ATET Document

HOW TO BUILD AND RUN

Once you open the project you should build it from **Debug** Once the build is finished, you can start the project to find the OpenGL window. Please build only in debug for now. There is no restart in the game implemented as of now so to restart you have to exit the application and run it again.

CLASSES AND FUNCTIONS

Json Class

This class deals with Read and writing player data. This class consists of Two major functions **ReadPlayerData(args..) and WritePlayerData(args..)**. These functions are the key to reading the player data and writing the player data.

Player Controller Class

This class deals with player skeletal animation, animation states and player movement. The major function for this class is the **AddState(args..) and Update(args..) override.** These functions are used to add state to the player like idle walk and much more. Update is the entity update inherited from entitymanager which updates all the animations. Skeletal animation update is also called in this function for the bone movements.

State Machine Pattern

As discussed in the UML diagram, every state has its own class which is inherited from a basestate class, states which deal with axis change for the player also inherit the inputlistener class which deals with player inputs. We've added a death state in our state machine for player death.

List Of States Used

AxisChangeState Class

This class is used to change the axis for the player when he collides with the axis changer.

CollisionState Class

This class handles all the collision responses for the player.

DeathState Class

This class deals with player death when he collides with traps or any other obstacle that damages the player.

IdleState Class

This class deals with a player when he is idle without any input.

RunState Class

This class deals with the player when he is running with inputs.

As mentioned earlier all these states inherit a BaseState class which has all the necessary virtual functions.

Camera Controller Class

This is the side scroller camera controller class which inherits from entity class for all the necessary functions to update. This class also deals with our main mechanic, the camera axis changing. The **HandlePosition(args..) and HandleRotation(args..)** deals with the position and rotation of the camera.

Scene_1 Class

This class deals with initialising all the objects for this demo scene. Consider it as demo level 1 in which all objects are placed. This class inherits from basescene class which has all the functions to start and update and much more. This class also contains the WorldObjectFactory class to spawn all the objects in the scene using factory pattern. All the objects are Created in the **Start()** function

WorldObjectsFactory

This folder contains all the object classes that need to be spawned into the world. It has two base classes which are **BaseAnimatedObjectClass and BaseWorldObjectClass**.The **SpikeTrap class** which has animation inherits the BaseAnimatedObjectClass for the object animation and the object associated functions like start and update. The **Axis Changer Class** and **Floor Class** inherits the BaseWorldObjectClass for spawning and all the necessary functions associated with the respective classes. Finally the **WorldObjectFactory Class** is the **factory** that creates these objects and spawns it in the world. You can find **CreateFloor(args..), CreateAxisChanger() and CreateTrap()** functions inside this class that creates the objects as the name suggests.

Atet_Application Class

This class is the application setup class that is called in the main to initialise the OpenGL window the scene loaded with all the objects and player and camera.

POST BUILD SETUP

```
<PropertyGroup>
  <outputFile>$(OutDir)</outputFile>
  <JsonConfig> $(OutDir)Assets\JsonConfig</JsonConfig>
  <PlayerFile> $(OutDir)Assets\Models\Player\</PlayerFile>
  <TrapFile> $(OutDir)Assets\Models\Traps\</TrapFile>
  <AnimationFile> $(OutDir)Assets\Animation\</AnimationFile>
  <PlayerTextureFile> $(OutDir)Assets\Models\Player\Player.fbm\</PlayerTextureFile>
  <ModelFile> $(OutDir)res\Models\</ModelFile>
  <ShaderFile> $(OutDir)res\Shader\</ShaderFile>
  <PostprocessingFile> $(OutDir)res\Shader\PostProcessing\</PostprocessingFile>
  <DefaultTextureFiles> $(OutDir)res\Textures\DefaultTextures\</DefaultTextureFiles>
  <IconsTextureFiles> $(OutDir)res\Textures\Icons\</IconsTextureFiles>
  <SkyBoxTextureFiles> $(OutDir)res\Textures\SkyBox\</SkyBoxTextureFiles>
</PropertyGroup>


<ItemGroup>
  <JsonConfigCopyFile Include="Assets\JsonConfig\**" />
  <PlayerModelCopy Include="Assets\Models\Player\**" Exclude="Assets\Models\Player\Player.fbm\**" />
  <TrapFileCopy Include="Assets\Models\Traps\**"/>
  <AnimationFileCopy Include="Assets\Animation\**"/>
  <PlayerModeltexturesCopy Include="Assets\Models\Player\Player.fbm\**" />
  <ModelFilesCopy Include="res\Models\**" />
  <ShaderFileCopy Include="res\Shader\**" Exclude="res\Shader\PostProcessing\**" />
  <PostprocessingFilesCopy Include="res\Shader\PostProcessing\**" />
  <DefaultTextureFilesCopy Include="res\Textures\DefaultTextures\**" />
  <IconTextureFilesCopy Include="res\Textures\Icons\**" />
  <SkyboxFilesCopy Include="res\Textures\SkyBox\**" />
  <assimpDLLCopyFile Include="assimp-vc143-mtd.dll" />
  <imGuiDefault Include="imgui.ini" />
</ItemGroup>
```

```
<Target Name="AtetBuild">
  <MakeDir Directories="$(outputFile)" />
  <MakeDir Directories="$(JsonConfig)" />
  <MakeDir Directories="$(PlayerFile)" />
  <MakeDir Directories="$(PlayerTextureFile)" />
  <MakeDir Directories="$(ModelFile)" />
  <MakeDir Directories="$(ShaderFile)" />
  <MakeDir Directories="$(PostprocessingFile)" />
  <MakeDir Directories="$(DefaultTextureFiles)" />
  <MakeDir Directories="$(IconsTextureFiles)" />
  <MakeDir Directories="$(SkyBoxTextureFiles)" />
  <MakeDir Directories="$(TrapFile)" />
  <MakeDir Directories="$(AnimationFile)" />

  <Copy SourceFiles="@(JsonConfigCopyFile)" DestinationFolder="$(JsonConfig)" />
  <Copy SourceFiles="@(PlayerModelCopy)" DestinationFolder="$(PlayerFile)" />
  <Copy SourceFiles="@(TrapFileCopy)" DestinationFolder="$(TrapFile)" />
  <Copy SourceFiles="@(AnimationFileCopy)" DestinationFolder="$(AnimationFile)" />

  <Copy SourceFiles="@(PlayerModeltexturesCopy)" DestinationFolder="$(PlayerTextureFile)" />
  <Copy SourceFiles="@(ModelFilesCopy)" DestinationFolder="$(ModelFile)" />
  <Copy SourceFiles="@(ShaderFileCopy)" DestinationFolder="$(ShaderFile)" />
  <Copy SourceFiles="@(PostprocessingFilesCopy)" DestinationFolder="$(PostprocessingFile)" />
  <Copy SourceFiles="@(DefaultTextureFilesCopy)" DestinationFolder="$(DefaultTextureFiles)" />
  <Copy SourceFiles="@(IconTextureFilesCopy)" DestinationFolder="$(IconsTextureFiles)" />
  <Copy SourceFiles="@(SkyboxFilesCopy)" DestinationFolder="$(SkyBoxTextureFiles)" />
  <Copy SourceFiles="@(assimpDLLCopyFile)" DestinationFolder="$(OutDir)" />
  <Copy SourceFiles="@(imGuiDefault)" DestinationFolder="$(OutDir)" />
</Target>
```

```
<Target Name="AfterBuild" DependsOnTargets="AtetBuild" Condition="'$(Configuration)'=='Debug'">

   <Message Text="Atet Application Build Successfully" />
 <!--All lossless pngs-->
   <ItemGroup>
      <PngFiles Include="$(OutDir)\**\*.png" />
   </ItemGroup>

     <!-- Lossy Compression with pngquant -->
   <Exec Command='"$(ProjectDir)pngquant.exe" --force --ext .png --quality=70-80 --skip-if-larger "%(PngFiles.Identity)"'
         Condition="Exists('%(PngFiles.Identity)')"
         ContinueOnError="false" ConsoleToMSBuild="false" />

   <!-- Lossless Compression with optipng -->
   <Exec Command='"$(ProjectDir)optipng.exe" -force "%(PngFiles.Identity)"'
         Condition="Exists('%(PngFiles.Identity)')"
         ContinueOnError="false" />

    <!-- Convert to PVR with PVRTexToolCLI -->
    <Exec Command='"$(ProjectDir)PVRTexToolCLI.exe" -f PVRTCI_4BPP_RGBA -i "%(PngFiles.Identity)" -o "$(OutDir)\%(RecursiveDir)%(Filename).pvr" -q pvrtcfast'
         Condition="Exists('%(PngFiles.Identity)')"
         ContinueOnError="true" />

</Target>
```

Added post build setup in visual studio as you can see in the screenshots i've attached above. Added Lossy Compression, Lossless Compression and Compression of all our PNG files. You can find the comments of where it is happening in the last picture.

Note : all questions attended.

Team Members,

Surya Prakash Ramu
Mohamed Ibrahim Sheik Dawood Salahudeen
Uphendhra Nagarajan