**Muniffic**

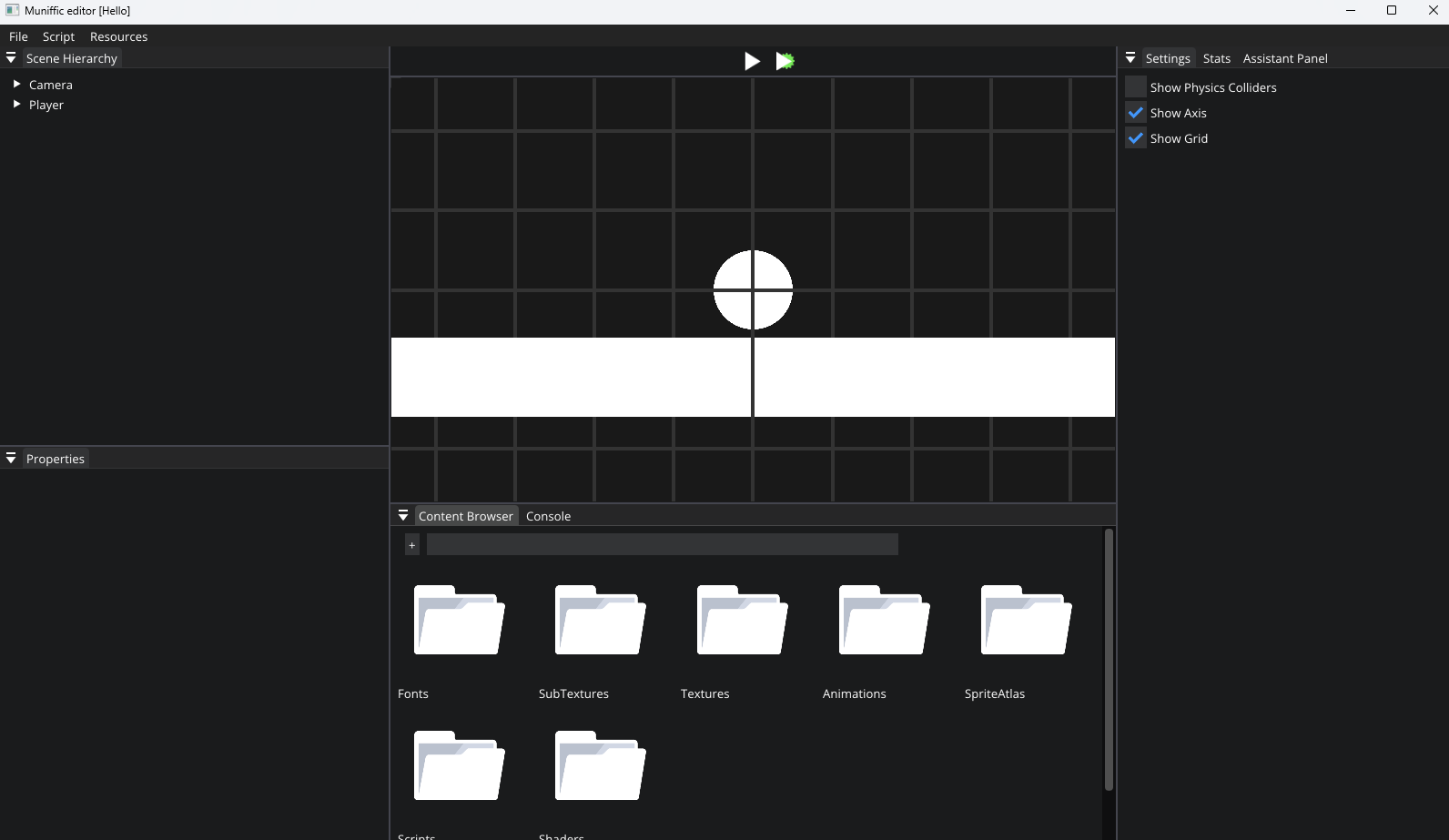
1. **Introduction**

Muniffic is innovative game engine integrating Artificial Intelligence to help users create games faster and more easily. Using Muniffic users can develop 2D games with help of engine’s comprehensive tools and features. Muniffic aims to enable users to enter new era of game development in which they can use AI assistant to speed up and make their work easier. Assistant is able to control the engine for the user, it can interpret user’s messages to create entities, modify components, write C# scripts and many more. The assistant can also help them learn and better understand functionalities of the engine. Combining all these features AI Assistant can make game development accessible for people with disabilities (e.g. paralysis) and people who are unable to properly use mouse and/or keyboard. These people can use the assistant to control the engine by text.

1. **Getting Started**

Games consist of scenes containing many different entities set up with various components defining its look and behavior and are controlled by C# scripts written by user.

* **Basic overview**



1. **Toolbar**

* File  
  Allows to open project, create new scene, save scene, save scene as and exit the application.
* Script  
  Allows to reload user’s scripts during runtime. It can be used to hot reload any script changes made by user.
* Resources  
  Provides various options to load resources and create new ones. It can be used to load images, text files and also create animations.

1. **Scene hierarchy**

Scene hierarchy panel shows all entities in the scene. It supports hierarchical entity structure – allows to add, move and remove children. By right clicking user can add entity, remove entity or directly add children entity.

1. **Properties**

Properties panel is the place for managing everything that defines the entity – here user can modify every component. It allows to add new components to currently selected entity and displays every component already added to it. On the top there is always input for entity name. First component is the default Transform Component which defines placement of the entity. Below every component and its properties is displayed. Each component has additional options menu which is shown by clicking plus button. Here user can remove the component and specify its settings related to hierarchical entity structure. User can specify options like:

* Inherit component in children – makes all children entities containing this component inherit component’s values from parent.
* Copy component to children – copies the component to all children entities with default values.
* Remove component from children – removes the component from all children entities.
* Copy values to children – copies values of the component to all children entities containing this component.
* Copy component with values to children – copies the component to all children entities with its values.

1. **Preview/Runtime**

This panel shows preview of the scene created by user. It has two buttons on top of it – play and simulate. Play button starts runtime scene and can be used to test current state of the game. On the other hand simulate button starts only physics simulation but no scripts are being executed. It can be used to check if any of the scripts are interfering with the game and are causing any problems. While running the scene user can pause runtime by clicking pause button.

1. **Content browser panel**

Content browser panel shows all assets added to the project. Every asset category has its own directory. User can also search for assets by their name using search bar on top of the panel.

1. **Console**

Console panel shows all logs from inside the engine (e.g. scene serialization progress, renderer initialization etc.) and from the scripts – script engine allows user to log important messages with DebugConsole class and specify message severity.