Reflection time – AssetStore

DOCUMENTATION

Online documentation: classes and its method, properties, variables you can get there https://goo.gl/BG3UfW

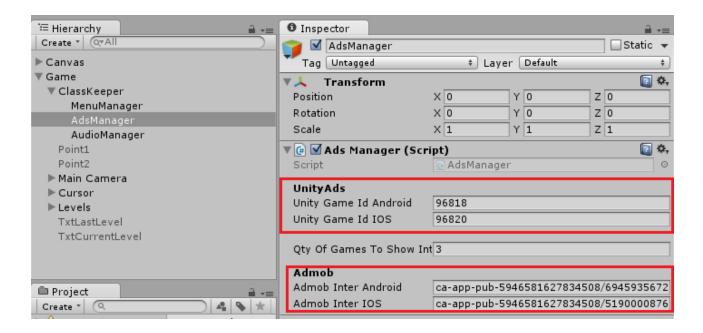
Class List

Here are the classes, structs, unions and interfaces with brief descriptions:

	Admoblmplement	Script, where admob interstitial ad is realized
	AdsManager	Class in charge to ads showing
	AudioManager	Class in charge to play and control all sound in the game
	BallBehaviour	In charge to moved the ball and control his behavior during the game
	Constants	Class holds constant variables
	Cursor	Control cursor physics
	ExtensionMethods	Class, that contains all of extension methods
	C Figure	Contain Figure properties and components
*	GameManager	Class that references a singleton object and consist of global variables and properties. Control level loading and actions, that are related with obstacles
	☑ InputManager	Class in charge of the input on different platforms.
	C Level	Contain level properties
	MenuManager	Class that references a singleton object and in charge to show and control menu gameobjects
	C LevelNumb	Struct, that contain properties of level number in scrollview
	MoverRotator	In charge to move and rotate gameobject
	Obstacles	Control obstacles behaviour
¥	PersistentSingleton	Persistent singleton.
	PlayerPrefsX	Store data
	C StrokeBehaviour	Control stoke behavior
	© Stroke	Contain component reference to stroke gameobject
	C Tools	Class contains static methods,that used in game
	UnityAdsImplement	In charge of UnityAds video realization

Ads Integration

Visit <u>admob, unityAds</u> sites to define your ID's and set them in *AdsManager* scripts, which is attached to *AdsManager* gameobject



Interstitial settings

qtyOfGamesToShowInter – value of frequency. Set how many interstitial ad will be after passed level.

For example, default *qtyOfGamesToShowInter* value show, that interstitial ad will be appear every 3-rd game.

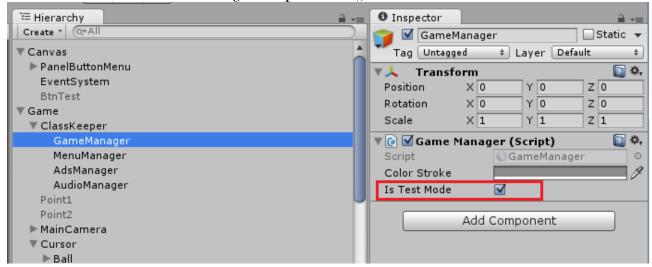
UnityAds settings

Set name of your integration id – *defaultZone*. Like on the picture

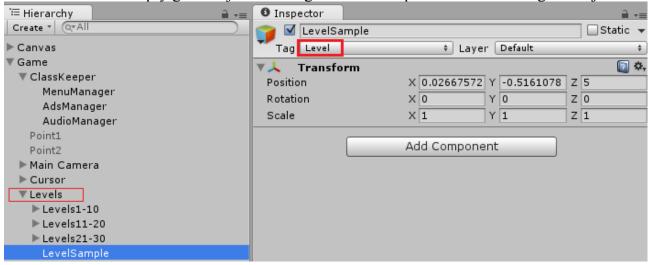


Customizing levels

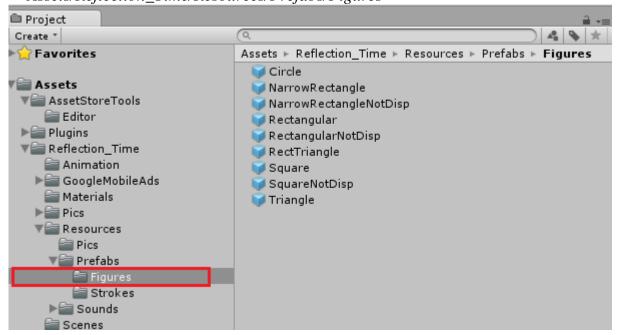
• For testing your level, **turn on test mode** in *GameManager* script. If "*test mode*" is off – the latest available level is load Find detail in *GameManager* script in *Start()* method



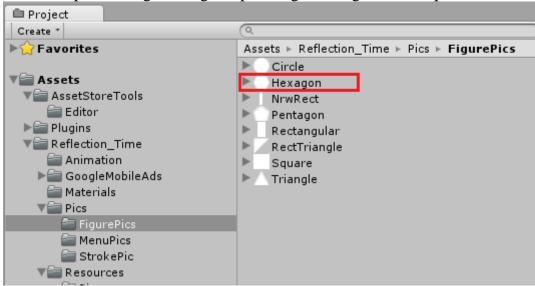
• Create empty gameobject with tag "Level" and place under Levels gameobject



• For building levels you can use figures' prefabs that are located in Assets/Reflection_Time/Resources/Prefabs/Figures



• For implementing new figure: paste figure image with unique name among other figures



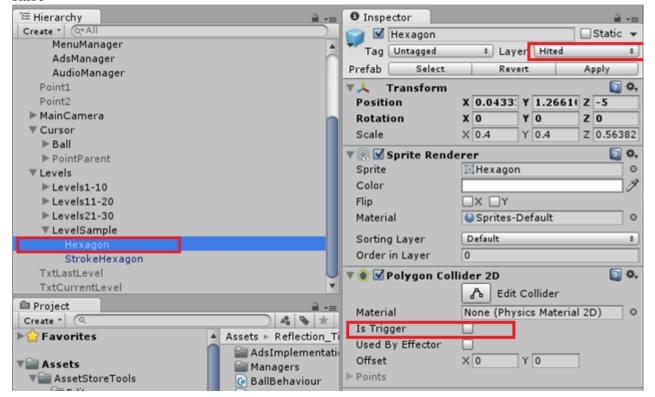
Hexagon stroke's name **must** consist with prefix of figure's name (in this case – "Hexagon") and annex "Stroke"

It made because, program need to know what kind of stroke is going to appear after hit. You can look code realization in script *StrokeBehaviour*. Method *StrokeReact()* called from script *Obstacles*

```
StrokeBehaviour.cs + X BallBehaviour.cs
C# Assembly-CSharp
                                                              StrokeBehaviour
                                                                                                                           public void StrokeReact(Figure hitedFig)
    42
    43
                   foreach (Stroke stroke in strokes)
    44
    45
                      if (!stroke.GameObj.activeInHierarchy && stroke.SpriteRen.sprite.name.StartsWith(hitedFig.SpriteRen.sprite.name)
    46
    47
                          stroke.Trans.position = hitedFig.Trans.position;
    48
                          stroke.Trans.localRotation = hitedFig.Trans.localRotation;
    49
    50
                          stroke.Trans.localScale = hitedFig.Trans.localScale;
                          stroke.GameObj.SetActive(true);
    51
    52
                          float time = 0.5F;
                          mono.StartCoroutine(Tools.MakeTransparent(stroke.SpriteRen, 0, time));
    53
                          mono.StartCoroutine(Tools.ChangeScale(stroke.Trans, hitedFig.Trans.localScale * 1.5F, time));
    54
    55
                          break:
    56
                      }
                  }
    58
     Obstacles.cs + X
                         StrokeBehaviour.cs
                                                     BallBehaviour.cs
                                                                                       🔩 Obstacles
     C# Assembly-CSharp
         228
                        void FigureDisappear(Figure hitedFig)
         229
         230
         231
                              AudioManager.Instance.LadderPlay(hitsQTY-1);
         232
                             hitedFig.Collider.isTrigger = true;
         233
                             mono.StartCoroutine(Tools.MakeTransparent(hitedFig.SpriteRen, 0, 0.1F));
         234
                              strokeB.StrokeReact(hitedFig);
         235
```

Save prefabs

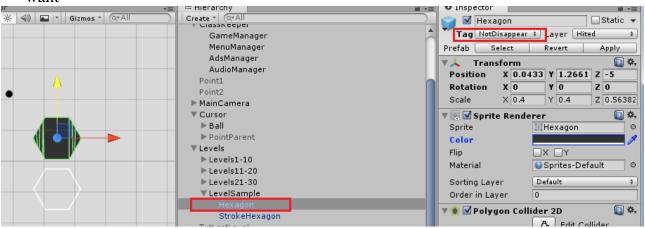
Hexagon gameobject must have layer with name "Hited" and 2d collider with "IsTrigger" false



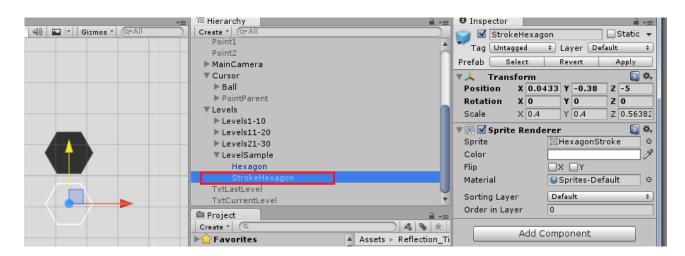


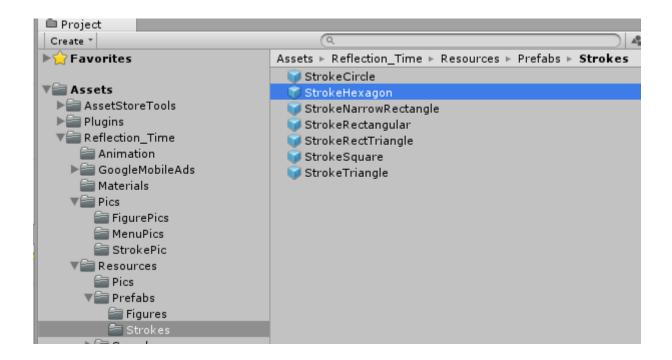
I suggest to save *Hexagon* as a prefab, if you want to use it in next levels

• For figures that, don't disappear, set "NotDisappear" tag, and different color, if you want



• Strokes must be save as a prefab to *Assets/Reflection_Time/Resources/Prefabs/Strokes*

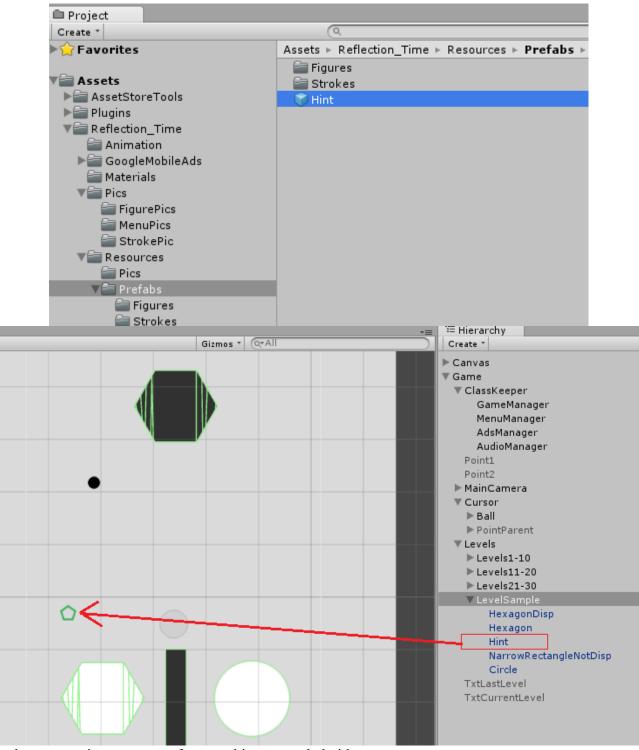




Strokes prefabs are loaded from *Resources* folder inside *StrokeBehaviour* constructor and after, used in pool system

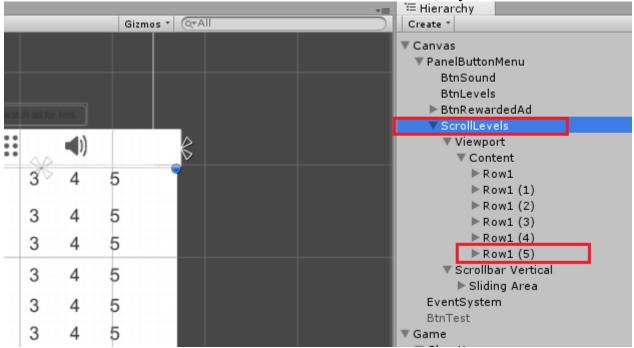
```
StrokeBehaviour.cs + X BallBehaviour.cs
Constants.cs
                  Cursor.cs
                                  Obstacles.cs
C# Assembly-CSharp
                                                                       🔩 StrokeBehaviour
                 MonoBehaviour mono;
     23
                 List<Stroke> strokes = new List<Stroke>();
                 - references
                 public StrokeBehaviour (MonoBehaviour mono)
     24
     25
                 {
     26
                     this.mono = mono;
                     Transform strokesParent = new GameObject().transform;
     27
                     strokesParent.name = "Strokes";
     28
     29
                     GameObject[] strokePrefabs = Resources.LoadAll<GameObject>("Prefabs/Strokes");
     30
                     for (int i = 0; i < strokePrefabs.Length; i++)</pre>
     31
     32
                     {
                         StrokesInit(strokePrefabs[i], strokesParent, strokePrefabs[i].name);
     33
     34
                     }
     35
                 }
```

 Hint Add hint gameobject from Assets/Reflection_Time /Resources / Prefabs to your level

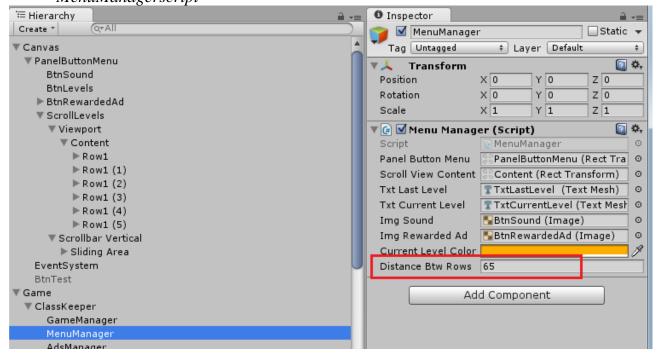


^{*}Hint become active on scene after watching rewarded video

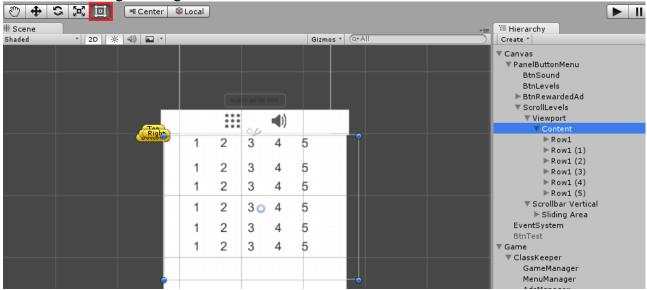
In ScrollLevels menu add new rows of levels' numbers, if you need



• You can regulate distance between rows: set value of variable *distanceBtwRows* in *MenuManagerscript*



• Don't forget change size of Content if all levels aren't shown



Errors

- Advertisement is defined multiple times

Comment all rows in *UnityAdsImplement* script, run the game(obviously, you got the error), then uncomment all rows in *UnityAdsImplement* script and run the games

- The name "Advertisement" does not exist in current context You can find solution her http://goo.gl/yzNWkt
- UnityException: Tag or error with layer Open *ReadMe* file, which is located in "*Reflection_Time*" root folder and you get answer

Thanks for your purchasing

Have a question? - shorkagames@gmail.com

More our assets: Publisher link

Follow us on Twitter: twitter.com/ShorkaGames

Liked us on Facebook: facebook.com/ShorkaGames