

Game Development

Subject Intro



Contents

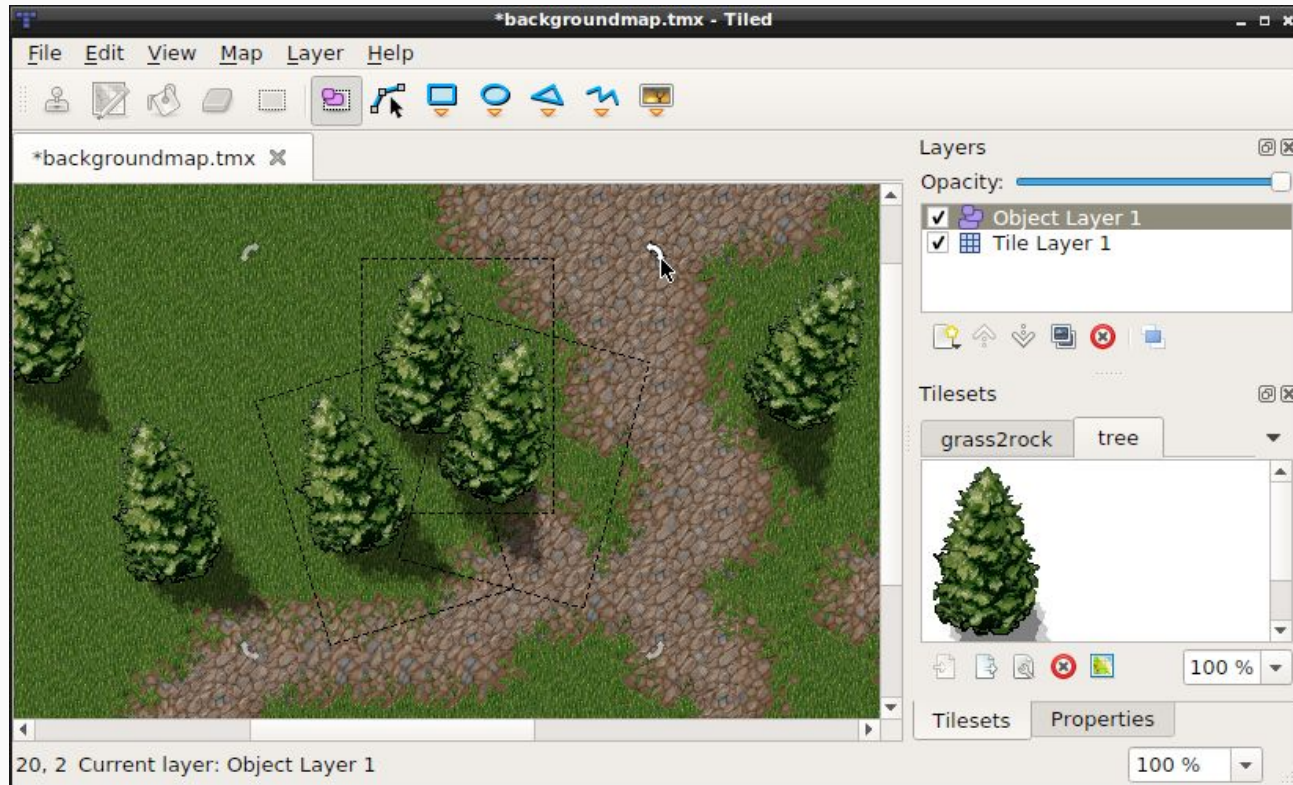
- Learning goals
- What we will learn
- Evaluation
- Rules
- Tools
- Observations



Learning Goals

- Become proficient with C++
- Understand the building blocks of video games
- Get into data driven programming
- Understand isometric perspective
- Develop debug tools
- First glimpse into GUI programming

Proper data read and write



What we will learn

- Data Formats (XML)
- Load / Save
- Loading maps

Map isometric rendering in [Diablo 2](#)



What we will learn

- Orthographic vs. Isometric
- Metadata information on maps
- Pathfinding!

Max Payne



What we will learn

- FPS control
- Game Logic Time control
- Entity Management

GUI - WoW



What we will learn

- In Game Graphical User Interfaces:
 - Labels
 - Buttons
 - Text Input
 - Windows
 - Scrollbars

Quake style console || Crysis console

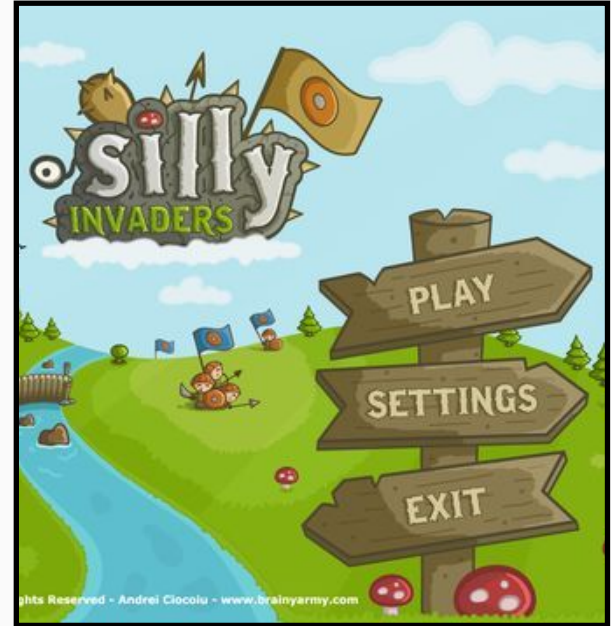
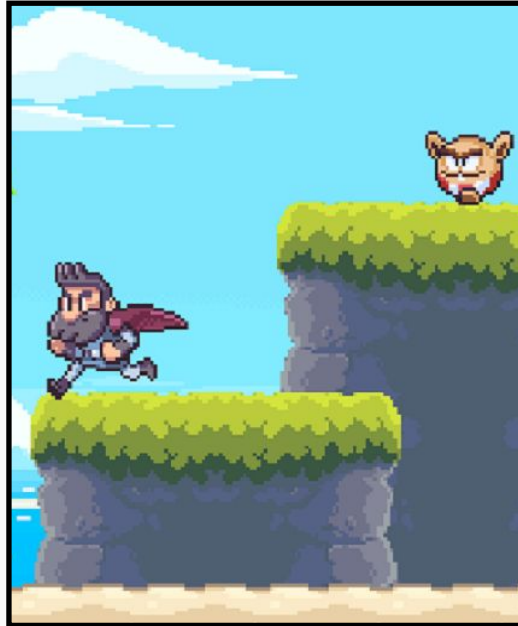
```
CGF streaming: Loaded:2352 InProg:0 All:2352 Act:2352 MemUsed:131.50 MemReq:0.00 PoolSize:24
TexStreaming: MemUsed:698.01MB(109%) PoolSize:640MB Trghput:0KB/s
  r_GetScreenShot = 2 []
Screenshot: c:\users\admin\saved games\crysis3\ScreenShots\ScreenShot0019.jpg
Drawcalls: 0
FPS: 16.4 (40.5 ms)
Tris: 0,000
CGF streaming: Loaded:2352 InProg:0 All:2352 Act:2352 MemUsed:131.50 MemReq:0.00 PoolSize:24
TexStreaming: MemUsed:698.01MB(109%) PoolSize:640MB Trghput:0KB/s
  r_GetScreenShot = 2 []
Screenshot: c:\users\admin\saved games\crysis3\ScreenShots\ScreenShot0020.jpg
Drawcalls: 0
FPS: 29.1 (33.4 ms)
Tris: 0,000
CGF streaming: Loaded:2352 InProg:0 All:2352 Act:2352 MemUsed:131.50 MemReq:0.00 PoolSize:24
TexStreaming: MemUsed:720.51MB(112%) PoolSize:640MB Trghput:0KB/s
hud_hide 1
hud_hide = 1 []
>hud_hide _
```



What we will learn

- Performance analysis (Profiler)
- Development console
- Cvars for configuration

We will build a platformer: Map + A* + UI



Worth mention "*Parallax Paradox*" from 2018



Marc & Pol project from 2018 ended up winning some contest !



Evaluation

First Assignment:

- Counts as **15%**
- Before October 20th 23:59
- *Save/Load*
- *XML Parsing*
- *Map Rendering*
- *Collision and Logic import*

Second Assignment:

- Counts as **15%**
- Before November 17th 23:59
- *Pathfinding*
- *Time control*
- *Performance monitoring*

Evaluation

Third Assignment:

- Counts as **20%**
- Before December 15th 23:59
- *All GUI*

Game Innovation:

- Counts as **10%**

Final Exam (theory):

- Counts as **40%**
- January 8th - 17th

Revaluation exam:

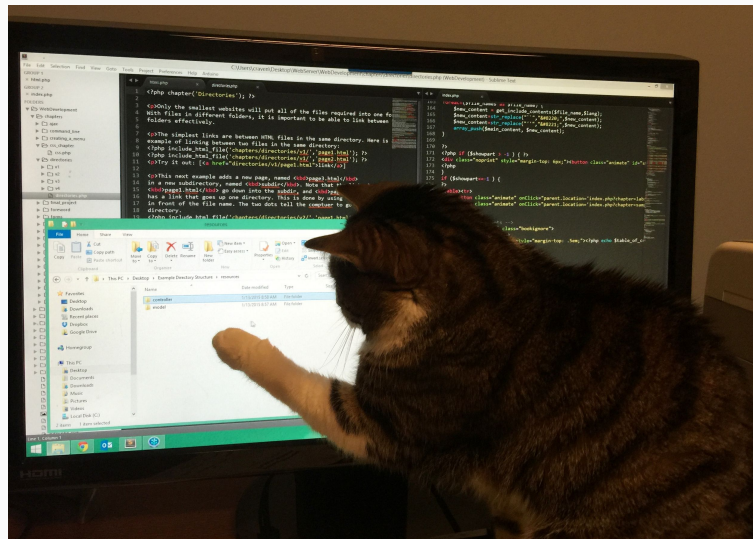
- Counts as **40%**
- Max grade is 5
- January 30th - February 5th

Rules

- Work is **done in groups of two** - spreadsheet [here](#)
 - Not to be changed later but in extreme circumstances
 - Each github user should be used only by its own student!
- Code is expected to be:
 - **Clear**
 - **Consistent**
 - **Optimal**
 - **Original:** only once can trigger a zero for the whole subject

Rules

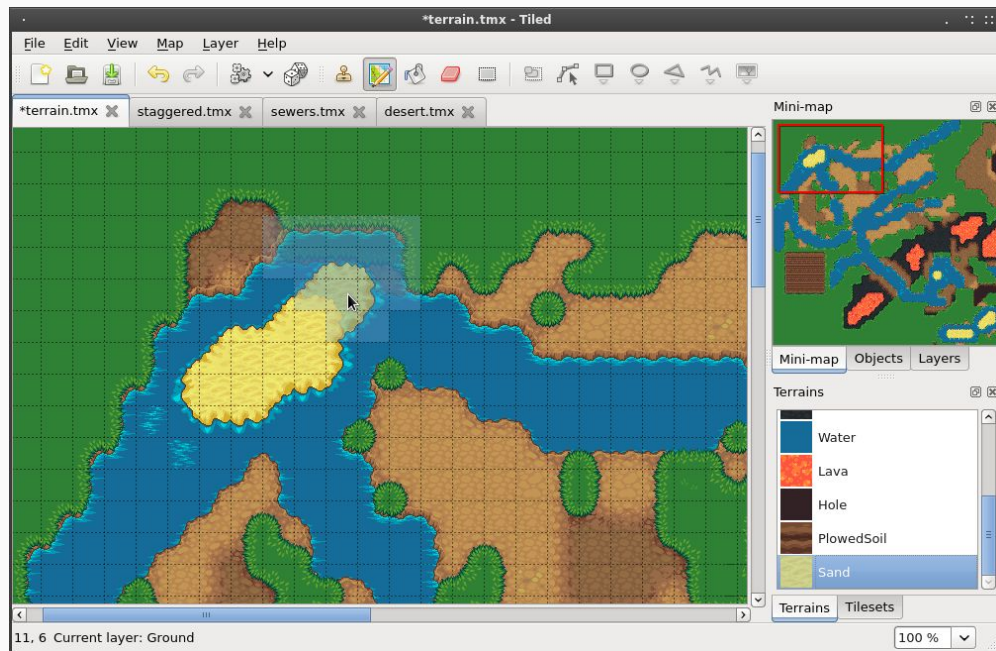
- For the 10% grade for innovation on assignments:
 - Go beyond what is requested
 - Do some interesting **tech**
 - It is not an easy grade!
- Final exam will be done on paper:
 - It is done individually
 - Can be revaluated
 - No teamwork :)



Tools

We will build on last year code structure:

- Visual Studio 2017
- Github.com
- [Tiled](#)
- Brofiler



Observations

- We will learn coding *for* video games
- Everything is about spending time coding
- You will start understanding the games you play
- All those building blocks will be used on **Project II** Subject next semester
- Have fun! :)
- <https://www.youtube.com/watch?v=nxtMnaDp6M4>

Homework

Let's test our C++ skills: Create a new vec3 class with

- Contains three values x,y,z with templated type
- Think of at least three handy constructors
- Operators +, -, +=, -=, =, ==
- Methods: normalize(), zero(), is_zero(), distance_to(vec3 ...)
- Mind references and const!

Upload it to your github and it will be reviewed next class.

Start forming groups [here!](#) :)