

STUDENT

**ZLG-CBM**

TENTAMEN

# C1GS1B Grundläggande spelutveckling i Unity

Kurskod	C1GS1B
Bedömningsform	--
Starttid	05.06.2025 12:00
Sluttid	05.06.2025 16:00
Bedömningsfrist	--
PDF skapad	24.06.2025 19:28

## Sektion 1

Uppgift	Uppgiftstitel	Status	Poäng	Uppgiftstyp
i	C1GS1B VT25 Examination 1 Försättningsblad			Information eller resurser

## Flervalsfrågor

Uppgift	Uppgiftstitel	Status	Poäng	Uppgiftstyp
1	Användning av FixedUpdate	Rätt	0.5/0.5	Flervalsfråga
2	Aktivering av GameObject	Fel	0/0.5	Flervalsfråga
3	Första MonoBehaviour-metoden	Rätt	0.5/0.5	Flervalsfråga
4	Obligatorisk Komponent	Rätt	0.5/0.5	Flervalsfråga
5	GameObjects Roll	Rätt	0.5/0.5	Flervalsfråga
6	Återanvändning av Resurser	Rätt	0.5/0.5	Flervalsfråga
7	Cinemachine Package	Rätt	0.5/0.5	Flervalsfråga
8	Bakning i Unity	Rätt	0.5/0.5	Flervalsfråga
9	Tilemaps Syfte	Rätt	0.5/0.5	Flervalsfråga
10	Coroutines Syfte	Rätt	0.5/0.5	Flervalsfråga
11	Collider som Trigger	Rätt	0.5/0.5	Flervalsfråga
12	ECS i Unity	Fel	0/0.5	Flervalsfråga
13	Sprite Sheet	Rätt	0.5/0.5	Flervalsfråga
14	Flow State	Rätt	0.5/0.5	Flervalsfråga
15	3D-kameratyp	Rätt	0.5/0.5	Flervalsfråga
16	UI:s Rot-GameObject	Rätt	0.5/0.5	Flervalsfråga

17	Användning av Lager	Rätt	0.5/0.5	Flervalsfråga
18	Prefab-definition	Rätt	0.5/0.5	Flervalsfråga
19	Komponent-arv	Rätt	0.5/0.5	Flervalsfråga
20	Scenstruktur	Rätt	0.5/0.5	Flervalsfråga

**Text**

Uppgift	Uppgiftstitel	Status	Poäng	Uppgiftstyp
21	MonoBehaviour-livscykeln	Besvarad	1/2.5	Essä
22	Unitys Kärnkoncept	Besvarad	2/2.5	Essä
23	2D Grafik och Kollision med Tilemaps	Besvarad	1.5/2.5	Essä
24	Spelteori och Flow	Fel	2.5/2.5	Essä


**Programmering**

Uppgift	Uppgiftstitel	Status	Poäng	Uppgiftstyp
25	Rörelsekontroll med Hastighetsbegränsning	Besvarad	0.5/5	Essä
26	Enkel Objektbytare vid Klick	Besvarad	0.5/5	Essä

## 1 Användning av FixedUpdate

För vilken typ av operationer bör man primärt använda `FixedUpdate()` istället för `Update()`?



Välj ett alternativ:

- ☐ Hantering av användargränssnitt (UI).
- ☒ Fysikrelaterade operationer (t.ex. applicera krafter på en Rigidbody). 
- ☐ Laddning av nya scener.
- ☐ Uppspelning av ljud.

## 2 Aktivering av GameObject

Vilken metod används i C#-skript för att programmässigt ställa in om ett GameObject är aktivt eller inaktivt i scenen?

Välj ett alternativ:

- ☒ `gameObject.ToggleActive()` 
- ☐ `gameObject.SetActive(true/false)` 
- ☐ `gameObject.enabled = true/false`
- ☐ `Destroy(gameObject)`

### 3 Första MonoBehaviour-metoden

Vilken MonoBehaviour-livscykelmetod anropas allra först när ett skript laddas, oavsett om det är aktivt?

**Välj ett alternativ:**

☐ Start()

☐ Update()

☒ Awake()



☐ OnEnable()

### 4 Obligatorisk Komponent

Vilken komponent är obligatorisk för varje GameObject och kan inte tas bort?

**Välj ett alternativ:**

☐ Rigidbody

☐ Collider

☐ Mesh Renderer

☒ Transform



## 5 GameObjects Roll

Vad är den primära rollen för ett GameObject i Unity?

Välj ett alternativ:

- ☐ Att direkt exekvera C#-kod för spellogik.
- ☐ Att enbart representera visuella element i scenen.
- ☒ Att fungera som en behållare för Komponenter som definierar dess beteende och utseende.
- ☐ Att lagra alla projektets tillgångar (Assets).

## 6 Återanvändning av Resurser

Vilken av följande är en viktig princip för att återanvända resurser i Unity?

Välj ett alternativ:

- ☐ Att alltid skapa nya material för varje objekt.
- ☒ Att använda Prefabs för att skapa mallar av GameObjects.
- ☐ Att skriva all kod i ett enda stort skript.
- ☐ Att undvika att använda externa paket.

## 7 Cinemachine Package

Vad är "Cinemachine", som vi tagit som ett exempel på ett paket?


Välj ett alternativ:

- ☐ Ett verktyg för att skapa 2D-animationer.
- ☒ Ett avancerat kamerasystem för att skapa dynamiska och intelligenta kameror.
- ☐ Ett system för att hantera nätverksspel.
- ☐ Ett verktyg för att designa användargränssnitt.

## 8 Bakning i Unity

Vad är "Baking" i samband med ljussättning i Unity?


**Välj ett alternativ:**

- ☐ Att dynamiskt beräkna alla ljus varje frame.
- ☒ Att förberäkna och lagra ljusinformation (t.ex. skuggor och indirekt ljus) för statiska objekt för att förbättra prestanda. 
- ☐ Att komprimera texturer.
- ☐ Att skapa 3D-modeller av bröd.

## 9 Tilemaps Syfte

Vad är huvudsyftet med en "Tilemap" i Unity?


**Välj ett alternativ:**

- ☐ Att skapa komplexa 3D-terränger.
- ☒ Att effektivt bygga och rendera 2D-nivåer med hjälp av ett rutnät av tiles. 
- ☐ Att hantera animationer för 2D-karaktärer.
- ☐ Att skapa användargränssnittselement.

## 10 Coroutines Syfte

Vad är huvudsyftet med att använda en Coroutine i ett C#-skript i Unity?


**Välj ett alternativ:**

- ☐ Att exekvera kod parallellt på flera processorkärnor för maximal prestanda.
- ☒ Att kunna pausa exekveringen av en funktion och återuppta den senare, ofta över flera bildrutor (frames), utan att hela spelet fryser. 
- ☐ Att permanent lagra data, såsom spelarens poäng, mellan olika spelsessioner.
- ☐ Att automatiskt hantera fysikaliska kollisioner och reaktioner mellan GameObjects.

## 11 Collider som Trigger

Vad är huvudskillnaden mellan en Collider markerad som "Is Trigger" och en som inte är det?

**Välj ett alternativ:**

- ☐ "Is Trigger" colliders är alltid osynliga.
- ☐ "Is Trigger" colliders orsakar fysisk kollision, medan vanliga bara detekterar.
- ☒ "Is Trigger" colliders detekterar överlappning utan fysisk respons, medan vanliga orsakar fysisk kollision. 
- ☐ "Is Trigger" colliders kan bara användas på 2D-objekt.



## 12 ECS i Unity

Vad står ECS för inom Unity spelutveckling?

Välj ett alternativ:

☐ Enhanced Controller System



☐ Entity Component System



☐ External Content Server

☐ Efficient Collision Scripting

## 13 Sprite Sheet

Vad är en "Sprite Sheet" i Unity?

Välj ett alternativ:

☐ Ett enskilt C#-skript som hanterar alla sprites.

☒ En enskild bildfil som innehåller flera mindre bilder (frames) för en animation eller en tile set.



☐ En 3D-modell som har konverterats till 2D.

☐ Ett dokument som beskriver hur sprites ska användas.

## 14 Flow State

Vad kännetecknar tillståndet "Flow" enligt Mihaly Csikszentmihalyi?

**Välj ett alternativ:**

- ☐ En känsla av uttråkning på grund av för lätta utmaningar.
- ☒ Fullständig uppslukning och fokusering i en lagom utmanande aktivitet.
- ☐ Frustration på grund av för svåra utmaningar.
- ☐ En passiv upplevelse där spelaren inte behöver agera.

## 15 3D-kameratyp

Vilken typ av kamera används vanligtvis för att skapa en känsla av djup och perspektiv i 3D-spel?

**Välj ett alternativ:**

- ☐ Orthographic Camera
- ☒ Perspective Camera
- ☐ Cinematic Camera
- ☐ 2D Camera

## 16 UI:s Rot-GameObject

Vilket GameObject är nödvändigt som rot för alla UI-element i en scen?

**Välj ett alternativ:**

☐ EventSystem

☐ Camera

☒ Canvas



☐ UI Manager

## 17 Användning av Lager

Vad används "Layers" (Lager) primärt till i Unity?

**Välj ett alternativ:**

☐ Att organisera ljudfiler i projektet.

☒ Att styra vilka GameObjects som kan kollidera med varandra och vilka kameror som renderar dem.



☐ Att definiera olika svårighetsgrader i spelet.

☐ Att gruppera C#-skript för bättre prestanda.

## 18 Prefab-definition

Vad är en "Prefab" i Unity?

**Välj ett alternativ:**

- ☐ Ett färdigt 3D-objekt från Unity Asset Store.
- ☒ En mall av ett GameObject med dess komponenter som kan återanvändas.
- ☐ Ett C#-skript som styr spelets logik.
- ☐ En inställning för hur fysik ska fungera i scenen.



## 19 Komponent-arv

Vilken klass måste ett C#-skript ärva från för att kunna kopplas som en komponent till ett GameObject i Unity?

**Välj ett alternativ:**

- ☐ GameObject
- ☐ Component
- ☒ MonoBehaviour
- ☐ System.Object



## 20 Scenstruktur

I vilket fönster i Unity Editor ser du en hierarkisk lista över alla GameObjects i den nuvarande Scenen?

Välj ett alternativ:

- ☐ Project Window
- ☐ Inspector Window
- ☒ Hierarchy Window
- ☐ Console Window



## 21 MonoBehaviour-livscykeln

**MonoBehaviour-livscykeln och Interaktion:** Ett C#-skript i Unity som ärver från `MonoBehaviour` har tillgång till flera speciella metoder som anropas automatiskt av Unity vid olika tidpunkter.

1. Beskriv syftet med och när följande livscykelmetoder typiskt anropas: `Awake()`, `Start()`, `Update()`, och `FixedUpdate()`. (1.5p)
2. Förklara hur man i ett skript kan få en referens till en annan komponent på samma `GameObject`, och varför det ofta är bra att göra detta i `Awake()` eller `Start()` istället för i `Update()`. (1p)

1. `Awake()` Är den alltid den första metoden som körs. Alltså den körs alltid dirket innan "spelet börjar" så att säga. Bra för att se till att vissa saker som absolut måste vara på plats i spelet och fungera alltid körs förts

`Start()` Körs efter awake så fort man "börjar spela" så att säga. nyckeln med både awake och start är det körs en gång endast när man börjar spela.

`Update()` innehåller logik och script för olika typer av gameobjects i scenen och körs i varje igen och igen i varje ny frame

2. Exempelvis kan man lägga till en rigid body fysik komponent på ett gameobject och referera till den i tillhörande script. Anledningen att inte göra det i update beror på att update renderingstid kan skilja sig från dator till dator

Ord: 133

## 22 Unitys Kärnkoncept

**Unitys Kärnkoncept:** Beskriv följande centrala koncept i Unity och hur de relaterar till varandra:

1. **GameObject:** Vad är det och vilken roll spelar det? (0.5p)
2. **Component:** Vad är det och hur relaterar det till ett GameObject? Ge exempel på minst två olika typer av komponenter. (1p)
3. **Prefab:** Vad är en Prefab och varför är de användbara i spelutveckling? Hur skapar man en Prefab? (1p)

1. Spelar enormt stor roll då dessa representerar allt som finns med i scen.

Inte nödvändigtvis visuellt men de existerar ändå alltid i scenen och påverkar på något sätt spelet. Allt som existerar i scenen är alltså ett gameobject (även kameran).

2. Komponenter är det man använder för tillgodose gameobjects med specifika funktionaliteter etc ( dvs det som karaktäriserar ett specifikt gameobjekt på olika sätt) . Huvudkomponenten som finns med i alla gameobjects heter transform och används primärt för att styra vart på kartan spelaren befinner m.m. Två andra komponenter är exempelvis Rigidbody/Rigidbody2D. Dessa är fysikkomponenter som gör det möjligt för ett gameobjekt att bli påverkat av fysikaliska effekter som gravitation, massa etc.. En annan komponent är exempelvis boxCollider som gör det möjligt för gameobjects att känna av kollision med andra gameobject

3. Prefab är sparade gameobject i exempelvis assets/prefab mappen. Dessa innehåller grundläggande funktionalitet och komponenter som man kan återanvända för att slippa skapa nya liknande gameobjects med snarlika funktionaliteter, dessa går istället att återanvända.

Ord: 167

## 23 2D Grafik och Kollision med Tilemaps

**2D Grafik och Kollision med Tilemaps:** Tilemaps är ett kraftfullt verktyg för att bygga 2D-nivåer i Unity.

1. Beskriv vad en **Tilemap** är och vad en **Tile Palette** används till i samband med Tilemaps. (1p)
2. Förklara processen för att lägga till **kollision** till en Tilemap så att en spelkaraktär kan gå på vissa tiles men inte passera genom andra. Vilka komponenter är vanligtvis involverade? (1p)
3. Vad är syftet med **Sorting Layers** och **Order in Layer** när man arbetar med 2D-grafik i Unity? (0.5p)

1. Används för att på ett enklare sätt "måla" exempelvis en level i en scene. Detta istället för att använda sig av massa sprite. Så används istället Tilemap och tile palette för ändamålet.
2. Man det två objects varsin boxcollider2D. Om dessa två komponenter också har rigid eller iaf en av dem kommer det orsaka en 2Dcollision med fysikalisk effekt
3. Syftet är bland annat att se till vilka object som kan kollidera med varandra och vilka som skall agera mer i bakgrunden. Man kan säga att det ersätter de djup man har lite i ett 3D spel.

Ord: 99

## 24 Spelteori och Flow

**Spelteori och Flow:** Flow är ett centralt begrepp inom speldesign och spelupplevelse.

1. Definiera begreppet **Flow**. (0.5p)
2. Beskriv minst tre viktiga **förutsättningar** som behöver vara uppfyllda i ett spels design för att en spelare ska kunna uppnå Flow. (1p)
3. Förklara hur en speldesigner kan **applicera kunskapen om Flow** för att skapa mer engagerande spel. Ge ett konkret exempel från ett spel (kan vara Super Mario Bros. som nämns i texten, eller ett annat spel du känner till) och förklara hur det spelet uppfyller några av förutsättningarna för Flow. (1p)

1 Fullständig uppslukning och fokusering i en lagom utmanande aktivitet ( i enkelhet kan ett flow förklaras genom att en spelare kommer in i ett mode, där hen upplever en känsla av tillfredsställelse, genom en mix av njutning, utmaning, prestation med mera som får spelaren att hela tiden vilja fortsätta.

2.

- Direkt feedback av något slag vid input eller motsvarande(dvs ljud, visuell effekt etc)

- Belöning spelaren skall bli belönad när man gör rätt.

- Konstruera spelet så det håller en bra balans i svårighetsgrad. För enkelt = spelaren blir uttråkad i brist på utmaning, för svårt/i princip omöjligt = spelaren blir frustrerad och arg vilket resulterar i dåligt flow

- Även faktorer som bristande eller otydligt mål med spelet eller i givna situationer orsakar tristess och frustration = dåligt flow

3. COD ( call of duty DMZ spelläget) är ett extrem bra exempel. Detta då spelet ger möjlighet till att skjutdueller mot andra spelare vilket känns spännande och tävlingsinriktat = flow. Faktorer som att man i spelet kan klara av missions som i sin tur ger belöningar och låser upp möjligheter till att skaffa andra vapen = flow/belöning/känsla av accomplishment. Faktumet att man i spelet kan hitta saker som bättre västar (eller ta från spelare man dödar, ännu bättre) som man kan ta med sig ut ur spelet för att sedan ta med sig in nästa gång man spelar = flow/skapar mervärde/spänning/koncentration då man inte vill bli av med sakerna man plockar upp .

Ord: 247



## 25 Rörelsekontroll med Hastighetsbegränsning

Skapa ett C#-skript i Unity som heter `LimitedMover`. Skriptet ska kopplas till ett `GameObject` med en `Rigidbody2D`-komponent.

1. Ha en publik `float`-variabel `moveSpeed` som kan ställas in i Inspector (t.ex. till 5.0f).
2. Ha en publik `float`-variabel `maxSpeed` som kan ställas in i Inspector (t.ex. till 3.0f).
3. I `FixedUpdate()`-metoden, läs av horisontell input (`Input.GetAxisRaw("Horizontal")`).
4. Applicera en kraft på `Rigidbody2D` i inputens riktning, multiplicerat med `moveSpeed`. Använd `ForceMode2D.Force`.
5. Se till att `Rigidbody2D`:s horisontella hastighet (`velocity.x`) aldrig överstiger `maxSpeed` (i absolutvärde, dvs. både positivt och negativt). Om hastigheten överstiger `maxSpeed`, ska den klampas (begränsas) till `maxSpeed` eller `-maxSpeed`.

**Skriv din kod här:**

```
using UnityEngine;
```

```
public class LimitedMover : MonoBehaviour
```

```
[serializedFields] moveSpeed;
```

```
[serializedFields] maxSpeed;
```

```
{
```

```
    public float moveSpeed = 5.0f;
```

```
    public float maxSpeed = 3.0f
```

```
public void Start()
```

```
{}
```

```
    FixedUpdate ()
```

```
{
```

```
    (Input.GetAxisRaw("Horizontal"))
```

```
}
```

```
}
```

## 26 Enkel Objektbytare vid Klick

Skapa ett C#-skript i Unity som heter `SpriteSwapper`. Detta skript ska kopplas till ett `GameObject` som har en `SpriteRenderer`-komponent. Skriptet ska ha två publika `Sprite`-variabler som kan ställas in i Inspector: `spriteOne` och `spriteTwo`.

När spelaren klickar med vänster musknapp var som helst på skärmen ska `GameObject`ets `SpriteRenderer` byta till den *andra* spriten. Om `spriteOne` visas, ska den byta till `spriteTwo`. Om `spriteTwo` visas, ska den byta tillbaka till `spriteOne`. Initialt ska `spriteOne` visas.

**Skriv din kod här:**

```
using UnityEngine;
```

```
public class SpriteSwapper: MonoBehaviour
{
    public Sprite spriteOne;
    public Sprite spriteTwo;

    public void Start()
    {

    }

    public void Update()
    {

    }

}
```

Ord: 25