanizam za upravljanje događajima i bolja struktuiranost elemenata projekta.

Osnovni elementi



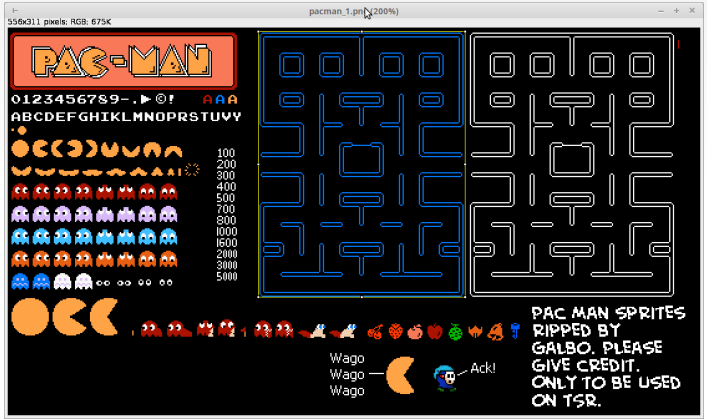
Mapa elemenata - pacman

w:14, h:14 | y: 90 | x: 3, 19, 33, 47, 61, 75, 92, 109, 126

Mapa elemenata - crveni duh

w:14, h:14 | y: 125 | x: 3, 20, 37, 54, 71, 88, 105, 122

Mapa elemenata - pozadina



w:165, h:214 | y: 2 | x: 201

Zadatak

Zadatak

Implementirati klase

- Sprite,
- Pozadina i
- Game

i povezati sa main funkcijom u jednu celinu.

klasa Sprite

Klasa Sprite koristi se za upravljanje jednom figurom.

klasa Pozadina

Klasa Pozadina koristi se za upravljanje pozadinom i prikazivanjem rezultata.

Klasa Game

U ovoj klasi se obrađuju svi događaji, raspoređuje upravljanje kretanjem i vrši prikazivanje svih komponenti igre.

Implementacija

Klasa Sprite

```
0     class Sprite{
1       public:
2         Sprite();
3
4         SDL_Texture* textura;
5         int dx, dy;
6         int x,y,w,h;
7         int sx, sy;
8         SDL_Rect boxes[8];
9
10        virtual void render(SDL_Renderer* gRenderer);
11        virtual void tick(int time);
12        virtual void left();
13        virtual void right();
14        virtual void up();
15        virtual void down();
16    };
```

Klasa Pozadina

```
0   class Pozadina:public Sprite{
1       public:
2           Pozadina(){
3               x = 640/2-165;
4               y = 480/2-214;
5               sx = 201;
6               sy = 2;
7               w = 165;
8               h = 214;
9           }
10
11          void render(SDL_Renderer* gRenderer){
12              SDL_Rect clip = { sx, sy, w, h }; // ovo su koordinate na slici
13              SDL_Rect renderQuad = { x, y, 2*w, 2*h }; // ovo su koordinate na ekranu
14              SDL_RenderCopy( gRenderer, textura, &clip, &renderQuad );
15          }
16  };
```

Klasa Game

```
0   class Game {
1       public:
2           Game();
3           ~Game();
4
5           bool loadFromFile(string, SDL_Renderer*);
6           void free();
7           void eventHandling(SDL_Event);
8           void tick(int);
9           void render(SDL_Renderer* gRenderer);
10          map<int, SpriteEventHandler*> event_listeners;
11
12      private:
13          Pozadina pozadina;
14          Sprite duhA;
15          Sprite duhB;
16          SDL_Texture* mTexture;
17          int mWidth;
18          int mHeight;
19  };
```

Event handling

```
0   void Game::eventHandling(SDL_Event event){
1       switch (event.type){
2               case SDL_KEYDOWN: //SDL_KEYUP:
3                       switch( event.key.keysym.sym ){
4                               case SDLK_LEFT:
5                       duhA.left();
6                                       break;
7                               case SDLK_RIGHT:
8                       duhA.right();
9                                       break;
10                              default:
11                                  break;
12                      }
13                      break;
14          }
15  }
```

Event handling - unapređen

Formirati mapu u kojoj će se nalaziti funkcije koje će se pozivati kao reakcija na događaj.

Event handling - unapređen

```
0   void Game::eventHandling(SDL_Event event){
1     switch (event.type){
2       case SDL_KEYDOWN: //SDL_KEYUP:
3         switch( event.key.keysym.sym ){
4           case SDLK_LEFT:
5             duhA.left();
6             break;
7           case SDLK_RIGHT:
8             duhA.right();
9             break;
10          default:
11            break;
12        }
13        if(event_listeners.count(event.key.keysym.sym)){
14          event_listeners[event.key.keysym.sym]->execute();
15        }
16        break;
17    }
18  }
```

Event handling - unapređen

```
0   typedef void (Sprite::*SpriteFunction)();
1
2   class SpriteEventHandler{
3     public:
4       int code;
5       Sprite* obj;
6       SpriteFunction function;
7
8       SpriteEventHandler(){};
9       SpriteEventHandler(int aCode, Sprite* aObj, SpriteFunction aFunction){
10        code = aCode;
11        obj = aObj;
12        function = aFunction;
13      };
14
15      void execute(){
16        (obj->*function)();
17      };
18  };
```

Event handling - unapređen

```
0
1    event_listeners[115] = new SpriteEventHandler(115, &duhB, &Sprite::right); // s
2    event_listeners[119] = new SpriteEventHandler(119, &duhB, &Sprite::up); // w
3    event_listeners[122] = new SpriteEventHandler(122, &duhB, &Sprite::down); // z
```

Literatura

Literatura

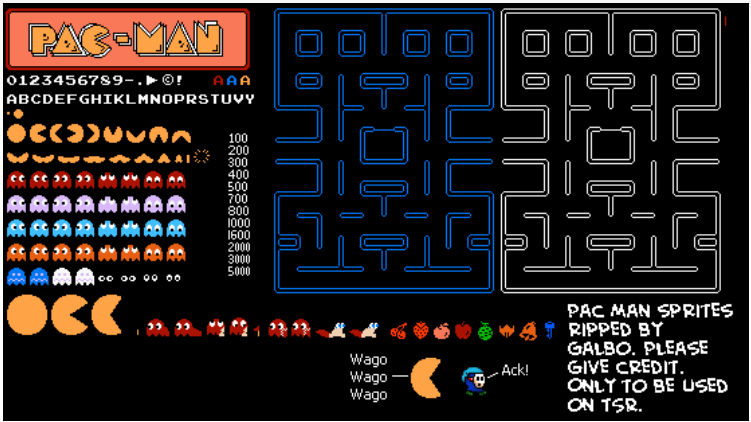
- https://www.libsdl.org/
- https://en.wikipedia.org/wiki/Simple_DirectMedia_Layer
- http://lazyfoo.net/tutorials/SDL/

Domaći

Zadaci za vežbu

1. Napraviti mapu slika za narandžaste duhove.
2. Implementirati funkcije za proveru mogućeg kretanja duhova i igrača.
3. Nacrtati tačkice za bodove.
4. Ispisati labelu za rezultat i ispod nje svoje ime.
5. Omogućiti duhovima kretanje.

Pakmen


PAC-MAN
0123456789-.▶©!
ABCDEFGHIKLMNOPRSTUVY
100
200
300
400
500
700
800
1000
1600
2000
3000
5000
PAC MAN SPRITES RIPPED BY GALBO. PLEASE GIVE CREDIT. ONLY TO BE USED ON TSR.
Wago
Wago
Wago
Ack!