Programiranje 2

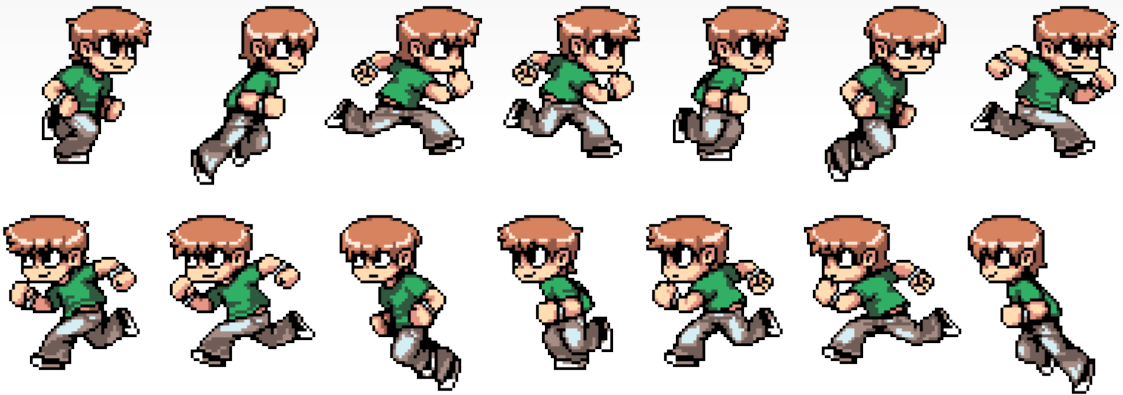
Kontruktor, destruktor indirektno adresiranje

dr Đorđe Obradović

Singidunum - Centar Novi Sad

Igra 02 - Animacija

Simulacija kretanja



Opis i cilj demonstracije

Demonstrirati tehnike potrebne za implementaciju kretanja pomoću SDL biblioteke

Implementacija

Biblioteke

```
0  #include <SDL2/SDL.h>
1  #include <SDL2/SDL_image.h>
2  #include <iostream>
3  #include <sstream>
4  #include <stdio.h>
5  #include <string>
6  using namespace std;
```

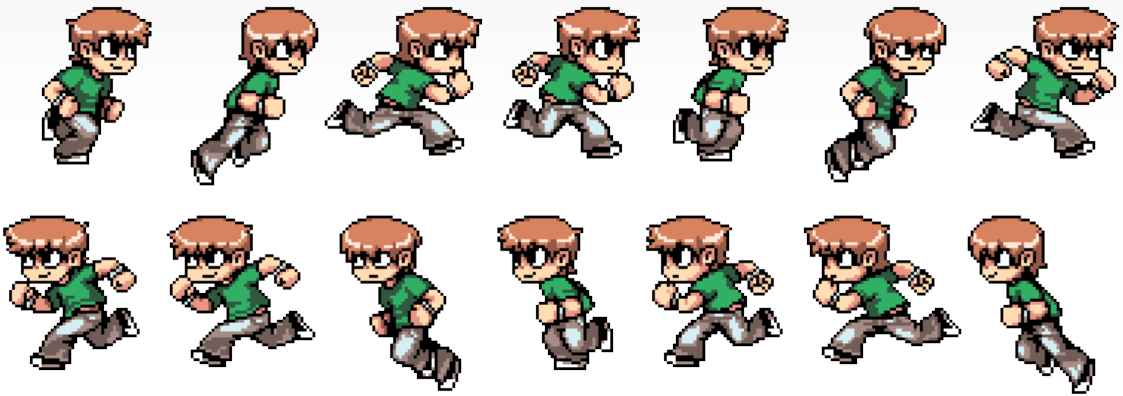
Main funkcija

```
0   int main(int argc, char ** argv) {
1       bool quit = false;
2       SDL_Event event;
3
4       // init SDL ===============================================
5       init();
6       loadMedia();
7
8       // main loop ==============================================
9
10      // cleanup SDL ============================================
11      close();
12      return 0;
13  }
```

Main loop

```
0  int dx = 1;
1  int frame = 0;
2  int x = 0;
3
4  // main loop
5  while (!quit){
6      SDL_Delay(10);
7      //------ event handling ----------
8      //------ render frame ---------
9  }
```

Render frame



Render frame

```
0  SDL_SetRenderDrawColor(gRenderer, 100, 100, 100, 255);
1  SDL_RenderClear(gRenderer);
2
3  int iClip = frame / 8;
4  if(dx<0)
5    iClip = 8+frame / 8;
6  SDL_Rect* currentClip = &gSpriteClips[ iClip ];
7  gSpriteSheetTexture.render(x, 100, currentClip );
8
9  ++frame;
10 if( frame / 8 >= WALKING_ANIMATION_FRAMES ){
11   frame = 0;
12 }
13 SDL_RenderPresent(gRenderer);
```

Event handling

```
0   switch (event.type){
1       case SDL_QUIT:
2           quit = true;
3           break;
4       case SDL_KEYDOWN: //SDL_KEYUP:
5           switch( event.key.keysym.sym ){
6               case SDLK_LEFT:
7                   dx = -1;
8                   break;
9               case SDLK_RIGHT:
10                  dx = 1;
11                  break;
12              case SDLK_ESCAPE:
13                  quit = true;
14                  break;
15              default:
16                  break;
17          }
18          break;
19    }
20  // move objects
21  x = x + dx;
22  if(x<0)
23    dx = 1;
24
25  if(x>640)
26    dx = -1;
```

Igrac

Klasa - H

```
0  class Igrac {
1      public:
2          Igrac();
3          ~Igrac();
4
5          bool loadFromFile(string path );
6          void free();
7          void render( int x, int y, SDL_Rect* clip = NULL );
8
9      private:
10         SDL_Texture* mTexture;
11
12         int mWidth;
13         int mHeight;
14 };
```

Igrac - CPP Kontruktor

```
0  Igrac::Igrac()
1  {
2      mTexture = NULL;
3      mWidth = 0;
4      mHeight = 0;
5  }
```

Igrac - CPP Destruktor

```
0  Igrac::~Igrac()
1  {
2      free();
3  }
```

Igrac - CPP loadFromFile

```
0   bool Igrac::loadFromFile( std::string path ){
1     free();
2     SDL_Texture* newTexture = NULL;
3     SDL_Surface* loadedSurface = IMG_Load( path.c_str() );
4     SDL_SetColorKey( loadedSurface, SDL_TRUE, SDL_MapRGB( loadedSurface->format, 0, 0xFF, 0xFF ) );
5     newTexture = SDL_CreateTextureFromSurface( gRenderer, loadedSurface );
6     mWidth = loadedSurface->w;
7     mHeight = loadedSurface->h;
8     SDL_FreeSurface( loadedSurface );
9     mTexture = newTexture;
10    return mTexture != NULL;
11  }
```

Globalne funkcije i promenljive

```
0  void init(){
1    SDL_Init(SDL_INIT_VIDEO);
2    gWindow = SDL_CreateWindow("Programiranje 2 - Singidunum",
3        SDL_WINDOWPOS_UNDEFINED, SDL_WINDOWPOS_UNDEFINED, 640, 480, 0);
4    gRenderer = SDL_CreateRenderer(gWindow, -1, 0);
5
6  }
```

Globalne funkcije i promenljive

```
0  void loadMedia(){
1      gSpriteSheetTexture.loadFromFile( "boy.png" );
2    for(int i=0; i<8; i++){
3      gSpriteClips[ i ].x =  i*108;
4          gSpriteClips[ i ].y =   0;
5          gSpriteClips[ i ].w = 108;
6          gSpriteClips[ i ].h = 140;
7    }
8    for(int i=0; i<8; i++){
9      gSpriteClips[ 8+i ].x =  i*108;
10         gSpriteClips[ 8+i ].y =   140;
11         gSpriteClips[ 8+i ].w = 108;
12         gSpriteClips[ 8+i ].h = 140;
13   }
14 }
```

Literatura

Literatura

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

Korisno Ubuntu

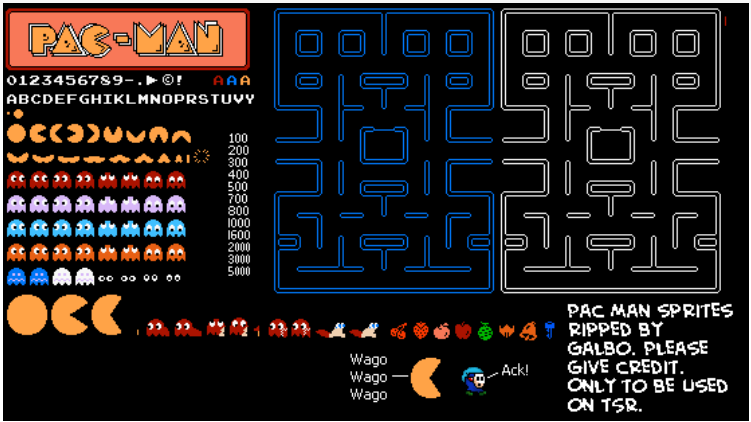
```
sudo apt-get install libsdl2-dev
sudo apt-get install libsdl2-image-2.0-0
sudo apt-get install libsdl2-image-dev
sudo apt-get install libsdl2-image-dbg
sudo apt-get install libsdl2-ttf-dev
```

Domaći

Zadaci za vežbu

1. Nacrtati dve horizotalne linije zelenu ispod igrača i plavu iznad
2. Odovjiti delove koda u logicne celine (h, cpp, main)
3. Omoguciti igracu da se zaustavi
4. Implementirati skok igrača (bez animacije)
5. Ubaciti belu loptu koja se nalazi ispred igrača
6. Implementirati kretanje pakmen igrača korišćenjem slike

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!