



# **School of Interactive Arts**

**Prospectus**

## **Intro**

### **Urban Arts Partnership**

Urban Arts Partnership's mission is to advance the intellectual, social and artistic development of underserved public school students through arts-integrated education programs to close the achievement gap. Founded in 1991, UAP has established itself as New York's largest and fastest growing arts-education organization, providing direct services to over 100 schools, 12,000 students, and 450 teachers. UAP delivers a suite of arts-integrated programs designed to accelerate student achievement. Our programs are framed within a youth development model that supports social and emotional learning, increases levels of student engagement, and prepares young people for college and careers. Core subject matter is taught by infusing the arts into the academic curriculum, engaging and inspiring youth to develop the critical thinking skills, confidence and fortitude needed for success in the 21st century. From using hip-hop to help students prepare for the Regents exam, to leveraging the art of storytelling as an ESL tool, to reinforcing in-school social studies lessons with afterschool filmmaking classes, UAP's programs unite the arts and academics to give students an opportunity to succeed at definitively breaking the cycle of poverty.

### **The School of Interactive Arts**

The School of Interactive Arts is a pre-college program in which gifted high school students learn professional game and app development skills. Founded in 2015 with the generous support of the Adobe Foundation, SIA reaches over 550 New York City public school students with in-school programming designed to provide an engaging introduction to computer science through game creation. From this group and from selected recruitment schools around the five boroughs, 25 students are selected to receive advanced training on weekends and after school in our Flatiron office. In addition to their education in game design, students receive individualized test tutoring, college advising, and financial advice. In our first two years, alumni of the program have earned admission to top colleges as game design or computer science majors, been awarded tens of thousands of dollars in scholarships, and raised thousands more in crowdfunding campaigns for their student work.

# The School of Interactive Arts

## I. Level 1 - Game Design

### Overview

The Game Design class is the core offering of the School of Interactive Arts. Over the course of a year, game design students learn to develop 2D projects using the professional-standard Unity Engine and the Playmaker FSM plugin. The year commences with a summer intensive in the month of July during which classes in fundamental Unity and Playmaker skills are held four hours a day, four days a week for four weeks. When the school year begins, the class meets for four hours of instruction every Saturday, ending in May with a culminating showcase in which students present their work. Students who have completed the requirements for the year are promoted to SIA's advanced program, the Game Studio.

### Admissions

SIA's admissions process is open to New York City high school students between 9th and 11th grades. The small size of the program and its accelerated curriculum mean that only those applicants who are best qualified can be admitted. Prospective students must demonstrate evidence of significant ability and motivation in one or more aspects of game development. These can include (but are not limited to) - programming, digital art and animation, music production, writing and storytelling, and gameplay design. Applicants must link to a portfolio of their creative work in any discipline and promising candidates are invited to an interview with an SIA faculty member to finalize their admission. Academic qualifications are also taken into account - strong performance in school can supplement or replace the portfolio requirement if sufficiently impressive.

### Badges

During their year of study, students in SIA's Game Design class earn badges indicating degrees of proficiency with the various hard and soft skills of game development. Badges are divided into two tiers - **Growth Badges** certify students in specific skills or knowledge areas used in a particular discipline, while **Master Badges** are awarded upon a student's completion of all Growth badges in a particular discipline. There are three skills for which badges may be awarded: **Unity Engineering**, **Finite State Machine Programming**, and **Program Design**. Successful completion of the three master skill badges in the Game Design class results in the student being promoted to the Game Studio and awarded the Year 1 SIA Diploma, which opens the student to further opportunities including access to advanced training and personal equipment. The badge structure and associated learning progression is given below:

# Game Design: Master Skills Overview

The Game Design class is a space where gifted high-school students learn the skills involved in the creation of original videogames. Participants are selected through a competitive admissions process at recruitment schools across the five boroughs of New York City.

MASTER SKILL	DESCRIPTION	GROWTH SKILLS
<b>Finite State Machine Programmer</b>	<p>Programming is an essential skill involved in game development. Playmaker is used by Unity programmers in the game industry to rapidly create and prototype game mechanics.</p> <p>Students will acquire proficiency in Finite State Machine (FSM) Programming using Playmaker for the Unity Engine and create FSMs involving the elements learned in the Level 1-3 FSM Programmer badges.</p>	<ol style="list-style-type: none"> <li><b>Level 1 FSM Programmer (Unity/Playmaker):</b> Navigate FSM interface and use states, actions, transitions, and variables.</li> <li><b>Level 2 FSM Programmer (Unity/Playmaker):</b> Use multiple FSMs on a single GameObject and generate inter-GameObject communications.</li> <li><b>Level 3 FSM Programmer (Unity/Playmaker):</b> Use advanced actions in FSM programming.</li> </ol>
<b>Unity Engineer</b>	<p>Game engine usage is an essential skill involved in game development. The Unity Engine is used by professional developers to create cross-platform games and mobile apps.</p> <p>Students will acquire proficiency in Unity Engine usage and assemble projects involving the elements learned in the Level 1-3 Unity Engine badges.</p>	<ol style="list-style-type: none"> <li><b>Level 1 Unity Engineer:</b> Create and arrange gameObjects into coherent scenes, Manipulate the Hierarchy, Inspector, Project, Console, Scene, e3eand Game panels in the Unity GUI.</li> <li><b>Level 2 Unity Engineer:</b> Manipulate the Transform, Rigidbody, ConstantForce, Renderer, and Collider Components in the Unity Engine.</li> <li><b>Level 3 Unity Engineer:</b> Perform standalone build creation in the Unity Engine; integrate functionality of multiple Components to produce complex behavior.</li> </ol>
<b>Program Designer</b>	<p>Program design is an essential cognitive skill involved in game development. Professional developers must convert ideas for game mechanics into algorithms that can be implemented by a computer.</p> <p>Students will acquire proficiency in the design and development of original games by abstracting game mechanics into algorithmic solutions involving the elements learned in the Level 1-3 Program Designer badges.</p>	<ol style="list-style-type: none"> <li><b>Level 1 Program Designer:</b> Abstract basic game mechanics using methods such as linear 2D movement, state transitions, and object creation and destruction.</li> <li><b>Level 2 Program Designer:</b> Abstract intermediate game mechanics using methods such as non-linear 2D movement, operator and global state transitions.</li> <li><b>Level 3 Program Designer:</b> Abstract advanced game mechanics using methods such as physics-based 2D movement, scene transitions, and animations.</li> </ol>

# Master Skill: Finite State Machine (FSM) Programmer

Programming is an essential skill involved in game development. Playmaker is used by Unity programmers in the game industry to rapidly create and prototype game mechanics. Students will acquire proficiency in Finite State Machine (FSM) Programming using Playmaker for the Unity Engine and create FSMs involving the elements learned in the Level 1-3 FSM Programmer badges.

Growth Skill	Sub-Skills	Criteria	Evidence
<b>Level 1 FSM Programmer (Unity/Playmaker)</b>	Students use States, Actions, Transitions, and Variables as they relate to FSM programming. Students navigate the FSM interface by creating and editing states, adding and removing actions, creating events and transitions, and declaring and modifying variables.	<ul style="list-style-type: none"> <li>Replicate an FSM involving Level 1 features from memory</li> <li>Modify that FSM in major ways</li> <li>Create an original FSM using similar features</li> </ul>	<ul style="list-style-type: none"> <li>Link - Level 1 FSM Programmer Replicate project</li> <li>Link - Level 1 FSM Programmer Modify project (Media file)</li> <li>Link - Level 1 FSM Programmer Create project (Media file)</li> </ul>
<b>Level 2 FSM Programmer (Unity/Playmaker)</b>	Students use multiple FSMs on a single GameObject, and generate inter-GameObject communication via Global Variables.	<ul style="list-style-type: none"> <li>Replicate usage of multiple FSMs on a single GameObject and performing intermediate actions from memory</li> <li>Modify those FSMs in major ways</li> <li>Create original FSMs that perform intermediate actions on a single GameObject</li> </ul>	<ul style="list-style-type: none"> <li>Link - Level 2 FSM Programmer Replicate project</li> <li>Link - Level 2 FSM Programmer Modify project</li> <li>Link - Level 2 FSM Programmer Create project</li> </ul>
<b>Level 3 FSM Programmer (Unity/Playmaker)</b>	Students use advanced actions (e.g. Animate Variable).	<ul style="list-style-type: none"> <li>Replicate usage of Events and advanced actions on an FSM from memory</li> <li>Modify usage of Events and advanced actions in major ways</li> <li>Create an original FSM that uses Events and advanced actions in novel ways</li> </ul>	<ul style="list-style-type: none"> <li>Link - Level 3 FSM Programmer Replicate project</li> <li>Link - Level 3 FSM Programmer Modify project</li> <li>Link - Level 3 FSM Programmer Create project</li> </ul>

# Master Skill: Unity Engineer

Game engine usage is an essential skill involved in game development. The Unity Engine is used by professional developers to create cross-platform games and mobile apps. Students will acquire proficiency in Unity Engine usage and assemble projects involving the elements learned in the Level 1-3 Unity Engine badges.

Growth Skill	Sub-Skills	Criteria	Evidence
<b>Level 1 Unity Engineer</b>	Students use the Hierarchy, Inspector, Project, Console, Scene, and Game panels, importing assets, moving, rotating, and scaling objects using the graphical user interface (GUI), Scene view movement, object focusing, flythrough mode, project and scene structure, and the Transform component.	<ul style="list-style-type: none"> <li>Replicate a Unity project involving Level 1 features from memory</li> <li>Modify the Unity project in major ways</li> <li>Create an original Unity project that uses Level 1 features</li> </ul>	<ul style="list-style-type: none"> <li>Link - Level 1 Unity Engine elements Replicate project</li> <li>Link - Level 1 Unity Engine elements Modify project</li> <li>Link - Level 1 Unity Engine elements Create project</li> </ul>
<b>Level 2 Unity Engineer</b>	Students use the Rigidbody, ConstantForce, Renderer, and Collider, panels, importing assets, moving, rotating, and scaling objects using the graphical user interface (GUI), Scene view movement, object focusing, flythrough mode, project and scene structure, and the Transform component.	<ul style="list-style-type: none"> <li>Replicate a Unity project involving Level 2 features from memory</li> <li>Modify the Unity project in major ways</li> <li>Create an original Unity project that uses Level 2 features</li> </ul>	<ul style="list-style-type: none"> <li>Link - Level 2 Unity Engine elements Replicate project</li> <li>Link - Level 2 Unity Engine elements Modify project</li> <li>Link - Level 2 Unity Engine elements Create project</li> </ul>
<b>Level 3 Unity Engineer</b>	Students employ standalone build creation in Unity Engine to create functional products.	<ul style="list-style-type: none"> <li>Replicate a Unity project involving standalone build creation from memory</li> <li>Modify the Unity project in major ways</li> <li>Create original Unity project that uses standalone build creation</li> </ul>	<ul style="list-style-type: none"> <li>Link - Level 3 Unity Engine elements Replicate project</li> <li>Link - Level 3 Unity Engine elements Modify project</li> <li>Link - Level 3 Unity Engine elements Create project</li> </ul>

# Master Skill: Program Designer

*Program design is an essential cognitive skill involved in game development. Professional developers must convert ideas for game mechanics into algorithms that can be implemented by a computer. Students will acquire proficiency in the design and development of original games by abstracting game mechanics into algorithmic solutions involving the elements learned in the Level 1-3 Program Designer badges.*

Growth Skill	Sub-Skills	Criteria	Evidence
<b>Level 1 Program Designer</b>	Students utilize methods for abstracting basic game mechanics - including linear 2D movement, object and prefab creation and destruction, and boolean on/off switches - into states, transitions, variables, and actions.	<ul style="list-style-type: none"> <li>• Replicate a process for abstracting game mechanics from memory</li> <li>• Modify the abstracting process in major ways</li> <li>• Create original abstracted game mechanic of similar complexity</li> </ul>	<ul style="list-style-type: none"> <li>• Link - Replicate project for abstracting a basic game mechanic</li> <li>• Link - Modify project for abstracting a basic game mechanic</li> <li>• Link - Create project for abstracting a basic game mechanic</li> </ul>
<b>Level 2 Program Designer</b>	Students utilize methods for abstracting intermediate game mechanics - including non-linear 2D movement, float variable manipulations, timers, string variables, and UI creation - into states, transitions, variables, and actions.	<ul style="list-style-type: none"> <li>• Replicate a process for abstracting intermediate game mechanics from memory</li> <li>• Modify the abstracting process in major ways</li> <li>• Create original abstracted intermediate game mechanic of similar complexity</li> </ul>	<ul style="list-style-type: none"> <li>• Link - Replicate project for abstracting an intermediate game mechanic</li> <li>• Link - Modify project for abstracting an intermediate game mechanic</li> <li>• Link - Create project for abstracting an intermediate game mechanic</li> </ul>
<b>Level 3 Program Designer</b>	Students utilize methods for abstracting advanced game mechanics - including force-based 2D physics movement, transitioning between multiple scenes, and complex win/lose conditions - into states, transitions, variables, and actions.	<ul style="list-style-type: none"> <li>• Replicate a process for abstracting advanced game mechanics from memory</li> <li>• Modify the abstracting process in major ways</li> <li>• Create original abstracted advanced game mechanic of similar complexity</li> </ul>	<ul style="list-style-type: none"> <li>• Link - Replicate project for abstracting an advanced game mechanic</li> <li>• Link - Modify project for abstracting an advanced game mechanic</li> <li>• Link - Create project for abstracting an advanced game mechanic</li> </ul>

## II. Level 2 - Game Studio

### **Overview**

The Game Studio is an advanced pre-professional class in which each student spends a full year conceiving and creating a single portfolio-quality game. It is open to advanced students from the Game Design class who have successfully earned the Year 1 SIA Diploma and its accompanying master and growth skill badges and runs on Wednesday afternoons during the school year. Students also have access to office hours on Saturdays to work on their projects with the help of program staff.

Each student's project must be an original mobile game or interactive application for iOS, and the student is responsible for all facets of its development, from asset creation to programming to launch. Once a playable demo has been created, the student spends several months bringing it to a high level of polish.

This rigorous structure allows students to grapple with every aspect of the software development process, from design to development to marketing. Successful students will have had the unusual experience of developing and marketing a commercial title while still in school, which will significantly aid their chances of admission to top postsecondary game design and computer science programs.

### **Test Prep and College Advising**

Game Studio students are between their sophomore and senior years of high school, and their opportunities after graduation will depend in part on strong academic performance and high standardized test scores. As part of its core curriculum, SIA provides comprehensive multi-year test prep and individual advising to all students. SAT workshops occur three times a year in week-long intensives for all sophomores, juniors, and seniors. Seniors and second-semester juniors meet once a week with their college advisor at SIA to develop a strategy for college admissions. By providing these services internally and integrating them with the rest of SIA's programming, students are assured consistent, effective support for their academic endeavors.



### III. Level 3 - Alumni

#### **Overview**

SIA Alumni who remain in New York City can continue their involvement with the school by pursuing training in fields outside of game and app design with professional mentors, completing crowdfunding campaigns for original projects, and taking on roles as program staff. Participation in the alumni program is contingent on the student maintaining status as a full-time college student and earning a sufficiently high GPA, or on the pursuit of another full-time postsecondary option such as an internship or bootcamp.

#### **Assistant Teaching Artists**

Especially qualified alumni are hired to help teach and develop curriculum for the Game Design or Game Studio classes. These Assistant Teaching Artists co-facilitate classroom activities in collaboration with the lead teacher and help design lesson plans. The involvement of alumni in SIA's curriculum design process helps ensure that activities and lessons are as timely, relevant, and engaging as possible.

#### **Professional Mentorship**

Each semester, alumni work with a new Project Code Mentor – a full-time tech professional whose job is to share expertise while training apprentices for paid internships in his or her field. The Mentor is responsible for teaching the fundamental skills of his or her primary technical discipline and for helping students complete a portfolio of work demonstrating those skills.

#### **Entrepreneurship**

Once a year, two alumni are chosen to receive support in putting together crowdfunding campaigns to underwrite the completion of full commercial games based on their Game Studio projects. The campaigns are launched in the winter and all funds raised go to the alumnus to help in completing the game. The completed game will be launched on the iTunes Store or Steam as a commercial product.

## **Contact**

For more information about the program, please contact us below:

Akshai Raj  
Director  
School of Interactive Arts  
39 West 19th Street, 5th Floor  
(917) 900-9024  
[akshai@urbanarts.org](mailto:akshai@urbanarts.org)