ARIZONA STATE UNIVERSITY

ARIZONA STATE UNIVERSITY

GENERAL STUDIES PROGRAM COURSE PROPOSAL COVER FORM

Courses submitted to the GSC between 2/1 and 4/30 if approved, will be effective the following Spring.

Courses submitted between 5/1 and 1/31 if approved, will be effective the following Fall.

(SUBMISSION VIA ADOBE.PDF FILES IS PREFERRED)

DATE 02/02/2008

1. ACADEMIC UNIT: School of Computing and Informatics

2. COURSE PROPOSED: CPI 111 Game Development I 03

(prefix) (number) (title) (semester hours)

3. CONTACT PERSON: Name: Ashish Amresh Phone: 5-1349

Mail Code: 8809 E-Mail: amresh@asu.edu

4. ELIGIBILITY: New courses must be approved by the Tempe Campus Curriculum Subcommittee and must have a regular course number. For the rules governing approval of omnibus courses, contact the General Studies Program Office at 965–0739.

5. AREA(S) PROPOSED COURSE WILL SERVE. A single course may be proposed for more than one core or awareness area. A course may satisfy a core area requirement and more than one awareness area requirements concurrently, but may not satisfy requirements in two core areas simultaneously, even if approved for those areas. With departmental consent, an approved General Studies course may be counted toward both the General Studies requirement and the major program of study. **(Please submit one designation per proposal)**

Core Areas

Literacy and Critical Inquiry–L ☐
Mathematical Studies–MA ☐ CS ☒
Humanities, Fine Arts and Design–HU ☐
Social and Behavioral Sciences–SB ☐
Natural Sciences–SQ ☐ SG ☐

Awareness Areas

Global Awareness–G ☐
Historical Awareness–H ☐
Cultural Diversity in the United States–C ☐

6. DOCUMENTATION REQUIRED.
(1) Course Description
(2) Course Syllabus
(3) Criteria Checklist for the area
(4) Table of Contents from the textbook used, if available NA

7. ***In the space provided below (or on a separate sheet), please also provide a description of how the course meets the specific criteria in the area for which the course is being proposed.***

In this course the students learn problem solving skills by applying a visual computer programming language inside the game design and development domain. The game design and development domain is a newly established area that is truly interdisciplinary and requires a strong background in computer programming and analytical reasoning. The course teaches the students the fundamental theory behind game design and the applied skills involved in learning the GameMaker programming language. Students then use these skills and apply them in various game design areas ranging from collision detection, camera movement, user interface controls, artificial intelligence and animation. Every student has to submit a fully functional 2D game as final project for this course. The process of a creating a fully functional game encompasses all the skills learnt during the course and culminates into a visual representation of student's grasp of the subject.

New Course

ARIZONA STATE UNIVERSITY

CROSS-LISTED COURSES: ☒ No ☐ Yes; Please identify courses: _____

Is this amultisection course?: ☐ No ☒ Yes; Is it governed by a common syllabus? yes

Gerald Farin, Associate Director SCI
Chair/Director (Print or Type)

Chair/Director (Signature)

Date: 2/12/09

CPI 111 GAME DEVELOPMENT I

The course introduces video game design, art, programming theory, and concepts as they apply to video game development. Topics range from fundamental video game art principles of 2-D and 3-D composition, color theory, modeling, and lighting techniques. The course demonstrates practical application of these art fundamentals in establishing style guides, concept art, storyboards, and in-game assets. It further combines hands-on programming skills with game design theory to educate concepts involved in the game development cycle. Students learn a visual programming language to implement the above concepts. Students create fully functional video game prototypes by the end of the course.

CPI 111 **GAME DEVELOPMENT I** **SPRING 2009**

INSTRUCTOR: Ashish Amresh
Office: BY 362
Phone: (480)-965-1349
Email: amresh@asu.edu
Office Hours: Wed 10 -12 @ BYE 350

LECTURES: **Tu**, Th 1:30 - 2:45 pm BYE 361
Tu, Th 3:00 - 4:15 pm BYE 361

CLASS WEBSITE: TBA

CATALOG DESCRIPTION:

This course introduces fundamental video game development concepts and techniques prevalent in video game industry. The students would have an opportunity to understand these techniques and learn how they are applied during the production of a video game. The areas covered include game design, game production, asset production, game programming, game art and animation. The students will have hands on experience by writing 2D game prototypes belonging to different genres and evaluating the techniques that they learn in class.

PREREQUISITES:

None

REQUIRED TEXT:

(B1) "GameMaker's Apprentice: Game Development for Beginners", by Jacob Habgood (Author), Mark Overmars (Author), ISBN-1-59059-615-3

RECOMMENDED TEXTS:

(B2) *"Introduction to Game Development"*, Game Development Series, Edited by Steve Rabin, Charles River Media, ISBN – 1-58450-377-7.

GRADING:

Assignments	5 INDIVIDUAL Assignments. 10 points each.	50
Final Project	Game Prototype creation	50
TOTAL		**100**

COURSE GRADE:

100+	A+
90 – 100	A
85 – 90	A-
80 – 85	B
75 – 80	B-
70 – 75	C
60 – 70	D
< 60	E

SCHEDULE LINE NUMBER: CSE 494 - 06820 / CSE 598 – 39198

Details:

This course is intended to serve as an introduction into the game production cycle. The course breaks down the complex process of game creation into a simple step by step program. No programming knowledge is required for the course. However some degree of computer knowledge is desirable. The course will go into detail the design and production methodologies used for creating games in various genres. Students will apply the skills taught during class in the various assignments. The assignments would lead into each other and the final project would be a simple game created by the student as a result of successfully completing the various assignments. The class will teach the students how to use Game Maker 7.0, which is a simple easy to use visual game design and development software. Students will learn the basic art principles which should be used in game development. Among the principles which will be covered are 2-D and 3-D composition, color theory, modeling and lighting techniques. Practical application of these art fundamentals will be demonstrated in establishing style guides, concept art, storyboards, and in-game assets.

Assignments:

Students will be assigned 5 game development assignments, each worth 10 points (50% of the final grade). The assignments will test a student's skill in implementing the concepts discussed in class.

Final project: (see final project guidelines document for more information)

The Final Project will the culmination of the efforts put by the student in the class assignments in developing a simple playable game. The Final project will have different deliverables over the time frame of the class

1. Initial Game Idea Document (see initial idea document sample for more details)
2. Game Design Document (see game design specifications document for more details)
3. Final Game implementation, Presentation and Demo.

Disability resource center:

Please check the website for ASU's Disability Resource Center (http://www.asu.edu/drc/) for assistance. Students with special needs should contact the center a priori in order to secure assistance.

Behavior in class:

We expect the students to maintain atmosphere conducive to teaching and learning in the class. It would be appreciated if all cell phones and pagers were switched off, or to vibrate mode. Active student participation is expected in all in-class discussions.

Honor policy:

The highest standards of academic integrity are expected of all students. The failure of any student to meet these standards may result in suspension or expulsion from the University or other sanctions as specified in the University Student Academic Integrity Policy. Violations of academic integrity include, but are not limited to, cheating, fabrication, tampering, plagiarism, or facilitating such activities.

Expected Workload:

The course is designed to distribute workload pretty evenly over the semester. Students would typically spend **3-5 hours per week** working on homework assignments and projects. Plan your schedule accordingly.

Arizona State University Criteria Checklist for

MATHEMATICAL STUDIES [CS]

Rationale and Objectives

The **Mathematical Studies** requirement is intended to ensure that students have skill in basic mathematics, can use mathematical analysis in their chosen fields, and can understand how computers can make mathematical analysis more powerful and efficient. The **Mathematical Studies** requirement is completed by satisfying both the **Mathematics [MA]** requirement and the **Computer/Statistics/Quantitative Applications [CS]** requirement explained below.

The **Mathematics [MA]** requirement, which ensures the acquisition of essential skill in basic mathematics, requires the student to complete a course in College Mathematics, College Algebra, or Precalculus, or demonstrate a higher level of skill by completing a mathematics course for which any of the first three courses in a prerequisite.

The **Computer/Statistics/Quantitative Applications [CS]** requirement, which ensures skill in real world problem solving and analysis, requires the student to complete a course that uses some combination of computers, statistics, and mathematics.

Approved: Feb. 2000

Proposer: Please complete the following section and attach appropriate documentation.

ASU--[CS] CRITERIA

A COMPUTER/STATISTICS/QUANTITATIVE APPLICATIONS [CS] COURSE MUST SATISFY ONE OF THE FOLLOWING CRITERIA: 1, 2, OR 3

YES	NO		Identify Documentation Submitted
		1. Computer applications*: courses must satisfy both **a** and **b**:	
☒	☐	**a.** Course involves the use of computer programming languages or software programs for quantitative analysis, modeling, simulation, animation, or statistics.	Syllabus, student assignments and projects
		b. Course requires students to analyze and implement procedures that are applicable to at least one of the following problem domains **(check those applicable)**:	Syllabus, student assignments and projects`
☐	☐	**i.** Spreadsheet analysis, systems analysis and design, and decision support systems.	
☒	☐	**ii.** Graphic/artistic design using computers.	Syllabus, student assignments and projects
☐	☐	**iii.** Music design using computer software.	
☒	☐	**iv.** Modeling, making extensive use of computer simulation.	Syllabus, student assignments and projects
☐	☐	**v.** Statistics studies stressing the use of computer software.	

*The **computer applications** requirement **cannot** be satisfied by a course, the content of which is restricted primarily to word processing or report preparation skills; learning a computer language or a computer software package; or the study of the social impact of computers. Courses that emphasize the use of a computer software package or the learning of a computer programming language are acceptable, provided that students are required to understand, at an appropriate level, **the theoretical principles embodied in the operation of the software and are required to construct, test, and implement procedures that use the software to accomplish tasks in the applicable problem domains.**

		2. Statistical applications: courses must satisfy both **a** and **b**.	
☐	☐	**a.** Course has a minimum mathematical prerequisite of College Mathematics, College Algebra, or Precalculus, or a course already approved as satisfying the MA requirement.	
☐	☐	**b.** The course must be focused principally on developing knowledge in statistical inference and include coverage of all of the following:	

ASU--[CS] CRITERIA

YES	NO		Identify Documentation Submitted
☐	☐	**i.** Design of a statistical study.	
☐	☐	**ii.** Summarization and interpretation of data.	
☐	☐	**iii.** Methods of sampling.	
☐	☐	**iv.** Standard probability models.	
☐	☐	**v.** Statistical estimation	
☐	☐	**vi.** Hypothesis testing.	
☐	☐	**vii.** Regression or correlation analysis.	
		3. Quantitative applications: courses must satisfy both **a** and **b**.	
☐	☐	**a.** Course has a minimum mathematical prerequisite of College Mathematics, College Algebra, or Precalculus, or a course already approved as satisfying the MA requirement.	
		b. The course must be focused principally on the use of mathematical models in quantitative analysis and design making. Examples of such models are:	
☐	☐	**i.** Linear programming.	
☐	☐	**ii.** Goal programming.	
☐	☐	**iii.** Integer programming.	

ASU--[CS] CRITERIA

YES	NO		Identify Documentation Submitted
☐	☐	Inventory models. Decision theory.	
☐	☐		
☐	☐	**vi.** Simulation and Monte Carlo methods.	
☐	☐	**vii.** Other (explanation must be attached)	

Course Prefix	Number	Title	Designation
CPI	111	GAME DEVELOPMENT I	CS

Explain in detail which student activities correspond to the specific designation criteria. Please use the following organizer to explain how the criteria are being met.

Criteria (from checksheet)	How course meets spirit (contextualize specific examples in next column)	Please provide detailed evidence of how course meets criteria (i.e., where in syllabus)
1.a	The programming language taught in class is used to create digital art, digital animation and game design content in all assignments and Final Project	Check Assignments 1 thru 5 and the Final project folders for more information
1.b	The Course teaches the students how to create digital art and animation content for use in video games. This content is created procedurally by applying visual programming skills and the students then use the content in creating game simulations that are fully functional and playable	Check the course syllabus and Final project folders for detailed information on applying the course content to the area of digital art and animation, modeling and simulation.

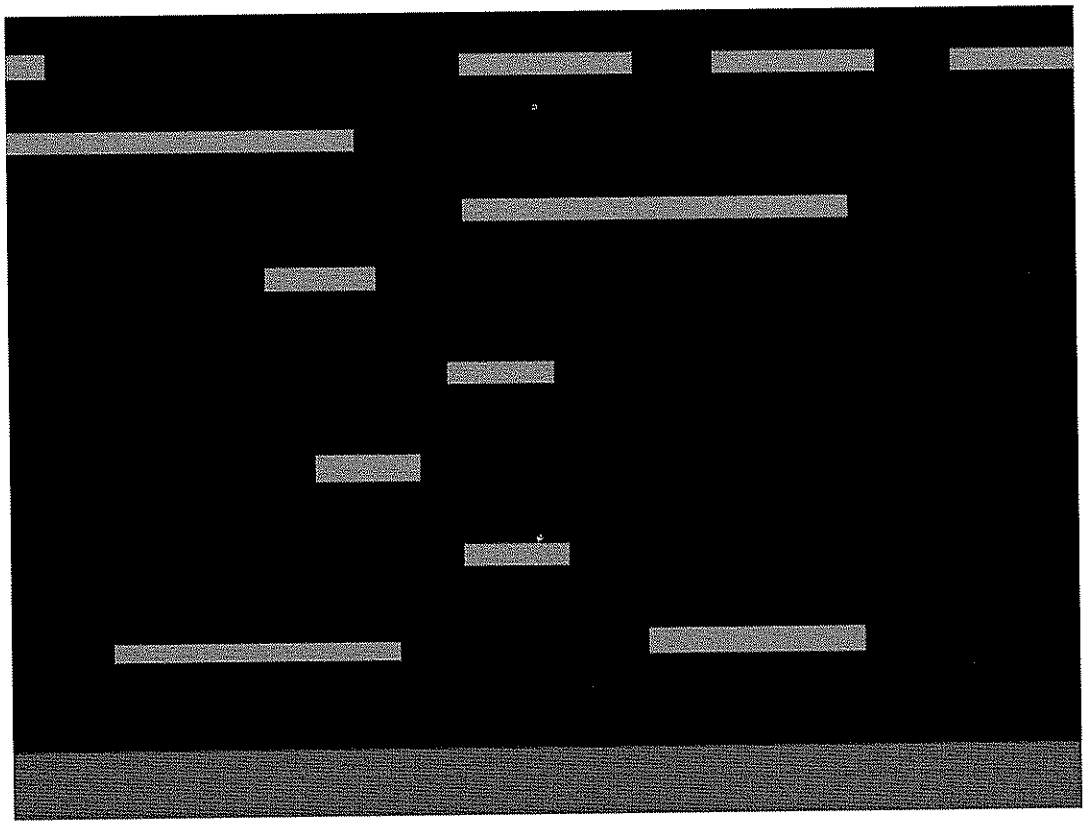
Assignment 1: Pixel art and seamless textures

What you need to do:

- Create the playfield of a full level made of 5 frames (640 x 480 square pixel)
- No more than 5 seamless geometrical tiles for the whole thing
- Black background

Pipeline:

- Create a file that is the size of the 1 Frame
- Using PS Rectangle Marquee tool create to platform you want to make for your game (it does not matter where, because your tiles will be tileable because they are seamless



Assignment 2: Collision detection and simple motion paths

In this assignment you will create a simple Maze game.

You will create three objects, a player object, a finish object and a wall object. The objective of the game is for the player to reach the finish object. There will be three levels (rooms) The first room will be a simple maze with the player learning the controls. left arrow moves player left, right arrow moves player right, up arrow moves player up and down arrow moves player down. The second room will be moderately tough and the final room will be the toughest maze. once player reaches finish object, you will move to the next room. If this is your last room(3rd) you will end game and display score.

Scoring is done the following manner. you will start with 0 and every frame(step) increment the score by 1. Person with lower score wins. Scoring needs to carry over from one level to another. (Tip: make a controller object, set its persistent value as true and attach the scoring to it. Keep that object in the first room.)

A Assignment1_template.gmk is provided with all the resources, you only have to create the objects and the rooms.
Once you are finished upload the gmk to the digital dropbox
Also included is a working example(Assignment1.exe no code) That should give you an idea of what you game should look like.

Grading: 2 For each object creation and functionality (6 points)
2 points rooms and 2 points for scoring

Extra credit:
3 points if you are able to create some bricks that are open to pass but close over time. you will need to create a new object (timed_wall). hint: Use the chance action with the step event to achieve this behavior

Assignment 3: Inheritance, 2D Sprites and simple animations

The main objective of this game is to destroy the bugs using a spray can while avoiding getting hit by the bugs.

All objects, sprites, sounds, backgrounds and rooms are in Assignment3_temp.gmk, All the events required for these objects are also included. All you have to do is write the necessary actions to finish the required functionality. You are not allowed to add any new events in the objects nor change any room properties.

1)Implement score health and lives. 100 health points and 3 lives. 10 points for every bug taken out and 10 negative health if when hit by tse tse, 20 if hit by dogstick and 30 if hit by wasp. When you reach 200 points go to the next room, when you reach 1000 points you win, if you loose all your lives game is over

2)Swamp generates tse tse, bushes generate dogstick and hive generates wasp

3) Insect is the parent for tse tse, dogstick and wasp

4)RoomManager Keeps track of score and Sounds are played when you take out a bug or if a bug hits you

Grading:

Player movement and spraying: 4 points
Insect behavior 2 points
Generators 2 points
Sounds, Scoring, health and lives 2 points

Assignment 4: Game object creation and control behavior

BUGs! *design document*

Description

In this game you control a player who is trying to collect an apple while avoiding the Bugs. You encounter an increasing number Bugs as you collect the apples. You should avoid these and collect the apples in all the three levels.

Game objects

Player object who is influenced by gravity. Key press Up moves him up with speed 10 and using the up sprite. Similarly for left, right and down. Make sure on key release the speed is set to 0. You will need to add a variable called bugSpeed and set value to 2 on create. Make the object persistent so you have only one player throughout the three levels (i.e. you only need to add a player instance in room0) bugSpeed is incremented by 2 as player finishes a level by collecting the apple and moves to the next level.

Bugs (Tick and Fly)

These are both assigned the zigzag time line in their create event. Health is decremented by 15 for collision with player. There are three lives that are available and they decrement by 1 when health reaches zero.

Apple

The apple is assigned the updown timeline. When you have collected the apple in the third level display a message called you win. If your lives are lost before that display a message you lose.

Wall

An invisible wall (Floor) is used to check the collision of the player and it so that the speed for the player can be set to zero on collision. Prevents moving the player outside of the bottom of the room.

Stats

An object for drawing the health score and lives, used only if you want to do the extra credit

Game Timelines

ZigZag

The zigzag timeline starts the instance at random and then moves the instance towards the player with the speed equal to the variable bugSpeed for one second and then moves it away from the player for another second.

UpDown

The updown timeline makes the instance move down for half a second and up for another half with a speed of 4.

Grading

5 points for all the objects and timeline code

5 Points for the easy and hard level creation

2 extra credit for creating a scoring system that shows the time it took to collect all three apples.

2 extra credit for drawing health, score and lives.

Assignment 5: Game debugging and tracking

Four problems need to be resolved; each has its own .gmk file that you will need to fix

1. When you click all the soccer balls the game should go to the next room that says You Win and end
2. The boss should die when the boss health variable is 0
3. One hit on the boss freezes and ends the game
4. The Explosions and collision testing for the planes do not happen at the current path location of the planes on the circular path.

Grading: 1: 2 points
2: 2 points
3: 2 points
4: 4 points
Extra Credit: In problem 4 make orient the left and right moving planes so they look correct: 2 points

CPI 111: GAME DEVELOPMENT I

Final Project Guidelines and Requirements

The final project for this class will test your skills in developing a 2D game. The final project carries 50% of your grade. All Students will develop a game based on your game concept entries which is submitted as the first deliverable for the project in the form of a plot outline document. The plot outline should contain the story, expected audience, the number of levels planned; genre the game belongs to, the game play elements and all the concept art showing the different levels.

The students will then submit their design document for the project. The design document should have the complete flow of the different levels planned for your game including details on how the difficulty is ramped, GameMaker implementation details, complete list of features (enemies, power-ups, resources, puzzles, etc.) that you will use every level. Complete description of the User Interface that will be used for your game.

I recommend that you divide the project load over the semester instead of getting crammed at the last minute. It will save you hours of debugging and caffeine drinks.

Here is a list of broad guidelines for the Students to follow while designing and developing their game:

(1) The game should have at least 2/3 levels with enough complexity. Shooting 5 balls through a loop and going to the next 'level' is a bad example of level design. A good example is a level that takes the player through a variety of obstacles

and roughly takes about 10 minutes of **continuous** game play to complete

(2) The levels should have detailed backgrounds

(3) The User Interface should be consistent throughout the game

(4) Top-Down or Side-Scrolling views are preferred

(5) Limit the number of different Units/Buildings/Attributes or Features but keep the game play interesting

(6) Don't spend too much time on introductory screens, etc. You don't have to make your graphics or sound and music yourself

Plot outline (due Feb 11th): Give a brief outline of the plot of the game you intend to develop. Try to be imaginative; points are also given for creativity!! I do not expect the students to submit a complete storyboard with illustrations etc. but it is recommended that the students also submit a preliminary sketch of the various game entities and floor (level) plan which would help me to visualize the end-product. Think of the document as a proposal to a producer – you want to sell your idea in the most effective way as possible. Use of catchy visual aids is highly recommended. The report should typically state the genre of the game, target audience, samples of artwork, script etc. (this list is neither complete nor exhaustive, feel free to add more information to get your idea across). If you have referred to a white paper or other document, please attach such documents as an appendix to your main outline. Refer to the online guidance document.

Design Document (due March 17th): The design document should build on the previous document, by giving a more concrete version of

the idea presented in the outline. The Student should specify what GameMaker features they are using to develop the game, the document should list in detail the resources they are going to develop/use using GameMaker. The Document should explain the user interface in detail and also describe the implementation details of all the game play elements using GameMaker.

Project Deliverables (Nov 28): The deliverables for the project include Game Maker code for the final implementation along with the plot outline and design document. The final game should build upon the individual assignments by using the ideas employed in the individual assignments. There will be an in-class demo at the end of the semester where every student will present their game. A 10 min time slot for their demo and 5 min for Q&A will be given for each student of the class.

Grading

The game will be evaluated and marked according to the following criteria:

Item	Points
Storyline	5
Documentation	5
Playability	5
Balance	5

Graphics and sound	7
User interface	5
AI/Simulation	5
Implementation	8
Originality	5
Bonus Points	
Total	50