

Unity Platformer

Eric Thuc Tran, Jonathan Mai,
Michael Chung, Justin Tang

Intro

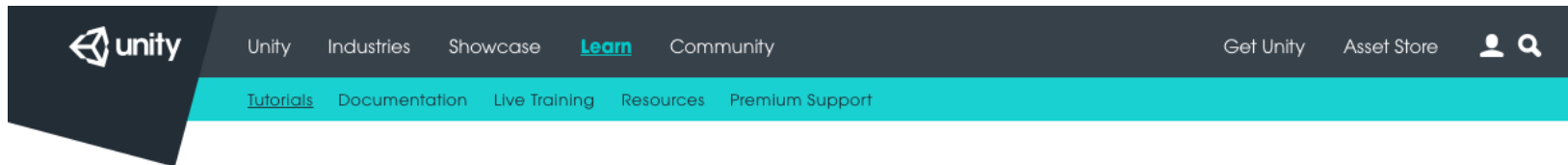
Vision: “To make a game that uses ideals from old-school platformers, while making it as playable and addicting as today’s mobile app games.”

IDE

Rating: 5/5 (10/10 #woulduseagain)



Unity Tutorials

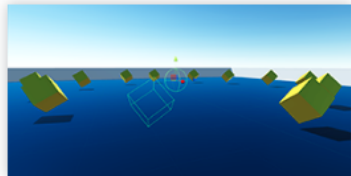


TUTORIALS

In this area we provide two paths to learning Unity. One is a set of linear Projects, where you step through creating an example game, the other is a set of lessons and assignments sorted by Topic.



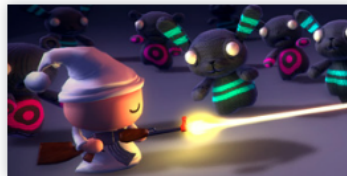
PROJECTS



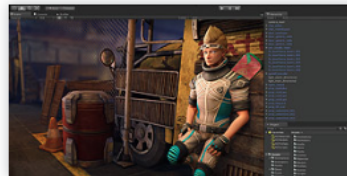
Project: Roll-a-Ball →
Beginner



Project: Space Shooter →
Beginner



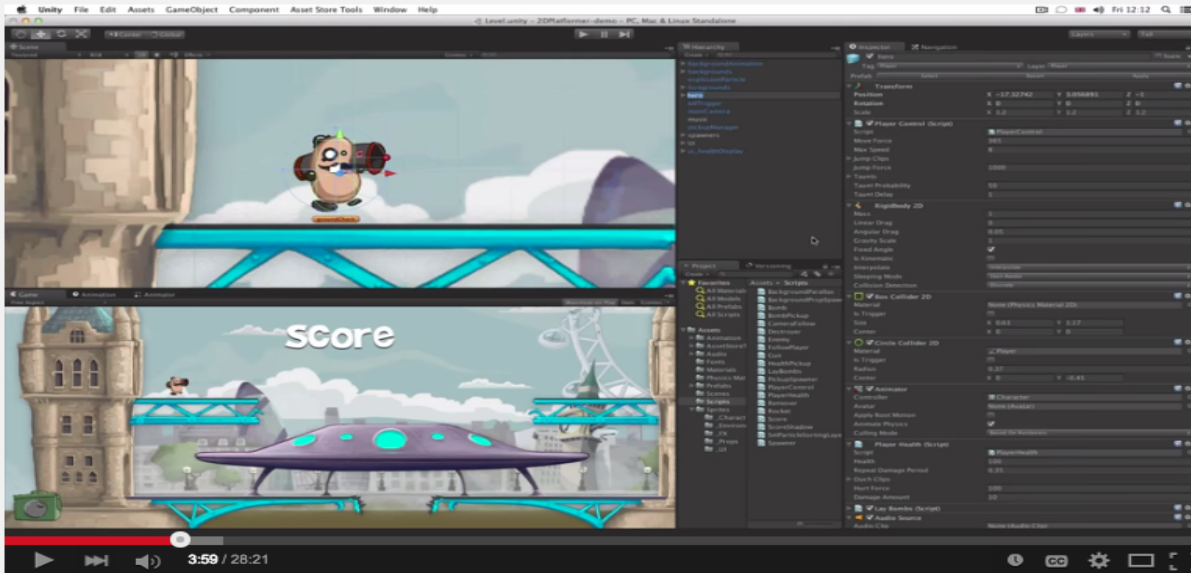
Project: Survival shooter →
Beginner



Project: Stealth →
Intermediate

Unity Tutorials cont.

YouTube



Unity 4.3 - 2D Game Development Walkthrough

Rating: 5/5

Agile and Pair Programming

Rating: 5/5



Rating: 2/5



Project Tracker



Rating: 3/5

Project Tracker cont.

Organization(cards/lists), Updates,

The screenshot displays a Trello board titled "CSCI 3308 Game Project" with a public visibility setting. The board is organized into several lists:

- Helpful Websites:** Contains two cards: "For art, use opengameart.org(we may want to establish a theme first)" and "Scripts: Google 2D Unity Tutorials(There's a lot of them)".
- Presentation Work Split up:** Contains five cards: "project Tracker" (assigned to MC), "VCS Repository" (assigned to ET), "DataBase" (assigned to JM), "Testing" (assigned to JM), and "AutoDOC".
- To-Do:** A list with a placeholder "Add a card...".
- Job Choice:** Contains two cards: "Position 1: Main" (0/5) and "Position 4: Enemy" (0/3).

On the right side, there is a sidebar with the following sections:

- Menu:** A dropdown menu.
- Members:** Shows members MC, ET, JM, and J, with an "Add Members..." button.
- Activity:** A log of recent actions by Eric Tran (ET):
 - Moved "Important" from To-Do to Working On at 4:43 pm.
 - Moved "Final Test" from To-Do to Working On at 4:42 pm.
 - Moved "Score System" from To-Do to Working On at 4:42 pm.
 - Moved "Tweaks" from To-Do to Working On at 4:42 pm.

VCS Repository

Rating: 4/5




Database

Started off making SQL table to organize

```
INSERT INTO `worker` (`Id`, `Name`) VALUES
(1, 'Jonathan Mai'),
(2, 'Eric Tran'),
(3, 'Justin Tang'),
(4, 'Michael Chung');

CREATE TABLE IF NOT EXISTS `Job` (
  `pId` int(1) NOT NULL auto_increment,
  `Position` varchar(40) NOT NULL,
  `Description` varchar(75) NOT NULL,
  PRIMARY KEY (`Id`)
) ENGINE=MyISAM DEFAULT CHARSET=utf8 AUTO_INCREMENT=7 ;
INSERT INTO `Job` (`Id`, `Name`, `Description`) VALUES
(1, 'Jonathan Mai', 'Sprite Artwork,animation,camera use, gameplay'),
(2, 'Eric Tran', 'Background art, obstacles & level design'),
(3, 'Justin Tang', 'Menu functionality,screens,settings,score'),
(4, 'Michael Chung', 'Variety of Enemies(Sprite,animation,ai)');
```



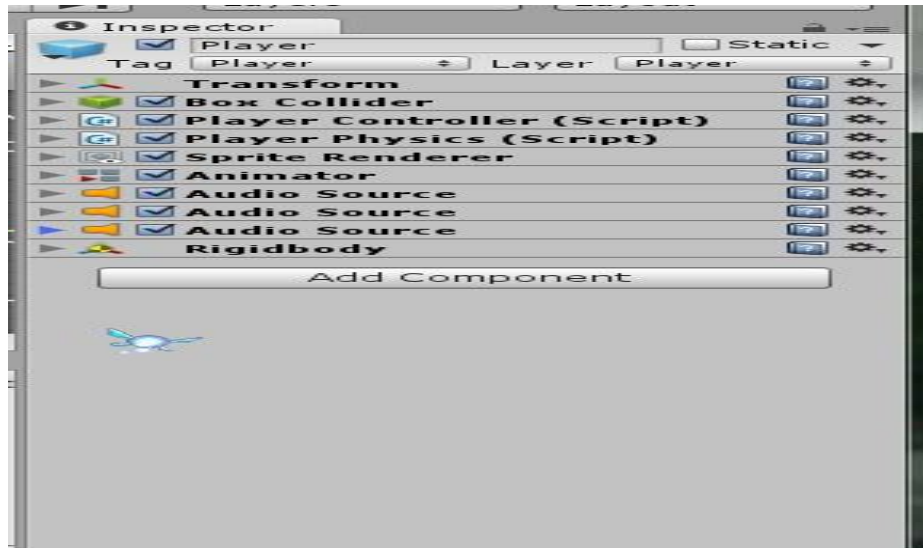
Unity allows Folder Organization

Works for small teams with few files



Unity & SQL

Prefabs are “keys” while components are values



Database for Large Scale Games

Can't rely purely on folders for organization

Each type of component(audio, model, animation,etc.) needs prefix/suffix recognition for searches

Example “bombDeathSD” for bomb's death sound

Rating: 4/5

Testing

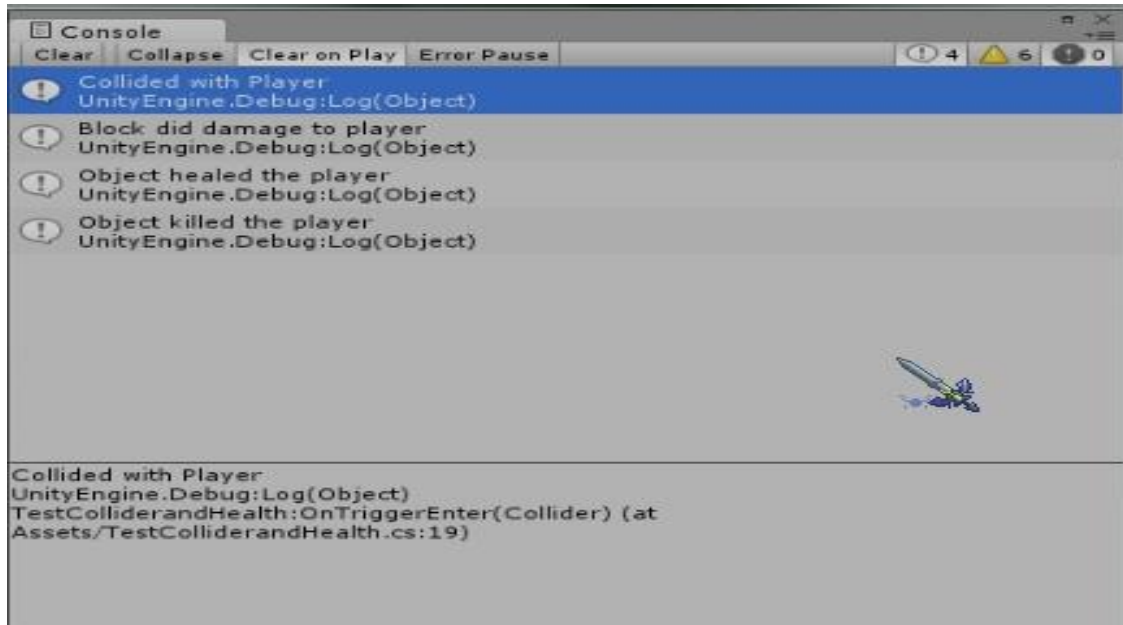
Automated testing by running game with test scripts & test prefabs

Use “debugs” to automatically detect errors

```
void OnTriggerEnter(Collider col){  
    if(col != null){  
        {if(col.gameObject.tag == "Player")  
        {Debug.Log ("Collided with Player");}  
        else{Debug.Log ("Failure:Object did not collide with player");}}  
        PreviousHealth += col.GetComponent <Health> ().currentHealth;  
        col.GetComponent<Health>().ReceiveDmg(5);  
        NewHealth += col.GetComponent <Health> ().currentHealth;  
        {if(NewHealth < PreviousHealth){  
            Debug.Log ("Block did damage to player");}  
            else{Debug.Log ("Failure:Object did not damage the player");}}    }  
}
```

Unity Console

Give compilation errors & runtime errors



Rating: 5/5

Auto-Doc

The logo for doxygen, featuring the word "doxygen" in a 3D, metallic font with a yellow-to-orange gradient. The letters are bold and have a slight shadow. A thick red horizontal line is positioned below the text, starting from the left and ending under the 'n'.

Rating: 5/5

Deployment Environment

Rating: 4/5



Challenges

- Over ambitious
- Learning C#
- Issues with Unity